

OTA APP: UMA SOLUÇÃO PARA ORGANIZAR TAREFAS ACADÊMICAS

Marcelo Veronez Ramos¹

Bruno Nascimento Brancalhão²

Daniel Facciolo Pires³

Resumo: a tecnologia embarcada nos *smartphones* com sistemas operacionais Android e iOS é evoluída, oferecendo mobilidade na utilização de *softwares* para a maioria dos usuários. O presente artigo apresenta os resultados com o desenvolvimento de um aplicativo para dispositivo móvel que permita a organização de tarefas acadêmicas para alunos, denominado OTA APP. A partir de pesquisas bibliográficas com artigos eletrônicos, em temas da área, e levantamento de requisitos dos dados e informações acadêmicas, foi proposto o desenvolvimento de um aplicativo, que possibilite uma organização de atividades baseado na metodologia Kanban. O processo metodológico passou por levantamento de requisitos, desenvolvimento e após o sistema estar pronto, iniciaram-se os testes.

Palavras-chave: Tecnologia. Híbridos. Atividades. Acadêmicas. OTA.

Abstract: *the technology embedded in smartphones with Android and iOS operating systems is evolved, providing mobility in the use of software for the majority of users. This article presents the results with the development of a mobile application that allows the organization of academic tasks for students, called OTA APP. Based on bibliographical researches with electronic articles, on subjects of the area, and survey of data requirements and academic information, it was proposed the development of an application, which enables an organization of activities based on Kanban methodology. The methodological process went through a survey of requirements, development and after the system was ready, the tests were started.*

Keywords: *Technology. Hybrids. Tasks. Academic. OTA.*

1 Introdução

Atividades acadêmicas são essenciais para formação de um aluno. Instituições de ensino superior dependem delas para testar os conhecimentos gerais do estudante. Pelo pouco tempo que a maioria dos estudantes possuem, torna-se difícil a organização das atividades.

Como as tecnologias de desenvolvimento de aplicativos móveis, somada às metodologias de gestão de projetos, podem melhorar a organização e gestão de tarefas

¹ Discente do curso de Bacharelado em Sistemas de Informação do Uni-FACEF

² Discente do curso de Bacharelado em Sistemas de Informação do Uni-FACEF

³ Docente do curso de Bacharelado em Sistemas de Informação do Uni-FACEF

acadêmicas provenientes de exercícios, trabalhos e demais atividades oriundas das disciplinas sendo realizadas por um aluno? A partir desta pergunta, pode-se definir o problema de pesquisa deste trabalho.

Este artigo tem como objetivo geral apresentar os resultados do desenvolvimento de um aplicativo móvel, que explora conceitos da metodologia Kanban, para facilitar a organização e execução das atividades relacionadas às disciplinas em que o aluno está matriculado. Utilizou-se do banco de dados do Uni-FACEF, para acesso dos usuários e a disciplina que lecionam.

Os procedimentos metodológicos adotados, para o desenvolvimento, foram o uso da ferramenta Ionic, com a vantagem de se programar em sistemas híbridos, as *API's* foram feitas em Node-JS e consumidas em Angular, usando os padrões de desenvolvimento nas respectivas linguagens, e adotou-se o uso do Firebase, para armazenamento de dados. Além disso, seguiu os conceitos da Engenharia de Software para o desenvolvimento do sistema, aplicando os padrões de gestão de projetos apresentados pelo (PMBOK® 2009).

Este trabalho ficou estruturado em seis seções: esta introdução, a segunda apresenta conceitos e definições teóricas que contextualizam o tema, a questão problema e a hipótese de solução desenvolvida; a seção três apresenta estudos de empreendedorismo, para colocação do produto no mercado de aplicativos móveis como o modelo de negócio em Canvas; a quarta apresenta as definições de processos envolvidas na execução do projeto; a seção cinco apresenta as telas explicativas do sistema, parte do código utilizado e testes feitos; finalizando com a sexta seção que apresenta a conclusão do artigo.

2 Referencial Teórico

Para o desenvolvimento deste projeto, são abordados temas de conceitos sobre gerenciamento de projetos e os processos fundamentais na criação de uma aplicação para dispositivos móveis. Trata-se de tecnologias que englobam, desde a interface gráfica apresentada ao usuário, até o armazenamento de dados inseridos na aplicação.

2.1 Gerenciamento de projetos

A prática de administração de projetos não é recente, existe desde o início da história, porém os projetos, atualmente, estão sujeitos a um maior grau de complexidade técnica, exigem uma maior diversidade de conhecimentos e habilidades. Os gerentes estão cada dia mais encontrando grandes problemas de como gerir as organizações, pois estão sujeitas a prazos e recursos um pouco limitados, e convivendo em um ambiente de incertezas. Sendo assim novas formas de organizações e práticas estão surgindo (GIMENES, 2011).

O processo em que se define gerenciamento de projetos está sendo cada dia mais aprimorado pelo guia PMBOK® (2009) como: aplicar os seus conhecimentos, suas habilidades e técnicas as atividades de um projeto, buscando atender todos seus requisitos.

Nos dias de hoje, quaisquer projetos implicam uma grande complexidade técnica, além de exigirem diversas habilidades. Para alinhar estas características, surgiram as formas de gestão e uma destas é a gestão de projetos.

Para Cruz (2011), atender a todas as solicitações do mercado de maneira eficaz, em um círculo que tem como característica principal a velocidade em que as mudanças ocorrem, é de maneira indispensável. Devido a isso o gerenciamento de projetos está crescendo de uma maneira intensa no mercado nos últimos anos.

Todos os projetos atuais podem utilizar de um gerenciamento, em quaisquer situações. Uma de suas principais vantagens é que ele não deve ser utilizado apenas em projetos grandes, pode ser utilizado em quaisquer projetos independentemente de suas complexidades e custos. Para Cruz (2011), dentre os principais benefícios podem se destacar: evitar surpresas durante o processo de execução dos trabalhos, permitir aperfeiçoar alguns diferenciais competitivos e novas técnicas, uma vez que a metodologia esteja estruturada e compreendida e facilitar a alocação de pessoas e equipamentos para trabalho.

2.2 Kanban

Segundo Curto (2010), o termo Kanban, inicialmente usado pela Toyota, cuja significado é “sinais” em japonês, é uma simbologia visual para registrar ações, e inicialmente aplicada na gestão de estoque e controle de fluxo de peças, tendo duas funções em um processo de produção: “desenhar os processos para que fabriquem produtos” e ensinar

manipuladores de materiais a deslocarem os produtos. Para projetos, a parte de desenhar processos é a definição para as colunas do Kanban, que normalmente são, “A fazer”, “Fazendo” e “Feito”, conforme ilustrado na Figura 1. Já a parte de retirada indica a mudança nas atividades do projeto, gerando assim alterações diretas no cronograma.

