

Especificação funcional de aplicativo *mobile* para apoio na abertura de chamados em um ambiente de ensino informatizado

Victor Hugo Franzini da Silva¹, Carlos Roberto¹

¹Instituto Federal de São Paulo - Campus Hortolândia (IFSP)

victorhugofsl@gmail.com, carlos9.rsj@gmail.com

Abstract. *Currently at the IFSP – Hortolândia Campus, when a professor identifies that an item in a computer lab may be in trouble, he must open a call to the CTI or inform the CAE about the problem. This process demands a lot of time from the teacher, who could be using the time in class. The purpose of this work is to describe the functional specification of a mobile application that helps teachers to inform the responsible areas about problems in the items that make up a computer lab. So that the process of opening calls does not demand a lot of time from the teacher.*

Resumo. *Atualmente no IFSP – Câmpus Hortolândia quando um professor identifica que um item em um laboratório de informática possa estar com problemas ele deve abrir um chamado para a CTI ou informar a CAE sobre o problema. Esse processo demanda muito tempo do professor, que poderia estar utilizando o tempo em aula. A proposta deste trabalho é descrever a especificação funcional de um aplicativo mobile que auxilie os professores a informar as áreas responsáveis sobre problemas nos itens que compõe um laboratório de informática. Para que o processo de abertura de chamados não demande muito tempo do professor.*

1. Introdução

Os itens que compõem um ambiente de ensino informatizado necessitam de cuidados específicos de modo que garanta a disponibilidade do ambiente e a rápida manutenção quando necessário. Além disso, quando se trata de especificamente de um espaço de ensino público, os cuidados com o ambiente devem considerar as especificidades para manutenção e aquisição de novos equipamentos.

Atualmente no IFSP-HTO (Instituto Federal de São Paulo – Campus Hortolândia) quando um professor em um laboratório de informática identifica que um item possa estar com problemas ou está em falta, dependendo o item, é necessário fazer a abertura de um chamado para a CTI (Coordenadoria de Tecnologia da informação) utilizando o sistema GLPI ou informar pessoalmente a CAE (Coordenadoria de Apoio ao Ensino) sobre o problema, ocupando o tempo de aula. Em alguns casos, para não interromper a sua aula, o professor deixa para fazer esse processo depois. Com isso pode ocorrer dele não lembrar de informar o setor responsável, podendo prejudicar o ambiente de ensino para o próximo professor que utilizará o laboratório.

No entanto, o GLPI não possui usabilidade boa o suficiente que permita a abertura de chamados de forma simples e rápida como é exigido nesse contexto. Tais dificuldades acarretam na não abertura de chamados pelos professores e na maioria das vezes os responsáveis não são notificados do problema.

Diante desse problema é necessário o aprimoramento da comunicação entre os professores e as áreas responsáveis. Esse aprimoramento geraria benefícios tanto para os

professores quanto para os alunos fazendo com que os itens dos laboratórios fiquem com sua manutenção em dia e que ocorra menos risco de indisponibilidade durante sua utilização

Com base nestes problemas, foi proposto a especificação funcional de um aplicativo *mobile*, que possibilite aos professores do IFSP-HTO uma forma rápida e simples de abrir chamados no GLPI e enviar E-mails para a CAE referente aos itens que estão com problemas nos laboratórios de informática.

Na seção 2 deste artigo é abordado a fundamentação teórica utilizada para este trabalho, na seção 3 será apresentado com mais detalhes o contexto do problema. Na seção 4 mostrará como o *plug-in FormCreator* pode colaborar para o processo de abertura de chamados no GLPI. Na seção 5 é mostrado o processo que será utilizado na elaboração da especificação do aplicativo e os diagramas utilizados. Na seção 6 será mostrado os requisitos funcionais e não funcionais do aplicativo. Nas seções 7, 8 e 9 serão apresentados os diagramas do aplicativo e suas descrições. Na seção 10 mostrará os protótipos das telas do aplicativo. Na seção 11 é abordado sobre o conceito de usabilidade e na seção 12 as considerações finais.

2. Fundamentação teórica

Nesta seção são abordados os principais conceitos que foram utilizados para o desenvolvimento deste trabalho.

2.1. Engenharia de Requisitos

Visando melhorar o processo de compreensão e solução para uma determinada finalidade, a engenharia de requisitos conta com uma série de técnicas e artefatos que permitem com que o analista, engenheiro de software ou até mesmo o arquiteto do sistema, possam compreender e descrever de forma clara, refinada e assertiva as necessidades do cliente, bem como seus impactos aos usuários finais. Essas técnicas vão desde reuniões, documentos, entrevistas, brainstormings até gravações de áudio. Além disso, a engenharia de requisitos serve como ponte entre descrições abstratas (necessidades do cliente) e o desenvolvimento técnico (definir os requisitos de maneira que os desenvolvedores consigam interpretar e implementar tais necessidades em forma de código) [PRESSMAN 2011].

Projetar e construir um programa de computador elegante que resolva o problema errado não atende as necessidades de ninguém. É por isso que é importante entender o que o cliente quer antes de começar a projetar e construir um sistema baseado em computador [PRESSMAN 2011].

2.2. Requisitos Funcionais e Não Funcionais

Requisitos funcionais são declarações dos serviços que o sistema deve fornecer, do modo como o sistema deve reagir a determinadas entradas e de como deve se comportar em determinadas situações [SOMMERVILLE, 2019]. Os requisitos funcionais de um sistema descrevem o que ele deve fazer e o que está sendo desenvolvido. Em alguns casos os requisitos funcionais podem descrever o que o sistema não deve fazer.

Requisitos não funcionais são restrições sobre os serviços ou funções oferecidas pelo sistema. Eles incluem restrições de tempo, restrições sobre o processo de desenvolvimento e restrições impostar por padrões [SOMMERVILLE, 2019]. Os requisitos não funcionais são

aqueles que não possuem relação direta com os serviços específicos do sistema, ou seja, descrevem características do sistema como um todo ao em vez de serviços e comportamentos individuais.

2.3. Unified Modeling Language

Unified Modeling Language (UML) é um conjunto de 13 tipos diferentes de diagramas que podem ser usados para modelar sistemas de software [SOMMERVILLE, 2019]. A UML é aceita universalmente como abordagem padrão para desenvolver modelos de sistemas de software. São utilizados para representar um sistema com diferentes visões e perspectivas.

