

**CENTRO UNIVERSITÁRIO DE ANÁPOLIS – UniEVANGÉLICA  
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO**

**INSTASK  
COMPARTILHAMENTO DE IMAGENS**

**MARCO AURELIO DE LIMA ALVES  
MARCOS NATÃ QUEIROZ DOS SANTOS**

**ANÁPOLIS  
2018-02**

**MARCO AURELIO DE LIMA ALVES  
MARCOS NATÃ QUEIROZ DOS SANTOS**

**INSTASK  
COMPARTILHAMENTO DE IMAGENS**

Trabalho de Conclusão de Curso II apresentado como requisito parcial para a obtenção de grau do curso de Bacharelado em Engenharia de Computação do Centro Universitário de Anápolis – UniEVANGÉLICA.

Orientador(a): Prof. Kleber Silvestre Diogo

**Anápolis  
2018-02**

**MARCO AURELIO DE LIMA ALVES**  
**MARCOS NATÁ QUEIROZ DOS SANTOS**

**INSTASK - COMPARTILHAMENTO DE IMAGENS**

Trabalho de Conclusão de Curso II apresentado como requisito parcial para a obtenção de grau do curso de Bacharelado em Engenharia de Computação do Centro Universitário de Anápolis – UniEVANGÉLICA.

Aprovado(a) pela banca examinadora em 11 de Dezembro de 2018, composta por:



---

Kleber Silvestre Diogo  
Presidente da Banca



---

Alexandre Moraes Tannus  
Prof(a). Convidado(a)



---

Viviane Carla Batista Pocivi  
Prof(a). Convidado(a)

## Resumo

Os acadêmicos possuem o hábito de registrarem os conteúdos passados pelos professores na lousa como fotografias armazenadas em seus smartphones ao invés de registrarem no caderno pela comodidade e agilidade que esta atitude proporciona. Como os *smartphones* são usados para fins pessoais, acadêmicos e profissionais tais registros se perdem em meio as demais fotografias dificultando a procura pelas mesmas, muitas vezes foram excluídas sem perceber ou os acadêmicos não se recordam que registraram o conteúdo pois não possuem nenhum lembrete informando aos mesmos de realizarem o estudo logo após concluírem o registro. Partindo deste pressuposto desenvolveu-se uma aplicação *mobile* para que os acadêmicos armazenem e compartilhem de forma simples e ágil o conteúdo acadêmico através de fotografias. Para que o objetivo dessa pesquisa fosse alcançado o primeiro passo realizado foi analisar as queixas e dificuldades enfrentadas pelos acadêmicos. Posteriormente o desenvolvimento de artefatos como modelagem dos diagramas da UML juntamente com a prototipação da aplicação e especificação dos requisitos direcionaram os trabalhos. A próxima etapa foi a implementação das funcionalidades e validação da aplicação no ambiente acadêmico para analisar a aceitação e os pontos a serem reavaliados para que o software esteja apto a ser utilizados pelos estudantes.

**Palavras-Chave:** Acadêmicos. Smartphones. Fotografias. Aplicativo.

## **Abstract**

Academics have a habit of recording the content passed by teachers on the board as photographs stored on their smartphones rather than recorded in the notebook for the convenience and agility this attitude provides. Because smartphones are used for personal, academic and professional purposes such records are lost in the midst of other photographs, making it difficult to search for them, they have often been excluded without realizing it or the academics do not remember that they registered the content because they have no reminder informing them to carry out the study as soon as they have completed the registry. Starting from this assumption a mobile application was developed so that the academic ones store and share of simple and agile form the academic content through photographs. In order to achieve the goal of this research the first step was to analyze the complaints and difficulties faced by the academics. Subsequently, the development of artifacts such as modeling UML diagrams along with application prototyping and requirements specification guided the work. The next step was to implement the functionalities and validation of the application in the academic environment to analyze the acceptance and the points to be reevaluated so that the software is able to be used by the students.

**Keywords:** Academics. Smartphones. Photos. App.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Modelo Incremental .....	13
Figura 2 – Ciclo de Vida da Atividade .....	19
Figura 3 – Fluxograma de Atividades .....	22
Figura 4 – Diagrama de Casos de Uso .....	23
Figura 5 – Adicionar Projeto Firebase .....	25
Figura 6 – Serviços e Planos .....	25
Figura 7 – Arquivo de Integração JSON.....	26
Figura 8 – Telas de Autenticação .....	27
Figura 9 – Tela de Perfil.....	28
Figura 10 – Tela de Seleção de Disciplinas .....	29
Figura 11 – Tela Principal .....	30
Figura 12 – Tela de Postagem .....	31
Figura 13 – Base de Dados .....	32
Figura 14 – Classe User.....	32
Figura 15 – Nó Users na Base de Dados.....	33
Figura 16 – Busca de Usuários.....	33
Figura 17 – Storage.....	34
Figura 18 – Aceitação de Usuários .....	35
Figura 19 - Protótipo de Login.....	42
Figura 20 - Protótipo de Menu Lateral.....	43
Figura 21 - Protótipo de Perfil De Usuário .....	44
Figura 22 - Protótipo –Captura Dos Conteúdos .....	45
Figura 23 - Protótipo –Realizando Post.....	46

## **LISTA DE TABELAS**

Tabela 1 – Descrição dos Requisitos não Funcionais .....	22
Tabela 2 – Descrição dos Casos de Uso.....	24
Tabela 3 – Descrição dos Casos de Uso Futuros .....	24
Tabela 4 – Priorização dos Requisitos .....	41

## LISTA DE ABREVIATURAS E SIGLAS

TIC's	Tecnologias de Informação e Comunicação
UC	<i>Use Case</i> /Casos de Uso
UML	<i>Unified Modeling Language</i>
V&V	Verificação e Validação
JSON	JavaScript Object Notation

## Sumário

1 INTRODUÇÃO .....	10
2 FUNDAMENTAÇÃO TEÓRICA .....	12
2.1 Engenharia de software .....	12
2.1.1 Processo de desenvolvimento de software .....	12
2.1.2 Processo incremental .....	12
2.2 Engenharia de requisitos .....	13
2.2.1 Levantamento de requisitos.....	14
2.2.2 Requisitos funcionais .....	14
2.2.3 Requisitos não funcionais.....	15
2.3 UML .....	15
2.3.1 Diagrama de atividades .....	15
2.3.2 Diagrama de casos de uso .....	16
2.3.3 Diagrama de classes .....	16
2.4 Metodologia scrum.....	16
2.5 Verificação e validação .....	17
2.6 Taiga.....	18
2.7 Android .....	18
2.8 Firebase .....	19
2.9 GIT .....	20
3 ANÁLISE/DESENVOLVIMENTO .....	21
3.1 Integração Android/Firebase .....	24
3.2 Implementação de Requisitos.....	26
3.3 Banco de Dados.....	31
3.4 Storage e Ferramentas .....	34
3.5 Verificação e Validação .....	35
4 CONSIDERAÇÕES FINAIS/CONCLUSÃO .....	37
5 REFERÊNCIAS BIBLIOGRÁFICAS .....	38
APÊNDICE A – Priorização dos requisitos e protótipos .....	41
APÊNDICE B – Documento de visão.....	47
APÊNDICE C – Histórias de usuários .....	53

## 1 Introdução

Segundo (LIMA JUNIOR, 2012) a sociedade contemporânea absorveu nas últimas décadas as Tecnologias de Informação e Comunicação (TIC's) em todos os seus segmentos. Os segmentos acadêmicos também estão neste contexto.

Notou-se atualmente o uso de tecnologia no ambiente acadêmico como retrata (VEEN; WRAKING, 2009, p.175): "[...] a escola, sendo meio para sistematização da educação e da aprendizagem, deve atender aos interesses e às necessidades da sociedade entre esses: a necessidade de acompanhar ou preconizar as tecnologias inerentes aos avanços da sociedade".

O uso de smartphones em sala de aula é cada vez mais comum. Alguns utilizam como passa tempo, outros para estudo e grupos de pesquisa. Segundo (PALFREY, 2011), os alunos são nativos digitais, ou seja, já nasceram em contato com as tecnologias digitais e se sentem bem mais à vontade com elas do que seus antecessores.

Verificou-se no dia a dia que muitos discentes fazem o uso de aparelhos *smartphones* para tirar fotos de tarefas, exercícios, trabalhos e da lousa. Entretanto o uso desta tecnologia gera a praticidade e rapidez no acesso e armazenamento do conteúdo quando comparado com o método convencional de escrita utilizando caneta e papel, porém esta mesma tecnologia é utilizada para fins pessoais, estudantis e profissionais, deste modo as informações se misturam no armazenamento dos *smartphones* o que dificulta o acesso e muitas vezes são perdidas na galeria de fotos, pois não há uma organização lógica. Diante da situação apresentada como então, armazenar tais fotografias de modo que seu acesso posteriormente seja simplificado?

É empírico o uso de *smartphones* para salvar imagens de exercícios, trabalhos, atividades e avaliações no cotidiano acadêmico. Entretanto é uma queixa muito comum dos estudantes dizer que não encontrou, perdeu ou até mesmo não se recordava daquela imagem/registro que havia realizado do conteúdo em seu smartphone. Com isso analisou-se especificamente essa queixa dos estudantes e desenvolveu-se esta proposta para facilitar o arquivamento destas imagens, onde o usuário possa organizá-las de forma lógica e tenha a facilidade de compartilhar e obter mais conteúdos relacionados as suas respectivas disciplinas.

A proposta do INSTASK armazena e compartilha com outros usuários as imagens tiradas de conteúdo acadêmico. A aplicação também emitirá lembretes aos usuários para que acessem o novo conteúdo armazenado de modo que auxilie no estudo.

Através deste trabalho desenvolveu-se uma aplicação mobile para arquivar e compartilhar as fotografias do conteúdo acadêmico em um ambiente propício e organizado. Entretanto, para que fosse possível, fez-se necessário objetivos específicos como o levantamento dos requisitos e desenvolvimento dos artefatos necessários, a priorização dos requisitos levantados e planejamento das entregas no processo incremental, a implementação de todos os requisitos essenciais levantados e documentos e como forma de garantir a qualidade da aplicação realização da verificação da aplicação.

## **2 Fundamentação Teórica**

### **2.1 Engenharia de software**

A Engenharia de Software surgiu como solução para a crise de software enfrentada em meados dos anos 2000 aonde os softwares possuíam inúmeros defeito, custos elevados e entregas fora do prazo.

Segundo (PRESSMAN, 2011), a engenharia de software permite que profissionais desenvolvam softwares de qualidade com apoio de ferramentas, métodos e processos bem definidos. Tais parâmetros direcionam o desenvolvimento de soluções de software com maior qualidade. É importante ressaltar que a Engenharia de Software utiliza dos mesmos princípios da uma disciplina de Engenharia tradicional, no qual se baseia na relação custo/benefício do produto oferecido e na sua eficiência.