A Digité (2018) explica que, enquanto o Kanban foi introduzido por Taiichi Ohno na indústria de manufatura, David J. Anderson foi o primeiro a aplicar o conceito de TI, desenvolvimento de software e trabalho de conhecimento em geral, no ano de 2004. O método Kanban é um processo para melhorar e evoluir, aos poucos, tudo o que é feito pelo usuário. O método deve ser aplicado diretamente no fluxo de trabalho, evitando mudanças repentinas, sempre buscando evoluções, respeitando os cargos, responsabilidades e papéis da equipe que está trabalhando no projeto e sempre incentivando atos de liderança, em todos os níveis, uma vez que a equipe toda pode fornecer ideias, para implementar e mudanças para melhorar a entrega final do produto.

Figura 1 – Quadro de tarefas Kanban



Fonte: UNASP, 2016, online.

Ao criar um modelo visual do projeto, é possível observar o fluxo de trabalho através do Kanban, levando a uma maior comunicação e colaboração. Possibilita também o

tempo de movimentação que leva um item no processo, diminuir problemas causados pela alternância de tarefas, além de reduzir a necessidade de priorizar constantemente itens, sem contar com uma melhora contínua, dada a possibilidade de medir a eficácia pelo rastreamento de fluxo, qualidade, produtividade, prazos de entrega entre outros (KAZAKA, 2016).

Novkov (2017) potencializa que o Kanban respeita papéis, títulos hierárquicos atuais de uma organização, a vantagem de estar sempre trabalhando na atividade mais importante primeiro. Concentra-se no fluxo e na conclusão das tarefas, evitando que tarefas não atinjam todo seu potencial de finalização. O Kanban não resolve todos os problemas, pois sempre haverá decisões, dilemas e complexidade que nenhum método pode consertar, mas com uso dele, a taxa de sucesso em projetos aumentando em um nível considerável.

A Bishop (2018) conclui que existem alguns benefícios quando o assunto é o Kanban, entre eles, a versatilidade, pois pode ser usado entre todos os membros da equipe, desde engenharia até marketing e produção, tornando mais fácil a movimentação de projetos. A responsividade que ele apresenta também se insere em benefícios, uma vez que possibilita uma resposta mais ágil às necessidades do negócio. O Kanban incentiva equipes acompanharem como está o andamento do trabalho, a qualquer momento, tornando assim as equipes mais unidas, com menos distrações e conseguindo fazer mais em menos tempo possível, com o produto final de maior perfeição, uma vez que ele coloca o controle de qualidade de volta no processo de gerenciamento de projetos.

2.3 Aplicativos híbridos

Os aplicativos móveis são *softwares* que exercem alguns objetivos específicos em celulares *smartphones* e *tablets*, conforme os programas de computadores. Geralmente são disponibilizados pelas lojas oficiais dos aplicativos de cada sistema operacional, por exemplo Android com a Play Store, Iphone com a Apple Store (PORTO, 2012).

Aplicativo híbrido é aquele que funciona em multiplataformas.

O desenvolvimento de aplicativos híbridos tem seus pontos positivos que são o rápido desenvolvimento, um baixo custo e a utilização de poucas linguagens e *frameworks* em relação a linguagens nativas. Um dos prós mais consideráveis para o desenvolvimento é o financeiro, com uma redução de custos, levando em consideração os aplicativos nativos, pois

não será necessária uma especialização em diversas linguagens e *frameworks* exclusivos além de um bom tempo para projetar e gerenciar os projetos em cada plataforma (AGUIAR, 2017).

Felix (2015) solidifica que aplicações nativas possuem uma interface e performance melhor do que um aplicativo híbrido, vendo que uma aplicação nativa possui um desempenho considerável. Porém, isso depende do sistema que está se desenvolvendo. Supondo-se que será desenvolvido um jogo, que exige um gráfico melhor, então se recomenda utilizar de uma linguagem nativa.

Utilizar de capacidades nativas dos aparelhos não é mais um problema para os aplicativos híbridos. Através de alguns *plug-ins*, como o Cordova, oferece-se um suporte em vários quesitos como acessar a câmera de seu dispositivo, ou a geolocalização do mesmo.

É possível criar aplicativos híbridos, utilizando HTML, CSS e JavaScript, e ainda utilizar de recursos nativos. Com isso, para conseguir desenvolver uma aplicação multiplataforma, que possua acesso aos recursos de hardware e software de diversos processadores (CHARLANDA; LEROUX, 2012). Assim, o tipo de aplicação definida para o desenvolvimento do presente trabalho foi de uma aplicação híbrida.

2.4 Android

Cidral (2011) afirma que o Android é um sistema operacional do Google para dispositivos móveis baseado em Linux. O Android é responsável pelo funcionamento correto de aplicativos no celular e do hardware, possibilitando com que o usuário consiga utilizar praticamente todas os recursos do sistema de forma intuitiva.

O Android é responsável por traduzir os comandos que você faz no celular, transformando estes em uma outra linguagem que o hardware é capaz de entender (ROGATTO, 2017).

2.5 IOS

O iOS, assim como o Android, é um sistema operacional para dispositivos móveis, porém, de outro fabricante (COUTINHO, 2014).

Apesar de ser um sistema operacional consideravelmente antigo, ele vem em constantes evoluções até os dias atuais, tornando-se assim um dos melhores do mercado. Com a forte concorrência do Android, é necessário que a Apple lance inovações para seu sistema, fazendo com que usuários sempre estejam servidos da melhor maneira possível (MAGRELLO, 2012).

2.6 Ionic

O Ionic (2018) é um *framework* para aplicativos móveis híbridos, que utiliza de tecnologias WEB, operando em linguagens como HTML, CSS e JavaScript. Também conta com sua própria ferramenta de interface de linha de comando, que ajuda no processo de criação de um aplicativo, economizando um precioso tempo do desenvolvedor (MITTAL, 2017).

Junior (2016) explica que o Ionic tem alguns pontos fortes, tais como: o uso de *tags* (etiquetas), para performance da tela criada, oferecendo velocidade na programação e a adaptação para a plataforma que será apresentada. O *framework* oferece alta produtividade, uma vez que este possui várias ferramentas que ajudam a testar a aplicação, a integração com outros programas e praticidade de utilizá-lo.

2.7 Angular

Angular (2017) é uma estrutura JavaScript, que auxilia os desenvolvedores a desenvolver aplicativos, conforme explica Vantoll (2018), podendo ser utilizado na construção de programas. Porém o seu melhor desempenho se dá em aplicativos moveis, com um melhor proveito em sistemas desenvolvidos com aplicações híbridas.