3. O Problema

Em reuniões, com a CTI e a CAE do campus Hortolândia, foram apontados que equipamentos com problemas ou danificados nos laboratórios de informática não estão sendo informado para as suas áreas responsáveis devidamente. Com isso os equipamentos ficam muito tempo sem levar a sua manutenção correta podendo acarretar no agravamento do problema ou prejudicando alguém que fosse utilizá-lo. Um dos principais motivos para esse inconveniente é a falta de usabilidade e agilidade no processo de abrir chamados e avisar a CAE sobre as adversidades.

Atualmente o processo inicia quando um professor identifica que um item no laboratório de informática está com problemas, danificado ou faltando. Identificado o problema, o professor deve abrir um chamado no sistema web GLPI pelo navegador ou informar pessoalmente a CAE sobre a complicação. O GLPI é um sistema web de gerenciamento de chamados *open source* [GLPI PROJET 2022] utilizado no IFSP-HTO para o controle interno das adversidades. Como GLPI não tem uma usabilidade boa na abertura de chamados ele ocupa mais tempo do que o necessário para o professor abrir um chamado simples. Essa demora acaba fazendo com que o professor não abra o chamado na hora que identificou o problema podendo, em alguns casos, esquecer de realizar este processo acarretando das partes responsáveis não saberem da complicação. E como o GLPI é considerado um sistema difícil de usar os professores na hora de abrirem o chamado podem acabar dando informações erradas ou imprecisas, prejudicando o atendimento.

Com relação a CAE, como o processo é informar pessoalmente, exige de o professor ter que se deslocar até a coordenadoria e voltar para o laboratório onde estava, demandando muito tempo. O processo de abrir chamados e informar pessoalmente a CAE pouco eficiente, tem sido o grande motivo para que as áreas responsáveis (CTI e CAE) não estejam sendo informadas devidamente sobre as complicações dos itens nos laboratórios de informática.

4. Trabalhos Correlatos

Existem algumas aplicações no mercado, que ajudam a agilizar o processo de abrir chamados como por exemplo a Drop Desk [DROPDESK 2022] que é um sistema de gerenciamento de chamados. Ele utiliza o WhatsApp para fazer o atendimento com os clientes agilizando na abertura dos chamados, mas com um custo mensal por atendente. Outro sistema é o Tomticket [TOMTICKET 2022] que também faz o gerenciamento de chamados. Ele possui um pacote grátis com um atendente utilizando o sistema para um departamento e possui suporte *mobile*. Porém para atender o nosso problema não seria o suficiente e precisaria utilizar os pacotes pagos para adicionar mais departamentos e mais atendentes.

O IFSP-HTO já utiliza um sistema de gerenciamento de chamados, o GLPI, que é um sistema web *open-source*, não gerando custo para a instituição. Com isso o Drop Desk e o Tomticket não seriam uma solução viável para o problema pois necessitaria de uma migração de sistemas na instituição e ainda geraria o custo mensal para utilização da plataforma.

Em seu Trabalho de Conclusão de Curso (TCC), Esteves (2021) apresentou uma solução para melhorar a usabilidade na abertura de chamados no GLPI. A dificuldade de utilização do sistema GLPI é alta e, como proposta de melhoria no processo de abertura de chamados, foi abordado o desenvolvimento de formulários personalizados de acordo com a categoria dos chamados. Cada formulário possui apenas os campos necessários para uma categoria específica de chamado. Para o desenvolvimento dos formulários foi utilizado o *plug-in FormCreator* e cada formulário é configurado separadamente. Neste artigo será abordado a utilização do *FormCreator* para a abertura de chamados dos itens com problemas nos laboratórios de informática. Pois para que o aplicativo, especificado neste artigo, faça a abertura do chamado no GLPI o sistema terá que preencher o formulário criado para então o formulário abrir o chamado.

5. Metodologia

Com os problemas relatados foi proposto para a CAE e a CTI a especificação funcional de um aplicativo que possa fazer a abertura de chamados e informar a CAE sobre os problemas dos itens dos laboratórios de forma rápida e simples. Para fazer a especificação funcional, para o melhor entendimento das funcionalidades do aplicativo, optou-se por três diagramas UML (*Unified Modeling Language*): Diagrama de Caso de Uso, Diagrama de Classe e o Diagrama de Sequência. Cada Diagrama proporciona uma visão e perspectiva única do sistema ajudando na sua compreensão. Desta forma dividimos a realização desse trabalho da seguinte forma:

- Análise e Levantamento dos Requisitos do aplicativo;
- Criação do Diagrama de Caso de Uso e sua descrição;
- Criação do Diagrama de Classe;
- Criação do Diagrama de Sequência;
- Criação dos Protótipos de Tela.

Para o Criação dos Diagramas UML foi utilizada a ferramenta *Lucidchart* [LUCIDCHART 2021] e para o Design dos Protótipos foi utilizado a ferramenta *Uizard* [UIZARD 2021]. A seguir segue o detalhamento de cada atividade listada.

6. Análise e Levantamento de Requisitos

A primeira fase para o levantamento de requisitos é o entendimento do problema. Para isso foram feitas reuniões junto com representantes da CTI e da CAE do campus de Hortolândia. Nelas foram esclarecidos quais itens tinham mais ocorrências e quais setores eles seriam direcionados. De acordo com as reuniões foram levantados os seguintes requisitos funcionais:

RF001: Deve ser criado no GLPI utilizando o *plug-in FormCreator* um formulário para os problemas nos itens dos laboratórios descritos na RF003. Os campos são os mesmos descritos para o aplicativo na RF005.

RF002: O aplicativo deve ser capaz de preencher o formulário criado no GLPI para fazer a abertura de chamados.

RF003: Ao abrir o aplicativo e realizar o login irá aparecer uma lista de itens para o usuário selecionar. Estes itens são: “cadeira”, “lousa”, “mesa”, “janela”, “ventilador”, “ar-condicionado”, “porta”, “projektor”, “computador”, “rede”, “teclado”, “mouse” e “monitor”. Na mesma tela um campo texto para o professor digitar o laboratório/sala que se encontra.

RF004: Após selecionar os itens com problemas e clicar em “avançar” o aplicativo deve redirecionar para outra tela onde mostra separadamente cada item selecionado para professor descrever seus problemas.