#### **2.1.1 Processo de desenvolvimento de software**

Para que um trabalho tenha êxito é necessário um padrão, um caminho a ser seguido, esta lógica também se aplica no processo de desenvolvimento de software. Segundo (PRESSMAN, 2011) o processo de desenvolvimento de software é o conjunto de atividades, ações e tarefas necessária para desenvolver um software de qualidade. É importante seguir passos previsíveis na elaboração de um sistema ou produto que garanta um resultado de alta qualidade. Esses passos que devem ser seguidos são denominados de processo de software

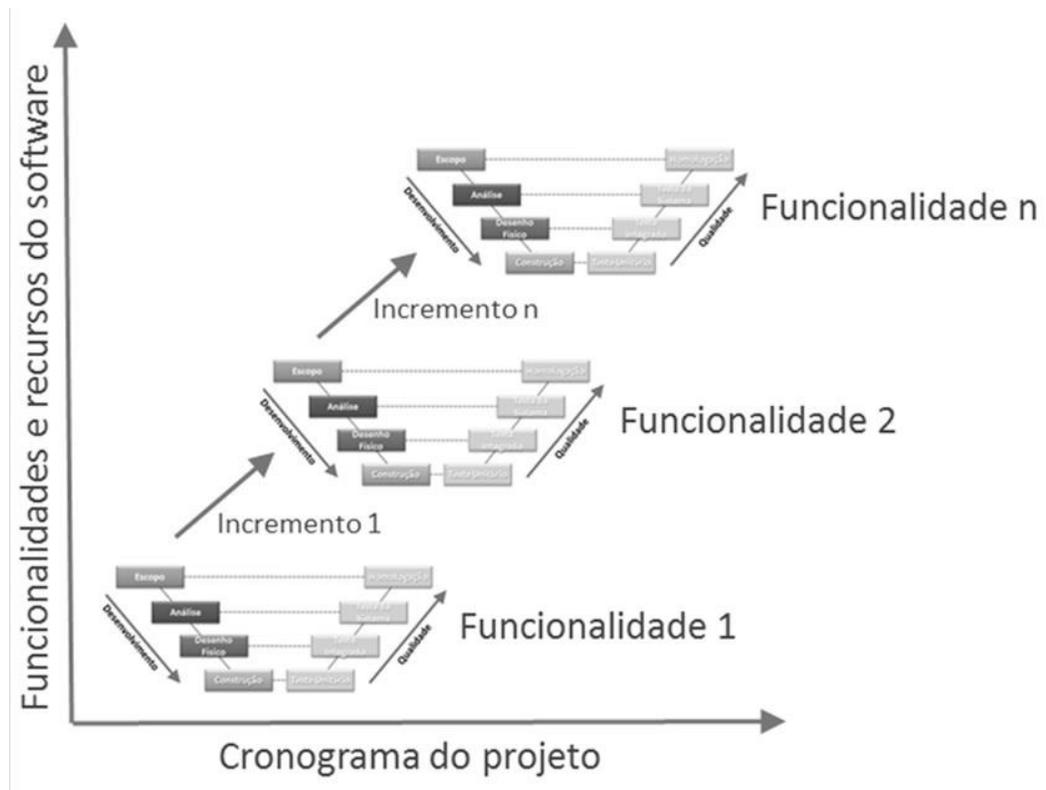
Segundo (HIRAMA, 2011), a importância dos métodos de desenvolvimento de software advém do fato de que eles definem, por meio de suas notações, um canal de comunicação uniforme entre os membros da equipe de desenvolvimento. Também, os métodos estabelecem produtos de trabalho padronizados que facilitam as atividades de manutenção de software. Em um projeto de software existem muitos *stakeholders* envolvidos, além dos envolvidos como clientes existem entre outros, gerentes, analistas, arquitetos, programadores e testadores. Para que haja uma eficácia nas atividades de desenvolvimento é necessário existir uma excelente comunicação entre os profissionais, tal comunicação advém de métodos definidos em processos de desenvolvimentos de software, podendo ser uma abordagem chamada Estruturada e a outra Orientada a Objetos.

#### **2.1.2 Processo incremental**

O modelo incremental é utilizado quando há necessidade de fortalecimento de algumas funcionalidades para o cliente e após este fortalecimento faz-se necessário expandir e melhorar tais funcionalidades em novas versões do software. Na utilização do método

incremental geralmente, a primeira entrega é um produto essencial, como pode ser visto na figura 1. Isto significa que os requisitos básicos foram atendidos, porém ainda existem outras funcionalidades complementares algumas conhecidas e outras desconhecidas que ainda não foram entregues (PRESSMAN, 2011).

**Figura 1: Modelo Incremental.**



Fonte: (PRESSMAN, 2011).

Após a validação e feedback do cliente é desenvolvido o planejamento do próximo incremento. Esse planejamento considera as modificações encontradas no produto essencial para adequar as necessidades do cliente e a adição de novas funcionalidades. Após cada incremento esse processo é repetido até que o produto completo seja produzido (PRESSMAN, 2011).

## 2.2 Engenharia de requisitos

De acordo com (PRESSMAN, 2011) para entender a necessidade do cliente a Engenharia de Requisitos fornece mecanismos apropriados para analisar a viabilidade, negociar uma solução razoável, especificar uma solução sem ambiguidade, validar a especificação e gerenciar essas necessidades à medida que são transformadas em software.

A Engenharia de Requisitos é o processo de desenvolvimento de uma especificação de software. As especificações destinam-se a comunicar as necessidades de sistema dos clientes para os desenvolvedores do sistema (SOMMERVILLE, 2011).

### **2.2.1 Levantamento de requisitos**

A resolução de problemas, elaboração, negociação e especificação são os elementos do levantamento de requisitos. Os interessados identificam o problema, propõe elementos da solução, negociam diferentes abordagens e especificam um conjunto de requisitos preliminares da solução (PRESSMAN, 2011). O alicerce de todo desenvolvimento de software é o levantamento de requisitos, pois nesta etapa o objetivo é procurar solucionar da forma mais otimizada possível o desafio enfrentado.

O levantamento de requisitos objetiva determinar as funcionalidades do software, ou seja, o que o software deve fazer. Durante esta etapa o engenheiro de software busca compreender a “dor” do cliente e o que ele deseja que o software faça. Para que tais informações sejam levantadas utiliza-se técnicas como por exemplo entrevistas nas quais o engenheiro tenta compreender como funciona o processo atual a ser informatizado e posteriormente as funcionalidades/serviços que o cliente necessita que o software forneça. Nesta etapa é comum a alteração e/ou inserção de novos requisitos e deste modo segundo (SOMMERVILLE, 2011) “...para seguir um padrão corretamente documentado qualquer alteração ou implantação de uma nova funcionalidade deve ser registrada na documentação para facilitar a detecção de erros e a manutenção do sistema.”.

### **2.2.2 Requisitos funcionais**

Define as funcionalidades que o sistema terá, considerando as entradas e saídas de dados, comportamento com entradas específicas e muitas vezes descreve o que o sistema não irá disponibilizar. Os requisitos funcionais vão ser construídos de acordo com o tipo de software que está sendo desenvolvido, quem são os usuários do sistema e qual a política de negócio adotada pela empresa (SOMMERVILLE, 2011).

Segundo (VAZQUEZ, 2016), o comportamento esperado pelo software e descrito em um requisito funcional se refere ao intercâmbio de informações entre o usuário, o software sendo descrito e os meios de armazenamento até que um objetivo específico seja alcançado. Esse objetivo específico é concluir a tarefa sob sua responsabilidade de tal maneira que seus resultados possam ser usados como insumos em outras tarefas por usuários com outras responsabilidades ou em outros momentos – seja por um usuário com as mesmas responsabilidades ou não.

### 2.2.3 Requisitos não funcionais

Define as limitações em que o sistema irá desempenhar considerando o sistema como um todo. Algumas dessas limitações podem ser restrições de tempo de resposta, no processo de desenvolvimento do produto, normas impostas pela empresa, limitações de orçamento, integração com outros sistemas, legislação de privacidade, etc. (SOMMERVILE, 2011).

Os requisitos não funcionais correspondem a restrições, condições, consistências e validações que devem ser levadas a efeito sobre os requisitos funcionais. Alguns requisitos não funcionais identificam regras de negócio, ou seja, políticas, normas e condições estabelecidas pela empresa que devem ser seguidas na execução de uma funcionalidade. (GUEDES, 2011).

## 2.3 UML

O principal objetivo da *Unified Modeling Language* é fornecer diferentes visões do sistema, modelando o sistema em diferentes aspectos, através dos recursos oferecidos pelos complementos dos diagramas. Quando se utiliza diversos diagramas é possível detectar falhas, o que diminui a ocorrência de erros futuros (GUEDES, 2011).

O UML não é uma linguagem de programação, e sim uma linguagem de modelagem, uma notação. Segundo (PRESSMAN, 2011) a *Unified Modeling Language* é uma linguagem padrão utilizada para descrever ou documentar projeto de software. Artefatos de um sistema de software podem ser visualizados, especificados, construídos e documentados através dos diagramas da UML. Esses diagramas facilitam o trabalho dos desenvolvedores pois ajudam no entendimento e na especificação do sistema. Diante disto permite-se destacar que a UML não é um processo de desenvolvimento de software, pode ser utilizada por diferentes processos de desenvolvimento adequando de acordo com a necessidade do engenheiro responsável.

### 2.3.1 Diagrama de atividades

O objetivo do diagrama de atividades é exibir as atividades e fluxo de controle referentes a um processo de software. Segundo (GUEDES, 2011), o diagrama de atividade preocupa-se em descrever os passos a serem percorridos para a conclusão de uma atividade específica, podendo esta ser representado por um método com certo grau de complexidade, um algoritmo, ou menos um processo completo.

Os diagramas de atividades são destinados a mostrar as atividades que compõem um processo de sistema e o fluxo de controle entre as atividades.

### 2.3.2 Diagrama de casos de uso

Segundo (GUEDES, 2011), o diagrama de casos de uso procura possibilitar a compreensão do comportamento externo do sistema (em termos de funcionalidades oferecidas por ele) por qualquer pessoa com algum conhecimento sobre o problema enfocado, tentando apresentar o sistema por intermédio de uma perspectiva dos usuários. Esse diagrama costuma ser utilizado, sobretudo, no início da modelagem do sistema, principalmente nas etapas de elicitação e análise de requisitos, embora venha a ser consultado e possivelmente modificado durante todo os processos de engenharia e sirva de base para a modelagem de outros diagramas.

Esse diagrama costuma ser utilizado em reuniões iniciais com o cliente como forma de apresentar o comportamento do sistema, facilitar a compreensão dos usuários, detectar possíveis falhas na especificação e verificar se os requisitos foram entendidos (GUEDES, 2011).

### 2.3.3 Diagrama de classe

O diagrama de classes é um dos mais importantes e utilizados da UML. Seu principal enfoque está em permitir a visualização das classes que comporão o sistema com seus respectivos atributos e métodos, bem como em demonstrar como as classes do diagrama se relacionam, complementam e transmitem informações entre si. Esse diagrama apresenta uma visão estática de como as classes estão organizadas preocupando-se em como definir a estrutura lógica delas. O diagrama de classes serve ainda como apoio para a construção da maioria dos outros diagramas da linguagem UML (GUEDES, 2011).

## 2.4 Metodologia scrum

Scrum segundo (SUTHERLAND, 2014) é uma metodologia ágil de desenvolvimento de software iterativa que tem como base a divisão do projeto em ciclos chamados de *sprints*. As funcionalidades a serem implementadas são listadas no *Product Backlog* e posteriormente a reunião, conhecida como *Sprint Planning Meeting*, as funcionalidades são priorizadas e transferidas para *Sprint Backlog*. Diariamente é realizado o *Daily Scrum*, uma reunião com objetivo de verificar o andamento da *Sprint*. Ao final da *Sprint* é realizada a *Sprint Review Meeting* e finalmente uma *Sprint Retrospective* e assim é planejada a próxima *Sprint*.

Para aprimorar o conhecimento referente da capacidade do *Scrum* (DIMES, 2014) informa que ainda que o *Scrum* seja geralmente usado para desenvolvimento de software, muito de seus conceitos poderiam ser usados para outros tipos de produtos ou mesmo

trabalho. O *Scrum* leva a tendência humana em consideração e cria fluxos de trabalhos muitos sustentáveis.

O time *Scrum* é composto pelo *Product Owner*, o Time de Desenvolvimento e o *Scrum Master*. Times *Scrum* são auto-organizáveis e multifuncionais. Times auto-organizáveis escolhem qual a melhor forma para completarem seu trabalho, em vez de serem dirigidos por outros de fora do time. Times multifuncionais possuem todas as competências necessárias para completar o trabalho sem depender de outros que não fazem parte da equipe. O modelo de time no *Scrum* é projetado para aperfeiçoar a flexibilidade, criatividade e produtividade (SCHWABER E SUTHERLAND, 2013).

O ponto chave do *Scrum* é a *Sprint*, um período de um mês ou menos, aonde são destinados todos os esforços da equipe no desenvolvimento de uma versão incremental potencialmente utilizável do produto. As durações das *sprints* são analisadas de acordo com o esforço de desenvolvimento relacionado com a prioridade das versões entregáveis aonde são iniciadas uma nova *sprint* imediatamente após a conclusão da *sprint* anterior.

## **2.5 Verificação e validação**

Segundo (PRESSMAN, 2011) o teste de software é um elemento de um tópico mais amplo, muitas vezes conhecido como verificação e validação (V&V). Verificação refere-se ao conjunto de tarefas que garantem que o software implementa corretamente uma função específica. Validação refere-se a um conjunto de tarefas que asseguram que o software foi criado e pode ser rastreado segundo os requisitos do cliente.