O Angular tem uma praticidade no quesito de aprendizado. Em questão de minutos, é possível aprender sobre o *framework*, sem contar com sua baixa manutenção, devido a toda essa praticidade. Uma de suas outras vantagens, é também a agilidade em se programar com Angular, uma vez que a comunidade que utiliza-o é bem ativa no mercado, facilitando as respostas para todas suas dúvidas e podendo dar sequência em seu trabalho (LOPES, 2013).

2.8 TypeScript

TypeScript (2018) é uma linguagem baseada em JavaScript, que suporta o uso de programação orientada a objetos, com uma sintaxe simplificada e mais clara. A programação orientada a objetos sempre foi um grande problema para ser implementada em JavaScript, mas o TypeScript contornou esses problemas (DIONISIO, 2016).

2.9 Node-JS

Node (2018) é uma plataforma que visa construir aplicações web com JavaScript. Construído em cima da “*engine V8*”, que auxilia na interpretação da linguagem JavaScript, é possível utilizar desta pelo lado do servidor. O Node também é capaz de fazer requisições assíncronas, não permitindo bloqueios, tornando-o assim rápido, para lidar com um número alto de requisições com o banco de dados (SANTOS, 2016).

O Node não é uma linguagem de programação, e sim um *framework*, que possui algumas vantagens, entre elas: utilizar da linguagem de JavaScript, ser leve, melhorando o desempenho e podendo ser empregado em multiplataformas. Pode se utilizar do Node na criação de *API's*, que são um tipo de “ponte” para conectar as aplicações. É também utilizado na criação de aplicações que consomem o servidor em tempo real, utilizando algumas extensões nativas do próprio Node (DUARTE, 2017).

2.10 Firebase

Firebase (2018) é uma plataforma de desenvolvimento que foi adquirida pela Google em 2004. Seu intuito é fornecer um *back-end* completo, com fácil uso, com diversos serviços para auxiliar no desenvolvimento, entre eles, está o banco de dados em tempo real transformando as aplicações em soluções mais ágeis nas respostas das requisições, como aplicações em tempo real (GASPERIN, 2017).

3 Empreendendo o OTA APP

Esta seção apresenta eventos relacionados a empreendedorismo, e diante o modelo de negócio desenvolvido para a solução OTA APP.

3.1 Startup

Startup é um momento na empresa, normalmente no começo, quando uma equipe busca desenvolver um produto ou serviço inovador, de âmbito tecnológico, que tenha um modelo de negócio fácil de ser replicado e que possa se expandir sem elevar proporcionalmente os custos (ABSTARTUPS, 2017).

A empresa ABSTARTUPS (2017) confirma que entre as características de uma *startup*, uma das mais importantes é sua capacidade de ganhar alta escala rapidamente, ou seja, conseguir que os produtos gerados por ela sejam utilizados por um alto número de usuários em pouco tempo. Além desta, outra é o modelo de negócio fácil de ser replicado, como já foi explicado anteriormente. Ainda é preciso que ela tenha flexibilidade e rapidez para se adaptar rapidamente às demandas, uma vez que o mercado está em constantes mudanças e geralmente tem estruturas enxutas, com equipes formadas por poucas pessoas.

O modelo de negócio apresenta informações importantes como clientes, custos, fontes de receita, atividades principais entre outros. Em uma empresa clássica, é comum a construção de um plano de negócios, que detalha os diversos aspectos de um modelo de negócios. Porém, em uma *startup*, devido às incertezas e rápidas mudanças geradas tem sido muito utilizado o Canvas, que descreve o modelo de negócios em blocos. Cada bloco representa um elemento deste e normalmente é preenchido com papel adesivo, permitindo a rápida montagem e modificação quando necessário (ABSTARTUPS, 2017).

As *startups* também podem contar com investimento-anjo, que é o investimento efetuado por pessoas físicas, com seu próprio capital em empresas nascentes com alto potencial de crescimento. É chamado de anjo, uma vez que ele não contribui só com dinheiro, mas também com conhecimentos, apoio, experiência e rede de relacionamento para orientar e aumentar a chance de sucesso (ABSTARTUPS, 2017).

3.2 Startup enxuta

O conceito de uma *startup* enxuta é minimizar o ciclo de desenvolvimento da empresa, pondo em prática várias ideias de forma rápida, com fim de alcançar a melhor possível, economizando e atingindo um produto de alta qualidade, que possa chegar ao mercado para venda o quanto antes (AMBRA, 2017).

O foco é o desenvolvimento do Produto Mínimo Viável, que consiste no resultado do teste das hipóteses. Esse teste é feito diretamente com o consumidor, sendo essencial um desenvolvimento com os clientes que normalmente é feito com um grupo de clientes selecionados. Não necessariamente corresponde ao produto final, mas é feito de forma ágil e econômica com tudo que necessita para ser um produto completo que possa ser lançado no mercado. Para não ocorrer um desequilíbrio nessa parte, é necessário o uso do mínimo de recursos investidos na criação com um máximo de viabilidade para ter um indicador de sucesso (AMBRA, 2017).

Por fim, o empreendedor encontra recursos para reduzir desperdícios, otimizar seus ativos e obter uma interação mais próxima e produtiva com seu cliente, ajudando assim a empresa a melhorar seu produto final antes de ser lançado. Afinal, é melhor errar no papel do que no produto já no mercado (SPINA, 2013).

3.3 Modelo Canvas

O Canvas é uma representação visual que possibilita as pessoas criarem modelos de negócios analisando 9 elementos que toda empresa ou organização possuem: 1) proposta de valor; 2) parcerias chaves; 3) atividades chaves; 4) recursos chaves; 5) relacionamento com clientes; 6) segmentos de clientes; 7) canais de distribuição; 8) estrutura de custos, e 9) fluxo de receitas. Normalmente, os processos são colocados em *post-its* em cada elemento separado (MOTA, 2018).

Os 9 componentes cobrem todas as áreas principais de um negócio, sendo elas, clientes, ofertas, infraestrutura e viabilidade financeira (MOTA, 2018).

Outro recurso que o Canvas oferece é a facilidade em fazer mudanças, essencial para *startups*, que no começo da jornada, enfrentam várias mudanças (UDACITY, 2018).

Os 9 elementos são melhor explicados, a seguir.

- 1) Clientes: onde é definido o segmento de mercado. Devem ser definidos os principais clientes para o projeto.
- 2) Valor provido: é respondido à questão de como se pode ajudar seus clientes, definindo quais serão os principais benefícios do produto.
- 3) Canais: onde e como os produtos serão divulgados e vendidos.
- 4) Relacionamento com clientes: aborda como irá ser feito a interação com os clientes.
- 5) Receitas e Benefícios: a parte que lida com o que será ganho com o projeto.
- 6) Recursos-chave: aborda o que se tem e quais recursos necessário para finalização do projeto.
- 7) Atividades-chave: o que será feito no projeto pela equipe de desenvolvimento.
- 8) Parcerias-chave: são itens ou atividades que serão fornecidos por outra empresa ou pessoas fora do projeto.
- 9) Custos: lida com o que será gasto pela equipe ao longo do desenvolvimento do projeto.