RF005: Na tela de descrição dos problemas alguns itens devem ter combos com problemas decorrentes para que o usuário não precise digitar sempre:

- Projetor: Campos *Toggle Switch* “A lâmpada está Queimada?” E “Este item está com defeito?”;
- Rede: Campo *Toggle Switch* “A rede está com oscilação ou sem conexão?”;
- Alguns itens tem o problema decorrente de estarem faltando e também vão ter o campo “Quantos itens faltam?” E “Quantos itens quebrados?” Onde o professor colocará a quantidade de itens que está faltando. Os itens que terão esse campo são: cadeira, mesa, teclado, computador, mouse;
- Os itens a seguir terão a opção “Este item está com defeito?” Para o usuário selecionar: “Projektor”;
- Todos os itens terão o campo texto “Descrição”, não obrigatório, para o professor colocar a descrição do problema, caso prefira.

RF006: Se o professor selecionar no formulário os itens “cadeira”, “lousa”, “mesa”, “janela”, “ventilador”, “ar-condicionado” ou “porta” ao finalizar o formulário o aplicativo deve mandar um e-mail para a CAE, com o professor em cópia, com as informações de cada item passadas pelo professor.

RF007: Se o professor selecionar no formulário os itens “projektor”, “computador”, “rede”, “teclado”, “mouse” ou “monitor” ao finalizar o formulário o aplicativo deve abrir um chamado no sistema GLPI para a CTI com as informações de cada item passadas pelo professor.

RF008: Se o professor selecionar itens de áreas diferentes o aplicativo agrupará os itens para fazer a requisição final. Ex.: O professor José selecionou “projektor” e “cadeira” o aplicativo conforme a RF006 e envia o e-mail para a CAE apenas com os detalhes do item “cadeira” e conforme a RF007 e abre um chamado no GLPI com os dados do item “projektor”.

RF009: Ao finalizar a requisição o aplicativo deve retornar uma mensagem de sucesso, caso tenha feito todo o processo sem erros. Se foi realizado um chamado deve retornar o número do chamado na mensagem.

RF10: Ao tentar finalizar a requisição caso aconteça algum erro em algum processo o aplicativo deve retornar mensagem “Falha no envio! Tente novamente!” E informar quais itens que não foram enviados.

RF011: Ao clicar em “Finalizar” o aplicativo não deve deixar o usuário interagir com o botão “Finalizar” para não enviar requisições duplicadas.

RF012: Caso o professor não preencha nenhum campo na tela de detalhamento e tentar finalizar. O sistema deverá emitir uma mensagem escrito “Obrigatório o preenchimento de um campo por item!” E não prosseguir com a finalização do formulário.

RF013: caso tenha feito 2 processos e 1 deles teve sucesso e o outro não, mostrar os itens que tiveram sucesso e os itens que não tiveram sucesso.

Para a escolha dos requisitos não funcionais foram feitas algumas considerações. O aplicativo teria que rodar tanto nos aparelhos com sistema operacional Android quanto IOS. E como o aplicativo não necessitaria fazer o armazenamento de dados não será necessária a utilização de um mecanismo de banco de dados. A seguir os requisitos não funcionais levantados:

RN001: O acesso dos usuários no aplicativo será através da integração com o protocolo LDAP (*Lightweight Directory Access Protocol*) do IFSP-HTO.

RN002: O aplicativo deve ser executar nos sistemas operacionais Android e IOS.

RN003: Arquitetura de Microserviços.

RN004: O aplicativo não necessita fazer o armazenamento de dados.

7. Diagrama de Caso de Uso

Diagramas de caso de uso proporcionam uma visão geral simples de interações que o usuário tem com o sistema [SOMMERVILLE, 2019]. Na figura 1 podemos ver o diagrama de caso de uso do sistema com as interações que o usuário tem com o aplicativo.

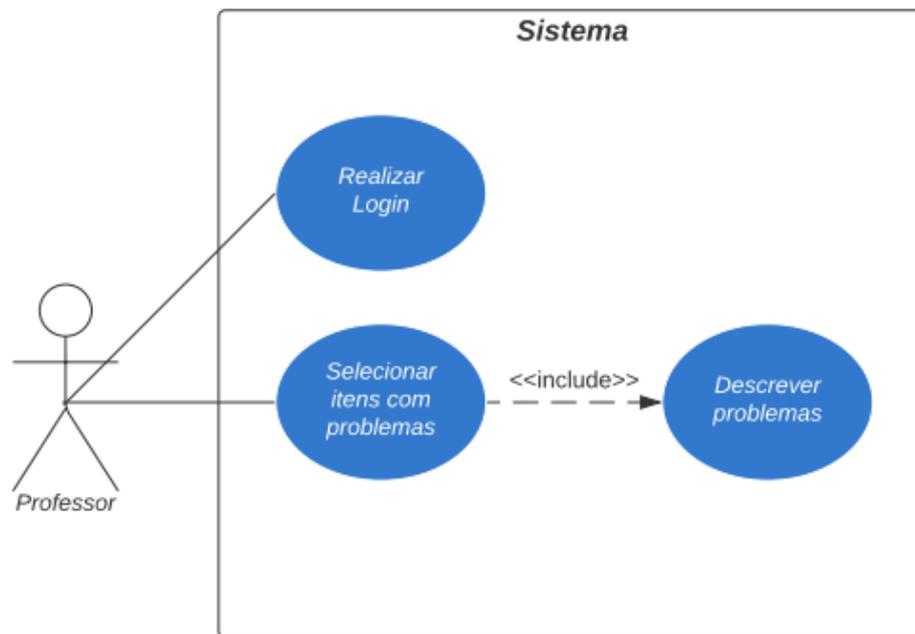


Figura 1. Diagrama de Caso de Uso

Como o diagrama apresenta informações simples e gerais é necessário apresentar mais detalhes explicando cada interação. A seguir as tabelas 1, 2 e 3 apresentam os detalhamentos dos casos de uso e seus fluxos.

Tabela 1. Caso de Uso: Realizar Login

Caso de Uso 01: Realizar Login
Descrição: Usuário realiza login no sistema.
Ator: Professor
Precondições: Professor não estar logado no sistema.
Fluxo Principal:
<ol style="list-style-type: none">1. O sistema apresenta a tela de login com os campos “Usuário” e “Senha” para serem preenchidos.2. O professor preenche os campos “Usuário” e “Senha”.3. O professor clica no botão “Login”.4. O sistema envia os dados informados para o LDAP fazer a autenticação do usuário. Caso os dados estiverem incorretos o sistema emite uma mensagem de erro com o retorno do LDAP.
Pós-condições: O login é efetuado e o sistema redireciona o professor para a tela “Seleção dos itens”.