O Teste Funcional ou teste de Caixa Preta de acordo com (INTHURN, 2001), consiste em verificar na interface do software se todas as funcionalidades estão operando corretamente.

Segundo (PRESSMAN, 2011) o teste de regressão é realizado nas funcionalidades que já estão implementadas, para que seja verificada se não houve introdução de falhas nas funcionalidades já codificadas.

Durante o desenvolvimento do software é bastante importante o foco na qualidade do mesmo e para tal o é necessário realizar o teste de aceitação com o cliente para analisar o nível de qualidade desejado. Segundo (SOMMERVILLE, 2011) isso nem sempre é possível, já que o cliente está envolvido com o negócio e tem pouco tempo para apoiar esta atividade.

Segundo (RIOS, 2003) o teste de compatibilidade valida a capacidade do software de executar em um particular ambiente de hardware/software/sistema operacional/rede etc.

Segundo (PRESSMAN, 2011) a Verificação e Validação incluem uma ampla gama de atividade de SQA (software quality assurance, garantia da qualidade de software): revisões técnicas, auditorias de qualidade e configuração, monitoramento de desempenho, simulação, estudo de viabilidade, revisão de documentação, revisão e base de dados, análise de algoritmo, teste de desenvolvimento, teste de usabilidade, teste de qualificação, teste de aceitação e teste de instalação. Embora a aplicação de teste tenha um papel extremamente importante em V&V, muitas outras atividades também são necessárias.

## 2.6 Taiga

Segundo a empresa Taiga Agile LLC o Taiga é uma plataforma de gerenciamento livre e de código aberto com o objetivo de auxiliar times ágeis de desenvolvimento de software. Taiga.io suporta metodologias ágeis que fazem times mais responsivos e softwares melhores e mais rápidos. Essa ferramenta foi escrita em Python e construída sobre os frameworks Django e AngularJS.

A plataforma disponibiliza um *Backlog* onde pode ser informado as histórias de usuários ou casos de usos. Sendo possível realizar a priorização dos requisitos e adicioná-los a Sprints preparando os entregáveis. Para rastreabilidade do andamento do projeto pode-se usar o *Kanban* ligando cada caso de uso aos seus devidos estágios durante o projeto. Outra funcionalidade disponível são os *Epics* usado para gerenciar e rastrear histórias de usuários ou casos de usos de múltiplos projetos relacionados entre si.

## 2.7 Android

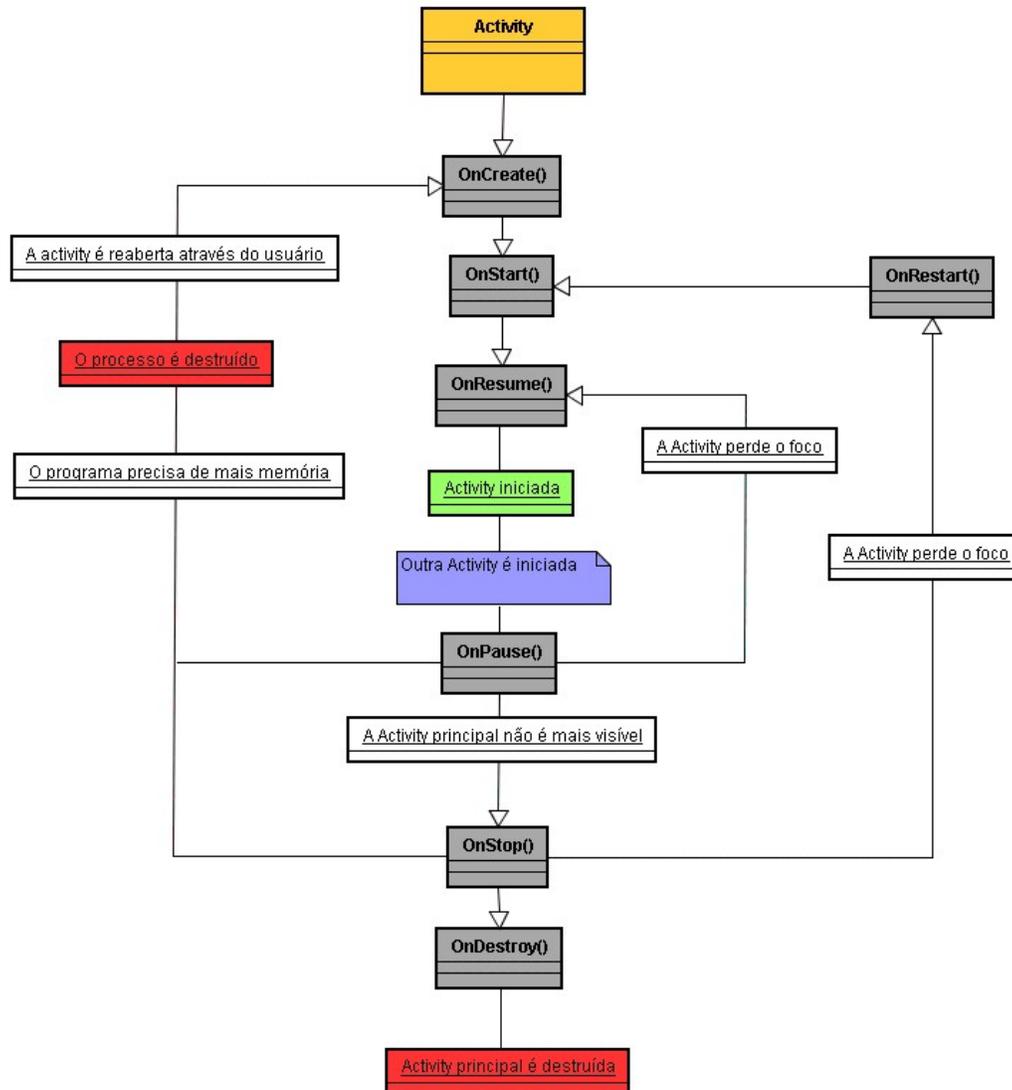
Segundo a empresa Google LLC o Android é um sistema operacional baseado no núcleo do Linux desenvolvido pela própria empresa. O Android é projetado principalmente para dispositivos móveis com tela sensível ao toque como smartphones, podendo ser usados também em TV, carros e relógios de pulso. O sistema operacional faz uso da tela sensível ao toque e de um teclado virtual.

A plataforma incorporou todas as características intrínsecas do sistema Linux como sistema de arquivos, o *kernel*, os servidores de terminais, *framework* de aplicação permitindo reuso de componentes, navegador web integrado, suporte para mídias de áudio, vídeo e imagens, suporte para armazenamento de dados e ambiente de desenvolvimento.

A arquitetura das aplicações Android são baseadas em atividades como ilustrado na figura 2. As atividades são os componentes de aplicativos que fornece uma tela com a qual os usuários podem interagir. Os aplicativos normalmente têm várias atividades pouco

vinculadas em entre si. Uma atividade pode chamar outras atividades com objetivo de executar diferentes tarefas. A arquitetura se utiliza de estados para controlar as mudanças de atividades utilizando uma pilha para guardar as atividades deixadas em segundo plano e posteriormente podendo ser restauradas pelos usuários.

Figura 2 – Ciclo de Vida da Atividade Android.



Fonte: <https://www.devmedia.com.br/entendendo-o-ciclo-de-vida-de-uma-aplicacao-android/22922>.

Acesso em 20 de maio de 2018.

## 2.8 Firebase

Segundo a empresa Google LLC o *Firebase* é uma plataforma de desenvolvimento de aplicações web e aplicações móveis desenvolvida pela empresa *Firebase* e adquirida pela

Google. A plataforma disponibiliza vários serviços que auxiliam no desenvolvimento e expansão de produtos, sendo os principais, banco de dados em tempo real, autenticação com suporte a provedores, cloud storage para subir e baixar arquivos.

A plataforma disponibiliza ferramentas para integração dos serviços para Android, iOS, Web, C++ e Unity fornecendo guias, referências, exemplos e bibliotecas para adequada integração das ferramentas.

O banco de dados em tempo real possibilita armazenar e sincronizar dados entre usuários e dispositivos em tempo real através de um banco de dados *NoSQL* hospedado na nuvem. Oferece uma disponibilidade dos dados independentemente da conectividade de rede.

O sistema métodos, incluindo e-mail, senha e provedores de terceiros, como Google, Facebook ou seu próprio sistema de contas. A ferramenta também disponibiliza uma interface de código aberto e totalmente personalizável.

O *cloud storage* oferece uma forma simples de armazenar e compartilhar o conteúdo gerado pelos usuários, como imagens, áudio e vídeo. A ferramenta utiliza-se da segurança do Google para subir e baixar arquivos nos aplicativos do Firebase, independentemente da qualidade da rede.

A plataforma tem planos de pagamento baseados no escalonamento do uso, sendo assim é possível começar a utilizar a plataforma de forma totalmente grátis e conforme seu negócio crescer pagar taxas baseadas no consumo da aplicação.

## 2.9 GIT

Criado em 2005 por Linus Torvalds (também criador do Linux), depois de um desentendimento com os criadores do BitKeeper (ferramenta terceirizada que versionava o kernel do linux). O Git é um sistema de versionamento de código baseado no BitKeeper que tem por características principais: a velocidade, o design simplificado, o suporte a desenvolvimento não linear, ser totalmente distribuído e consegue versionar também grandes projetos eficientemente. (Software Freedom Conservancy, 2011).

### 3 Análise/Desenvolvimento

Esse capítulo apresentará os passos realizados para o desenvolvimento do software utilizando como base o modelo incremental.

Inicialmente será apresentado como foi levantado os requisitos do aplicativo, em seguida será apresentado os artefatos que foram gerados após a definição e validação dos requisitos, por fim, será apresentado como foi realizada a implementação da aplicação.

Os requisitos foram levantados ouvindo os relatos/queixas dos discentes. Os mesmos nos informaram o quanto era comum esquecer de olhar as fotos do conteúdo e principalmente de perder em meio as fotos pessoais armazenadas no smartphone.

Diante destes relatos mais recorrentes e outros notou-se que o aplicativo deveria de alguma maneira ser um ambiente específico para armazenamento dessas fotos dos conteúdos de simples e bastante organizada além de enviar lembretes de estudos das fotos postadas. Com esta percepção em mãos elaborou-se o diagrama de casos de uso para visualizar as funcionalidades e seu(s) respectivo(s) ator(es).

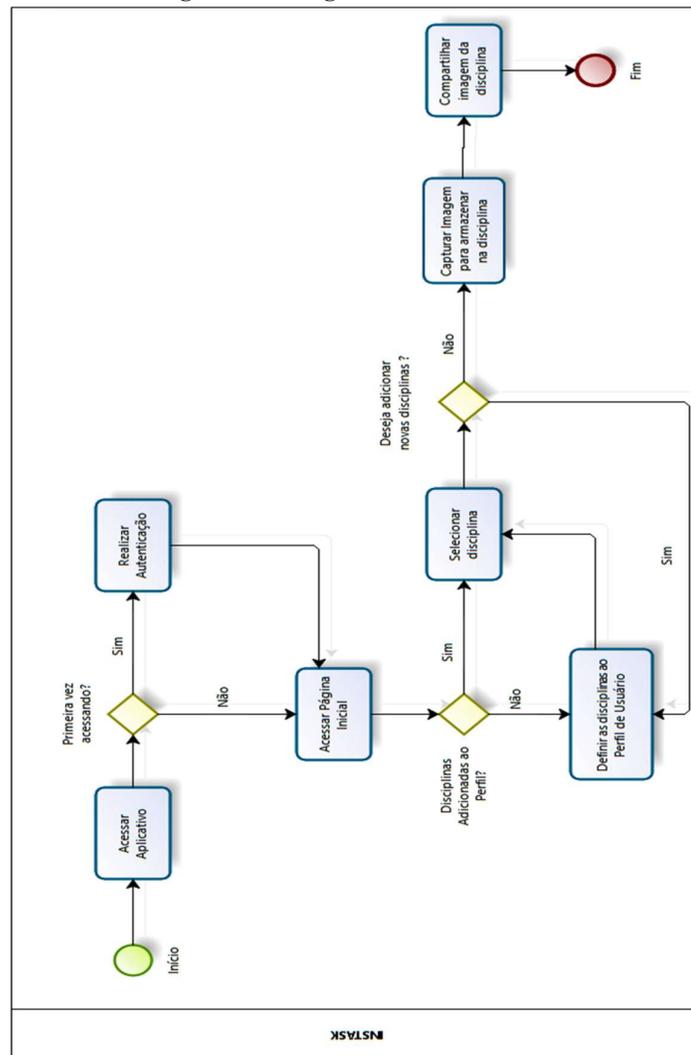
A próxima iteração foi através do fluxograma de atividades aonde visualizou-se as ações a serem realizadas dentro do aplicativo. Em seguida os protótipos para representar a ajustar a realização do fluxo definido, botões e demais funcionalidades. Durante o desenvolvimento dos protótipos ao analisar com alguns discentes notou-se a necessidade de facilitar o entendimento das funcionalidades do aplicativo de modo que fosse mais intuitivo utilizar suas respectivas funcionalidades. Aderimos as sugestões dos discentes e reavaliamos os protótipos.

