

LEANDRO MASCARO FERNANDES

**MODELAGEM DE FERRAMENTA PARA SUPORTE TÉCNICO DE  
LABORATÓRIOS DE INFORMÁTICA**

Limeira

2019

LEANDRO MASCARO FERNANDES

**MODELAGEM DE FERRAMENTA PARA SUPORTE TÉCNICO DE  
LABORATÓRIOS DE INFORMÁTICA**

Monografia apresentada à Faculdade de  
Tecnologia da Universidade Estadual de  
Campinas como parte dos requisitos  
exigidos para a obtenção do título de  
Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Plínio Roberto Souza Vilela

Limeira

2019

## FOLHA APROVAÇÃO

## **AGRADECIMENTOS**

Gostaria de agradecer primeiramente a Deus, que me deu forças para realizar meus sonhos e objetivos.

Ao meu orientador, Prof. Dr. Plínio Roberto Souza Vilela, por todo o empenho em me auxiliar no desenvolvimento deste trabalho.

A todos os professores que durante meus anos de faculdade contribuíram para meu crescimento pessoal, acadêmico e profissional.

Aos alunos, funcionários da faculdade e docentes que colaboraram com o projeto ao participarem do levantamento de requisitos.

Aos meus pais, Antônio e Mônica, e minha irmã, Andressa, que sempre me apoiaram e me alegraram nos dias difíceis.

## RESUMO

Atualmente na Faculdade de Tecnologia da UNICAMP falta um método simples de comunicação entre os alunos e a equipe responsável pelos laboratórios de informática. Este trabalho pretende resolver esse problema modelando um aplicativo que será um método direto para criar uma solicitação e disponibiliza à equipe de suporte uma forma mais eficaz e com transparência de realizar a gestão dos laboratórios. O projeto aqui apresentado propõe, através da utilização de importante conceitos de engenharia de software, a modelagem e implementação de um protótipo de um aplicativo que auxilie no suporte e gerenciamento de TI para laboratórios de informática. Os resultados apresentados são a entrega da documentação do software, tais como os requisitos, diagramas e o protótipo de telas.

Palavras-chave: Engenharia de Software, Levantamento de Requisitos, Modelagem de Sistemas.

## **ABSTRACT**

Nowadays UNICAMP's School of Technology is lacking on a simple method of communication between the students and the staff responsible for the computer labs. This paper aims to solve this problem by giving the students a straightforward method to create a request and providing the support team with a more effective way to manage the laboratories. The project presented here proposes, by using important software engineering concepts, the modeling and implementation of a prototype of an application that assists in IT support and management for computer labs. The result was the delivery of software documentation such as requirements, diagrams and screen prototypes.

**Keywords:** Software Engineering, Requirements Gathering, System Modeling.

## LISTA DE ILUSTRAÇÕES

- Figura 1: Fases do RUP e a ênfase de cada fase.
- Figura 2: Ciclo de Vida Iterativo RUP.
- Figura 3: Pergunta 1 do Questionário.
- Figura 4: Pergunta 2 do Questionário.
- Figura 5: Pergunta 3 do Questionário.
- Figura 6: Pergunta 4 do Questionário.
- Figura 7: Pergunta 5 do Questionário.
- Figura 8: Diagrama de Casos de Uso.
- Figura 9: Diagrama de Classes.
- Figura 10: Diagrama de Sequência - Realizar Login.
- Figura 11: Diagrama de Sequência - Cadastrar Aluno.
- Figura 12: Diagrama de Sequência - Cadastrar Equipamento.
- Figura 13: Diagrama de Sequência - Cadastrar Funcionários.
- Figura 14: Diagrama de Sequência - Consultar Funcionários.
- Figura 15: Diagrama de Sequência - Gerar Relatório de Solicitações.
- Figura 16: Diagrama de Sequência - Finalizar Solicitação.
- Figura 17: Diagrama de Sequência - Consultar Solicitação.
- Figura 18: Diagrama de Sequência - Identificar Problema.
- Figura 19: Diagrama de Sequência - Identificar Equipamento.
- Figura 20: Diagrama de Entidade e Relacionamento.
- Figura 21: Protótipo - Tela de Login.
- Figura 22: Protótipo - Tela de Cadastro de Aluno.
- Figura 23: Protótipo - Tela de Gerar Solicitação.
- Figura 24: Protótipo - Tela de Identificar Problema.
- Figura 25: Protótipo - Tela de Funcionários.
- Figura 26: Protótipo - Tela de Consultar Solicitação.
- Figura 27: Protótipo - Tela de Cadastrar Funcionários.
- Figura 28: Protótipo - Tela de Consultar Solicitação Funcionário.
- Figura 29: Protótipo - Tela de Consultar Funcionário.
- Figura 30: Protótipo - Tela de Solicitação Gerada.
- Figura 31: Protótipo - Tela de Relatório de Solicitações.
- Figura 32: Protótipo - Tela de Cadastrar Equipamento.

Figura 33: Programação por Blocos Back-end.

Figura 34: Programação por Blocos Front-end.

## **LISTA DE ABREVIACÕES E SIGLAS**

UML – Unified Modeling Language

RUP – Rational Unified Process

FT – Faculdade de Tecnologia

DER – Diagrama de Entidade-Relacionamento

MIT – Massachusetts Institute of Technology

## SUMÁRIO

<b>1. Introdução</b> .....	11
1.1. Estrutura do Trabalho .....	11
<b>2. Fundamentação Teórica</b> .....	13
2.1 Processo de Desenvolvimento de Software.....	13
2.2 Engenharia de Requisitos .....	15
2.3 Diagramas UML .....	15
2.3.1. Diagrama de Caso de Uso.....	16
2.3.2 Diagrama de Classes.....	16
2.3.3. Diagrama de Sequência.....	17
2.3.4. Diagrama de Entidade e Relacionamento.....	17
<b>3. Levantamento de Requisitos</b> .....	18
<b>4. Especificação do Sistema</b> .....	23
4.1. Requisitos Funcionais.....	24
4.1.1. Realizar Login .....	24
4.1.2. Identificar Problema.....	24
4.1.3. Identificar Equipamento.....	24
4.1.4. Consultar Solicitação .....	25
4.1.5. Finalizar Solicitação .....	25
4.1.6. Gerar Relatório de Solicitações .....	26
4.1.7. Cadastrar Aluno.....	26
4.1.8. Cadastrar Funcionários.....	26
4.1.9. Consultar Funcionários.....	27
4.1.10. Cadastrar Equipamento .....	27
4.2. Requisitos Não Funcionais .....	27
<b>5. Modelagem do Sistema</b> .....	28
<b>6. Protótipo de Telas</b> .....	40
<b>7. Resultados e Considerações Finais</b> .....	46

## 1. Introdução

Atualmente na Faculdade de Tecnologia da UNICAMP existem seis laboratórios de informática utilizados para aulas e para estudos pelos alunos. Sempre que existe uma falha de hardware com algum dos equipamentos, como problemas na inicialização, nos periféricos, ou software, como necessidade de instalação, por exemplo, é necessário buscar presencialmente ajuda da equipe de TI. Após alertá-los sobre o problema, o aluno não consegue acompanhar o status da correção do problema, e outros alunos não têm a visibilidade de que o equipamento pode não estar operando normalmente. Falta um método simples de comunicação entre os alunos e a equipe responsável.

Com isso em mente, este trabalho propõe um facilitador para os alunos e para os responsáveis pelos laboratórios de informática, aplicável não apenas à FT como a lugares com o mesmo tipo de cenário-problema. É aqui apresentada a concepção de um aplicativo *mobile* que gere as solicitações. Seja via *QR Code* ou, de forma alternativa, fazendo uso de uma *Asset Tag*, será possível fazer a abertura de uma solicitação de serviço. O aplicativo também será responsável pela geração de relatórios sobre o histórico de atendimento das solicitações.

Para a realização das atividades deste projeto, foram utilizados três softwares: Astah, Draw.io e MIT App Inventor. O Astah e o Draw.io são softwares que permitem a criação de diagramas e esquemas do processo de desenvolvimento de sistemas. Neste caso, o Astah foi utilizado na elaboração e construção dos diagramas de caso de uso, sequência e de classes. O Draw.io foi utilizado para a construção do diagrama de entidades e relacionamentos. Já o MIT App Inventor, criado pela universidade americana Massachusetts Institute of Technology (MIT), é um software que permite o desenvolvimento de aplicativos Android, utilizado para a criação do protótipo do aplicativo.

### 1.1. Estrutura do Trabalho

Este trabalho foi dividido em sete capítulos. No Capítulo 2 é apresentada a fundamentação teórica do projeto.

No Capítulo 3 será apresentado sobre os problemas existentes nos laboratórios de informática da faculdade, assim como os métodos realizados para o levantamento de requisitos. O Capítulo 4 é responsável por apresentar a especificação do sistema.

No Capítulo 5 apresenta os diagramas que compõe a documentação do sistema apresentado. O Capítulo 6 contém as informações de como foi feito o protótipo de telas e a representação das telas do aplicativo.

No Capítulo 7 contém as considerações finais e resultados desse projeto.

## 2. Fundamentação Teórica

Neste capítulo são apresentadas as teorias que embasam este projeto, tanto referente ao modelo de desenvolvimento escolhido, tanto acerca da engenharia de requisitos e gerenciamento de serviços.

### 2.1 Processo de Desenvolvimento de Software

Para o desenvolvimento de um software existem vários modelos de processos, um modelo é uma estratégia que já foi desenvolvida e aceita no mercado. O modelo cascata, também conhecido como ciclo de vida clássico, possui uma abordagem sequencial e sistemática para o desenvolvimento de software, iniciando com a especificação de requisitos e avançando para as fases de planejamento, modelagem, construção e entrega do software. PRESSMAN (2011) Para esse modelo, em princípio, é necessário planejar e programar todas atividades do processo antes de iniciar o trabalho nelas. SOMMERVILLE (2011)

Como esse projeto é uma modelagem de um sistema, o modelo cascata não seria interessante de ser aplicado, porque logo no início do projeto seria necessário que os requisitos estivessem claros e completamente definidos.

O modelo escolhido para o desenvolvimento desse projeto é uma adaptação do Processo Unificado da Rational conhecido como RUP (Rational Unified Process). O RUP é uma metodologia criada pela Rational Software Corporation, adquirido pela IBM em 2003 e foi criado para dar apoio ao desenvolvimento orientado a objetos, fornecendo uma forma sistemática para obter vantagens no uso da UML, que será descrito melhor nos próximos capítulos. SOMMERVILLE (2011)

O RUP é composto por três princípios básicos: arquitetura, iteração e orientação a casos de uso. Os casos de uso são responsáveis por orientar o desenvolvimento do projeto, porque eles são responsáveis por especificar o que o sistema deve fazer, independentemente de como ele o irá fazer. O RUP é um modelo centrado em arquitetura, porque defende a criação do esqueleto da aplicação que ganha corpo gradualmente ao longo de seu desenvolvimento e é um processo iterativo e incremental que oferece um tratamento para particionar o trabalho em mini-projetos. SOMMERVILLE (2011)

O ciclo de vida adotado no RUP é evolutivo, diferentemente dos modelos habituais de planejamento, levantamento de requisitos, análise e projeto, o RUP é adaptado nas seguintes fases:

- **Concepção:** é estabelecido o escopo do projeto, determinando os principais casos de uso do sistema. Nesta fase a ênfase recai sobre o planejamento e,

consequentemente, é necessário o levantamento do sistema e analisá-los. Ao término dessa fase, são examinados os objetivos do projeto.