Tabela 2. Caso de Uso: Selecionar itens com problemas

Caso de Uso 02: Selecionar itens com problemas
Descrição: Usuário seleciona os itens que estão com problemas.
Ator: Professor
Precondições: Professor estar logado no sistema e estar na tela principal.
Fluxo Principal:
<ol style="list-style-type: none">1. Na tela “Seleção dos itens” o sistema apresenta campos do tipo <i>checkbox</i> com os nomes dos itens do laboratório (“Cadeira”, “Lousa”, “Mesa”, “Janela”, “Ventilador”, “Ar-condicionado”, “Porta”, “Projeter”, “Computador”, “Rede”, “Teclado”, “Mouse” e “Monitor”) para o professor selecionar quais estão com problemas.2. O professor seleciona os itens que estão com problemas.3. O professor clica no botão “Avançar”. Caso o professor não tenha selecionado nenhum item o sistema emite um alerta solicitando a seleção de ao menos um item.
Pós-condições: O sistema redireciona o professor para a tela “Descrição dos problemas”.

Tabela 3. Caso de Uso: Selecionar itens com problemas

Caso de Uso 03: Descrever problemas
Descrição: Usuário descreve os problemas dos itens selecionados.
Ator: Professor
Precondições: Professor tenha selecionado ao menos um item na tela “Seleção dos itens”.
Fluxo Principal:
<ol style="list-style-type: none"> 1. Na tela “Descrição dos problemas” o sistema apresenta para cada item que o professor selecionou na tela “Seleção dos itens” campos para o professor descrever os problemas. E apresenta um campo texto para o professor informar o laboratório que está. 2. O professor descreve os problemas de cada item apresentado na tela. 3. O professor seleciona o laboratório que está. 4. O professor clica no botão “Finalizar”. Caso o professor não tenha preenchido nenhum campo de algum item o sistema vai emitir um alerta para que seja preenchido ao menos um campo por item. 5. O sistema verifica qual setor cada item pertence (CAE ou CTI). Caso o professor tenha selecionado um ou mais dos itens entre “projektor”, “computador”, “rede”, “teclado”, “mouse” ou “monitor” o sistema irá fazer uma requisição no formulário do GLPI para abrir um chamado com os dados informados destes itens. Caso o professor tenha selecionado um ou mais itens entre “cadeira”, “lousa”, “mesa”, “janela”, “ventilador”, “ar-condicionado” ou “porta” o sistema irá gerar e enviar um e-mail para a CAE e para o professor.
Pós-condições: O Sistema exibe uma mensagem de sucesso com os retornos recebidos.

8. Diagrama de Classe

Os diagramas de classes são utilizados no desenvolvimento de um modelo de sistema orientado a objetos para mostrar as classes em um sistema e as associações entre elas [SOMMERVILLE, 2019]. Na figura 2 podemos ver o diagrama de classe do aplicativo. Nela pode-se ver as subclasses com os nomes dos itens do laboratório de informática (“Cadeira”, “Lousa”, “Mesa”, “Janela”, “Ventilador”, “Ar-condicionado”, “Porta”, “Projektor”, “Computador”, “Rede”, “Teclado”, “Mouse” e “Monitor”), que herdam os atributos da classe pai “Item”. As classes “Chamado” e “Email” são onde os itens serão agrupados para serem redirecionados para suas respectivas áreas responsáveis. Os itens que estão na classe “Chamado” serão redirecionados para o preenchimento do formulário do GLPI onde fará a abertura do chamado para atendimento da CTI. Para os itens que estão na classe “Email” o sistema irá pegar suas informações dos seus atributos e formulará um e-mail que será enviado para o professor que está fazendo a solicitação e a CAE.