3.4 Modelo de negócio do Organize Tasks Application (OTA APP)

Com o aumento excessivo de tarefas exigidas das pessoas no dia a dia, o tempo está ficando cada vez mais curto e a organização de cada uma delas mais precária. Outro fator que aumenta mais o fluxo constante de atividades é que com a concorrência do mercado atual, a sociedade tende a sempre buscar melhor competitividade, capacitando-se com cursos, bacharelados, pós-graduações, entre outros, fazendo assim surgir em mais tarefas a serem feitas, exigindo assim *softwares* que possam apoiar a organização.

Tratando deste assunto, a proposta desse modelo traz a ideia de um aplicativo móvel, que possa incentivar a organização das tarefas acadêmicas de cada aluno, que tem seu

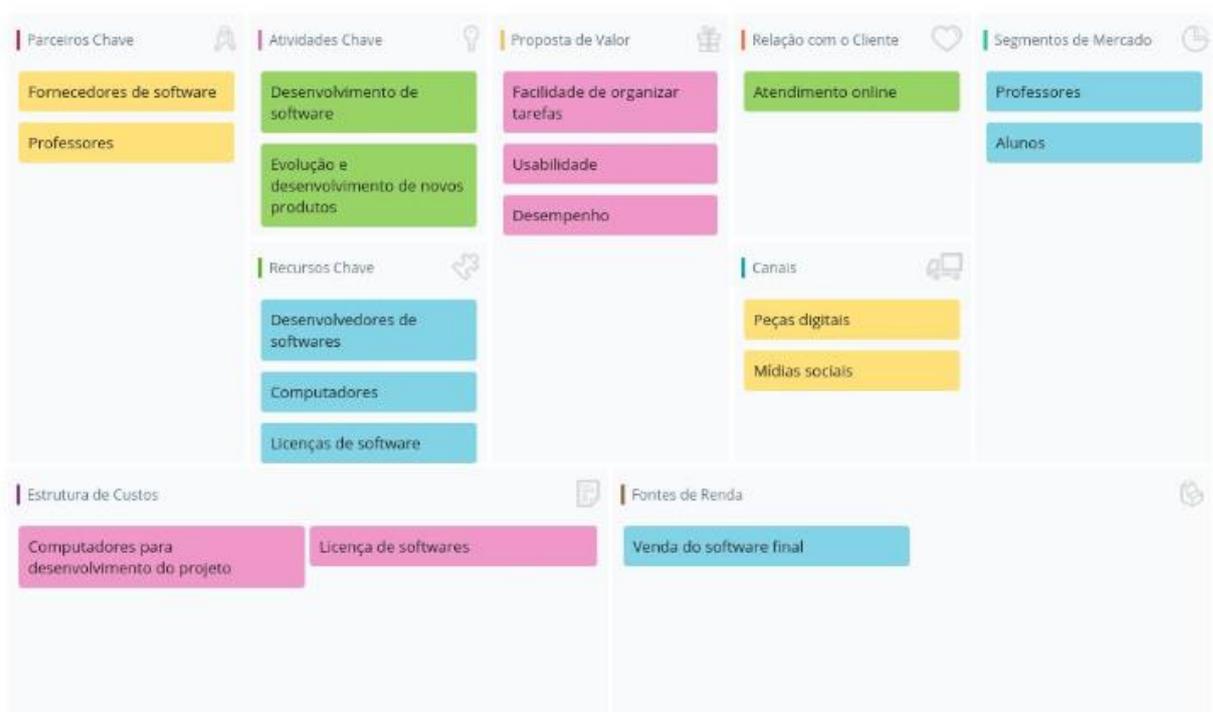
tempo corrido devido a trabalhos e atividades. São detalhados os 9 componentes do modelo Canvas, para este aplicativo, conforme ilustrado na Figura 2.

Segmento de mercado e proposta de valor: focado no segmento de nicho de mercado para alunos e professores, o projeto irá viabilizar a organização de atividades, aumentando assim o desempenho nas mesmas. Através de um aplicativo para celulares, será fácil atingir nosso objetivo final.

Canais: nosso aplicativo será divulgado de maneira direta em mídias sociais e peças digitais, os dois que abrangem um grande público alvo para nosso projeto, no caso, estudantes de faculdades, cursos, entre outros.

Figura 2 – Modelo de negócio - Aplicativo

TCC - Marcelo e Bruno



Fonte: Os autores.

Relação com o cliente: nosso foco é a facilidade e usabilidade, sendo assim, na parte de relacionamento, será realizada uma assistência social, totalmente por meio de atendimento online, seja via e-mail ou chat.

Fontes de renda: nossa principal fonte de renda será com a venda final de licença de uso do nosso projeto, através de uma licença, para faculdades e cursos que querem adquirir uma maneira fácil de seus alunos para organizarem suas atividades.

Recursos-chave: para desenvolvimento de tal projeto, serão necessários recursos físicos e intelectuais como, softwares para programação base, computadores para rodar tais softwares, além de licenças para uso desses softwares que irão nos auxiliar.

Atividades-chave: nossa principal atividade será a produção através do desenvolvimento de um aplicativo mobile, contando sempre com novas evoluções para o mesmo, por meio de manutenção e inovação tecnológica.

Parceiros Chave: os professores do curso irão nos auxiliar em todo o desenvolvimento, reduzindo riscos para o sucesso final do projeto. Além disso, as empresas que disponibilizam os softwares necessários para que tal desenvolvimento ocorra, também vão nos ajudar com aquisições de recursos.

Estrutura de custo: nosso custo fixo será com os computadores necessários para rodar softwares de auxílio para o desenvolvimento do projeto e com as licenças de tais programas.

4 Análise e projeto do OTA APP

Para disponibilizar e compartilhar com a comunidade os processos de nosso desenvolvimento, foi feito um repositório no Github (2018) que está disponível em uma conta de autoria própria (RAMOS; BRANCALHÃO, 2018). A identificação dos requisitos consiste em analisar as características a serem atendidas pelo sistema, atendendo às necessidades e às expectativas do cliente.

4.1 Categoria dos requisitos

Para estabelecer a categoria dos requisitos foram adotadas as denominações “evidente” e “oculta”.

- **Evidente:** é a categoria sem o qual o sistema não entra em funcionamento. Categorias evidentes são os requisitos imprescindíveis, que têm que ser implementados e são mostrados para o usuário na aplicação.
- **Ocultas:** é a categoria sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Categorias ocultas devem ser implementadas e não são mostradas visualmente para usuário na aplicação.