- **Elaboração:** nesta fase, é necessário analisar mais refinadamente o problema do que o projeto está responsável solucionar, estabelecendo uma arquitetura mais sólida. Nessa fase também se desenvolve um plano para o sistema que será construído eliminando os riscos.
- **Construção:** o produto é completo e desenvolvido de maneira iterativa e incremental, para que a transição com o usuário esteja pronta.
- **Transição:** é nessa fase que o software é disponibilizado ao usuário. Depois que o produto é disponibilizado, normalmente surgem novas alterações que vão demandar novas versões para ajustes no sistema ou corrigir alguns problemas encontrados.

A Figura 1 abaixo mostra que apesar de possuir mais de uma iteração, a cada iteração a ênfase sobre cada atividade muda. A cor mais forte indica o foco principal enquanto a cor mais fraca mostra onde possui o menor foco na ênfase do projeto.

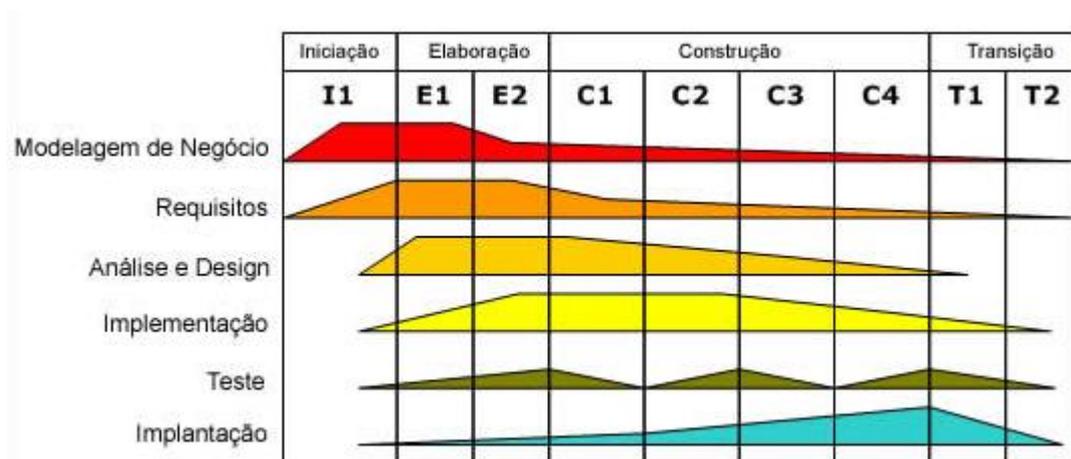


Figura 1 - Fases do RUP e a ênfase em cada fase (RUP,2003).

A Figura 2 mostra o Ciclo de Vida proposto pelo RUP, valendo destacar que cada fase pode não necessariamente, possuir a mesma duração que as outras. O tempo utilizado em cada fase varia em função de circunstâncias específicas do projeto. Esse projeto não apresentará a entrega do software, portanto foi realizada uma adaptação do modelo proposto pelo RUP retirando a fase de Transição.

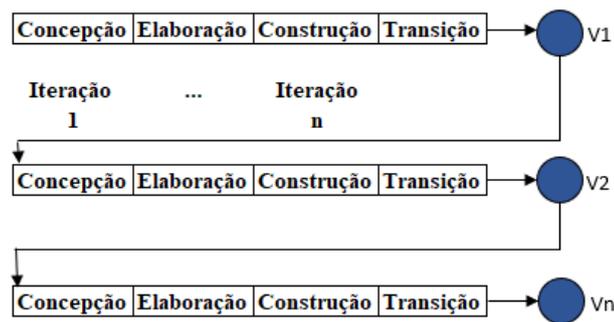


Figura 2 - Ciclo de vida iterativo do RUP

## 2.2 Engenharia de Requisitos

A Engenharia de requisitos é cada vez mais necessária para resolver problemas encontrado nas organizações com relação à definição de sistemas. HOLFMAN, LEHNER (2001) citam que a maior causa de falhas nos projetos de software provém de requisitos deficientes. A engenharia de requisitos é o processo no qual requisitos de um produto de software são analisados, documentados, coletados e gerenciados ao longo do ciclo de vida do software.

O levantamento de requisitos é descrito por SOMMERVILLE (2011) como o processo de reunir as informações necessárias sobre o sistema que será desenvolvido, separando os requisitos do sistema e os dos usuários. Os requisitos são a base de qualquer projeto, eles definem os usuários de um novo sistema e o que esse sistema deve fazer para realizar as necessidades desse projeto. DICK, HULL, JACKSON (2017)

Os requisitos de sistema possuem uma classificação como: requisitos funcionais, não funcionais e de domínio. Os requisitos funcionais descrevem o que o sistema deve fornecer e como ele trata as entradas do sistema. Deve explicar o que o sistema pode ou não pode permitir. Requisitos não funcionais especificam as restrições do sistema sobre as funções fornecidas como custo, manutenção, tempo, desempenho, segurança, usabilidade entre outros. Os requisitos de domínio podem conter os novos requisitos funcionais ou limitações nos sistemas já existentes. LAPLANTE (2013)

## 2.3 Diagramas UML

UML, *Unified Modeling Language*, é uma linguagem-padrão para elaboração da modelagem do software. Deve ser usada para especificar, documentar, visualizar e construir os artefatos de um sistema. BOOCH, RUMBAUGH, JACOBSON (2000)

Na UML existem nove diagramas: diagrama de caso de uso, diagrama de classes, diagrama de sequência, diagrama de objetos, diagrama de estados, diagrama de atividades, diagrama de implementação, diagrama de colaboração e diagrama de componentes. BOOCH, RUMBAUGH, JACOBSON (2000)

Iremos utilizar nesse projeto os diagramas de caso de uso, diagrama de classes, diagrama de sequência e o diagrama de entidade e relacionamento.

### 2.3.1. Diagrama de Caso de Uso

O diagrama de caso de uso são utilizados para demonstrar funcionalidades que o novo sistema irá apresentar. Esse diagrama mostra como o sistema e os atores vão interagir através de fluxos. Para entender melhor as necessidades do software, usuários e desenvolvedores podem criar conjunto de cenários que ajudam a visualizar o software que será construído. O engenheiro de software também pode utilizar esse diagrama como um *checklist* para o trabalho de especificação do sistema.

Os componentes de um caso de uso são:

- Ator: é representado por um *stick man*, especifica um papel executado por um usuário do sistema. O ator deve ser externo ao sistema.
- Caso de uso: é representado por uma elipse, com o nome dentro. Especifica um conjunto de ações executadas por um sistema.
- Include: é uma forma de interação, um caso de uso pode incluir outro. Esta relação indica uma obrigatoriedade, assim sempre que o primeiro ocorrer o que está incluído deverá ocorrer.
- Extend: é outra forma de interação entre casos de uso. Esta relação indica que este caso de uso pode ou não ser estendido.

### 2.3.2 Diagrama de Classes

O diagrama de classes denota a estrutura estática de um sistema e as classes representam o que vai ser manipulado por esse sistema. Uma classe é a descrição dos objetos criados a partir daquela classe. Os diagramas de classes são importantes para visualização, especificação e documentação do sistema. BOOCH, RUMBAUGH, JACOBSON (2000)

O objetivo de fazer um diagrama de classes é:

- Indicar quais abstrações fazem parte do sistema e quais estão fora do sistema.
- Mostrar como as classes trabalham juntas, fazendo assim com que tenha uma maior compreensão das classes.

- Fazer a modelagem do esquema lógico de um banco de dados.

### 2.3.3. Diagrama de Sequência

O diagrama de sequência, também chamado de diagrama de realização dos casos de uso, descreve as mensagens passadas entre os objetos num programa de computador. Esse diagrama tem como objetivo principal representar graficamente o comportamento de uma funcionalidade, considerando a interação entre todos os componentes de software relacionados ao seu uso. O diagrama é composto por atores, objetos, uma linha de vida e as mensagens que os objetos transmitem.

### 2.3.4. Diagrama de Entidade e Relacionamento

O diagrama de entidade e relacionamento utiliza elementos gráficos para demonstrar o modelo de dados de um sistema com alto nível de abstração. O DER parte do princípio que o mundo real é composto por conjuntos de objetos chamados de entidades e relacionamentos entre esses objetos. NAVATHE (2005)

### 3. Levantamento de Requisitos

Neste capítulo será abordado os métodos utilizados para o levantamento de requisitos do sistema e os resultados dos requisitos encontrados.

A etapa de concepção do desenvolvimento desse projeto foi realizada em três partes. A primeira delas foi realizada com o método de questionário com os alunos da faculdade, para levantar os principais requisitos do sistema e a necessidade do projeto. Na segunda parte foi realizado a análise das Atas de avaliação de curso, procurando ano a ano as necessidades dos alunos do curso e suas reclamações. Na última parte foi a realização de uma entrevista com o responsável pelo laboratório de informática da FT - Unicamp, com essa coleta analisamos o ponto de vista dos responsáveis pela gestão do laboratório.

#### 3.1 Questionário

A primeira técnica utilizada é o questionário, que consiste em uma técnica de levantamento que permite ao engenheiro de software obter respostas de várias pessoas interessadas no sistema. Um dos objetivos de utilizar um questionário como levantamento de requisitos é poder examinar uma grande amostra de usuários do sistema e entender os problemas ou levantar questões importantes, antes de se programar para entrevistas. KENDALL (1992)

O questionário foi disponibilizado aos alunos via o grupo da FT-Unicamp no Facebook e informado que a participação seria feita de forma voluntária e anônima. O resultado foi de trinta e cinco respostas. As perguntas feitas e os resultados obtidos encontram-se abaixo.

A primeira pergunta do questionário buscava conhecer o curso do público que está respondendo ao questionário.

#### Qual curso você faz?

35 respostas

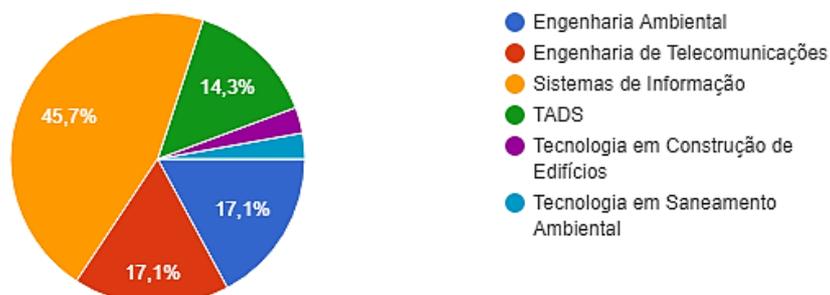


Figura 3 - Pergunta 1 do Questionário

A segunda pergunta busca levantar os requisitos dos principais problemas que o usuário enfrenta ao utilizar um computador nos laboratórios de informática da FT, tentando buscar dos usuários os atributos para a criação de uma solicitação de serviço. O usuário tinha a possibilidade de escolher mais de uma opção dentro do questionário e também poderia citar algum outro problema que enfrentou ao utilizar o laboratório.

#### Quais os principais problemas que você encontra quando utiliza um computador na FT?

35 respostas

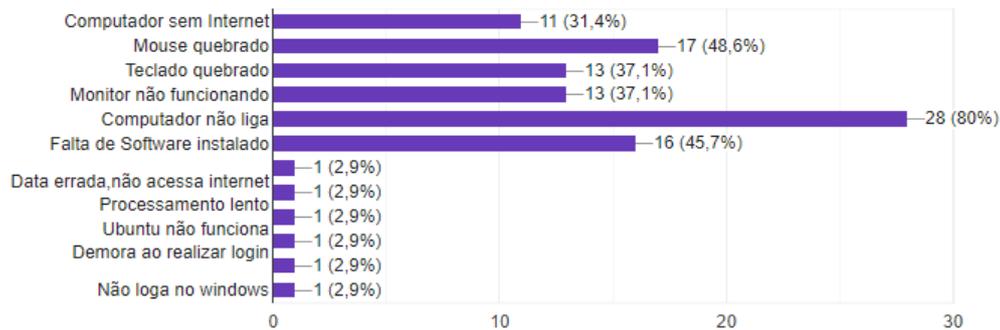


Figura 4 - Pergunta 2 do Questionário

Na opção que o usuário poderia digitar foram citados alguns problemas como:

- Computadores que inicializam muito devagar;
- Computador apresentando data incorreta, o que não permite utilizar a Internet;
- Um dos sistemas operacionais não está funcionando, por exemplo quando o Windows funciona e o Linux não;
- Erro ao fazer login no sistema do Windows;
- Processamento do computador lento.