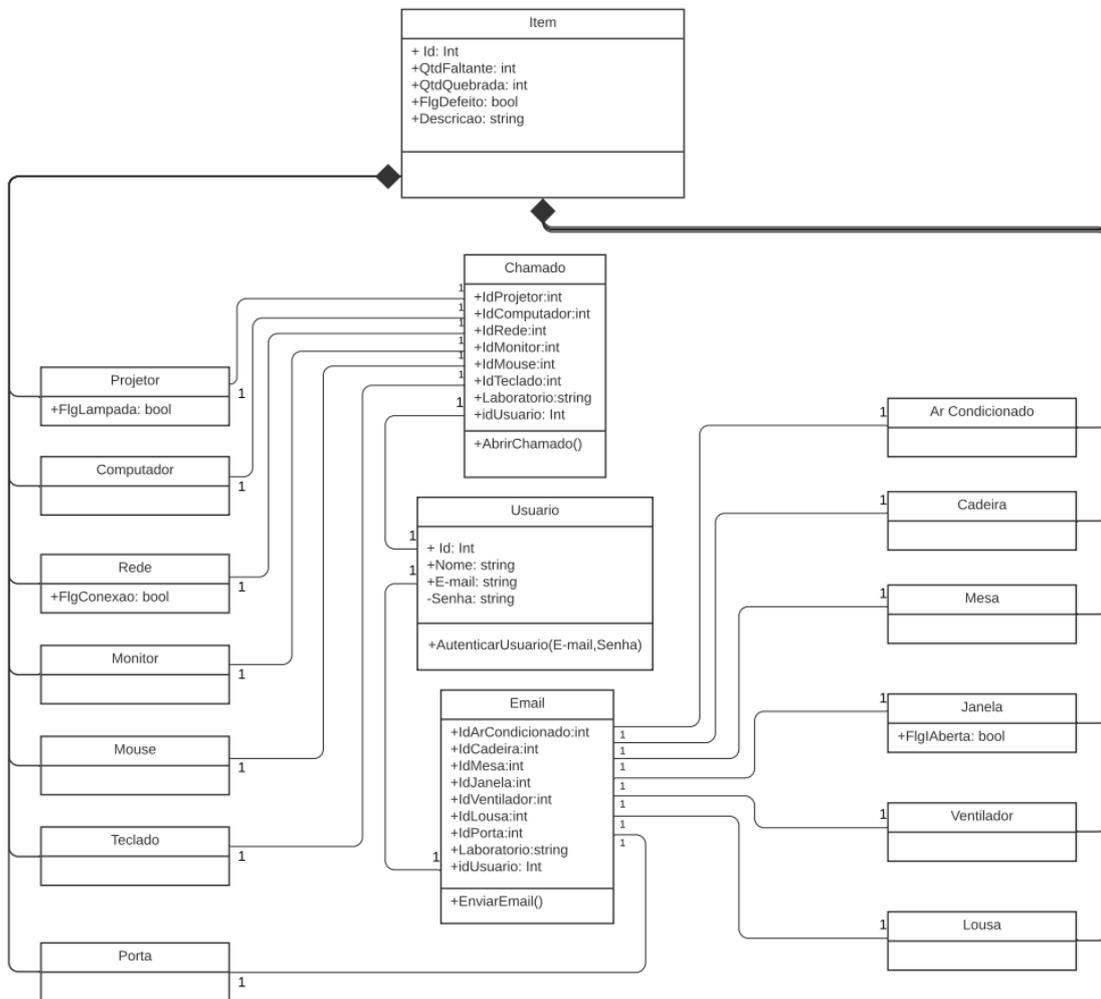


Figura 2. Diagrama de Classe

9. Diagrama de Sequência

Os diagramas de sequência são utilizados principalmente para modelar as interações entre os atores e objetos em um sistema e as interações entre os próprios objetos [SOMMERVILLE, 2019]. O LDAP é um protocolo aberto utilizado para acessar e manter serviços de informação de diretórios e usuários distribuídos sobre uma rede de *Internet Protocol (IP)* [LDAP 2022]. Ele é utilizado no IFSP-HTO para centralizar o login de seus sistemas. A figura 3 apresenta a interação que o sistema faz quando o usuário está realizando o login. Quando o professor informa suas credenciais na tela de login o sistema consulta o LDAP para ver se os dados informados estão corretos.

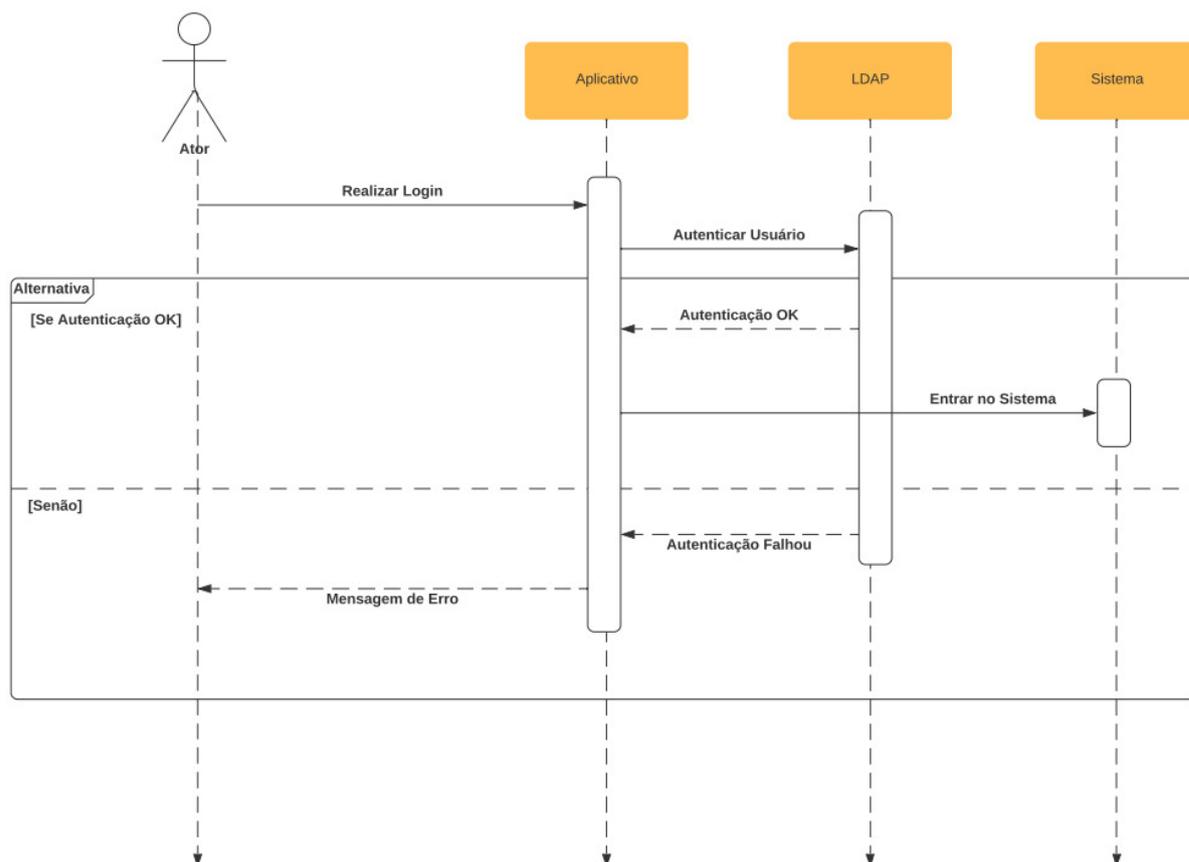


Figura 3. Diagrama de Sequência do Fluxo de Login

Na figura 4 é possível ver as interações do sistema quando o usuário quer informar o setor responsável por algum problema no laboratório. O usuário descreve os problemas que identificou, o sistema verifica se possui algum item que necessita abrir chamado. Caso a existência, o sistema acessa o formulário do GLPI criado com o *plug-in FormCreator* e preenchendo o formulário com apenas os dados dos itens que forem fazer a abertura do chamado. Feito o envio do formulário para o GLPI e a abertura do chamado o sistema recebe a resposta do formulário. Caso o houver itens para serem enviados para a CAE o sistema gera um e-mail e envia, utilizando o servidor de e-mail, para a CAE e para o professor que fez a solicitação e retorna a mensagem de sucesso. No final o sistema retorna todas as mensagens de retorno recebidas (caso o sistema teve que executar mais de um processo) para o usuário.