Após esta etapa realizou-se a especificação informal dos requisitos funcionais e não funcionais, documento de visão e histórias de usuários com o objetivo de registrar e manter uma coerência entre as funcionalidades do início ao fim da implementação da aplicação. Partindo deste princípio a implementação foi realizada na plataforma *Android* utilizando serviços de infraestrutura do Google por meio da plataforma do *Firebase*, que disponibiliza serviço de banco de dados onde serão armazenados todos os dados dos usuários, serviço de *storage* no qual utilizaremos para armazenar as imagens capturadas, serviço de autenticação para segurança de acesso as informações, serviço de monetização que foi utilizado com anúncios do Google.

A seguir será apresentado os principais artefatos produtos para que o processo de desenvolvimento do software fosse alcançado. Fez-se necessário o desenvolvimento do

fluxograma de atividades para construir as ações dentro a aplicação de forma coerente e lógica. As principais ações realizadas no aplicativo estão representados na figura 3

**Figura 3: Fluxograma de Atividades.**



**Fonte: Autores da Pesquisa**

Definiu-se os requisitos não funcionais do aplicativo como o intuito de limitar os comportamentos da aplicação objetivando apenas duas características essenciais. Os principais requisitos não funcionais do sistema estão descritos na tabela 1.

**Tabela 1- Descrição dos Requisitos Não Funcionais**

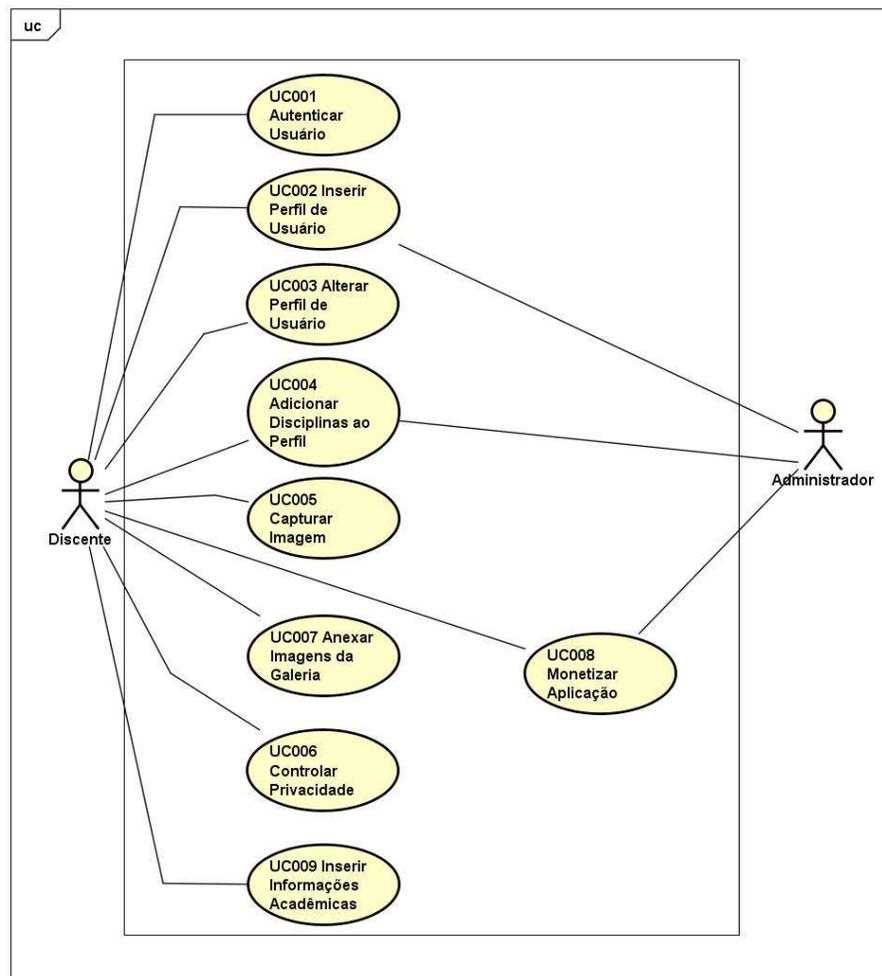
Requisitos Não Funcionais	Descrição
Usabilidade	O aplicativo deve oferecer facilidade para a utilização e memorização dos recursos oferecidos. A tela deve ser

	objetiva e oferecer poucos cliques para que o usuário acesse sua funcionalidade desejada.
Recuperabilidade	Em caso de falhas o aplicativo será capaz de recuperar os dados.

Fonte: Autores da Pesquisa

O diagrama de caso de uso do sistema representado na figura 4 apresenta as funcionalidades do aplicativo juntamente com seus atores (perfil de usuário que terá acesso a(s) respectiva(s) funcionalidade(s)). As descrições informais dos casos de uso estão definidas na tabela 2 e tabela 3 respectivamente e o documento de visão e histórias de usuários estão nos Apêndices B e C respectivamente.

Figura 4: Diagrama de Casos de Uso.



powered by Astah

Fonte: Autores da Pesquisa

**Tabela 2 – Descrição dos Casos de Uso**

Caso de Uso 01 – Autenticar Usuário (descrição informal)
Os usuários poderão realizar autenticação no aplicativo de duas maneiras. Primeira opção utilizando conta do Google. Segunda opção utilizando o e-mail como <i>login</i> , aonde caso seja necessário poderá redefinir a senha para acessar o aplicativo
Caso de Uso 02 – Inserir Perfil de Usuário (descrição informal)
Os usuários ao autenticarem pela primeira vez no aplicativo serão obrigados a definir suas informações de Perfil de Usuário como por exemplo: Universidade, Curso, Período. Deste modo as demais funcionalidades dependentes destas informações estarão disponíveis.
Caso de Uso 03 – Alterar Perfil de Usuário (descrição informal)
Os usuários poderão alterar suas informações de perfil sempre que necessário. Caso altere suas informações acadêmicas todos os registros respectivos ao curso previamente definido serão visualmente alterados para os registros referentes ao novo curso definido.
Caso de Uso 04 – Adicionar Disciplinas ao Perfil (descrição informal)
Os usuários após definirem o tipo do curso desejado serão disponibilizadas todas as opções de disciplinas cadastradas na base de dados referentes ao curso especificado aonde serão adicionadas ao perfil do usuário a partir do seu critério.
Caso de Uso 05 – Capturar Imagem (descrição informal)
Os usuários ao acessarem as matérias desejadas poderão realizar a captura dos conteúdos passados em sala de aula para servir de estudo posteriormente tanto pessoal quanto para demais usuários que compartilham desta mesma matéria.

Fonte: Autores da Pesquisa

**Tabela 3 – Descrição dos Casos de Uso Futuros**

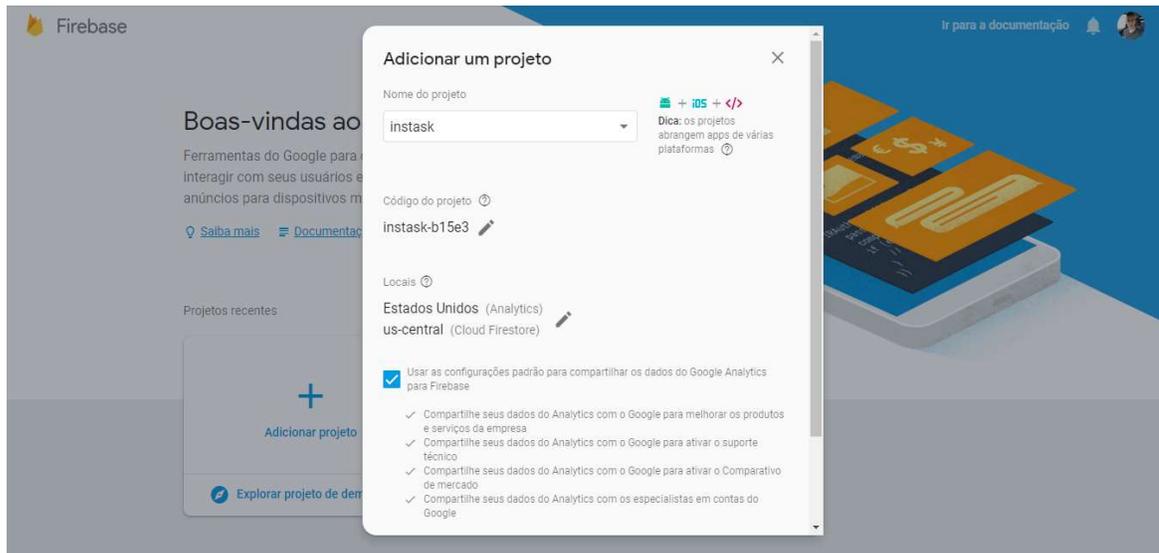
Caso de Uso 06 – Controlar Privacidade (descrição informal)
Os usuários poderão optar por não permitir que o conteúdo seja público, deste modo, apenas o mesmo terá acesso.
Caso de Uso 07 – Anexar Imagens da Galeria (descrição informal)
Os usuários poderão anexar conteúdos originados da galeria de seus smartphones.
Caso de Uso 08 – Monetizar Aplicação (descrição informal)
A monetização da aplicação será através de anúncios segmentados utilizando a ferramenta <i>AdMob</i> do <i>google</i> .

Fonte: Autores da Pesquisa

### 3.1 Integração Android/Firebase

A arquitetura da aplicação se baseia na integração da plataforma Android com os serviços do Firebase. Para realizar integração o processo é bastante simples, foi necessário apenas criar um projeto na plataforma conforme demonstra a figura 5.

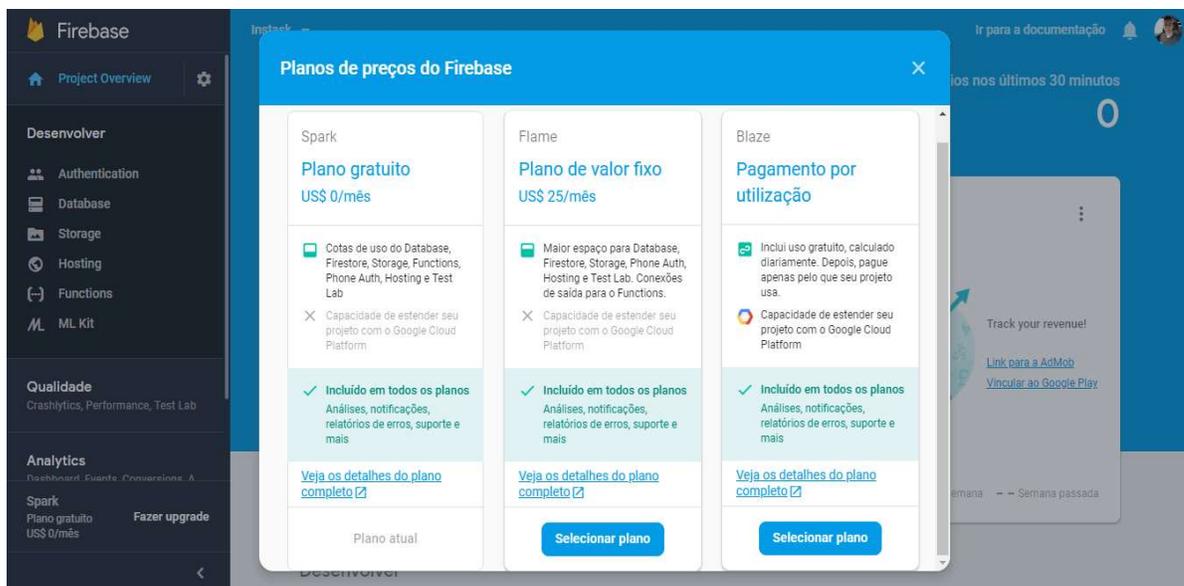
**Figura 5: Adicionar Projeto Firebase**



**Fonte: Autores da Pesquisa.**

Ao adicionar o projeto a plataforma já é possível ter acesso a todos os serviços disponível sem pagar taxa alguma, mas sempre deve ser respeitado os limites do plano. Para a aplicação o plano gratuito atendeu inicialmente conforme figura 6. É importante também ter ciência que os serviços do Firebase têm como pré-requisito um dispositivo com Android 4.0 (Ice Cream Sandwich) ou uma versão mais recente e o Google Play Services 15.0.0 ou posterior.