A terceira pergunta do questionário buscava entender o que o usuário busca fazer quando se depara com algum dos problemas. Qual saída ele tem disponível ou conhecimento do que ele pode fazer. Existe um site com funções semelhantes com às propostas desse projeto, mas só está disponível para professores e funcionários da faculdade.

Atualmente, o que você faz quando tem algum problema no computador?

35 respostas

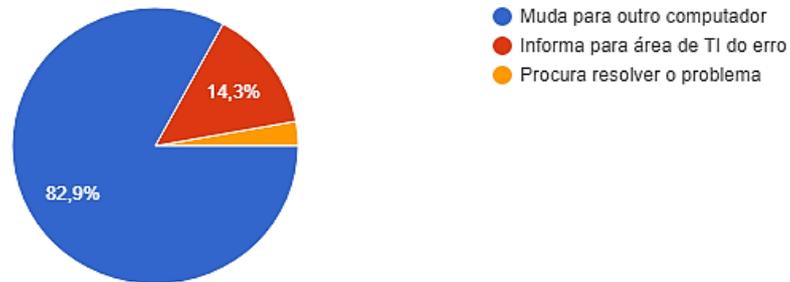


Figura 5 - Pergunta 3 do Questionário

A quarta pergunta teve o objetivo de entender do usuário qual a melhor forma dele poder relatar um problema:

Na sua opinião, qual é a melhor forma para relatar um problema?

35 respostas

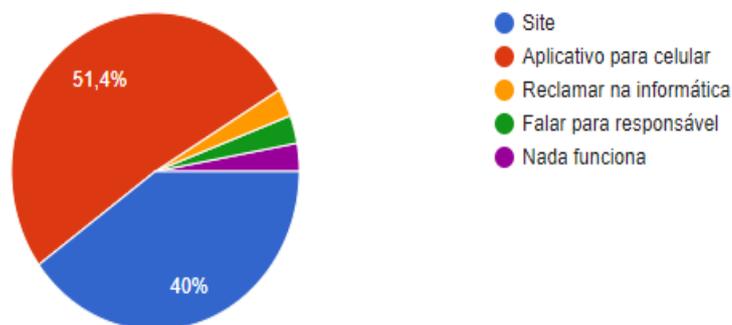


Figura 6 - Pergunta 4 do Questionário

A última pergunta teve o objetivo de saber se o usuário acha necessário um sistema para poder relatar quando se depara com um problema. Procurando verificar se o projeto é necessário:

Você acha importante um método para relatar o problema?

35 respostas

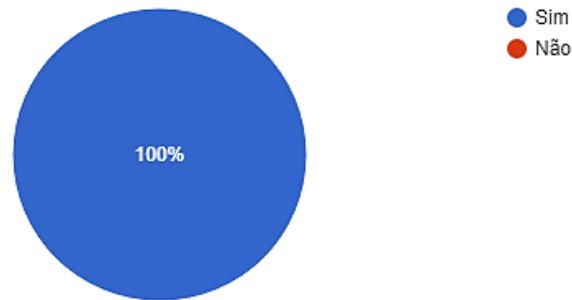


Figura 7 - Pergunta 5 do Questionário

### 3.2 Análise de Atas de Avaliação de Curso

Foi utilizado para o levantamento de requisitos as Atas de avaliação de curso dos cursos de Sistemas de Informação e Tecnologia em Análise e Desenvolvimento de Sistemas do ano de 2015 a 2018. Com a análise feita dos resultados foi possível rastrear novos requisitos para o sistema como:

- Ata 2015: Foi informado por alunos que existe uma carência de laboratórios de informática, também foi solocitado uma melhoria de informação sobre a utilização dos laboratórios, porque muitas vezes são requisitados por professores assim gerando uma falta de recursos.
- Ata 2016: Foi relatado pelos alunos problemas existentes nos laboratórios como: problemas com a rede LINUX, problemas para realizar o login e a falta de softwares instalados. Neste ano também foi solicitado por um professor que os alunos realizassem uma pesquisa dos softwares faltantes nos equipamentos e que fossem enviados periodicamente para coordenação de curso.
- Ata 2017: Neste ano o diretor de Informática, Ricardo Henrique Cândido abordou os tópicos problemáticos afetos à informática. O diretor comentou sobre os encaminhamentos já realizados, problemas solucionados e os que ainda persistem.
- Ata 2018: Foi informado por alunos as deficiências encontradas nos laboratórios, afirmando que muitos equipamentos apresentam problemas com o Sistema Operacional. Um dos professores comentou que existem dificuldades para acompanhar os problemas que surgem, porque existem três turnos de atividades.

Foi solicitado que os alunos entrem em contato diretamente com a informática por meio de um formulário de Solicitação de Serviço. Os alunos que estavam presentes comentaram que desconheciam a informação que existe esse formulário, com isso surgiram ideias para informar esses problemas como: fixar nos laboratórios um comunicado informando os canais de comunicação, sugeriram utilização de uma ficha de controle para cada computador e também foi sugerido a criação de um aplicativo para a manutenção dos laboratórios.

### 3.3 Entrevista

Por último, o método escolhido foi a entrevista. Uma entrevista é uma conversa direcionada que obtém um propósito e utiliza o formato de pergunta-resposta. É importante em uma entrevista que o engenheiro de software construa rapidamente uma base de confiança com o entrevistado.

A entrevista foi realizada com o atual responsável pelos laboratórios da FT no dia 14 de fevereiro de 2018. Ela teve duração de 20 minutos. Foram listados abaixo os principais pontos coletados nessa fase da concepção.

- Atualmente, quando ocorre algum problema nos computadores do laboratório os alunos podem entrar em contato pela chave de e-mail do laboratório ou o aluno pode ir direto até a sala da equipe de TI.
- A frequência de solicitações para o laboratório varia dependendo da semana, não é algo constante.
- Quando um problema é relatado por um aluno, os integrantes da equipe de TI que estão disponíveis tentam resolver o na mesma hora. Se for um problema relacionado a hardware pode demorar um tempo maior devido a manutenção.
- Quando está ocorrendo algum evento na faculdade, como a semana da Tecnologia em Foco, os integrantes do laboratório ficam focados no atendimento para o evento. Assim podendo nessas semanas ocorrer uma demora ao resolver um problema.
- Atualmente, não existe nenhum aplicativo para gestão de solicitações do laboratório.

#### 4. Especificação do Sistema

Com base nos resultados coletados na fase de concepção, neste tópico será descrito os requisitos funcionais e não funcionais do sistema. Com o objetivo de auxiliar e guiar no desenvolvimento final do software proposto.

Na Figura 8 está o diagrama com os casos de uso que serão descritos na Seção 4.1.

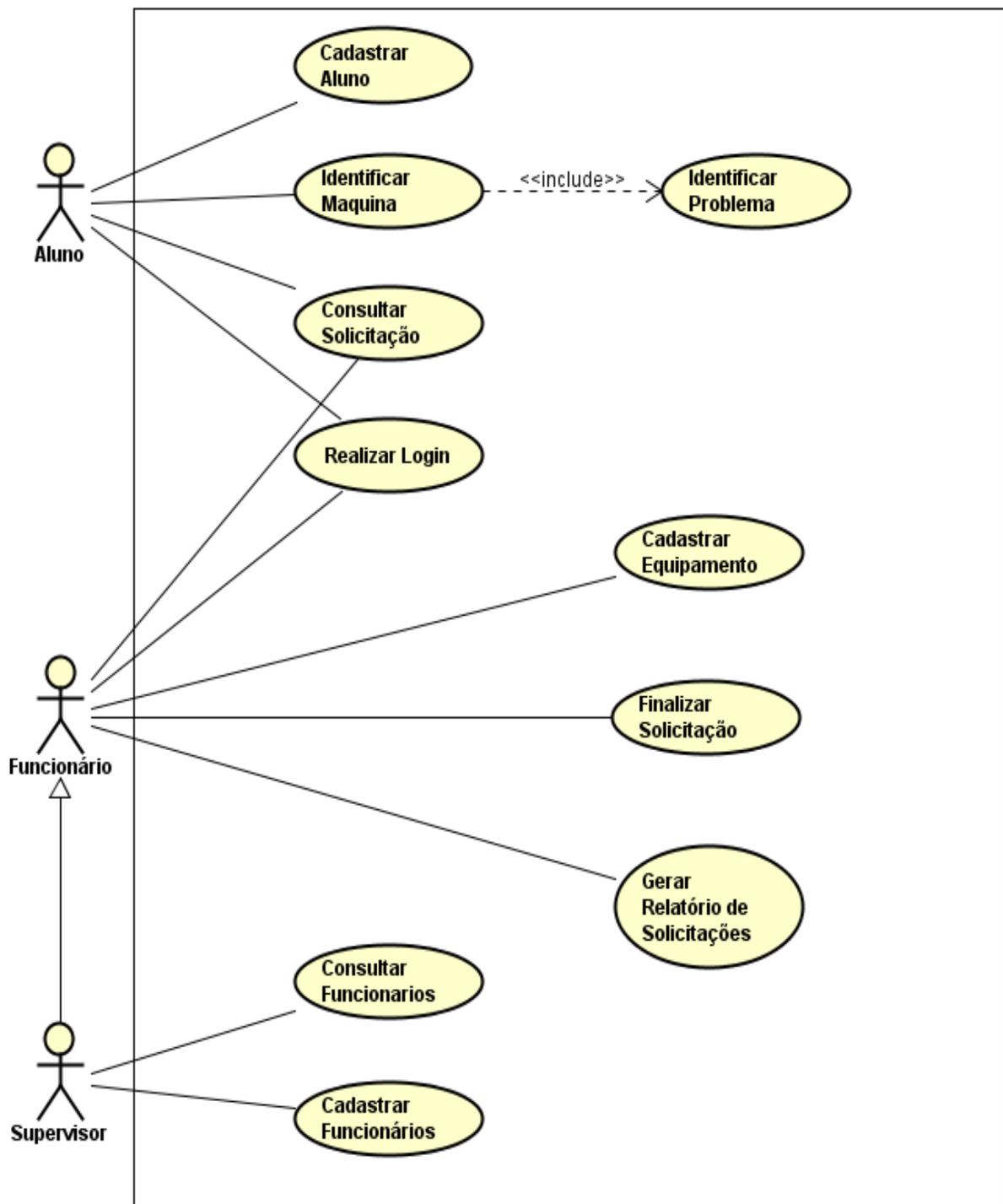


Figura 8 - Diagrama de Casos de Uso

#### 4.1.Requisitos Funcionais

Nesta seção, será apresentado as funcionalidades do sistema, ou seja, os casos de uso descritivos.

