****

**Documentação Inicial de produto de Software**

**Versão 1.0**

**Projeto CoordBox**

Equipe

Joao Santanna

Matheus Ataide

Documento faz parte do TCC do aluno Matheus – Licenciatura em Computação UFRA

[Introdução 3](#_Toc30452886)

[Modelo da Documentação 3](#_Toc30452887)

[1. Introdução ao Documento 3](#_Toc30452888)

[1.1 Tema 3](#_Toc30452889)

[1.2 Objetivo do Projeto 4](#_Toc30452890)

[1.3 Delimitação do Problema 4](#_Toc30452891)

[1.4 Justificativa da Escolha do Tema 4](#_Toc30452892)

[1.5 Método de Desenvolvimento 4](#_Toc30452893)

[1.6 Organização do Trabalho 4](#_Toc30452894)

[2. Descrição geral do sistema 5](#_Toc30452895)

[2.1 Descrição do problema. 5](#_Toc30452896)

[2.2. Principais Envolvidos e suas características 5](#_Toc30452897)

[2.2.1. Usuários do Sistema 5](#_Toc30452898)

[2.2.2 Stakeholders do sistema 5](#_Toc30452899)

[2.3 Regras de Negócio 5](#_Toc30452900)

[3. Requisitos do sistema 7](#_Toc30452901)

[3.1 Requisitos Funcionais 7](#_Toc30452902)

[3.1 Casos de USO 10](#_Toc30452903)

[3.2 Requisitos não funcionais 11](#_Toc30452904)

[4. Análise e Design 12](#_Toc30452905)

[4.1 Arquitetura do sistema 12](#_Toc30452906)

[4.2 Arquitetura do sistema modelo de visão 4 + 1 13](#_Toc30452907)

[4.2.1 Visão de Desenvolvimento 13](#_Toc30452908)

[4.2.2 Visão de processo 13](#_Toc30452909)

[4.2.3 Visão Lógica 14](#_Toc30452910)

[4.2.4 Visão Física 14](#_Toc30452911)

[4.3 Ambiente de desenvolvimento 14](#_Toc30452912)

[4.4 Sistemas e componentes externos utilizados 15](#_Toc30452913)

[5 - Implementação de interface com o usuário (IHC) 15](#_Toc30452914)

[6 - Bibliografia Utilizada 16](#_Toc30452915)

## Introdução

O objetivo deste documento é fornecer um roteiro para o desenvolvimento de sistemas de software utilizando os princípios desenvolvimento ágil (kanbam) juntamente com notação UML (Unified Modeling Language).

Esta é a versão 1.0 do documento, criada para utilizar a notação UML e Metodologias ágeis. A escolha da orientação a objetos é devido à tendência de mercado, mas nada impede que o roteiro seja seguido no caso de opção pela Modelagem Estruturada (ver item Comentários sobre a Documentação).

Sugestões e Comentários podem ser enviados para o professor no e-mail: [joao.santanna@ufra.edu.br](mailto:joao.santanna@ufra.edu.br)

## Modelo da Documentação

A modelagem utilizada para fazer a documentação do sistema foi os cartões de requisitos e a modelagem representada por diagramas UML. Foram Utilizados diagramas de casos de uso no qual foi demonstrado o funcionamento do sistema.

Esse documento representa a Elicitação do projeto para compartilhamento da equipe e compreensão dos requisitos

### 1. Introdução ao Documento

O presente projeto pretende apresentar o aplicativo de suporte ao atendimento dos alunos pela coordenação de curso , tema do trabalho te conclusão de curso do aluno

#### 1.1 Tema

Aplicativo desktop para acompanhamento dos atendimentos de requisitos feitos a coordenação, o software se destina a ser uma memoria documentada dos atendimento, e que seja capaz de gerar diversos relatórios dos atendimentos , aferindo assim os atendimentos feitos pela equipe da coordenação.

### 1.2 Objetivo do Projeto

Aplicativo com um intuito de ajudar a coordenação a controlar os atendimentos, numero de atendimentos, clientes , tempo de resposta , etc.

### 1.3 Delimitação do Problema

O aplicativo irá ajudar a coordenação a controlar e aferir os atendimentos de demandas

### 1.4 Justificativa da Escolha do Tema

O tema foi escolhido por uma necessidade da coordenação de curso em ter uma ferramenta de medição e controle

### 1.5 Método de Desenvolvimento

O método a ser utilizado para a realização do projeto foi o método ágil, com o paradigma orientado a objetos.