**Figura 6: Serviços e Planos**



**Fonte: Autores da Pesquisa.**

Depois de integrado um arquivo JSON é baixado para seu projeto no Android com todas informações para acesso aos serviços conforme demonstra a figura 7.

**Figura 7: Arquivo de Integração JSON**

```

1 [{"project_info": {
2   "project_number": "422503297579",
3   "firebase_url": "https://instask-d52dc.firebaseio.com",
4   "project_id": "instask-d52dc",
5   "storage_bucket": "instask-d52dc.appspot.com"
6 },
7
8 "client": [
9   {
10    "client_info": {
11     "mobilesdk_app_id": "1:422503297579:android:ccae4b699881bf50",
12     "android_client_info": {
13      "package_name": "br.com.instask.instask"
14     }
15    },
16    "oauth_client": [
17     {
18      "client_id": "422503297579-08331b6p5decpn70is5gntik39k55pa4.apps.googleusercontent.com",
19      "client_type": 1,
20      "android_info": {
21       "package_name": "br.com.instask.instask",
22       "certificate_hash": "ddd011a93268518f4727adc4c1c40cc534a0c592"
23      }
24     },
25     {
26      "client_id": "422503297579-mm751u7f1o4r6mgtutcl1662j8tkp0g3.apps.googleusercontent.com",
27      "client_type": 1,
28      "android_info": {
29       "package_name": "br.com.instask.instask",
30       "certificate_hash": "fb10ba11f295b130fcb83a8bd21af9cca3d8d472"
31      }
32     }
33    ]
34   }
35 ]

```

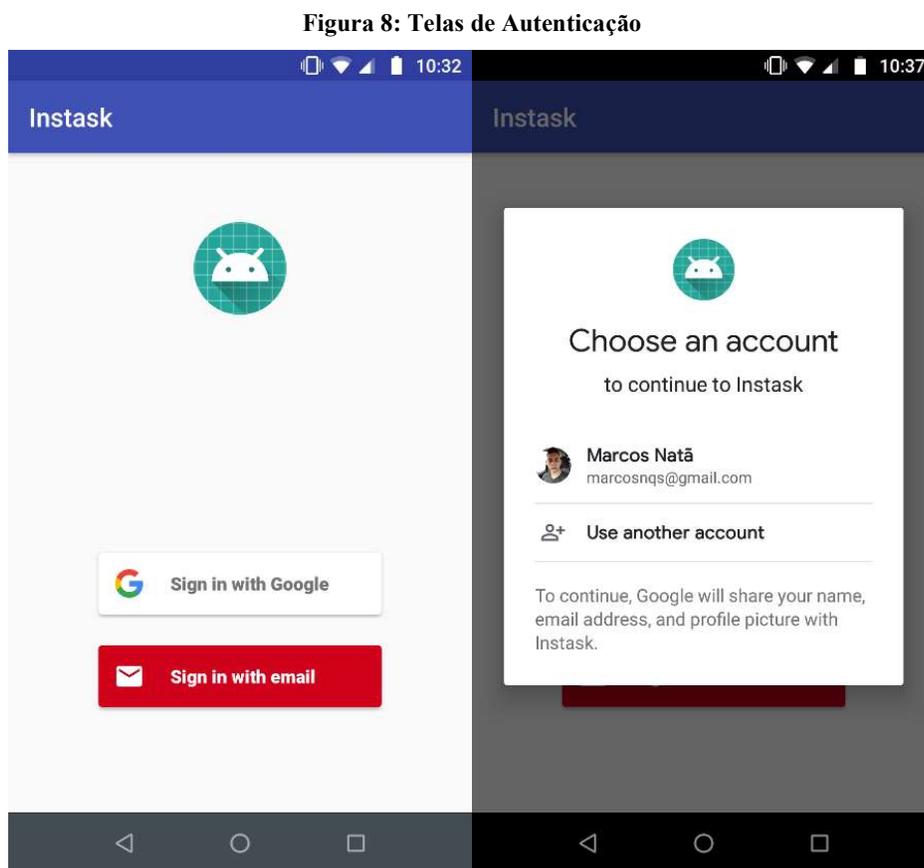
**Fonte: Autores da Pesquisa.**

No processo de configuração da arquitetura e dependências do projeto houve um contratempo numa das atualizações da plataforma de desenvolvimento do Android, onde a mudança na estrutura das aplicações fez com que o gerenciador não conseguisse mais construir a aplicação. Assim foi necessário retrabalhar as atividades já criadas para se adequar no novo padrão.

### 3.2 Implementação de Requisitos

Integração configurada partimos para implementação dos requisitos. No desenvolvimento da autenticação dos usuários ativamos na plataforma do Firebase os provedores por e-mail/senha e acesso por meio da conta do Google. Na plataforma é possível realizar todas customizações de e-mails para verificação de endereço de e-mail, redefinição

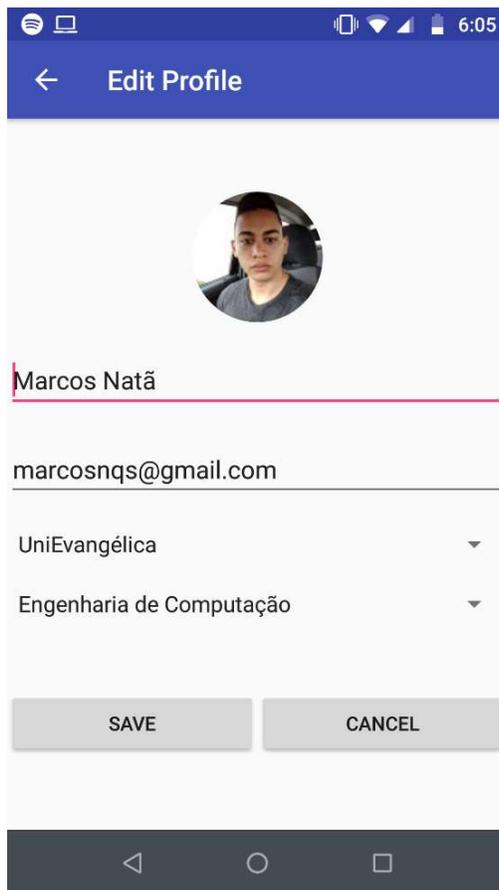
de senha, alteração de endereço de e-mail e gerenciar os usuários autenticados. No desenvolvimento da tela de *login* foi utilizado uma biblioteca do Firebase chamada *FirebaseUI* que disponibiliza uma variedade de modelos de autenticação que podem ser facilmente implementadas e personalizadas conforme demonstra a figura 8.



**Fonte: Autores da Pesquisa**

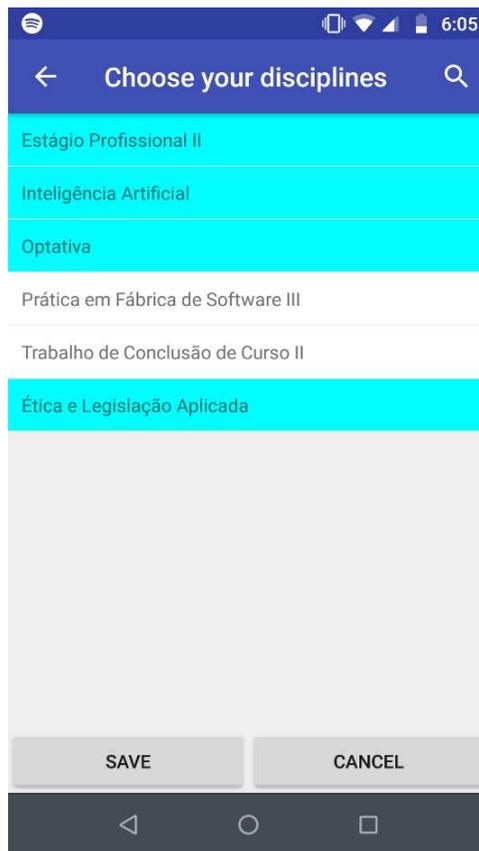
Após o processo de autenticação foi desenvolvido o cadastro do perfil do usuário. Logo após a primeira autenticação o usuário é levado a informar nome, e-mail, instituição e curso. No processo de autenticação a aplicação já carrega imagem, nome e e-mail na tela perfil, dados que vem do provedor, seja autenticação por e-mail ou pela conta do Google. Depois também é possível alterar essas informações editando o perfil. Todos estes dados são gravados diretamente no banco do Firebase na nuvem. A figura 9 apresenta a tela de perfil.

Figura 9: Tela de Perfil



Fonte: Autores da Pesquisa

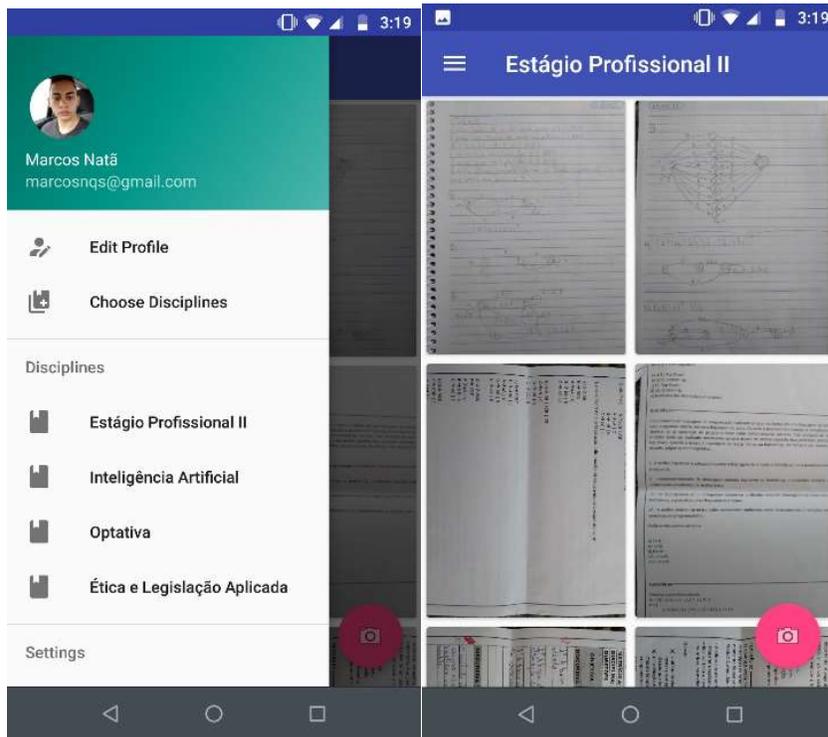
Perfil cadastrado seguimos para implementação da seleção de disciplinas. Nesta tela é possível selecionar as disciplinas as quais o usuário está cursando, independente do período do curso. As disciplinas apresentadas são baseadas no cadastro de instituição e curso no perfil do usuário. Foi também adicionado a opção de pesquisar, pois o número de disciplinas disponíveis pode ser grande. É importante salientar que todos os cadastros de instituições, cursos e disciplinas são de responsabilidade do administrador, sendo este processo realizado de forma manual no banco de dados. A figura 10 apresenta a tela de seleção de disciplinas.

**Figura 10: Tela Seleção de Disciplinas**

**Fonte: Autores da Pesquisa**

Desta forma com perfil cadastrado e disciplinas selecionadas o usuário pode navegar pelo aplicativo podendo visualizar os artefatos de cada disciplina. Buscando uma melhor usabilidade foi criado um menu lateral, também chamado de “gaveta de navegação”, onde é possível realizar todas ações principais como alterar de uma disciplina para outra. A gaveta de navegação para as disciplinas foi criada de forma dinâmica onde a apresentação das disciplinas depende das seleções do usuário, assim qualquer inclusão ou alteração nas disciplinas influenciará na disposição na gaveta de navegação. O usuário também pode contribuir postando mais fotos, pode alterar as disciplinas ou perfil a qualquer momento apenas acessando a gaveta de navegação, caso o usuário necessite trocar de conta isso possível ser realizada apenas desconectando na sessão de configurações na gaveta. A figura 11 apresenta a tela principal do aplicativo.