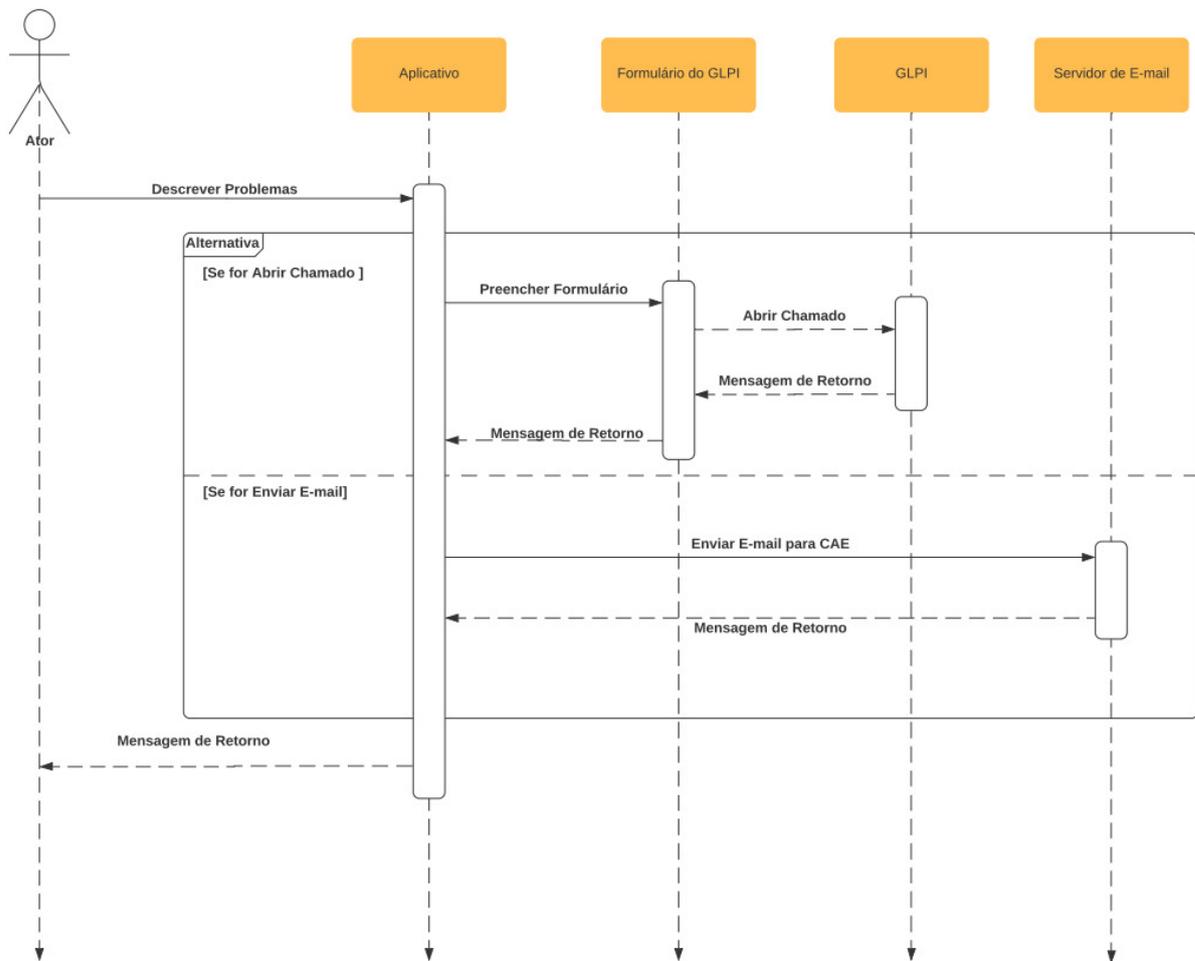


Figura 4. Diagrama de Sequência

10. Protótipos de Tela

Após a fase de levantamento e definição dos requisitos, foram criados os primeiros protótipos das telas do sistema. O protótipo serve para simular o modelo real do aplicativo, sendo assim uma ponte entre os desenvolvedores e os *stakeholders* do sistema, facilitando o desenvolvimento do aplicativo. Os protótipos também servem para validar o nível de usabilidade do sistema.

Na Figura 5 podemos ver o protótipo da “Tela de Login” onde o usuário deverá inserir suas credenciais para o sistema envia-las para o LDAP fazer a autenticação.



Figura 5. Protótipo: Tela de Login

Na figura 6 está o protótipo da “Tela Seleção dos itens” que é a tela inicial do sistema quando o usuário já está logado. Nela o usuário vai selecionar quais são os itens que estão com problemas. Nesta tela tem a opção de o usuário fazer o *log out* do sistema selecionando o botão “Log out”.

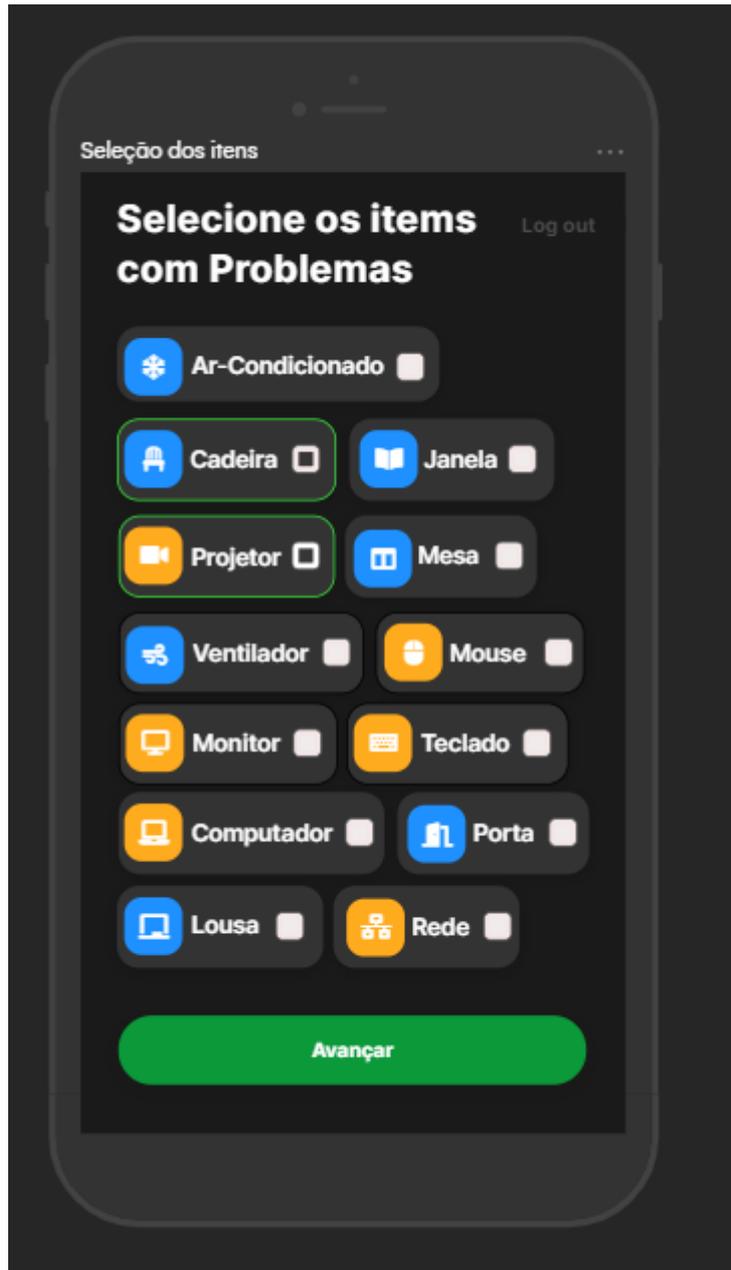


Figura 6. Protótipo: Tela Seleção dos itens

Na figura 7 podemos ver o protótipo da “Tela Descrição dos Problemas” onde aparecerão setores para o usuário preencher para cada item que foi selecionado na “Tela Seleção dos itens”. O usuário deve preencher ao menos um campo por item para prosseguir para o envio.