### 1.6 Organização do Trabalho

Este projeto foi dividido em tópicos para uma melhor execução individual dos membros da equipe.

## 2. Descrição geral do sistema

O sistema irá interagir com o usuário de forma a registrar os dados de atendimento e seus encaminhamentos.

### 2.1 Descrição do problema.

Esse aplicativo resolverá grande parte dos problemas de memoria de atendimento e aferição do trabalho da coordenação , futuramente um aplicativo via web e celular também deveria ser feito para poder ser acessado de forma ubíqua.

### 2.2. Principais Envolvidos e suas características

#### 2.2.1. Usuários do Sistema

O sistema será desenvolvido para os membros da coordenação e secretaria, os clientes da coordenação que o software visa atender são os seguintes: alunos ( na maioria) , professores ( do curso ou não) , outros órgãos da UFRA ( principalmente a PROEN ) e por ventura clientes externos.

#### 2.2.2 Stakeholders do sistema

Coordenador, vice coordenador, secretaria do curso , diretoria do ICIBE

### 2.3 Regras de Negócio

O aplicativo funcionará para desktop , de preferencia uma solução multiplataforma( Windows, Linux e MacOS) , os dados entre os usuários devem poder ser centralizados para um relatório geral , ou particularizados por usuários, os canais de comunicação também devem ser monitorados ( email, whatsapp, pessoalmente, contato telefônico, etc), O aplicativo poderá fornecer um backup mensal para salvar dados importantes.

## 

## 3. Requisitos do sistema

### 3.1 Requisitos Funcionais

|  |
| --- |
| Nº: 1  Título: **Login do usuario**  Descrição: O sistema deve ter um login para identificar o usuário  Nota: O login pode ser persistente ( varias semanas) para não ter que pedir o login sempre que o usuário se identificar. |

|  |
| --- |
| Nº: 2  Título: **Adicionar novo atendimento**  Descrição: Novos atendimentos devem poder ser adicionados sempre que necessário.  Notas:  Dados dos atendimentos:  - nome do solicitante  - Se aluno a matricula  - contatos ( email , whats App, cel, etc)  - data de abertura de pedido  - classe do pedido: ... ver cartão 3  - anamnese : anotações sobre o pedido que ajudem a coordenação a escolher os melhores encaminhamentos  - encaminhamento : anotações das providencias tomadas  -prioridade: usar escala de likert para a prioridade ( 1 a 5) , um pouco , 5 urgente. |

|  |
| --- |
| Nº: 3  Título: **Contados cadastrados**  Descrição: Os contatos já cadastrados devem ser aproveitados  - Os contatos cadastrados devem ser reaproveitados |

|  |
| --- |
| Nº: 4  Título: **Classe de pedidos**  Descrição: Os pedidos devem ser classificados em classes  Nota:  - informação, documentação, ajuste de matricula , etc.  - Essa lista pode ser editada ...  - itens dessa lista podem ser apagados e adicionados |

|  |
| --- |
| Nº: 5  Título: **Status dos pedidos**  Descrição: Os pedidos devem possuir status ( novo, em andamento, parado, concluído )  Nota:   * todo novo pedido inicia com status novo * pedido em andamento é aquele que já foi tomado alguma medida * pedido parado é aquele que está parado por algum motivo que deve ser documentado * pedido concluído é aquele fechado |

|  |
| --- |
| Nº: 6  Título: **Listar pedidos abertos**  Descrição: Organizar os pedidos abertos em uma tabela  Nota:  - seria bom poder organizar essa tabela por nome, data de atendimento, prioridade, etc... |

|  |
| --- |
| Nº: 7  Título: **Editar pedidos**  Descrição: Os pedidos devem ser editáveis, os campos podem ser atualizados |

|  |
| --- |
| Nº: 8  Título: **Mecanismo de busca**  Descrição: O sistema deve ter um mecanismo de busca  Nota:  - Vários campos podem ser marcados, no mínimo 1 deve ser marcado para a busca e uma caixa de texto para informar o termo buscado  - a busca deve resultar numa lista onde uma vez selecionada deve abrir o pedido para leitura. |

|  |
| --- |
| Nº: 9  Título: **Fechar pedidos**  Descrição: Uma vez atendidos os pedidos devem ser fechados  Nota:  - pedidos fechados entram em uma lista de pedidos fechados  - a data de fechamento deve ser indicada  - o tempo de atendimento deve ser computado |