##### 4.1.1. Realizar Login

- a) Descrição: O usuário (aluno, funcionário ou supervisor) deverá fazer login para acessar o sistema.
- b) Pré-condição: Usuário não estar logado no sistema. Usuário ter conexão com a internet.
- c) Fluxo Básico:
  - Usuário inicializar o aplicativo.
  - Usuário digita seu login.
  - Usuário digita sua senha.
  - Usuário clica no botão “Entrar”.
- d) Fluxo Alternativo:
  - Usuário inicializar o aplicativo.
  - Usuário escolhe a opção de Logar com Gmail.
  - Sistema redireciona o usuário para validação via Gmail.
- e) Fluxos exceção:
  - FE1: Usuário ou senha digitados está incorreto.
    - Sistema retorna mensagem “Usuário e/ou senha incorretos”
    - Sistema retorna para a tela inicial.
  - FE2: Campos vazios.
    - Sistema retorna mensagem “Por favor, preencher todos os campos”.
    - Sistema retorna para a tela inicial.
  - FE3: Usuário não efetua a validação da entrada por Gmail.
    - Sistema retorna mensagem “Escolha outra opção de Login ou tente novamente”

##### 4.1.2. Identificar Problema

- a) Descrição: O aluno deverá cadastrar um problema encontrado no laboratório.
- b) Pré-condição: Usuário estar logado no sistema e identificar o equipamento.
- c) Fluxo Básico:
  - Usuário localiza a equipamento por *QR Code ou Asset Tag*.
  - Usuário seleciona a opção “Identificar Problema”.
  - Sistema redireciona o usuário para Tela Identificar Problema.
  - Usuário seleciona a opção que contém o problema ocorrido ou digita o problema.
  - Usuário seleciona a prioridade do problema (alta, média ou baixa).
  - Usuário clica no botão “Enviar Solicitação”.
  - Sistema verificar nível de criticidade da Solicitação.
  - Sistema envia código da Solicitação para o e-mail do Aluno.
  - Sistema envia notificação de nova Solicitação para os Funcionários.
  - Sistema redireciona o usuário para Tela de Solicitação.

##### 4.1.3. Identificar Equipamento

- a) Descrição: O aluno deverá escolher se localiza o equipamento por *QRCode ou por Asset Tag*.

- b) Pré-condição: Usuário logado.
- c) Fluxo Básico:
  - Usuário entra na Tela Gerar Solicitação.
  - Usuário seleciona a opção “Identificar por *Asset Tag*”.
  - Usuário digita o código do equipamento.
  - Usuário clica no botão “Enviar.
  - Sistema redireciona usuário para tela Identificar Problema.
- d) Fluxo Alternativo:
  - Usuário entra no menu principal
  - Usuário seleciona a opção “Scan *QRCode*”
  - Usuário localiza o código pela câmera do celular”
  - Sistema redireciona usuário para Tela Identificar Problema
- e) Fluxo de Exceção:
  - Sistema não localiza equipamento pelo *QRCode ou Asset Tag*.
  - Sistema retorna a mensagem “Verificar *Asset Tag* da equipamento”.

#### 4.1.4. Consultar Solicitação

- a) Descrição: O usuário deverá digitar o código da solicitação que criou no sistema, para os dados serem apresentados.
- b) Pré-condição: Usuário logado. Existir uma solicitação criada.
- c) Fluxo Básico:
  - Usuário entra na Tela de Aluno/Tela Funcionários.
  - Usuário seleciona a opção “Consultar Solicitação”.
  - Usuário digita o código da solicitação.
  - O sistema redireciona o usuário para Tela Consultar Solicitação e apresenta as informações solicitação.
- d) Fluxo de Exceção:
- f) FE1: Erro de conexão com o banco de dados
  - Sistema retorna a mensagem “Erro de conexão. Tente novamente mais tarde”.
  - Sistema retorna para o menu principal.

#### 4.1.5. Finalizar Solicitação

- a) Descrição: O usuário (supervisor ou funcionário) deverá digitar o código da solicitação que pretende finalizar.
- b) Pré-condição: Usuário logado. Existir uma solicitação criada.
- c) Fluxo Básico:
  - Usuário entra na Tela de Funcionários.
  - Usuário seleciona a opção “Finalizar Solicitação”.
  - Usuário digita o código da solicitação.
  - Usuário clica no botão “Finalizar”.
  - Sistema retorna a mensagem “Solicitação finalizada com sucesso”
- d) Fluxo de Exceção:
- g) FE1: Erro de conexão com o banco de dados
  - Sistema retorna a mensagem “Erro de conexão. Tente novamente mais tarde”.
  - Sistema retorna para a tela inicial.

#### 4.1.6. Gerar Relatório de Solicitações

- a) Descrição: O usuário (supervisor ou funcionário) deverá selecionar o tempo escolhido para geração do relatório de solicitações.
- b) Pré-condição: Usuário logado.
- c) Fluxo Básico:
  - Usuário entra na Tela de Funcionários.
  - Usuário seleciona a opção “Gerar Relatório de Solicitações”.
  - Usuário seleciona o período de tempo escolhido.
  - Usuário clica no botão “Gerar Relatório”
  - Sistema redireciona usuário para Tela Relatório Solicitação e apresenta as informações das solicitações no tempo selecionado.
- d) Fluxo de Exceção:
- h) FE1: Erro de conexão com o banco de dados
  - Sistema retorna a mensagem “Erro de conexão. Tente novamente mais tarde”.
  - Sistema retorna para a tela inicial.

#### 4.1.7. Cadastrar Aluno

- a) Descrição: O aluno deverá realizar o cadastro com seus dados.
- b) Fluxo Básico:
  - Usuário inicializa o aplicativo.
  - Usuário clica no botão “Cadastrar”.
  - Sistema redireciona o aluno para a Tela de Cadastro.
  - Usuário digita o RA.
  - Usuário digita uma senha e a verificação da senha.
  - Usuário seleciona o curso.
  - Usuário clica no botão “Cadastrar”
  - Sistema salva as informações no banco de dados.
  - Sistema retorna a mensagem “Usuário cadastrado com sucesso”
- c) Fluxo de Exceção:
- i) FE1: Erro de conexão com o banco de dados
  - Sistema retorna a mensagem “Erro de conexão. Tente novamente mais tarde”
  - Sistema retorna para a tela inicial.
- j) FE2: Senhas não coincidem.
  - Usuário digitou senhas que não são idênticas.
  - Sistema retorna a mensagem “Senhas digitadas não são idênticas”
  - Sistema retorna para Tela de Cadastro.

#### 4.1.8. Cadastrar Funcionários

- a) Descrição: O usuário (supervisor) deverá realizar o cadastro de novos funcionários.
- b) Pré-condição: Usuário logado.
- c) Fluxo Básico:
  - Usuário entra no Tela Supervisor.
  - Usuário seleciona a opção “Cadastrar Funcionários”.
  - Sistema redireciona o usuário para Tela Cadastro Funcionários.
  - Usuário preenche os campos necessários.
  - Usuário clica no botão “Cadastrar”.
  - Sistema salva as informações no banco de dados.

- Sistema apresenta mensagem “Usuário cadastrado com sucesso”
- d) Fluxo de Exceção:
- k) FE1: Erro de conexão com o banco de dados
- Sistema retorna a mensagem “Erro de conexão. Tente novamente mais tarde”
  - Sistema retorna para a tela cadastro de funcionários.

#### 4.1.9. Consultar Funcionários

- a) Descrição: O usuário (supervisor) deverá digitar o nome do funcionário para visualizar os dados do funcionário.
- b) Pré-condição: Usuário logado. Funcionário Cadastrado.
- c) Fluxo Básico:
- Usuário entra no Tela Supervisor.
  - Usuário seleciona a opção “Consultar Funcionários”.
  - Usuário digita o nome do funcionário.
  - Usuário clica no botão “Consultar”
  - Sistema redireciona o usuário para Tela Consultar Funcionários e apresenta as informações do funcionário.
- d) Fluxo de Exceção:
- FE1: Erro de conexão com o banco de dados.
    - Sistema retorna a mensagem “Erro de conexão. Tente novamente mais tarde”
    - O sistema retorna para o menu supervisor.

#### 4.1.10. Cadastrar Equipamento

- a) Descrição: O usuário (funcionário ou supervisor) deverá cadastrar os dados de um novo equipamento.
- b) Pré-condição: Usuário logado.
- c) Fluxo Básico:
- Usuário entra na Tela de Funcionários.
  - Usuário seleciona a opção “Cadastrar Hardware”.
  - O sistema redireciona o usuário para Tela de Cadastro de Hardware.
  - Usuário preenche os dados do novo equipamento.
  - Usuário clica para gerar o QRCode.
  - Usuário clica em “Cadastrar Hardware”.
  - Sistema retorna a mensagem de “Equipamento cadastrado com sucesso”
- d) Fluxo de Exceção:
- FE1: Erro de conexão com o banco de dados.
    - Sistema retorna a mensagem “Erro de conexão. Tente novamente mais tarde”
    - O sistema retorna para a Tela de Funcionários.

#### 4.2.Requisitos Não Funcionais

- Interoperabilidade: O sistema precisa ter acesso à internet para o envio de informações e conexão com o banco de dados.
- Portabilidade: O sistema tem que ser multi plataforma. É importante que o sistema possa ser utilizado em Android e iOS.

## 5. Modelagem do Sistema

Nessa seção será apresentada a etapa de construção, essa etapa foi realizada com os diagramas de sequência, diagrama de classes e diagrama de entidade-relacionamento. Esses diagramas são parte da documentação do sistema e o modelo de dados.

O diagrama de classe é de extrema importância, porque nele está mapeado toda a estrutura do sistema com suas classes e relacionamentos entre si, contendo seus atributos e métodos. Está apresentado na Figura 9 abaixo:

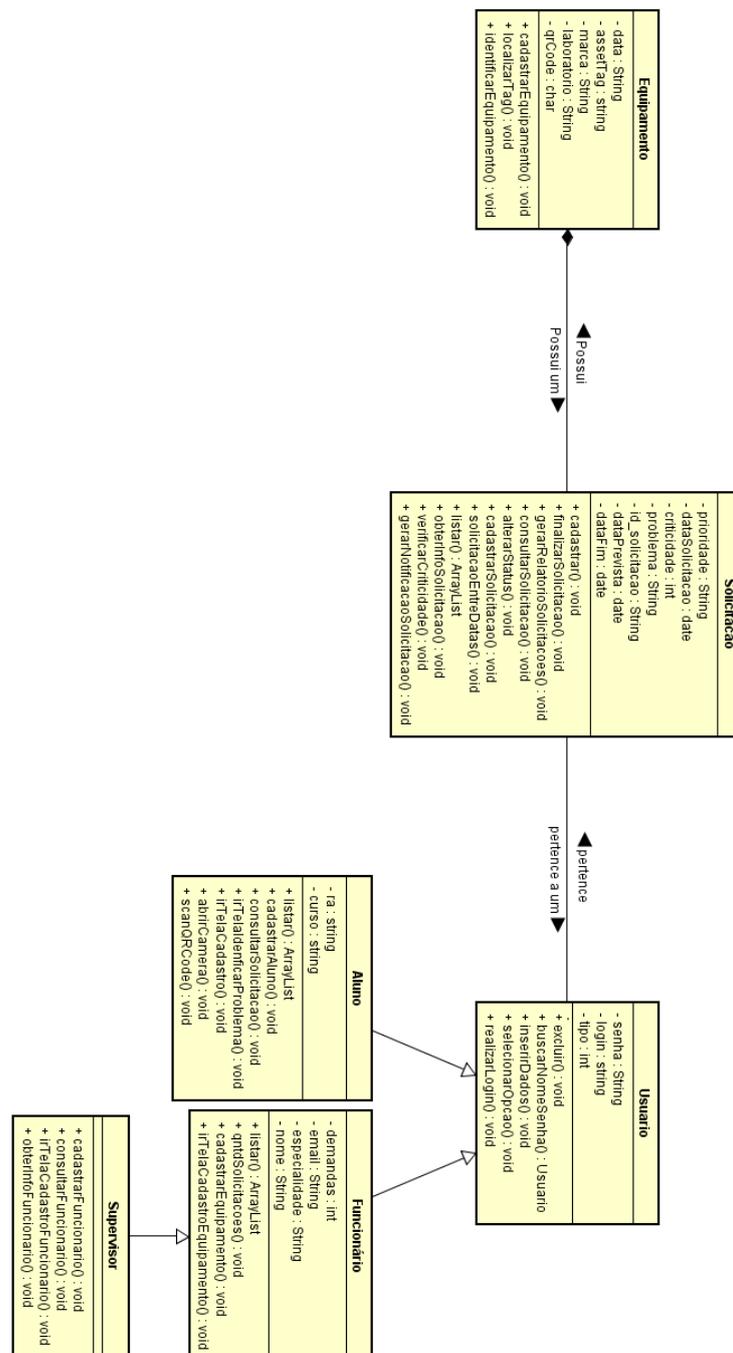


Figura 9 - Diagrama de Classes

Os diagramas de sequência fazem parte da documentação da aplicação, porque auxiliarão a observar quais são os comportamentos que os objetos deverão possuir durante a execução do sistema que será desenvolvido. Estão apresentados abaixo:

Diagrama de sequência do caso de uso Realizar Login, apresentado na Figura 10.