Figura 7. Protótipo: Tela Descrição dos Problemas

Na figura 8 está o protótipo de mensagem de confirmação quando o sistema termina de enviar as informações passadas pelo usuário na “Tela Descrição dos Problemas” para as áreas responsáveis. Após o usuário clicar no botão “OK” ele será redirecionado para a tela inicial (Tela Seleção dos itens).

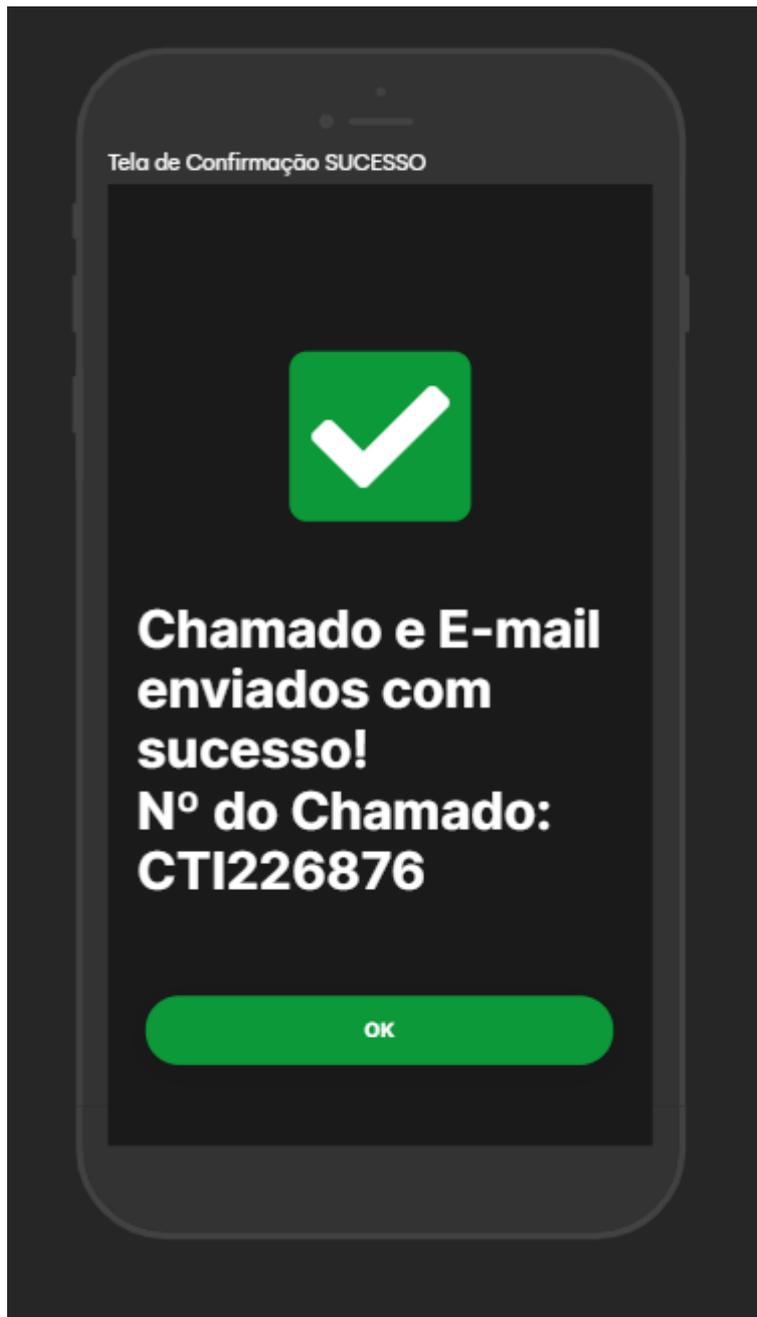


Figura 8. Protótipo: Mensagem Confirmação

Na figura 9 mostra a mensagem de confirmação e mensagem de erro. Isso se acontece quando um dos processos (enviar e-mail CAE, abrir chamado CTI) falha. Na figura mostra que o processo de abrir chamado para o item “Projektor” falhou, porém foi enviado o e-mail para a CAE sobre o item “Cadeira”.