|  |
| --- |
| Nº: 10  Título: **Armazenar informações em banco de dados**  Descrição: Os dados devem ser armazenados em banco de dados SQL  Nota:  - Incialmente usar SQLite em arquivo local |

|  |
| --- |
| Nº: 11  Título: **Gerar relatórios**  Descrição: Os dados devem ser usados para geral relatórios  Nota:  - de preferência a saída dos dados deve ser em documento PDF e ou HTML  - os relatórios devem ter numero de atendimentos por semestre , numero de atendimentos por classe, tempo médio de resposta , etc ...  - esse cartão deve ser revisado antes de implementado , nessa altura já vai se ter uma ideia de como os relatórios devem ser.  - talvez exportar os dados em formato do pandas o que permite fazer report customizáveis usando data science |

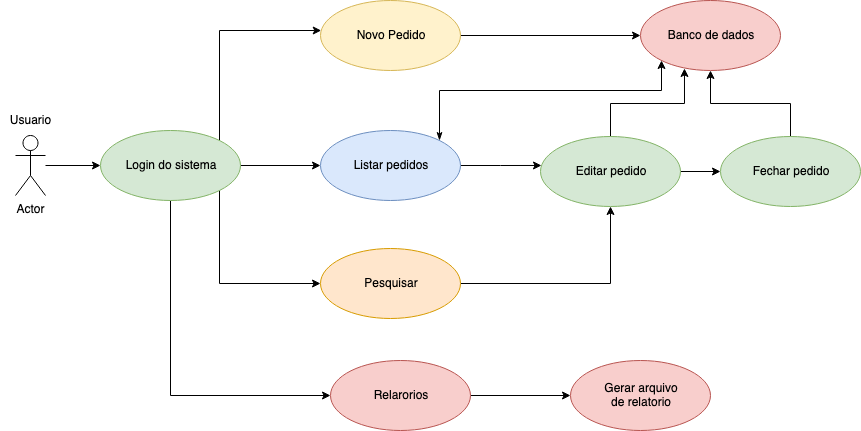
|  |
| --- |
| Nº: 12  Título: **Gerar executáveis nas plataformas alvo**  Descrição: Geras executáveis em todas as plataformas que o software pode rodar |

|  |
| --- |
| Nº: 13  Título: **Centralização de informações**  Descrição: Estudar formas de centralizar as informações  Notas:  - vamos supor que o coordenador , vice coordenador e secretaria tenha, cada um, seus arquivos com os atendimentos, estudar como adicionar esses dados a uma mesma tabela e atualizar nos computadores ( unificar as informações ) |

### 3.1 Casos de USO

**Caso 1: Gerenciar pedidos**

Um coordenador ( usuário ) se loga no sistema , cadastra novos pedidos , atualiza o status de alguns( fechar ).



**3- Pré-condições**

Utilizar tecnologia multiplataforma que permita a ferramenta ser utilizada nas seguintes plataformas: Windows, Linux, Mac Os

**4- Pós-Condições**

Não há.

**5- Observações**

Sem observações.

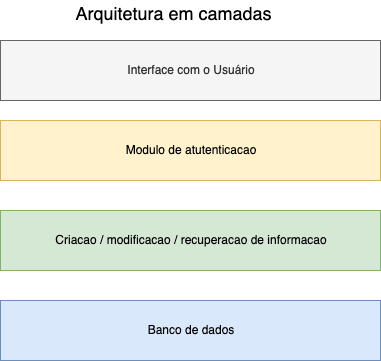
### 3.2 Requisitos não funcionais

* Confiabilidade - O aplicativo tem a capacidade de se manter em um estado estável, exatamente como se foi pensado no projeto.
* Tempo de execução - Dependeria da capacidade de processamento do dispositivo do usuário.
* Multiplataforma desktop - O aplicativo terá a capacidade de ser executado em vários sistemas operacionais em ambiente desktop
* Requisitos de usabilidade - O aplicativo não necessita de um amplo conhecimento na área, pois a sua interface é muito simples e de fácil entendimento.