4.2 Prioridade dos requisitos

Para estabelecer a prioridade dos requisitos, foram adotadas as denominações “alta”, “média” e “desejável”.

- **Alta:** é o requisito sem o qual o sistema não entra em funcionamento. Requisitos altos são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.
- **Média:** é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos médios devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.
- **Desejável:** é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar, de forma satisfatória sem ele. Requisitos desejáveis são requisitos que podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los.

4.2.1 Requisitos funcionais

Requisitos funcionais do sistema definem como ele deve agir perante diversas situações, podendo também declarar o que o sistema não deve fazer (SOMMERVILLE, 2008). Os requisitos funcionais do projeto estão disponíveis no Github (RAMOS; BRANCAHÃO, 2018).

4.2.2 Requisitos não funcionais

Um requisito funcional é aquele que não descreve o que o sistema fará, mas sim como ele fará, possuindo grande importância igual o requisito funcional (FILHO, 2008). Os requisitos não funcionais deste projeto estão disponíveis no Github (RAMOS; BRANCALHÃO, 2018).

4.3 Regra de negócios

Para Ventura (2016), regras de negócios são restrições que o sistema precisa seguir, para que funcione de maneira correta, de acordo com o pedido do cliente. Caso o sistema tenha uma regra de negócio mal especificada, poderá haver defeitos de uso no futuro. As regras de negócio deste projeto estão localizadas no Github (RAMOS; BRANCALHÃO, 2018).

4.4 Matriz de rastreabilidade

Project Builder (2017) explica que a matriz de rastreabilidade tem papel fundamental na otimização do gerenciamento de projetos, sendo possível analisar alterações de maneira rápida e eficaz. Ter uma análise mais profunda dos requisitos é também uma forma de conferir se todos os requisitos estão sendo utilizados no sistema. A matriz de rastreabilidade deste projeto está disponível no Github (RAMOS; BRANCALHÃO, 2018).

4.5 BPMN

O *Business Process Modeling Notation* (BPMN) é um quadro de notações que representa processos de negócios por meio de diagramas. O BPMN descreve corretamente a lógica dos passos usados em um processo. Com esta modelagem, é fácil ter uma notação gráfica, mesmo com processos complexos. A partir do BPMN, também já é possível descobrir possíveis falhas no sistema (NOGUEIRA, 2014). O BPMN deste projeto está disponível no Github (RAMOS; BRANCALHÃO, 2018).

4.6 UML

Unified Modeling Language (UML) é uma linguagem que define uma série de artefatos que ajudam a modelar o projeto. Ao total, ele tem nove diagramas, porém, no desenvolvimento deste projeto foram usados só quatro que são: caso de uso, diagrama de atividade, máquina de estado e diagrama de sequência. São descritos e apresentados a seguir.

4.6.1 Caso de uso

No caso de uso, a parte principal não é o diagrama, mas sim a especificação, a descrição dos seus cenários. Possui três elementos principais, o ator, que fará a execução do caso de uso, os casos de uso, que é a ação em si e os relacionamentos, para firmar o contato destas ações (VENTURA, 2016). O caso de uso, e a documentação dele, pode ser encontrado no Github (RAMOS; BRANCALHÃO, 2018).

4.6.2 Diagrama de atividade

O diagrama de atividades ilustra graficamente como será o funcionamento do *software*, suas execuções e a atuação do sistema no negócio em que ele será implementado. É parecido com um fluxograma, para mostrar o comportamento do *software*, no ponto de vista funcional (VENTURA, 2016). O diagrama de atividade que representa este projeto está no Github (RAMOS; BRANCALHÃO, 2018).

4.6.3 Máquina de estado

O diagrama de máquina de estado procura acompanhar as mudanças sofridas no estado das ações do sistema, demonstrando o comportamento do elemento nestas transições de estados (BARRETO, 2009). O diagrama de máquina de estado presente neste projeto, está disponível no Github (RAMOS; BRANCALHÃO, 2018).

4.6.4 Diagrama de sequência

O diagrama de sequência mostra e reforça a troca de mensagens dentro de uma linha de tempo sequencial (MELO, 2009). Um diagrama de sequência, normalmente, identifica-se através de um caso de uso, mostrando a linha de vida deste (BARRETO, 2009). A exposição do diagrama de sequência deste projeto está no Github (RAMOS; BRANCALHÃO, 2018).

4.7 Modelo de entidade de relacionamento

É um modelo conceitual que descreve os objetos envolvidos em um domínio de negócio, mostrando seus atributos e as relações que ele apresenta (RODRIGUES, 2014). O modelo do projeto está disponível em Github (RAMOS; BRANCALHÃO, 2018).

4.8 Diagrama entidade-relacionamento

É o diagrama que representa o modelo de entidade de relacionamento de maneira gráfica, facilitando a comunicação dos integrantes da equipe, oferecendo uma linguagem comum (RODRIGUES, 2014). O diagrama deste projeto pode ser acessado no Github (RAMOS; BRANCALHÃO, 2018).

5 Resultados

Nesta seção, são apresentados os códigos fonte, as telas e os resultados dos testes da solução proposta. A solução fez uso de *API's* do Uni-FACEF, para que se pudesse acessar informações da conta do usuário e disciplinas vigentes do aluno.

5.1 Códigos

A Figura 3 ilustra o código que foi criado para o sistema consumir a *API* para acessar o banco de dados do Uni-FACEF e fazer o *login* com o código do aluno e os 6 primeiros dígitos do CPF.

Já, na Figura 4, é ilustrado o código criado para acessar a *API* que busca automaticamente as matérias que o aluno está matriculado e mostra para ele na tela de matérias da aplicação. As linhas 3 a 5 mostram o consumo da *API* feita em Angular.

A Figura 5 ilustra o código da *API* desenvolvido para buscar os dados das atividades no Firebase, em que é possível salvar uma nova tarefa, atualizar a descrição ou removê-la. Também é salvo o status das atividades para mudá-la entre as abas “A fazer”, “Fazendo” e “Feito”. As tarefas não estão sincronizadas com o banco de dados do Uni-FACEF sendo guardadas apenas no banco de dados da solução.

Figura 3 – *Login* do usuário

```
1 login(username: string, password: string) {
2     return new Promise((resolve, reject) => {
3         let url = this.API_URL + 'matriculadoGrad/';
4         this.http.get(url+username )
5             .subscribe((result: any) => {
6                 console.log(result);
7                 resolve(result.json());
8             },
9             (error) => {
10                reject(error.json());
11            });
12    });
13 }
```

Fonte: Os autores.