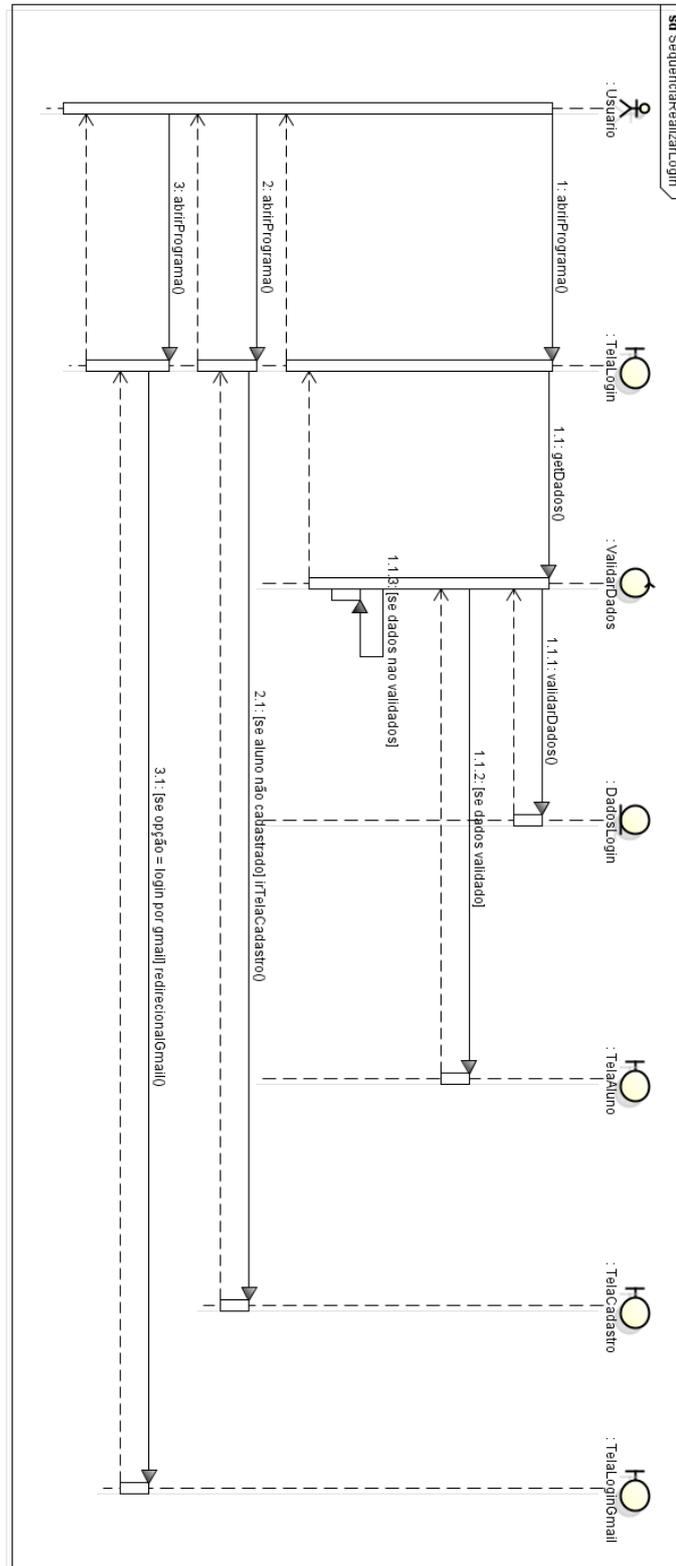


Figura 10 – Diagrama de Sequência - Realizar Login

Diagrama de seqüência do caso de uso Cadastrar Aluno, apresentado na Figura 11.