Figura 11: Tela Principal

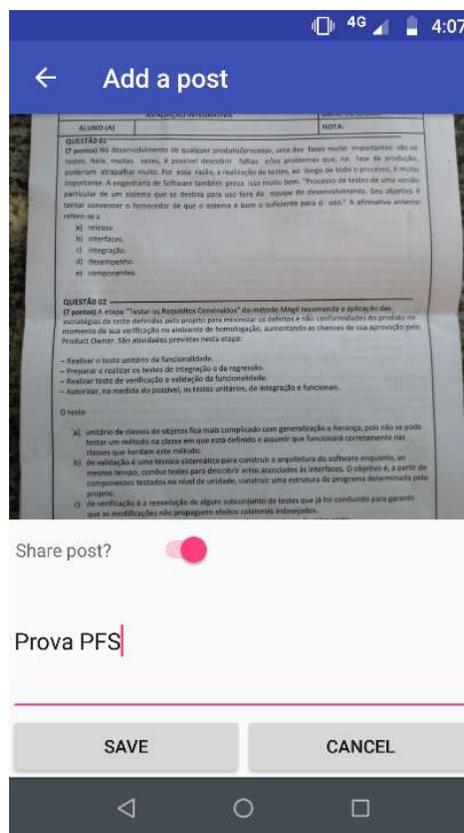


Fonte: Autores da Pesquisa

Para realizar a postagem ou captura de imagem foi criado um botão flutuante para realizar a chamada da câmera. Ao realizar a captura o usuário poderá adicionar um comentário a imagem e decidir se quer compartilhar com outros usuários. Se a opção compartilhar for selecionada o aplicativo vai subir a imagem para o *storage* do Firebase e assim estará disponível para todos, caso não, a imagem ficará apenas localmente no aparelho do usuário. Na implementação da galeria de imagens, inicialmente, a equipe tentou desenvolver utilizando apenas o *storage* na nuvem, sem utilizar um *storage* local para renderização das imagens. Entretanto nos deparamos com uma dificuldade técnica, pois as imagens retiradas pelo *smartphone* criam arquivos com tamanhos entre 3 e 6 *megabytes*. Então essa situação inviabilizou essa abordagem de desenvolvimento, pois tanto para subir quanto para baixar esse arquivo do *storage* na nuvem seria necessária uma alta banda de internet. Desse modo partimos para uma segunda abordagem que seria carregar as imagens do *storage* localmente, criando um serviço para subir e baixar as postagens em segundo plano. Nessa abordagem foi necessário realizar um estudo para realizar a compressão das imagens sem perder muita qualidade, pois estamos tratando de imagens onde o usuário precisa de um nível de qualidade adequado para conseguir ler as informações na imagem.

Então para conseguirmos minimizar o uso de internet é realizado uma compressão da imagem para posteriormente subir para *storage*. Também nos deparamos outros problemas técnicos como quando usuário fecha o aplicativo ou muda do INSTASK para outro aplicativo, como posteriormente continuar a subir ou baixar as imagens que estavam em processo anteriormente? Para conseguir estes comportamentos na aplicação foi necessário implementações específicas utilizando a biblioteca de *Cloud Storage* do Firebase. A figura 12 apresenta a tela de postagem.

Figura 12: Tela de Postagem



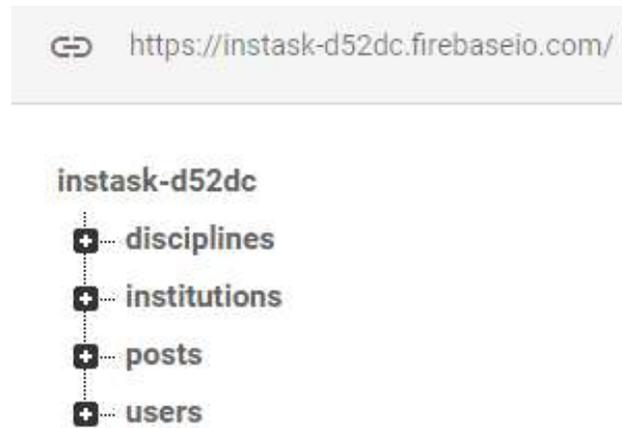
Fonte: Autores da Pesquisa

### 3.3 Banco de Dados

A base de dados utilizada na aplicação foi o banco de dados em tempo real do Firebase. A estrutura do banco é baseada numa árvore JSON onde cada tabela é na verdade um nó da árvore. Esse banco tem características diferentes de um banco relacional sendo por padrão desnormalizado para conseguir alcançar um maior desempenho nas buscas e ser mais

escalável. O Firebase também disponibiliza um *script* para tratamento das regras de segurança onde é possível assegurar que cada usuário tenha o acesso devido. A regra de segurança básica aplicada no projeto é que para o usuário acessar qualquer informação deve estar corretamente autenticado. A seguir na figura 13 apresentamos os principais nós da aplicação.

**Figura 13: Base de Dados**



**Fonte: Autores da Pesquisa**

O Firebase disponibiliza uma biblioteca específica para manipulação da base de dados prática e ágil. O banco de dados aceita uma variedade de tipos de dados, na aplicação mapeamos as tabelas por meio de objetos java. Nas figuras a seguir nota-se uma parte da aplicação que realiza uma busca de usuários na base de dados.

**Figura 14: Classe User**

```
public class User {

    public String imageUrl;
    public String name;
    public String email;
    public String institution;
    public String course;
    public List<String> disciplinesList;

    public User(){}

    public User(String imageUrl, String name, String email, String institution, String course) {

        if (name.isEmpty() || email.isEmpty()) {
            throw new IllegalArgumentException("Name/Email not informed");
        }

        this.imageUrl = imageUrl;
        this.name = name;
        this.email = email;
        this.institution = institution;
        this.course = course;
    }
}
```

**Fonte: Autores da Pesquisa**

Figura 15: Nó Users na Base de Dados



Fonte: Autores da Pesquisa

Figura 16: Busca de Usuários

```

public void populateUpdateUser(String uid) {
    mDatabase.getReference("users").child(uid)
        .addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                User user = dataSnapshot.getValue(User.class);
                mDisciplinesList = user.disciplinesList;
                displayUserForUpdate(user);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
}

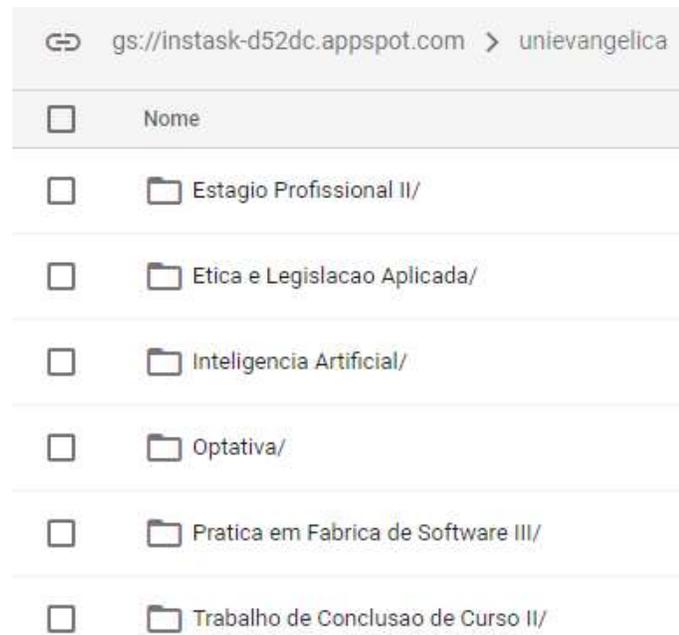
```

Fonte: Autores da Pesquisa

### 3.4 Storage e Ferramentas

O armazenamento das imagens foi implementado utilizando o storage do Firebase que disponibiliza um serviço para subir e baixar as imagens da nuvem. Utilizando o Firebase é possível tratar os processamentos em segundo plano sem comprometer a usabilidade da aplicação. A estrutura de pastas usada foi bem simples, onde cada instituição tem uma pasta que contém todas as disciplinas oferecidas. A figura 17 apresenta a estrutura do storage.

**Figura 17: Storage**



**Fonte: Autores da Pesquisa**

Para gerenciar todas as dependências do projeto como as bibliotecas do Firebase, bibliotecas para manipulação de imagem foi utilizada a ferramenta Gradle que vem integrada com Android Studio bastando apenas declarar a dependência no arquivo de configuração que a ferramenta realiza todo processo de baixar e disponibilizar a biblioteca. O Gradle também realiza todo processo de construção da aplicação para o Android. No versionamento foi utilizado a ferramenta Git juntamente com repositório remoto no GitLab.

### 3.5 Verificação e Validação

Ao término de cada funcionalidade realizou-se o Teste Funcional para certificar que as ações dentro da aplicação estavam operando corretamente e as inconsistências encontradas foram analisadas, corrigidas e posteriormente revisadas.

Após finalizar a implementação de todas as funcionalidades essenciais da aplicação realizou-se o Teste de Regressão para certificar que não houve avarias nas funcionalidades anteriormente implementadas após a adesão de novas e o resultado deste teste permitiu verificar na prática que todo o processo de arquitetura bem planejado resultou no aumento da consistência da aplicação.

Posteriormente realizou-se o Teste de Compatibilidade para verificar diferenças entre os artefatos desenvolvidos e a aplicação finalizada e vice-versa, deste como as inconsistências encontradas foram analisadas e atualizadas nos artefatos.

Com o aplicativo em mãos realizou-se o Teste de Aceitação com os futuros usuários (acadêmicos) simulando sua utilização em seu ambiente real, ou seja, em sala de aula para obter dos mesmos um feedback com sugestões e/ou melhorias mais apurado. Na figura abaixo nota-se a aceitação do aplicativo.

**Figura 18: Aceitação dos Usuários**



**Fonte: Autores da Pesquisa**

A realização da Verificação e Validação do aplicativo contribuiu para elevar a qualidade e confiabilidade do aplicativo, garantindo assim que os discentes possam utilizá-lo tal qual foi projetado.

#### **4 Considerações Finais/Conclusão**

Quando se vivencia o meio acadêmico observa-se atitudes de forma “automática” ou seja, em determinadas circunstâncias não são analisadas certos comportamentos e através de um estudo e observação das principais atitudes cometidas e principalmente dificuldades nos estudos do conteúdo desde então elaborou-se a ideia do desenvolvimento deste aplicativo para criar um ambiente característico aos estudantes para armazenar as fotos do conteúdo ministrado pelo docente de modo simples, mais atrativo e mais do que capaz de armazenar mas que agregasse nos estudos dos discentes.

A utilização da tecnologia mobile foi definida através de uma observação simples, analisando rapidamente o comportamento dos discente (e também pessoas no geral) nota-se o uso constante do smartphone para ações de lazer, profissional e acadêmicas, e a partir de então decidiu-se unir os dois mundos.

Os discentes verificaram que as funcionalidades do aplicativo permitem solucionar os principais problemas ao tirar fotos dos conteúdos e principalmente a praticidade da sua utilização despertou nos mesmos interesses em utilizá-lo.

Ao elaborar e desenvolver os artefatos necessários do processo de desenvolvimento de software permitiu-se enxergar de forma mais clara e objetiva cada funcionalidade, principalmente o potencial da aplicação e com a criação dos protótipos do aplicativo os discentes também abstraíram da mesma percepção mais concreta auxiliando em dicas e principalmente aceitação do projeto.

Ao finalizar a implementação das funcionalidades realizou-se a validação com os discentes aonde os mesmos puderam usar o aplicativo e certificar na versão final que o mesmo é bastante prático e útil no dia a dia acadêmico para estudos e controle das fotos dos conteúdos das disciplinas. Eventuais correções foram necessárias, porém já foram solucionadas e retornadas para continuação dos testes de aceitação.

O desenvolvimento deste aplicativo do levantamento dos requisitos e regras até a implementação e testes agregaram uma experiência prática bastante importante para os autores deste trabalho de como que habilidades pessoais e principalmente profissionais foram colocadas em prova e novas foram adquiridas para superar desafios vivenciados durante os processos, tal experiência tornou-se bastante gratificante.