Figura 4 – Busca matérias

```
1 buscaMaterias(id_aluno: string){
2     return new Promise((resolve, reject) => {
3         let url = this.API_URL + 'inscricoesDisciplinaGrad/';
4         this.http.get(url+id_aluno).subscribe((result: any) =>{
5             resolve(result.json());
6         },
7         (error) =>{
8             reject(error.json());
9         })
10    });
11 }
```

Fonte: Os autores.

Figura 5 – Firebase acessando as atividades do aluno

```
1 async getAtividades(id_aluno : string, id_disciplina : string) {
2     var atividades = [];
3     await this.db.database.ref('/atividades/').orderByChild('idx_aluno_disciplina').equalTo(id_aluno +
4     '_' + id_disciplina).once('value', (snapshot) => {
5         let result = snapshot.val();
6         for(let k in result){
7             atividades.push({
8                 key : k,
9                 id_aluno : result[k].id_aluno,
10                id_disciplina : result[k].id_disciplina,
11                titulo : result[k].titulo,
12                descricao : result[k].descricao,
13                status_atividade : result[k].status_atividade,
14                atividade_custom : result[k].atividade_custom,
15                idx_aluno_disciplina : result[k].idx_aluno_disciplina,
16                idx_atvTitulo : result[k].idx_atvTitulo
17            })
18        }
19    })
20    return atividades;
21 }
```

Fonte: Os autores.

5.2 Telas

Após o desenvolvimento dos códigos, é trabalhado o visual do aplicativo, conforme citado a seguir.

5.2.1 Login

A tela inicial da aplicação é a parte de *login*, como ilustra a Figura 6, onde o usuário entra com seu código e os seis primeiros dígitos do CPF. Após apertar em “Acessar”, o aplicativo verifica se o aluno está matriculado e efetua o *login*. Caso o usuário não saia de

sua conta, a mesma fica salva em um banco de dados local do aplicativo, permitindo assim que não seja necessário fazer *login* toda vez que acessar o aplicativo.

5.2.2 Disciplinas

Após efetuar o *login*, o usuário é transferido para a tela de disciplinas, ilustrada na Figura 7. Nessa parte, as disciplinas são recuperadas automaticamente. Também é possível acessar o menu nesta tela para ver mais informações do aplicativo, do usuário e sair da aplicação, retornando assim a tela de *login* para que seja possível entrar com outra conta.

Figura 6 – Tela do *login*



Fonte: Os autores.

Figura 7 – Tela de matérias



Fonte: Os autores.

5.2.3 Atividades

Ao clicar em qualquer disciplina, o usuário é transferido para a tela de atividades, conforme ilustra a Figura 8. Nessa tela, o usuário pode selecionar entre “A fazer”, “Fazendo” ou “Feito” e mover as atividades entre elas conforme ilustrado na Figura 9. Também é possível alterar a descrição de cada atividade, remover e adicionar novas. Quando adicionado uma nova tarefa, esta automaticamente já vai para a aba de “A fazer”. Ao clicar em “Voltar”, o usuário é levado para a tela de disciplinas.

5.2.4 Cadastro de atividades

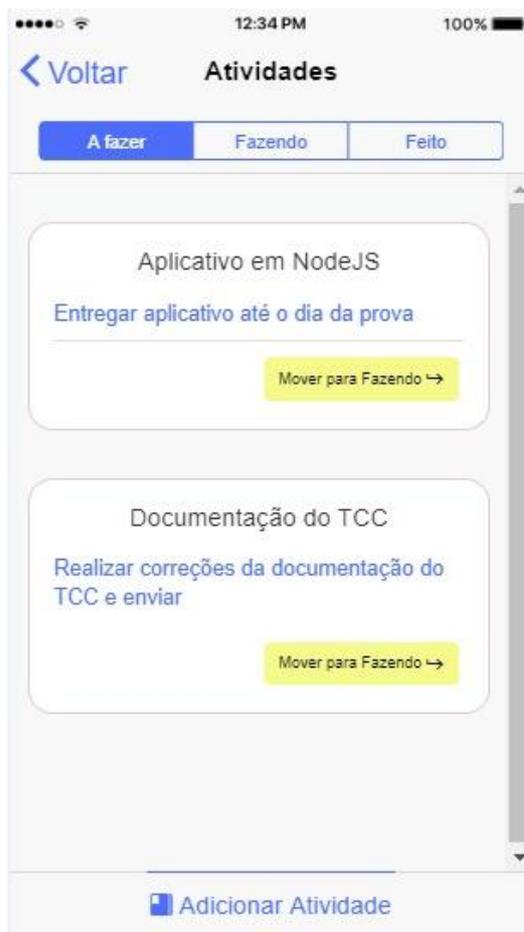
Na tela de atividades, ao acessar a área de adicionar atividade, o usuário é levado para uma tela, onde é possível criar novas tarefas, conforme ilustrado na Figura 10. Nessa tela, é possível criar um título para a atividade, a descrição e selecionar se quer mesmo salvar ou cancelar.

5.2.5 Menu

Na tela de disciplinas, é possível acessar o menu do aplicativo, como ilustrado na Figura 11. Neste menu, o usuário pode ver seus dados, sair do aplicativo, limpar todas as atividades de todas as disciplinas e ver mais informações sobre o aplicativo, como sua atual versão.

Figura 8 - Tela de atividades “A fazer”
“Fazendo”

Figura 9 – Tela de atividades



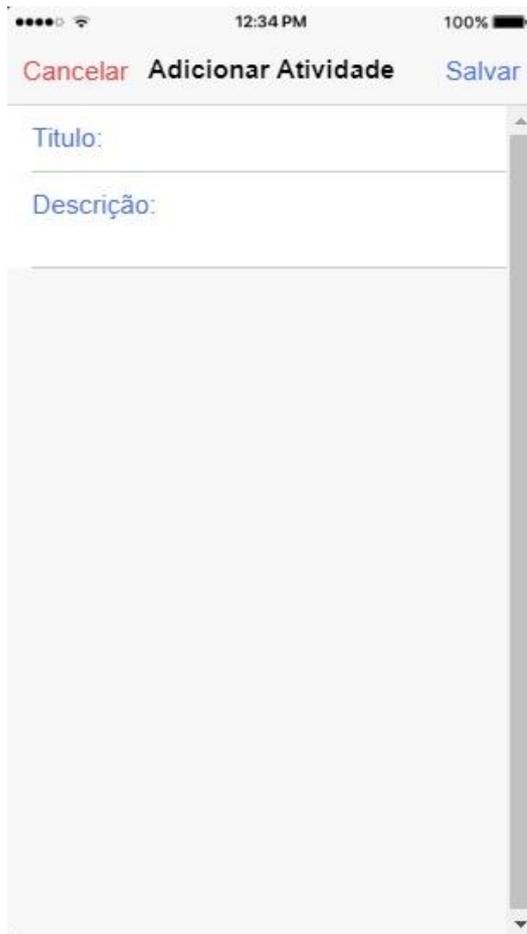
Fonte: Os autores.