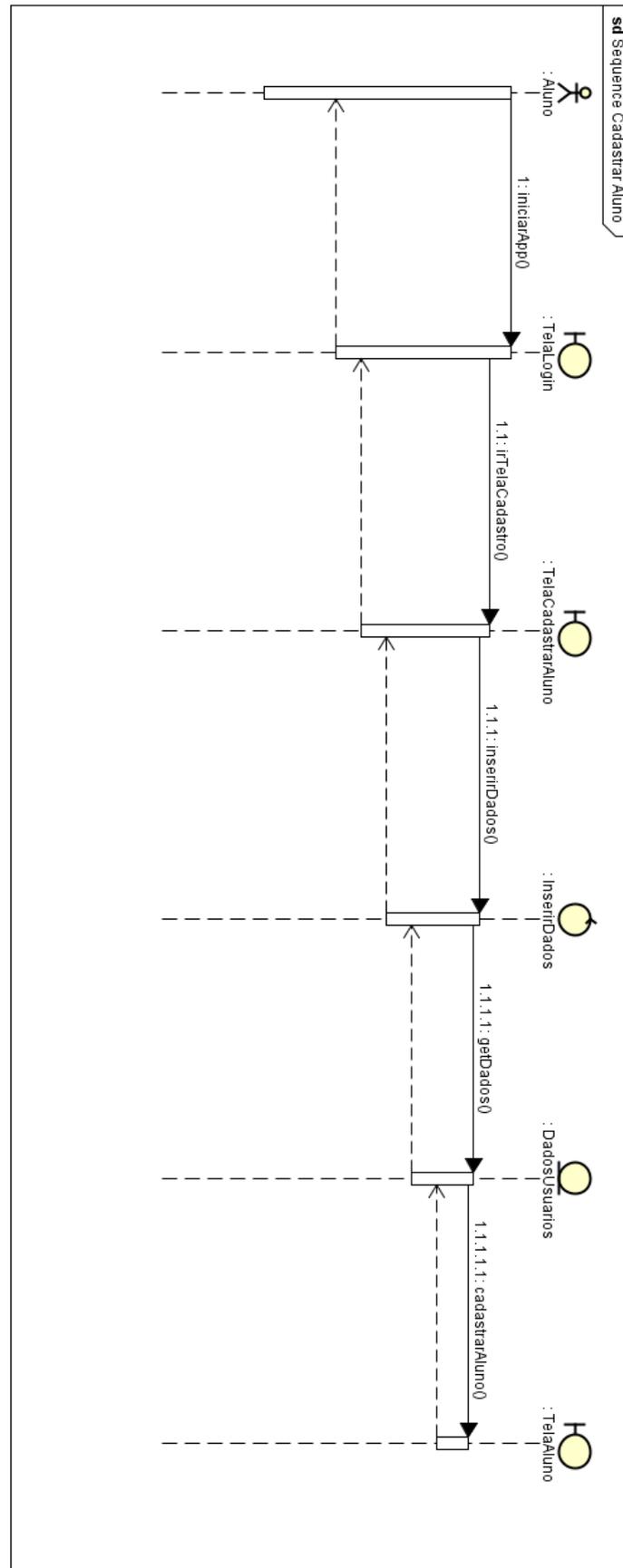


Figura 11 – Diagrama de Sequência - Cadastrar Aluno

12. Diagrama de seqüência do caso de uso Cadastrar Equipamento, apresentado na Figura

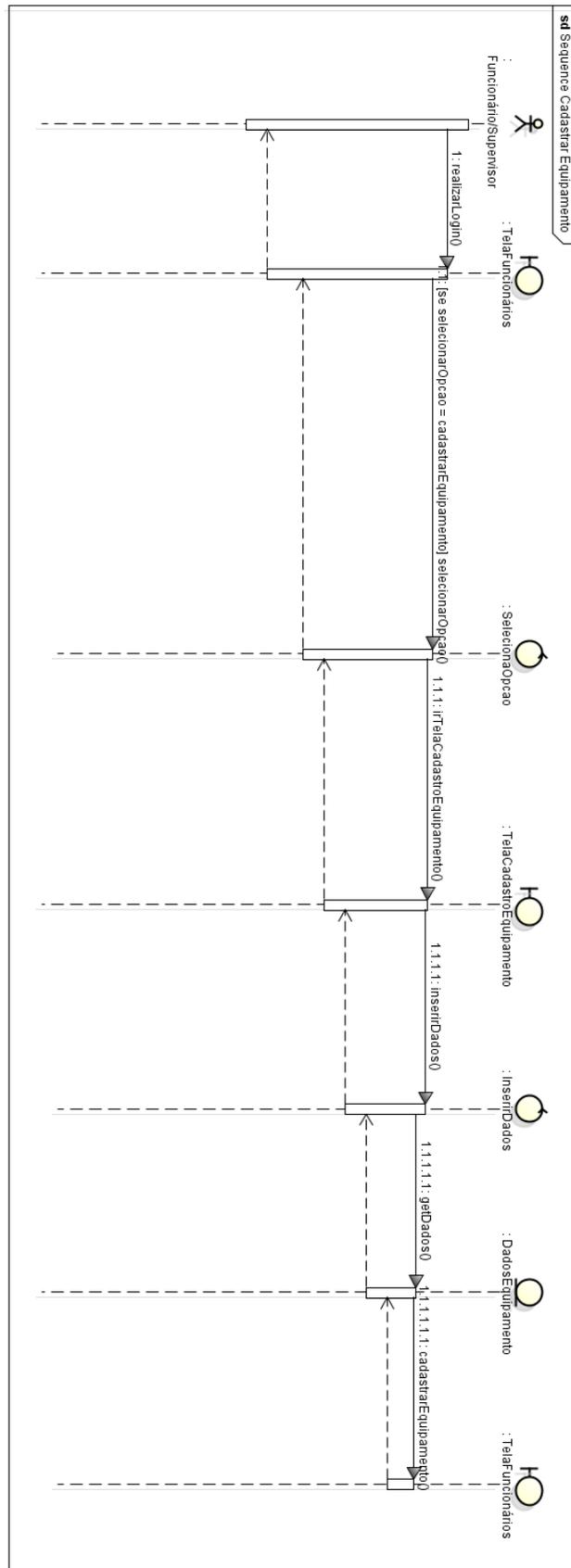


Figura 12 – Diagrama de Seqüência - Cadastrar Equipamento

13. Diagrama de seqüência do caso de uso Cadastrar Funcionário, apresentado na Figura

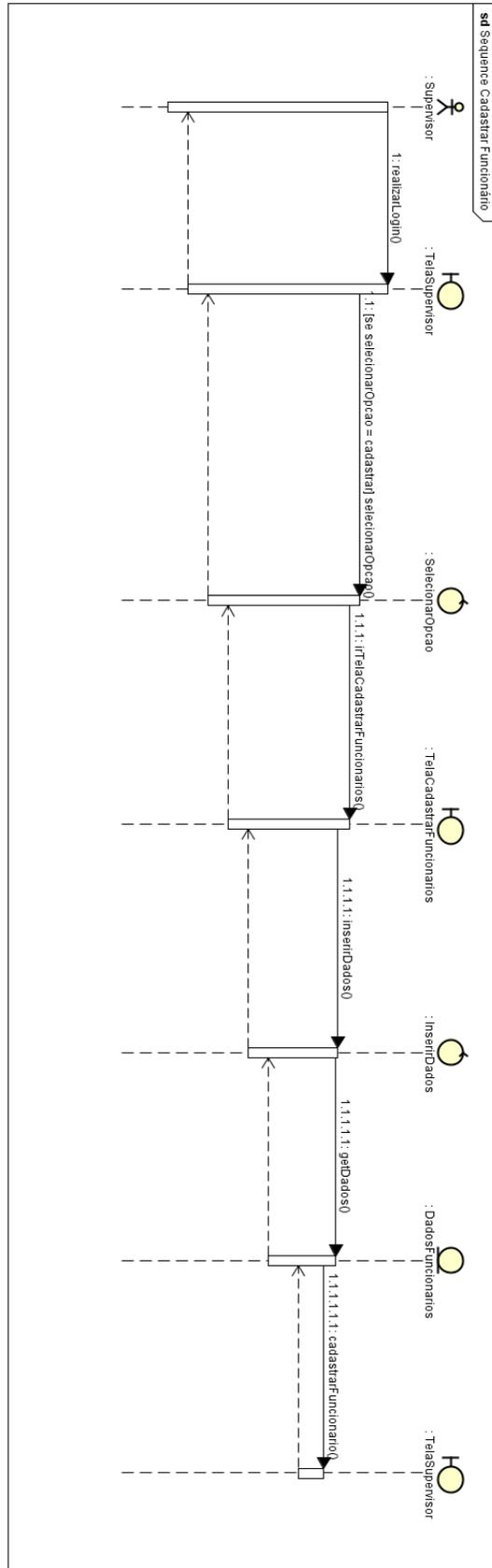


Figura 13 - Diagrama de Seqüência - Cadastrar Funcionários

14. Diagrama de seqüência do caso de uso Consultar Funcionários, apresentado na Figura

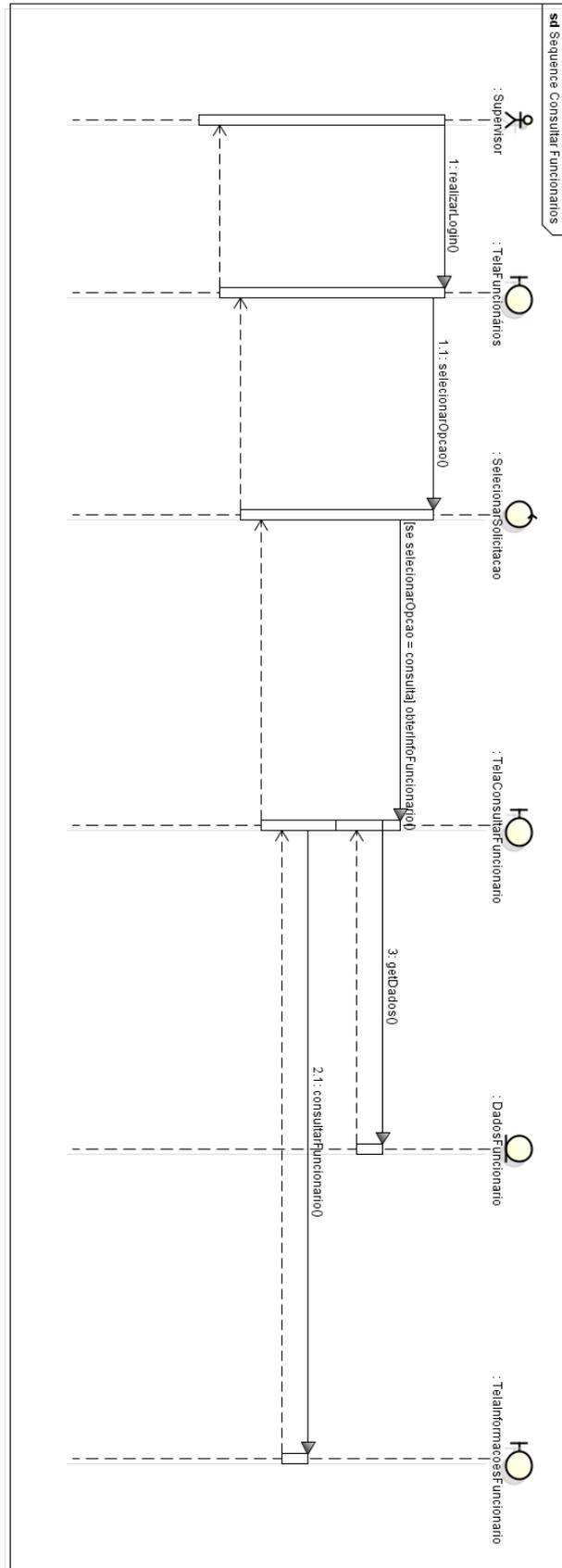


Figura 14 – Diagrama de Seqüência - Consultar Funcionários

Diagrama de seqüência do caso de uso Gerar Relatório de Solicitações, apresentado na Figura 15.

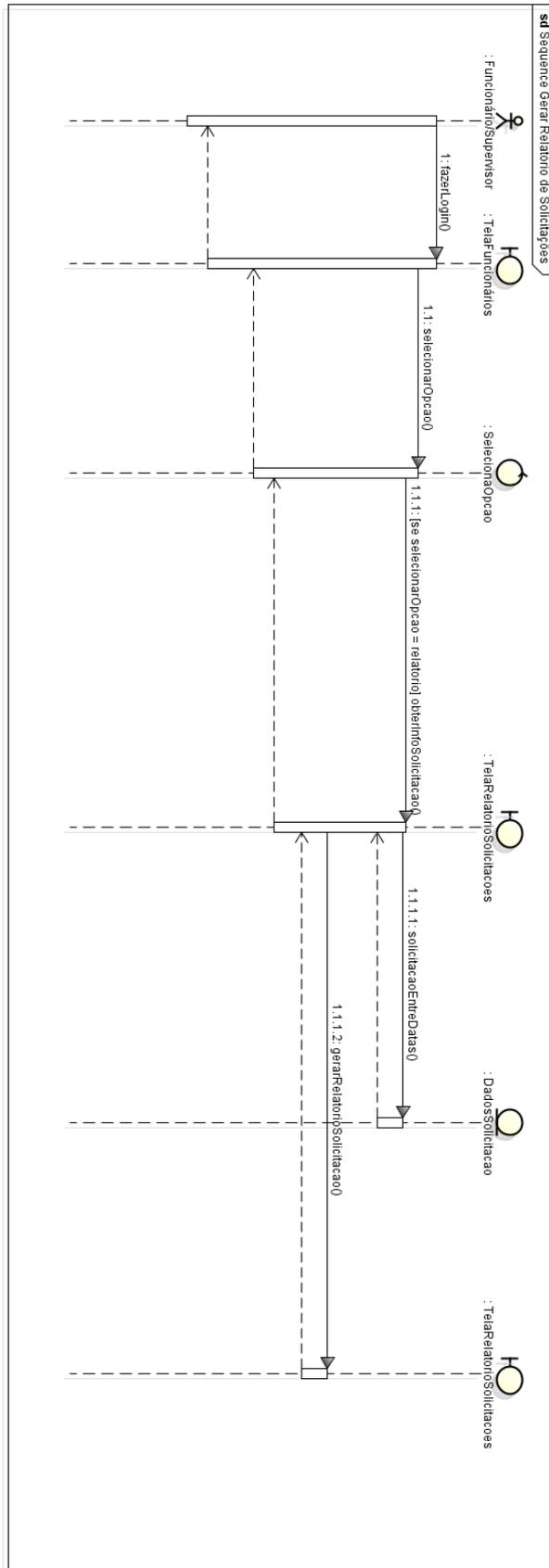


Figura 15 - Diagrama de Seqüência - Gerar Relatório de Solicitações

Diagrama de seqüência do caso de uso Finalizar Solicitação, apresentado na Figura 16.

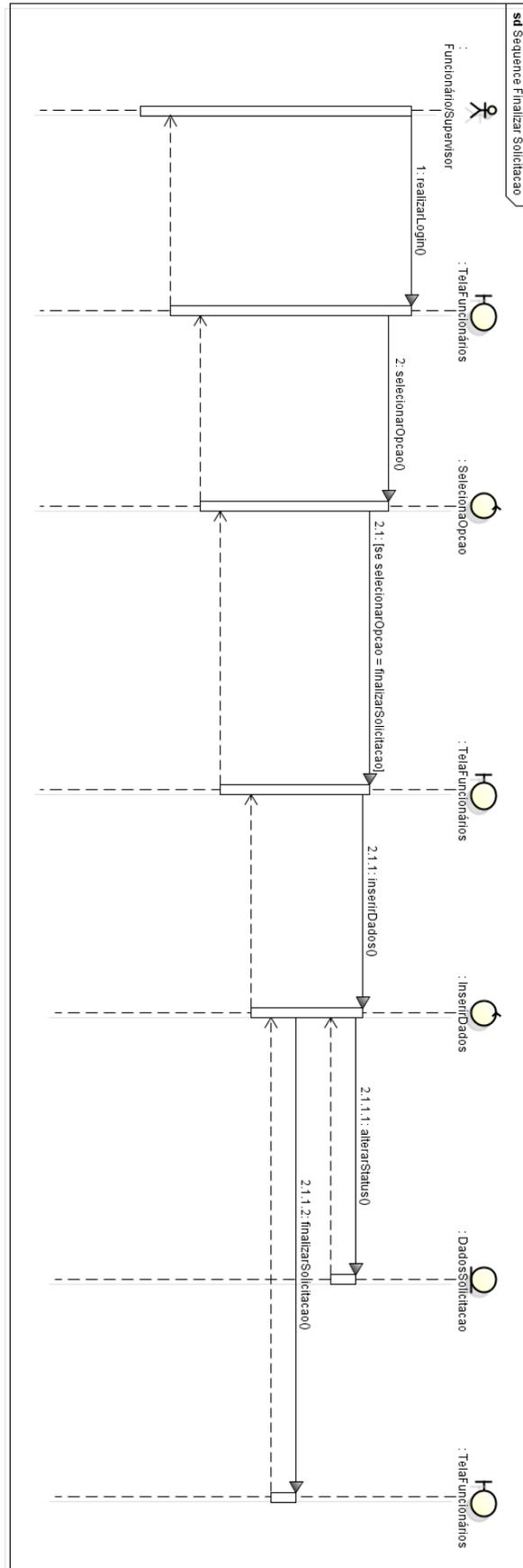


Figura 16 - Diagrama de Seqüência - Finalizar Solicitação

Diagrama de sequência do caso de uso Consultar Solicitação, apresentado na Figura 17.