## 5 Referências Bibliográficas

DIMES, TROY. **Scrum Essencial**. Canadá: Babelcube Inc, 2014.

GOOGLE LLC. **Google firebase**. Disponível em: <<https://firebase.google.com/>>. Acesso em 20 de abril de 2018.

GOOGLE LLC. **Android para Desenvolvedores**. Disponível em: <<https://developer.android.com/>>. Acesso em 12 de maio de 2018.

GUEDES, Gilleanes. UML 2. **Uma Abordagem Prática**. 2ª Edição. Novatec Editora. 2011

INTHURN, Cândida. **Qualidade & teste de software**. Florianópolis: Visual Books, 2011

LIMA JUNIOR, W. T. **Big Data, Jornalismo Computacional e Data Journalism: estrutura, pensamento e prática profissional na Web de dados**. Disponível em: <<http://www.ec.ubi.pt/ec/12/pdf/EC12-2012Dez-11.pdf>>. Acesso em 17 de fevereiro de 2018.

PALFREY, Jhon, GASSER, Uyr. **Nativos, Imigrantes e Colonizadores digitais: Nascidos na era digital: entendendo a primeira geração de nativos digitais**. Tradução de Magda França Lopes. Porto Alegre: Artmed, 2011.

PRESSMAN, Roger S. **Engenharia de Software: uma abordagem profissional**. 7ed. São Paulo: Pearson Makron Books, 2011.

RIOS, Emerson; MOREIRA, Trayahú R. F.. “Projeto & Engenharia de Software – Testes de Software”. - Ed. Atlas Book, Rio de Janeiro, 2003.

SOFTWARE FREEDOM CONSERVANCY. Git: About. 2011. Disponível em: <<https://git-scm.com/about>>. Acesso em 12 de novembro de 2018.

SOMMERVILLE, Ian. **Engenharia de Software**. 9ed. São Paulo: Pearson Prentice Hall, 2011.

SCHWABER, K.; SUTHERLAND J. **Um guia definitivo para o scrum: as regras do jogo**. Scrum Inc, 2013. Disponível em: <<https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-PortugueseBR.pdf>>. Acesso em 20 de abril de 2018

SUTHERLAND, JEFF. **Scrum: a arte de fazer o dobro do trabalho na metade do tempo**. 2014.

VAZQUEZ, Carlos Eduardo. **Engenharia de Requisitos: software orientado ao negócio**/ Carlos Eduardo Vazquez; Guilherme Siqueira Simões – Rio de Janeiro: Brasport, 2016.

VEEN, Win; WRAKKING, Bem. **Homo Zappiens**: educando na Era Digital. Tradução Vinícius Figueira. Porto Alegre: Artmed, 2009.

# APÊNDICES

## Apêndice A

### Priorização dos requisitos

Tabela 4 – Priorização dos Requisitos.

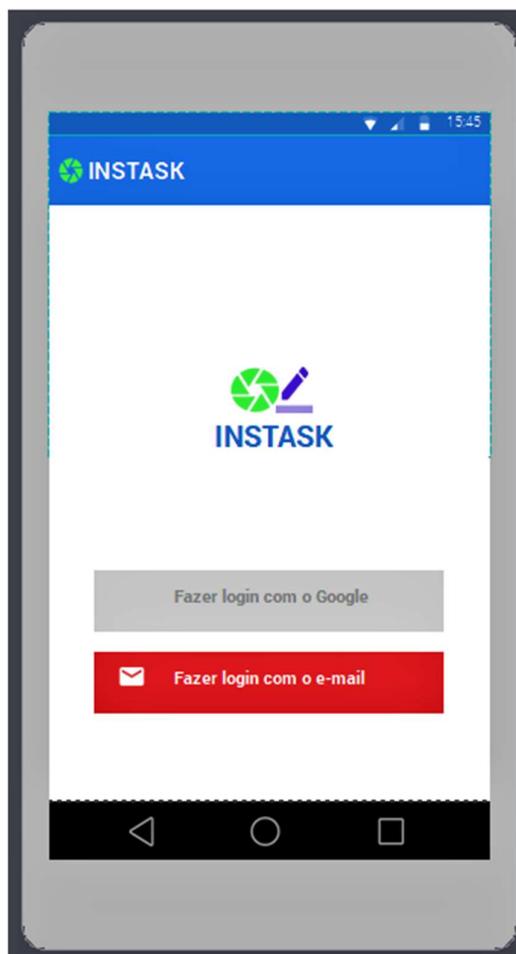
Item	Requisitos	Pré-Requisito	Status	Prioridade
RF.001	Autenticar Usuário	-	Finalizado	Essencial
RF.002	Inserir Perfil de Usuário	RF.001	Finalizado	Essencial
RF.003	Alterar Perfil de Usuário	RF.002	Finalizado	Essencial
RF.004	Adicionar Disciplinas ao Perfil	RF.003	Finalizado	Essencial
RF.005	Capturar Imagem	RF.004	Finalizado	Essencial
RF.006	Controlar Privacidade	-	Finalizado	Desejável
RF.007	Anexar Imagens da Galeria	-	A realizar	Desejável
RF.008	Monetizar Aplicação	-	A realizar	Desejável
RF.009	Inserir Informações Acadêmicas	-	A realizar	Desejável

Fonte: Autores da Pesquisa

## Protótipos do aplicativo

### Protótipo – login

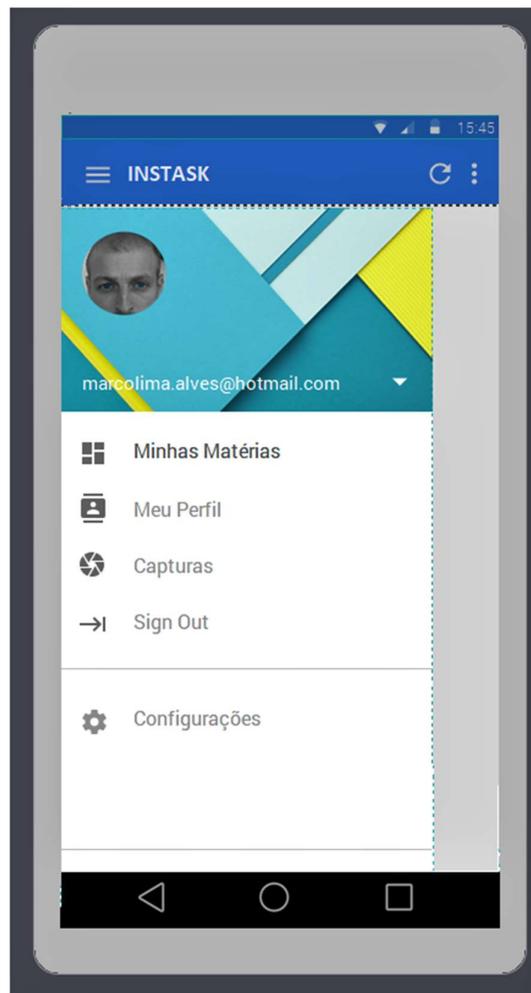
Figura 19: Protótipo de Login



Fonte: Autores da Pesquisa

## Protótipo – menu lateral

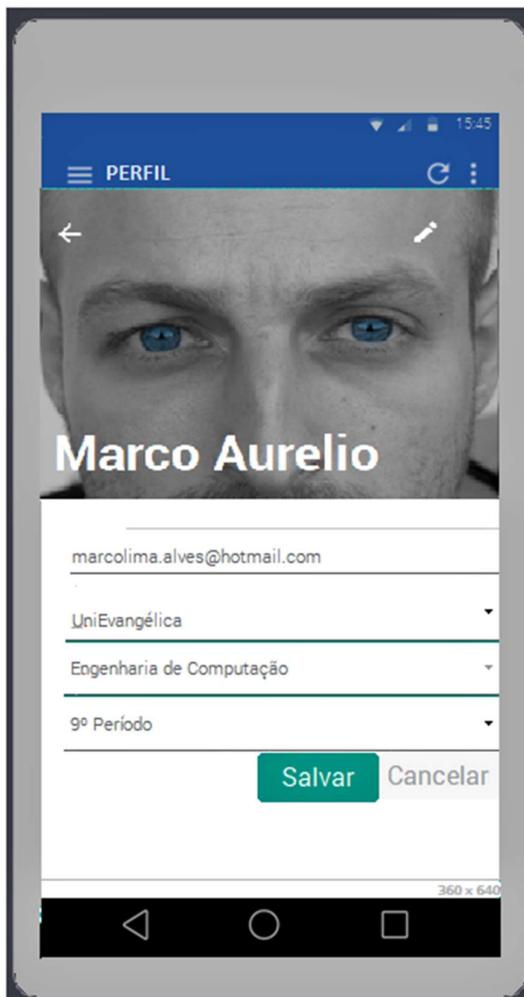
Figura 20: Protótipo Menu Lateral



Fonte: Autores da Pesquisa

## Protótipo – perfil de usuário

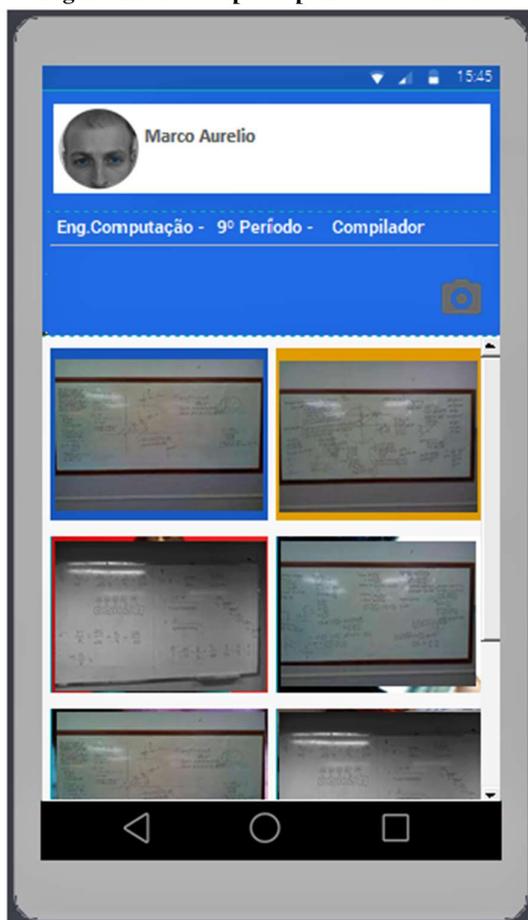
Figura 21: Protótipo Perfil de Usuário



Fonte: Autores da Pesquisa

## Protótipo – capturas dos conteúdos

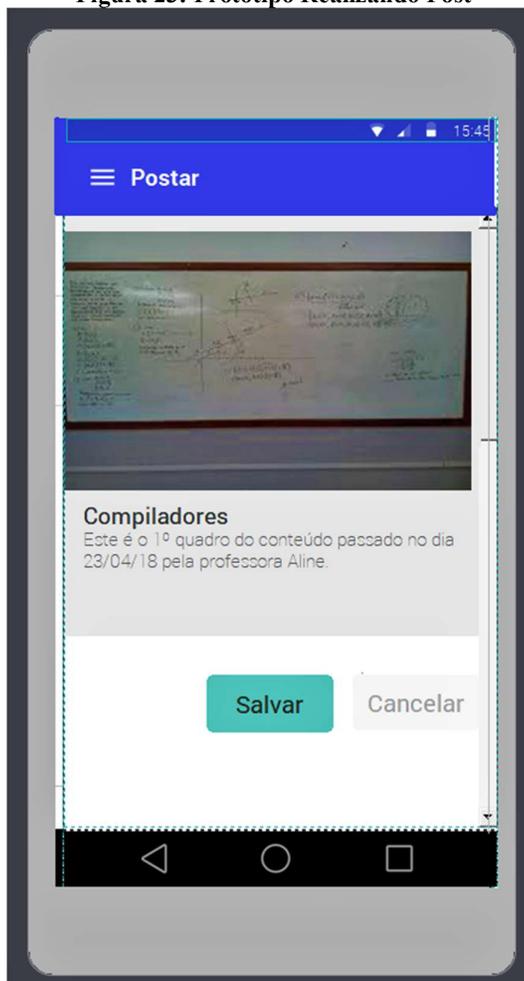
Figura 22: Protótipo Captura dos Conteúdos