Figura 10 – Tela de adicionar atividades

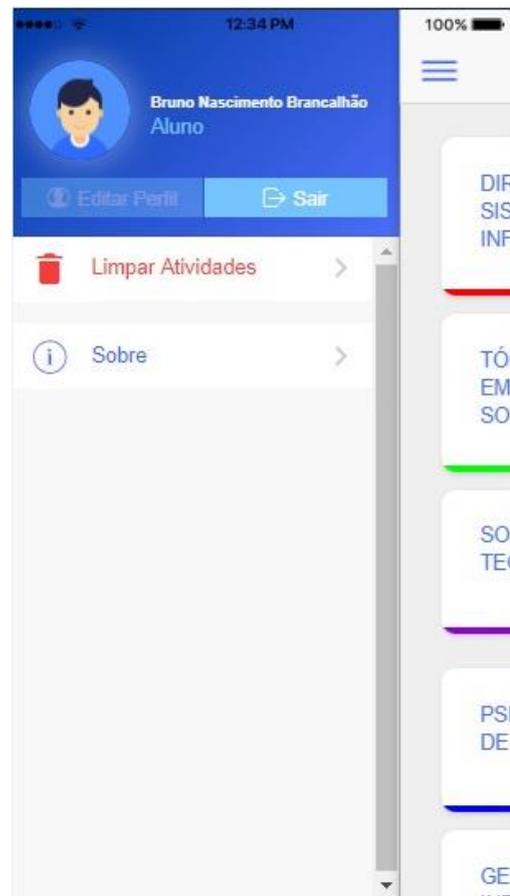


Fonte: Os autores.

Figura 11 – Menu



Fonte: Os autores.



Fonte: Os autores.

5.3 Testes

Conforme ilustrado na Quadro 1 a seguir, foram realizados testes para avaliar o projeto, sob a percepção de possíveis usuários.

De acordo com as entrevistas realizadas, notou-se que o aplicativo foi aceito pelos usuários com uma usabilidade prática, porém, serão necessárias implementações futuras para melhorá-lo.

Quadro 1 – Testes feitos para o projeto

TESTES	Qual seu curso?	Você usaria o aplicativo? Com qual frequência?	O que achou do visual do aplicativo?	O que melhoraria no aplicativo?	O que achou da usabilidade do aplicativo?
Pessoa 1 20 anos Masculino	Sistemas de Informação	Sim, todos os dias.	Bonito.	Nada.	Fácil de entender por todos.
Pessoa 2 22 anos Feminino	Engenharia Civil	Sim, pelo menos uma vez por semana.	Simple e fluído, mas poderia ter outras cores.	Mudaria as cores, de resto, nada.	Fácil de maneira bem simples de ser usado.
Pessoa 3 20 anos Masculino	Engenharia de Software	Sim, provavelmente todos os dias.	<i>Design</i> moderno e bonito.	Nada.	Uma maneira prática de organizar tarefas.
Pessoa 4 23 anos Feminino	Matemática	Sim, quando tiver atividades.	Bonito, com cores boas.	Um jeito melhor de salvar usuário para não fazer <i>login</i> sempre.	Entendi facilmente sem muitas explicações.

Fonte: Os autores.

6 Conclusão

Em virtude dos fatos mencionados, pode-se concluir que o desenvolvimento de um aplicativo para organização de tarefas acadêmicas para auxiliar o aluno pode ser desenvolvido com as tecnologias atuais de Engenharia de Software.

A conclusão deste projeto foi devido ao estudo de diversas áreas do ramo de Sistemas de Informação, o que pode trazer inovação para as instituições de ensino superior, uma vez que pode ajudar seus alunos em suas responsabilidades e execução de tarefas importantes.

Em futuras implementações do sistema, será feito o acesso das atividades diretamente com a instituição de ensino superior, recuperando as tarefas do aluno que o professor inseriu no AVA, sem precisar que ele as adicione manualmente, tornando assim ainda mais rápido e prático. Também será possível que o usuário edite seu perfil, colocando o nome que deseja ser exibido e alterar sua foto.

Referências

ABSTARTUPS. **Tudo que você precisa saber sobre startups.** Disponível em: <<https://abstartups.com.br/2017/07/05/o-que-e-uma-startup/>> Acesso em 17 abr. 2018.

AGUIAR, Anderson. **Aplicativo Nativo ou Aplicativo Híbrido: Qual a melhor opção?** Disponível em: <<https://secaoweb.com.br/blog/aplicativo-nativo-vs-aplicativo-hibrido/>> Acesso em 15 abr. 2018.

AMBRA, Redator. **Startup enxuta: entenda o conceito e aplique na sua empresa.** Disponível em: <<https://blog.ambracollege.com/startup-enxuta/>> Acesso em 06 jun. 2018.

ANGULAR. **What is Angular?** Disponível em: <<https://angular.io/guide/releases>> Acesso em 17 abr. 2018

BISHOP, Karri. **6 Benefits of Kanban for Project Management.** Disponível em: <<https://leankit.com/blog/2017/03/6-benefits-kanban-project-management/>> Acesso em 05 abr. 2018.

CHARLAND, Andre.; LEROUX, Brian. **Mobile Application Development: Web vs. Native.** Disponível em: <<https://queue.acm.org/detail.cfm?id=1968203>> Acesso em 15 abr. 2018.

CIDRAL, Beline. **Afinal, o que é Android?** Disponível em: <<https://www.techtudo.com.br/artigos/noticia/2011/01/afinal-o-que-e-android.html>> Acesso em 15 abr. 2018

COUTINHO, Dário. **O iOS é o sistema da Apple para os seus dispositivos móveis.** Disponível em: <<https://www.techtudo.com.br/tudo-sobre/ios.html>> Acesso em 15 abr. 2018.

CRUZ, Fábio. **Scrum e o gerenciamento de projetos.** Disponível em: <<https://www.devmedia.com.br/scrum-e-o-gerenciamento-de-projetos/22526>> Acesso em 05 abr. 2018.

CURTO, Hayala. **O Kanban no Gerenciamento de Projetos.** Disponível em: <<http://netproject.com.br/blog/o-kanban-no-gerenciamento-de-projetos/>> Acesso em 05 abr. 2018.

DIGITÉ. **What is Kanban?** Disponível em: <<https://www.digite.com/kanban/what-is-kanban/>>. Acesso em 05 abr. 2018.

DIONISIO, Edson José. **Introdução ao TypeScript.** Disponível em: <<https://www.devmedia.com.br/introducao-ao-typescript/36729>> Acesso em 17 abr. 2018.