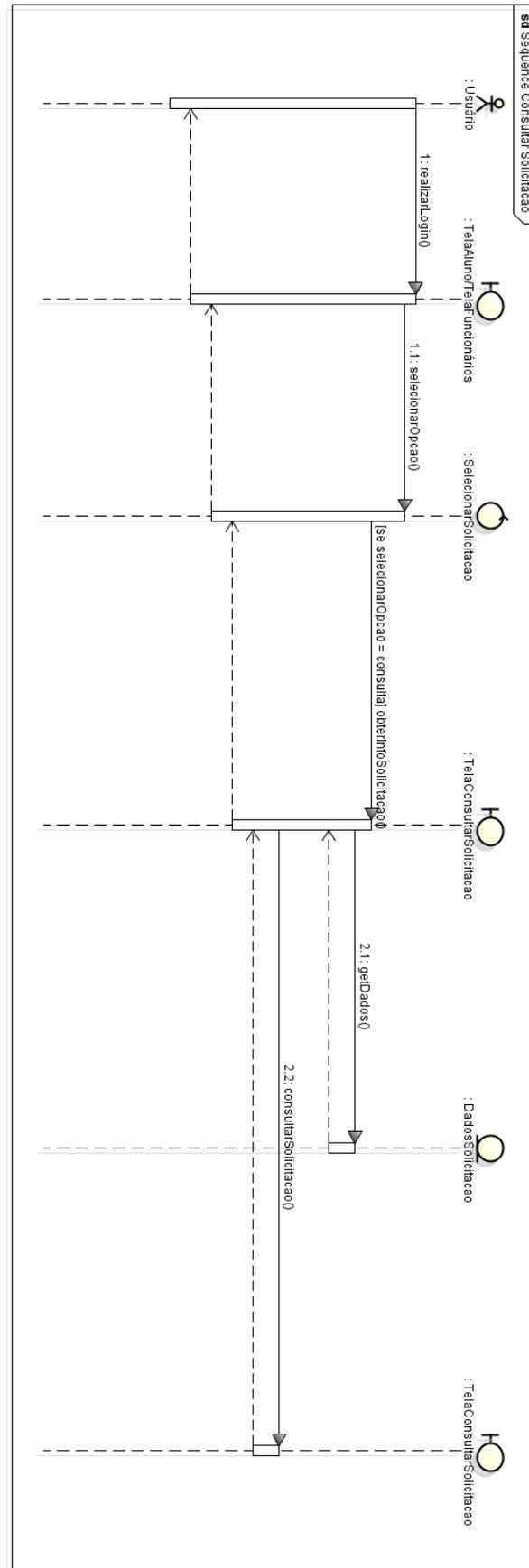


Figura 17 - Diagrama de Sequência - Consultar Solicitação

Diagrama de seqüência do caso de uso Identificar Problema, apresentado na Figura 18.

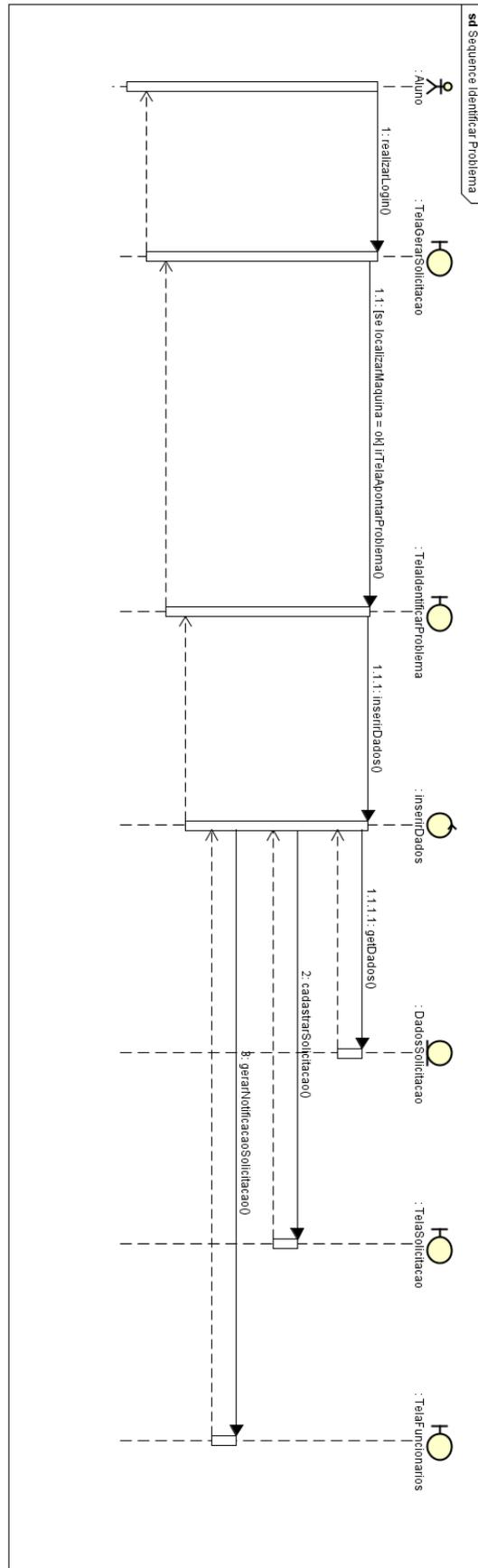


Figura 18 - Diagrama de Sequência - Identificar Problema

19. Diagrama de seqüência do caso de uso Identificar Equipamento, apresentado na Figura

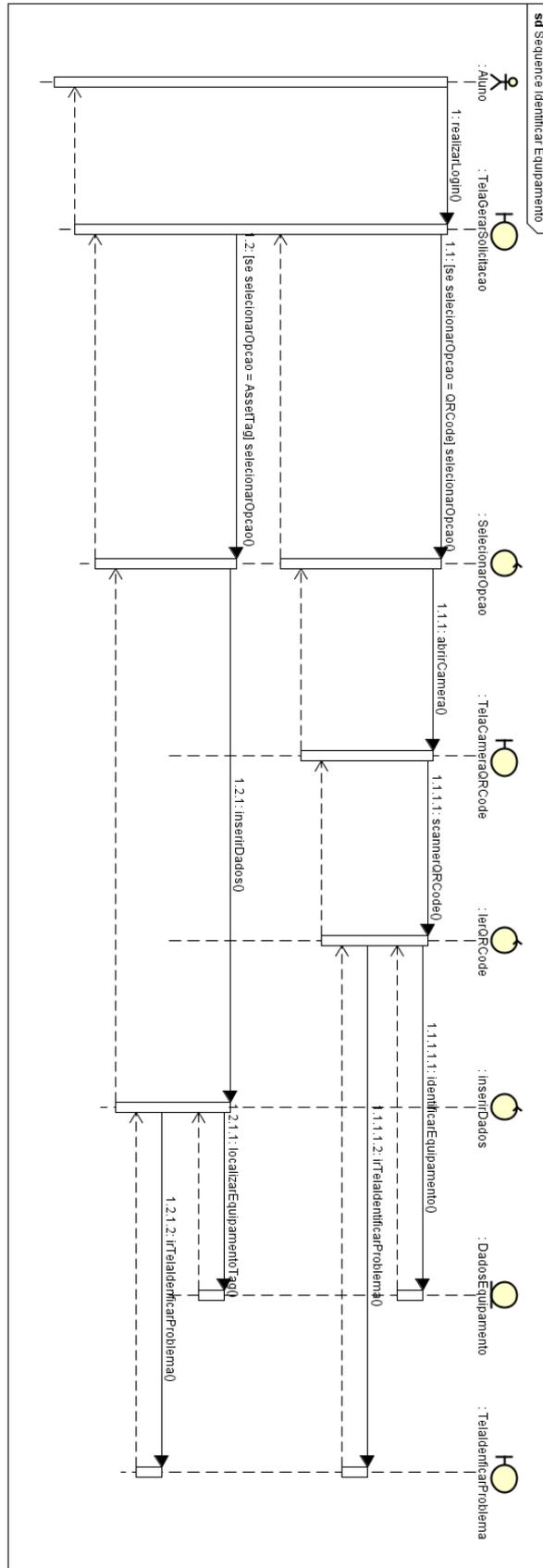


Figura 19 - Diagrama de Seqüência - Identificar Equipamento

O diagrama de entidades e relacionamentos realizado para demonstrar o relacionamento entre os objetos, é apresentado na Figura 20

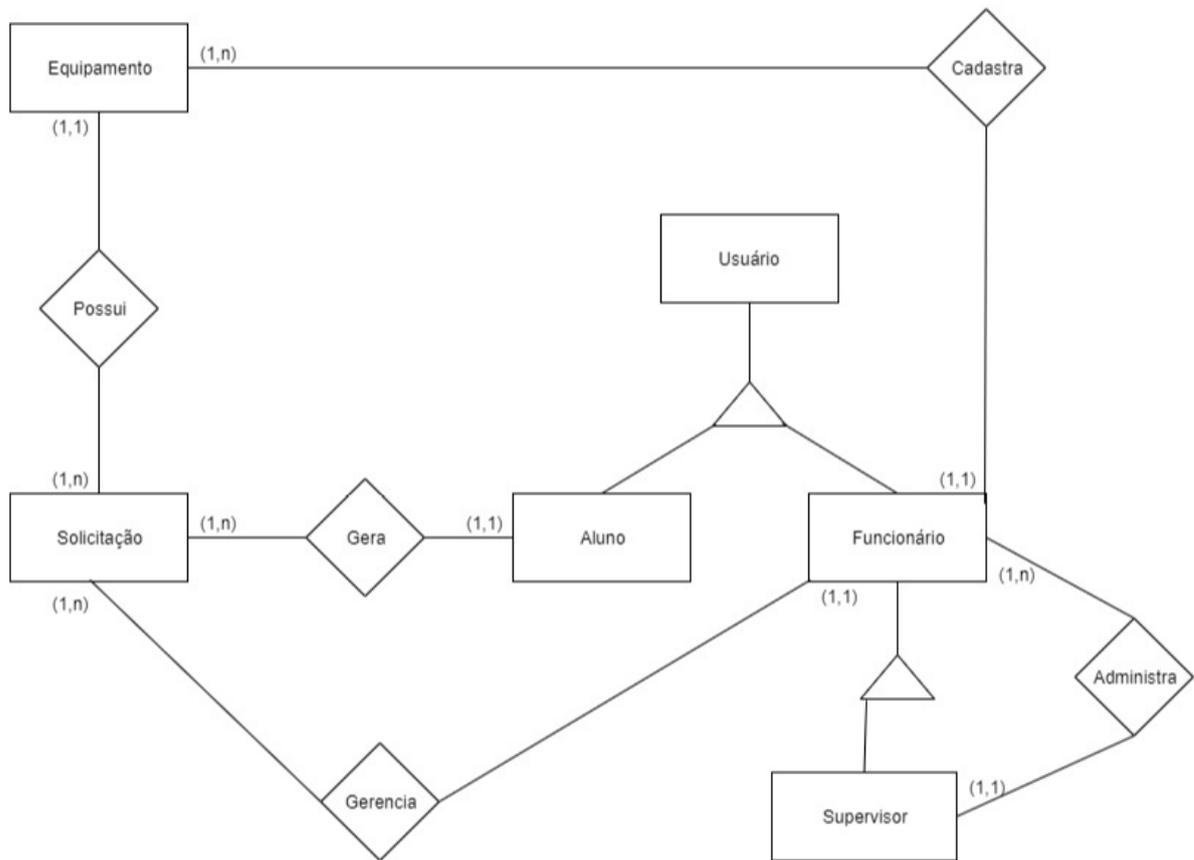


Figura 20 - Diagrama de Entidade e Relacionamento

## 6. Protótipo de Telas

Com base na análise realizada nas seções anteriores, foi criado um protótipo com objetivo de apresentar uma representação do sistema. A criação desse protótipo nos permite identificar oportunidades de melhorias no sistema e identificar algum problema antes de uma implementação final do sistema.