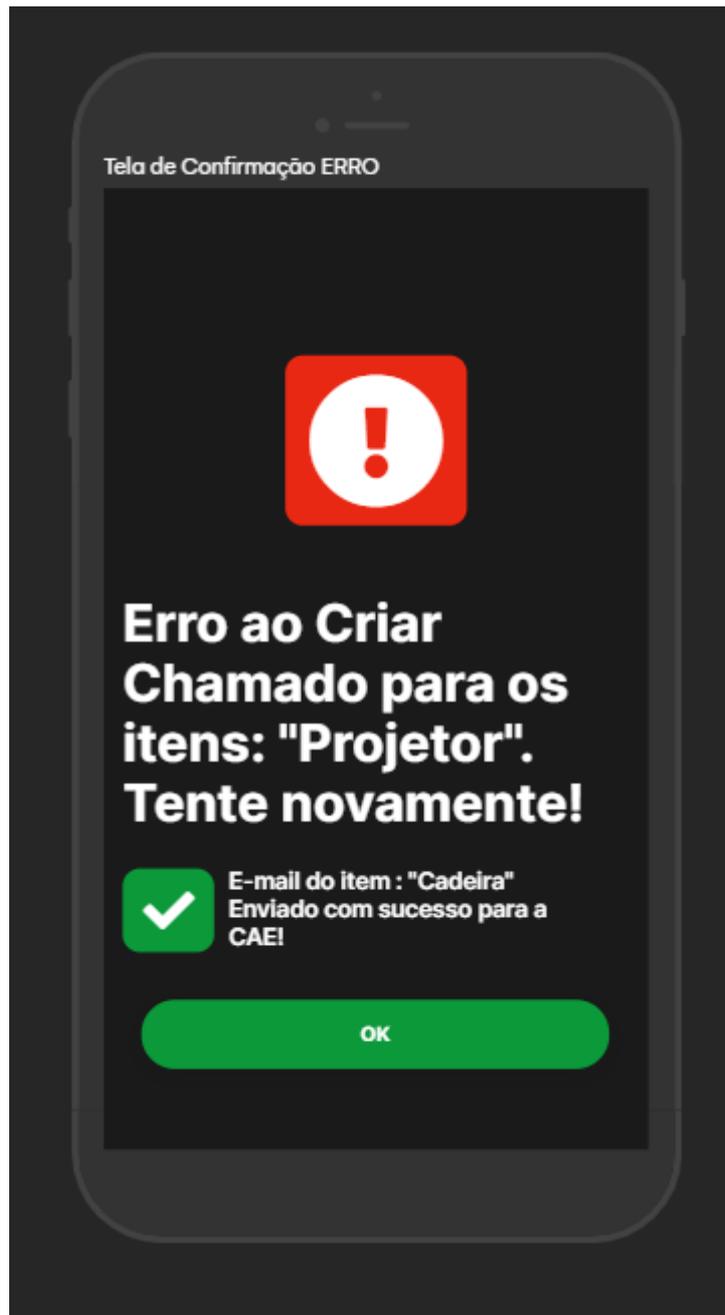


Figura 9. Protótipo: Mensagem Confirmação ERRO

11. Usabilidade

A definição de usabilidade é de que os sistemas devem ser fáceis de usar e de aprender, flexíveis e devem despertar nas pessoas uma boa atitude [BENYON 2011]. Segundo Benyon para que um sistema tenha um alto grau de usabilidade ele deverá ter essas características:

- Ser eficiente mediante a quantidade de esforço que as pessoas utilizarão;
- Ser eficaz no sentido de conter as funções e o conteúdo de informações e organizá-las apropriadamente.
- Ter um processo fácil de aprender e fácil de lembrar como após algum tempo;

- Ter um alto grau de utilidade no sentido que fará as coisas que as pessoas querem que sejam feitas.

Baseado nessas características o aplicativo, especificado neste artigo, possui um grau de usabilidade alto pois ele:

- Demandará pouco tempo e esforço dos usuários para realizar a abertura de chamados;
- Apresenta as informações necessárias, para que o usuário utilize o aplicativo, de forma organizada.
- Tem um processo intuitivo e de pouco cliques para que seja fácil de aprender e de lembrar.
- É uma solução que melhora o processo atual de abertura de chamados.

12. Conclusão

O objetivo deste trabalho foi descrever a especificação funcional de um aplicativo *mobile* que auxiliasse os professores a informar as áreas responsáveis sobre problemas nos itens que compõe um laboratório de informática. Considera-se que o objetivo foi alcançado. O Artigo expressa as necessidades do cliente e propõe uma solução mostrando com detalhes os serviços que o aplicativo fornecerá. Neste trabalho é possível compreender as necessidades do cliente e definir os requisitos e funcionalidades do sistema de maneira que desenvolvedores consigam compreender e desenvolver o aplicativo.

O aplicativo foi pensado para facilitar a comunicação dos professores com as áreas responsáveis quando um item no laboratório de informática tem algum problema. Com design simples e com poucos cliques o professor será capaz de abrir um chamado para a CTI e enviar um e-mail para CAE em poucos minutos, um processo que atualmente tem menos flexibilidade e ocupa muito tempo do professor e torna esse processo desgastante.

Realizar este trabalho nos permitiu adquirir um conhecimento maior sobre levantamento de requisitos com um cliente real e como é importante a compreensão dos seus problemas para que o sistema seja assertivo e solucione o problema. E esclareceu a importância do design e dos diagramas para que tanto o usuário quanto arquitetos e desenvolvedores possam ter uma visão ampla e objetiva das funcionalidades do sistema.

Neste trabalho foram aplicados os conhecimentos adquiridos na disciplina Engenharia de Software utilizados na especificação dos requisitos e na criação dos Diagramas UML. Já os conhecimentos da disciplina Interação Humano-Computador colaboraram no design dos protótipos e na usabilidade do aplicativo. Para a escrita deste trabalho foram utilizados os conhecimentos das disciplinas Projeto de Sistemas I e Projeto de Sistemas II.

Referências

- BENYON, D. (2011). *Interação Humano-Computador*. 2. ed. São Paulo: PEARSON, 2011.
- DROPDESK. (2022). Funcionalidades, disponível em: <<https://www.dropdesk.com.br>>, acesso em : 22 jan. 2022.
- ESTEVE, R. S. (2021). *Manutenção do GLPI para Facilitar Abertura de Chamados no IFSP Câmpus Hortolândia, Trabalho de Conclusão de Curso em Tecnologia em Análise e Desenvolvimento de Sistemas, IFSP Câmpus Hortolândia*, disponível em : <<https://hto.ifsp.edu.br/cloud/s/6WcpQ7diXtefizw>>, acesso em: 24 jan. 2022.
- GLPI PROJECT. (2022). *Gerenciamento de TI: Com o poder da liberdade*, disponível em: <<https://glpi-project.org/pt-br/>>, acesso em: 07 jan. 2022.
- LDAP. (2022). *Learn About LDAP*, disponível em: <<https://ldap.com/learn-about-ldap/>>, acesso em: 11 jan. 2022.
- LUCIDCHART. (2021). *Galeria de Modelos de Diagramas*, disponível em: <<https://www.lucidchart.com/pages/pt/templates>>, acesso em: 10 dez. 2021.
- PRESSMAN, Roger. S. *Engenharia de software: Uma abordagem profissional*. 7. ed. São Paulo: AMGH, 2011.
- SOMMERVILLE, Ian. (2019). *Engenharia de software*. 10. ed. São Paulo: PEARSON, 2019.
- TOMTICKET. (2022). *Recursos*, disponível em: <<https://www.tomticket.com>>, acesso em: 22 jan. 2022.
- UIZARD. (2021). *The design tool for non-designers*, disponível em: <<https://uizard.io>>, acesso em: 18 dez. 2021.

Documento Digitalizado Público

Anexo I (Artigo) - Victor Hugo Franzini da Silva - HT1520369

Assunto: Anexo I (Artigo) - Victor Hugo Franzini da Silva - HT1520369
Assinado por: Carlos Junior
Tipo do Documento: Outro
Situação: Finalizado
Nível de Acesso: Público
Tipo do Conferência: Documento Digital

Documento assinado eletronicamente por:

- **Carlos Roberto dos Santos Junior, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 23/03/2022 17:11:39.

Este documento foi armazenado no SUAP em 23/03/2022. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifsp.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 925151

Código de Autenticação: 5a93055e51