Fonte: Autores da Pesquisa

## Protótipo – realizando post

Figura 23: Protótipo Realizando Post



Fonte: Autores da Pesquisa

## Apêndice B- DOCUMENTO DE VISÃO

---

### FÁBRICA DE TECNOLOGIAS TURING

---



## DOCUMENTO DE VISÃO

INSTASK – COMPARTILHAMENTO DE IMAGENS

**VERSÃO: 0.2**

Autores:

Marco Aurelio de Lima Alves

Marcos Natã Queiroz dos Santos

Anápolis – GO

2018

**HISTÓRICO DE REVISÃO**

<b>Versão</b>	<b>Data</b>	<b>Responsável</b>	<b>Descrição</b>
0.1	27/08/2018	Marco Aurelio de Lima Alves; Marcos Natã Q. dos Santos	Início do Documento de Visão.
0.2	05/11/2018	Marco Aurelio de Lima Alves	Revisão do Documento

**SUMÁRIO**

1.	STAKEHOLDERS .....	50
2.	VISÃO DO PRODUTO .....	50
3.	ABREVIATURAS, CONVENÇÕES E SIGLAS .....	50
3.1	Prioridade dos Requisitos .....	50
4.	REQUISITOS FUNCIONAIS.....	51
4.1	Autenticar Usuário .....	51
4.2	Inserir Perfil de Usuário .....	51
4.3	Alterar Perfil de Usuário .....	51
4.4	Adicionar Disciplinas ao Perfil .....	51
4.5	Capturar Imagem .....	51
4.6	Controlar Privacidade .....	51
4.7	Anexar Imagens da Galeria .....	51
4.8	Monetizar Aplicação .....	51
4.9	Inserir Informações Acadêmicas .....	51
5.	REQUISITOS NÃO FUNCIONAIS .....	52

## 1. STAKEHOLDERS

JANEIRO - 2018/2		
Nome	Cargo	Responsabilidade
Marco Aurelio de Lima Alves	Analista de Requisitos e Tester	Responsável pela análise e especificação de requisitos e testes das funcionalidades implementadas
Marcos Natã Queiroz dos Santos	Analista e desenvolvedor	Responsável pela arquitetura e implementação dos requisitos

## 2. VISÃO DO PRODUTO

Para acadêmicos que possuem o hábito de registrarem os conteúdos passados pelos professores na lousa como fotografias armazenadas em seus smartphones ao invés de registrarem no caderno **que necessitam** da agilidade dos *smartphones* também para fins, acadêmicos e profissionais. **É um** aplicativo *mobile* para que os acadêmicos. **Possibilita** que armazenem e compartilhem de forma simples e ágil o conteúdo acadêmico através de fotografias. **O diferencial** do nosso produto está no levantamento do que os acadêmicos mais se queixavam para desenvolver uma aplicação sob medida para o ambiente específico. **Nosso produto** irá agilizar, lembrar e agregar no aprendizado dos acadêmicos utilizando a tecnologia como aliada neste ambiente.

## 3. ABREVIATURAS, CONVENÇÕES E SIGLAS

Todas as siglas, abreviações, convenções e estrangeirismo existem nos documentos do projeto estarão relatados na tabela abaixo.

Siglas	Descrição
Manter	Permite ao ator as ações de alterar, buscar, inserir, ativar e inativar.

### 3.1 Prioridade dos Requisitos

Para estabelecer a prioridade dos requisitos, foi adotada a técnica **MOSCOW**, onde:

- **Essencial: “MUST”**, referem-se aos requisitos que necessitam ser implementados prioritariamente, sem eles não há o funcionamento do sistema.
- **Importante: “SHOULD”**, são referidos aos requisitos que deveriam ser implementados no sistema, ele pode funcionar sem ele ainda que de forma não satisfatória.

- **Desejável: “COULD”**, são os requisitos que poderiam ser implementados no sistema, não altera o funcionamento caso não esteja implementado, o sistema pode funcionar de forma satisfatória sem ele.
- **Interessante: “WANT”**, são os requisitos que seriam interessantes ter no sistema.

#### 4. REQUISITOS FUNCIONAIS

##### 4.1 Autenticar Usuário

UC.	Descrição	Status	Prioridade
UC.001	Autenticar Usuário	Finalizado	Essencial
<b>Detalhamento:</b> Os usuários poderão realizar autenticação no aplicativo de duas maneiras. Primeira opção utilizando conta do <i>google</i> . Segunda opção utilizando o e-mail como <i>login</i> , aonde caso seja necessário poderá redefinir a senha para acessar o aplicativo			

##### 4.2 Inserir Perfil de Usuário

UC.	Descrição	Status	Prioridade
UC.002	Inserir Perfil de Usuário	Finalizado	Essencial
<b>Detalhamento:</b> Os usuários ao autenticarem pela primeira vez no aplicativo serão obrigados a definir suas informações de Perfil de Usuário como por exemplo: Universidade, Curso, Período. Deste modo as demais funcionalidades dependentes destas informações estarão disponíveis.			

##### 4.3 Alterar Perfil de Usuário

UC.	Descrição	Status	Prioridade
UC.003	Alterar Perfil de Usuário	Finalizado	Essencial
Os usuários poderão alterar suas informações de perfil sempre que necessário. Caso altere suas informações acadêmicas todos os registros respectivos ao curso previamente definido serão visualmente alterados para os registros referentes ao novo curso definido.			

##### 4.4 Adicionar Disciplinas ao Perfil

UC.	Descrição	Status	Prioridade
UC.004	Adicionar Disciplinas ao Perfil	Finalizado	Essencial
Os usuários após definirem o tipo do curso desejado serão disponibilizadas todas as opções de matérias cadastradas na base de dados referentes ao curso especificado aonde serão adicionadas ao perfil do usuário a partir do seu critério.			

##### 4.5 Capturar Imagem

UC.	Descrição	Status	Prioridade
UC.005	Capturar Imagem	Finalizado	Essencial
Os usuários ao acessarem as matérias desejadas poderão realizar a captura dos conteúdos passados em sala de aula para servir de estudo posteriormente tanto pessoal quanto para demais			

usuários que compartilham desta mesma matéria.

#### 4.6 Controlar Privacidade

UC.	Descrição	Status	Prioridade
UC.006	Controlar Privacidade	Finalizado	Desejável

Os usuários poderão optar por não permitir que o conteúdo seja público, deste modo, apenas o mesmo terá acesso.

#### 4.7 Anexar Imagens da Galeria

UC.	Descrição	Status	Prioridade
UC.007	Anexar Imagens da Galeria	A Realizar	Desejável

Os usuários poderão anexar conteúdos originados da galeria de seus smartphones.

#### 4.8 Monetizar Aplicação

UC.	Descrição	Status	Prioridade
UC.008	Monetizar Aplicação	A Realizar	Interessante

A monetização da aplicação será através de anúncios segmentados utilizando a ferramenta *AdMob* do *google*.

#### 4.9 Inserir Informações Acadêmicas

UC.	Descrição	Status	Prioridade
UC.009	Inserir Informações Acadêmicas	A Realizar	Desejável

Os próprios usuários (discentes) poderão inserir através do app sua faculdade, curso e respectivas disciplinas.

### 5. REQUISITOS NÃO FUNCIONAIS

RNF	Descrição
<b>Usabilidade</b>	O aplicativo deve oferecer facilidade para a utilização e memorização dos recursos oferecidos. A tela deve ser objetiva e oferecer poucos cliques para que o usuário acesse sua funcionalidade desejada.
<b>Recuperabilidade</b>	Em caso de falhas o aplicativo será capaz de recuperar os dados

## Apêndice C- HISTÓRIAS DE USUÁRIOS

---

### FÁBRICA DE TECNOLOGIAS TURING

---



Cursos superiores de  
**Computação**



Fábrica de Tecnologias  
Turing



# HISTÓRIAS

# DE

# USUÁRIOS

INSTASK – COMPARTILHAMENTO DE IMAGENS

**VERSÃO: 0.2**

Autores:

Marco Aurelio de Lima Alves  
Marcos Natã Queiroz dos Santos

Anápolis – GO

2018

**HISTÓRICO DE REVISÃO**

<b>Versão</b>	<b>Data</b>	<b>Responsável</b>	<b>Descrição</b>
0.1	13/10/2018	Marco Aurelio de Lima Alves; Marcos Natã Q. dos Santos	Início do Documento de Histórias de Usuários.
0.2	05/11/2018	Marco Aurelio de Lima Alves	Revisão do Documento.

## AUTENTICAÇÃO POR LOGIN E SENHA

- **Como** discente;
- **Quero** autenticar no aplicativo;
- **Para** que eu possa acessar a tela inicial;
- **Dado** que estou na tela de autenticação;
- **E** possuo um usuário registrado;
- **Quando** informo usuário;
- **E** informo senha;
- **E** clico em autenticar;
- **E** os dados estão validos
  - **Então** visualizo a tela inicial
- **Ou** o usuário ou senha estão inválidos;
  - **Então** visualizo a mensagem "usuário ou senha inválidos"

## AUTENTICAÇÃO DO USUÁRIO ATRAVÉS DO GOOGLE

- **Como** discente;
- **Quero** autenticar no aplicativo;
- **Dado** que estou na tela de autenticação;
- **E** quero logar usando minha conta do google;
- **Quando** clico em entrar com o google;
- **E** já estou logado no google;
  - **Então** visualizo o a tela inicial
- **Ou** não estou logado no google;
- **E** preencho o e-mail;
- **E** preencho senha;
  - **Então** visualizo a tela inicial;

## INSERIR PERFIL DE USUÁRIO

- **Como** discente;
- **Quero** inserir minhas informações no perfil;
- **Para** que meu perfil fique completo e atualizado;

- **Dado** que estou na tela inicial do app;
- **E** desejo inserir minhas informações de perfil;
- **Quando** realizo login pela primeira vez o app disponibiliza as informações para serem preenchidas imediatamente.
- **E** informo: E-mail, Faculdade, Curso, Período;
- **E** clico em “Salvar”;
  - **Então** a app salva as informações na base de dados.
- **Ou** alguma informação está incorreta;
  - **Então** visualizo a mensagem "Verifique as informações novamente"

## ALTERAR PERFIL DE USUÁRIO

- **Como** discente;
- **Quero** alterar minhas informações no perfil;
- **Para** que meu perfil fique atualizado;
- **Dado** que estou na tela inicial do app;
- **E** desejo alterar minhas informações de perfil;
- **Quando** seleciono o ícone do menu lateral
- **E** seleciono a opção “Alterar Perfil”;
  - **Então** o app desabilita as informações inseridas
- **E** realizo as alterações necessárias
- **E** clico em “Salvar”;
  - **Então** a app altera as informações na base de dados.
- **Ou** alguma informação está incorreta;
  - **Então** visualizo a mensagem "Verifique as informações novamente".

## ADICIONAR MATÉRIAS AO PERFIL

- **Como** discente;
- **Quero** inserir minhas informações no perfil;
- **Para** que meu perfil fique completo e atualizado;
- **Dado** que estou na tela inicial do app;

- **E** desejo inserir minhas informações de perfil;
- **Quando** seleciono o ícone do menu lateral;
- **E** seleciono a opção “Escolha as Disciplinas”
  - **Então** o app disponibiliza todas as disciplinas pertinentes ao período selecionado no perfil do discente.
- **E** seleciono as disciplinas desejadas
- **E** clico em “Salvar”;
- **Então** a app vincula das disciplinas selecionadas ao meu perfil.

## CAPTURAR IMAGEM

- **Como** discente;
- **Quero** capturar as imagens do conteúdo;
- **Para** que fique possa utilizar como estudo depois;
- **Dado** que estou na tela inicial do app;
- **E** desejo capturar novas imagens do conteúdo;
- **Quando** seleciono o ícone do meu lateral”;
- **E** seleciono a disciplina desejada;
- **E** o app me redireciona para visualizar todas as imagens do conteúdo da respectiva disciplina
- **E** seleciono o ícone “Câmera”
- **E** capturo a imagem
  - **Então** informo um comentário referente a imagem;
  - **E** seleciono “Salvar”
  - **Então** o app realiza o envio da imagem para a galeria da respectiva disciplina.
- **Ou** ocorre um erro na hora do envio;
  - **Então** visualizo a mensagem "Ocorreu um erro, tente novamente".