Para realizar a criação desse protótipo foi utilizado o software MIT App Inventor, que é uma ferramenta de programação baseada em blocos que permite desenvolver aplicativos para telefones Android. As telas do protótipo serão apresentadas abaixo:



Figura 21 – Protótipo - Tela de Login



Figura 22 – Protótipo - Tela de Cadastro de Aluno



Figura 23 – Protótipo -Tela de Gerar Solicitação

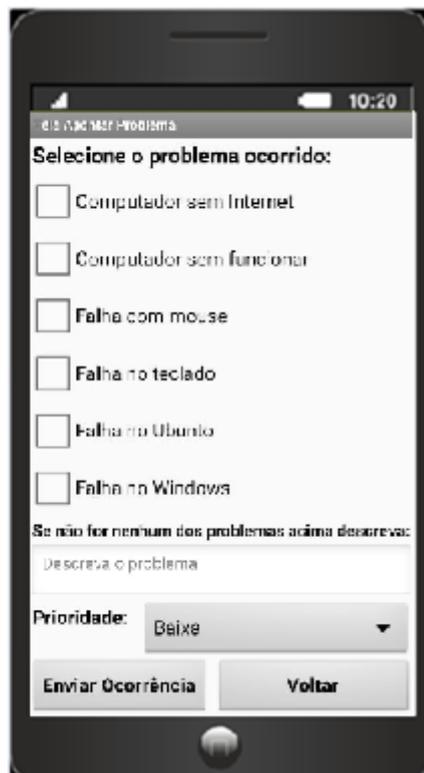


Figura 24 – Protótipo -Tela de Identificar Problemas



Figura 25 – Protótipo - Tela de Funcionários



Figura 26 – Protótipo - Tela de Consultar Solicitação

Tela Cadastro de Funcionário

Nome do funcionário:  
Nome

Senha:  
Senha

Confirme sua senha:  
Confirmação de senha

Email do funcionário:  
Email

Especialidade: Adm de Redes

Cadastrar Voltar

Figura 27 – Protótipo - Tela de Cadastrar Funcionários

Tela Consultar Solicitação

Código da Solicitação: 915017

Data de Abertura: 10/04/2019  
Previsão de Conclusão: 11/04/2019

Equipamento (TAG): LP01/1288

Criticidade: Baixa

Problema:  
Computador sem Internet

Status da Solicitação: Concluída

Voltar

Figura 28 – Protótipo - Tela de Consultar Solicitação Funcionários

Tela Consultar Funcionário

Nome do Funcionário: Rafael M

Email: rafaelm@ft.unicamp.br

Quantidade de Demandas atendidas:  
4 demandas

Código das demandas: 971213

Voltar

Figura 29 – Protótipo - Tela de Consultar Funcionários

Ocorrência gerada

Ocorrência Gerada com sucesso:

915017

Sair

Figura 30 – Protótipo - Tela de Solicitação Gerada

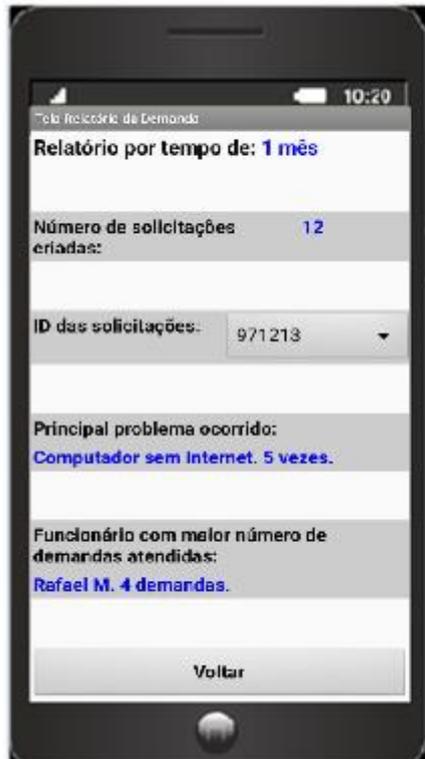


Figura 31 – Protótipo - Tela de Relatório de Solicitações



Figura 32 – Protótipo - Tela de Cadastrar Equipamento

A Figura 33 representa como é feito o back-end da programação por blocos do aplicativo para a tela de login. Na primeira etapa do código o sistema recebe as informações digitadas pelo usuário, já na segunda etapa o sistema faz a verificação se o usuário já está cadastrado. Na terceira parte do código é a chamada de uma função para direcionar a outra tela.

```
inicializar global id para "1uyzHD1uVowZ2_TrVjbb2zFPmlM_xFfPWLjwOmh9S"

quando btnLogin .Clique
fazer
  ajustar ControleDeFusiontables1 . Consulta para
  juntar "SELECT * FROM "
  obter global id
  "WHERE Login = "
  txtLoginUsuario . Texto
  "'AND Senha = "
  txtLoginSenha . Texto
  "'AND Tipo = "
  1
  "1"
  chamar ControleDeFusiontables1 . EnviarConsulta

quando ControleDeFusiontables1 . RecebeuResultado
resultado
fazer
  se
  é vazio? txtLoginUsuario . Texto
  então
  chamar ntfVazio . MostrarAlerta
  aviso "Campos Vazios"
  senão, se
  contém texto obter resultado
  parte txtLoginUsuario . Texto
  então
  abrir outra tela nomeDaTela "MenuPrincipal"
  senão
  chamar ntfIncorretos . MostrarAlerta
  aviso "Usuário e/ou Senha Incorretos"

quando btnCadastro .Clique
fazer
  abrir outra tela nomeDaTela "TelaCadastro"
```

Figura 33 - Programação por Blocos Back-end

Na Figura 34 é apresentado como é realizado o front-end pelo MIT app inventor.

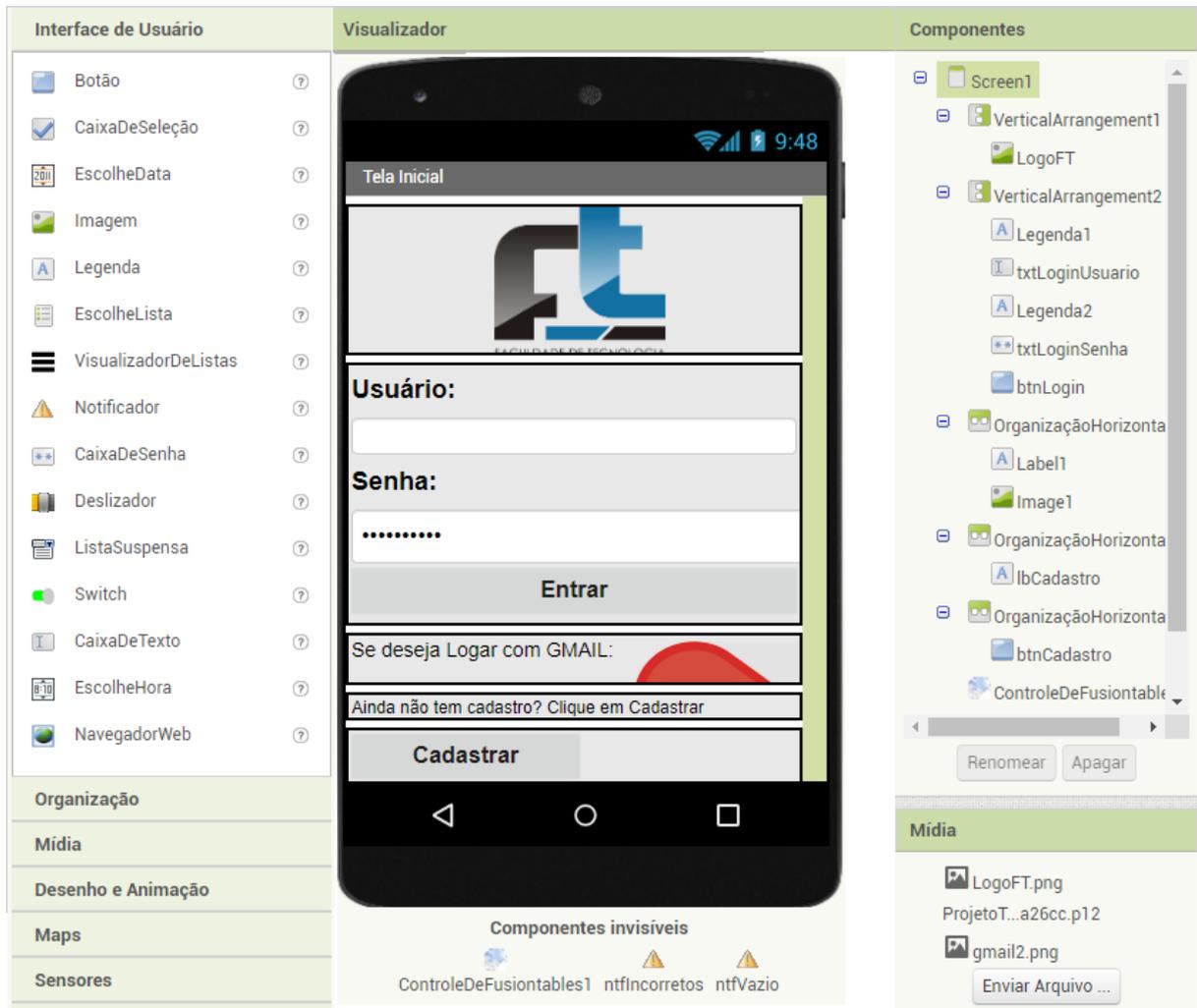


Figura 34 - Programação por Blocos Front-end

## 7. Resultados e Considerações Finais

O desenvolvimento do presente estudo apresentou uma proposta de um aplicativo *mobile* em que os alunos possam informar os problemas enfrentados ao utilizar os laboratórios de informática da faculdade. Além disso, permite que o supervisor do laboratório monitore os chamados atendidos por seus funcionários.

Para alcançar esses resultados, foram utilizados três métodos de levantamento de requisitos. O questionário disponibilizado para resposta dos alunos conseguiu demonstrar a necessidade que hoje existe nos laboratórios de informática. Com a análise das Atas de reunião de curso foi possível comprovar a carência de um método simples para informar problemas a equipe de suporte. Já com a entrevista realizada com o supervisor dos laboratórios foi possível compreender como funciona o fluxo de uma solicitação.

Aliado às técnicas de levantamento de requisitos, foram utilizados os diagramas UML de caso de uso, diagrama de classes, diagrama de sequência e diagrama de entidade-relacionamento para concluir a especificação do software.

Com esses resultados, foi criado um protótipo *mobile* que apresenta e auxilia na visualização do fluxo de um aluno criar uma solicitação de serviço e o fluxo de um funcionário para atender a essa solicitação. O protótipo também apresenta o fluxo para criação dos usuários e o acompanhamento de uma solicitação.

Para projetos futuros, este trabalho aqui apresentado serve como documentação de requisitos do sistema para a criação de um aplicativo *mobile* que possa ser utilizado pelos alunos e funcionários da faculdade que promova a agilidade no suporte e uma melhoria na gestão dos laboratórios de informática.

## REFERÊNCIAS BIBLIOGRÁFICAS

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML – Guia do Usuário. Editora Campus, São Paulo 2000.

DICK, Jeremy; HULL, Elizabeth; JACKSON, Ken. Requirements Engineering: 4. ed. Nova Iorque: Springer, 2017.

LAPLANTE, Phillip. Requirements Engineering for Software and Systems: 2. ed. Boca Raton: CRC Press, 2014.

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

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

NAVATHE, Shamkant B.; Elmasri, Ramez. Sistemas de Banco de Dados. 4. ed. São Paulo: Pearson Addison Wesley, 2005.

RUP. Rational Unified Process. 2003. Disponível em <<<http://www.wthreex.com/rup>>>. Acessado em: 24/04/2006>