DUARTE, Luiz. **O que é Node.js e outras 5 dúvidas fundamentais.** Disponível em: <<http://www.luiztools.com.br/post/o-que-e-nodejs-e-outras-5-duvidas-fundamentais/>> Acesso em 17 abr. 2018.

FELIX, Waldyr. **6 aspectos essenciais para decidir entre aplicações mobile híbridas e nativa.** Disponível em: <<https://medium.com/@waldyrfelix/6-aspectos-essenciais-para-decidir-entre-aplica%C3%A7%C3%B5es-mobile-h%C3%ADbridas-e-nativas-51bce0dace68>> Acesso em 15 abr. 2014.

FILHO, Antonio Mendes da Silva. **Artigo Engenharia de Software 3 - Requisitos Não Funcionais.** Disponível em: <<https://www.devmedia.com.br/artigo-engenharia-de-software-3-requisitos-nao-funcionais/9525>> Acesso em 16 ago. 2018

FIREBASE. **Notas da versão.** Disponível em: <<https://firebase.google.com/support/releases?hl=pt-br>> Acesso em 17 abr. 2018.

GASPERIN, Carlos Alberto. **Firestore: O Que é e Como Funciona.** Disponível em: <<http://micreiros.com/firebase-o-que-e-e-como-funciona/>> Acesso em 17 abr. 2018

GIMENES, Nicholas. **Gerenciamento de Projetos - Guia para Iniciantes - 16 Passos!** Disponível em: <<http://www.nicholasgimenes.com.br/2009/06/gerenciamento-de-projetos-guia-para.html>> Acesso em 05 abr. 2018.

GITHUB. Disponível em: <<https://github.com>> Acesso em 05 abr. 2018.

IONIC. **Ionic Releases.** Disponível em: <<https://libraries.io/npm/ionic/versions?page=3>> Acesso em 17 abr. 2018.

JUNIOR, Lazaro. **5 vantagens do Ionic para desenvolver suas aplicações mobile.** Disponível em: <<http://blog.alura.com.br/5-vantagens-do-ionic-para-desenvolver-suas-aplicacoes-mobile/>> Acesso em 17 abr. 2018.

KOZAKA, Ezio. **Uma curta história do Kanban e como ele funciona.** Disponível em: <<https://pt.linkedin.com/pulse/uma-curta-hist%C3%B3ria-do-kanban-e-como-ele-funciona-ezio-kozaka>> Acesso em 05 abr. 2018.

LOPES, Camilo. **Série AngularJS – Por onde começar?** Disponível em: <<http://blog.camilolopes.com.br/serie-angularjs-por-onde-comecar-qual-o-melhor-livro/>> Acesso em 17 abr. 2018.

MAGRELLO, Weverton. **IOS básico - 1 Introdução.** Disponível em: <<http://www.ebah.com.br/content/ABAAafeGAAH/ios-basico>> Acesso em 15 abr. 2018.

MITTAL, Aman. **What is Ionic?** Disponível em: <<https://hackernoon.com/what-is-ionic-c1da6eab0d8a>> Acesso em 17 abr. 2018.

MOTA, Gleison. **Canvas: o que é e para que serve?** Disponível em: <<http://www.administradores.com.br/artigos/empreendedorismo/canvas-o-que-e-e-para-que-serve/109236/>> Acesso em 06 jun. 2018.

NODE. **Previous Releases.** Disponível em: <<https://nodejs.org/en/download/releases/>> Acesso em 17 abr. 2017.

NOVKOV, Alex. **Debunking the Biggest Myths About Kanban.** Disponível em: <<https://kanbanize.com/blog/guest-post-debunking-the-biggest-myths-about-kanban/>> Acesso em 05 abr. 2018

PORTO, Fabiano. **Aplicativos mobile: definições, história e previsões.** Disponível em: <<http://tectriadebrasil.com.br/blog/mercado-de-midias-sociais-blog/aplicativos-mobile-definicoes-historia-e-previsoes/>> Acesso em 15 abr. 2018.

PROJECT BUILDER. **Veja como a matriz de rastreabilidade de requisitos pode ajudá-lo nos processos de gerenciamento.** Disponível em <<https://www.projectbuilder.com.br/blog/veja-como-a-matriz-de-rastreabilidade-de-requisitos-pode-ajuda-lo-nos-processos-de-gerenciamento/>> Acesso em 03 set. 2018.

PROJECT MANAGEMENT INSTITUTE, Inc. **Um Guia do Conhecimento do Gerenciamento de Projetos (GUIA PMBOK®).** 4. ed. Newtown Square, Pennsylvania: PMI, 2009.

RAMOS, Marcelo Veronez; BRANCALHÃO, Bruno Nascimento. **Repositório da Aplicação do OTA APP no Github contendo sua documentação e seu desenvolvimento.** Disponível em: <<https://github.com/MarceloVeronez/TCC>>

RODRIGUES, Joes. **Modelo Entidade Relacionamento (MER) e Diagrama Entidade-Relacionamento (DER).** Disponível em <<https://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332>> Acesso em 23 ago. 2018.

ROGATTO, Antônio. **Para que serve o Android no celular?** Disponível em: <<https://www.telefonescelulares.com.br/para-que-serve-android-celular/>> Acesso em 15 abr. 2018.

SANTOS, Guilherme. **Node.js—O que é, por que usar e primeiros passos.** Disponível em: <<https://medium.com/thdesenvolvedores/node-js-o-que-%C3%A9-por-que-usar-e-primeiros-passos-1118f771b889>> Acesso em 17 abr. 2018.

SOMMERVILLE, Ian. **Engenharia de Software:** 8. ed. Rio de Janeiro: Prentice-Hall. 2008.

SPINA, Cassio. **5 pontos básicos para entender a startup enxuta.** Disponível em: <<https://exame.abril.com.br/pme/5-pontos-basicos-para-entender-o-lean-startup/>> Acesso em 06 jun. 2018.

UDACITY. **Entenda o que é um modelo Canvas.** Disponível em: <<https://br.udacity.com/blog/post/modelo-canvas>> Acesso em 06 jun. 2018.

UNASP. **Organize e otimize seus estudos com a técnica japonesa do quadro Kanban.** Disponível em: <<https://www.unasp.br/blog/organize-seus-estudos-com-a-tecnica-do-quadro-kanban/>> Acesso em 18 set. 2018.

VANTOLL, TJ. **What is angular?** Disponível em: <<https://developer.telerik.com/topics/web-development/what-is-angular/>> Acesso em 17: abr. 2018.

VENTURA, Plínio. **O que é Regra de Negócio?** Disponível em: <<https://www.ateomomento.com.br/o-que-e-regra-de-negocio/>> Acesso em 15 ago. 2018.