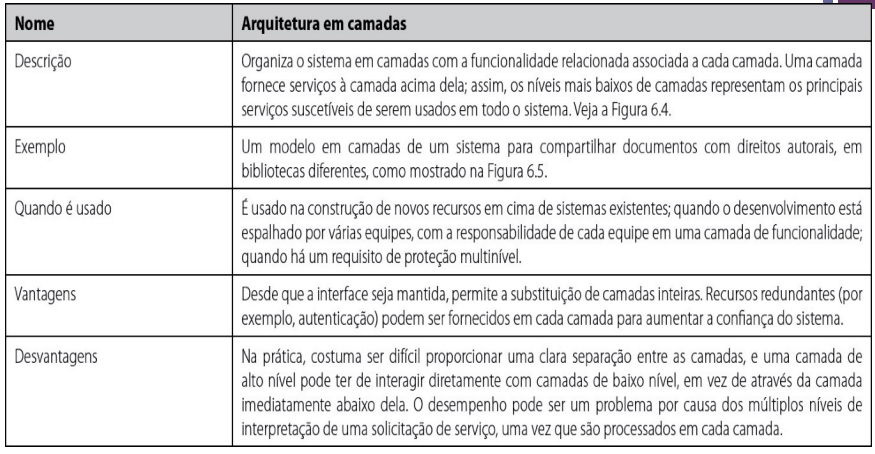
## 4. Análise e Design

### 4.1 Arquitetura do sistema

**Arquitetura de sistema de informação em camadas**

****

O usuário interage com a interface do aplicativo, e uma vez identificado pelo sistema de login, o usuário pode verificar ou modificar informações passadas, informações essas que estão armazenadas no banco de dados do aplicativo.

****

### 4.2 Arquitetura do sistema modelo de visão 4 + 1

Lógica

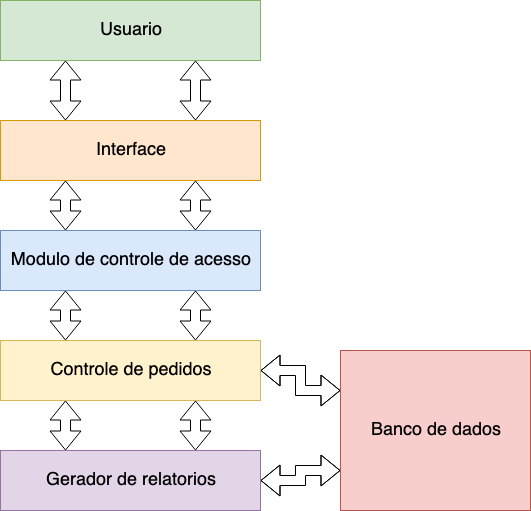
Uso

Desenvolvimento

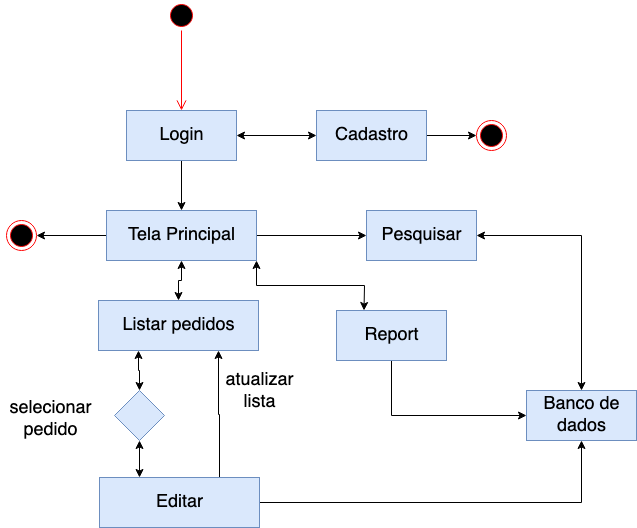
Processo

Física

#### 4.2.1 Visão de Desenvolvimento

****

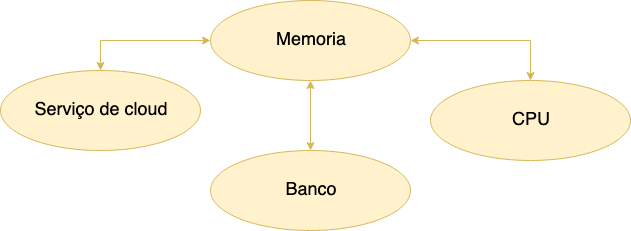
#### 4.2.2 Visão de processo



#### 4.2.3 Visão Lógica

Diagramas UML adicionados conforme as classes forem implementadas

#### 4.2.4 Visão Física



### 4.3 Ambiente de desenvolvimento

O aplicativo será desenvolvido usando a linguagem Python e SQLite, além de uma biblioteca de UI nativa do python, os executáveis serão gerados com o uso do Pyinstaller.

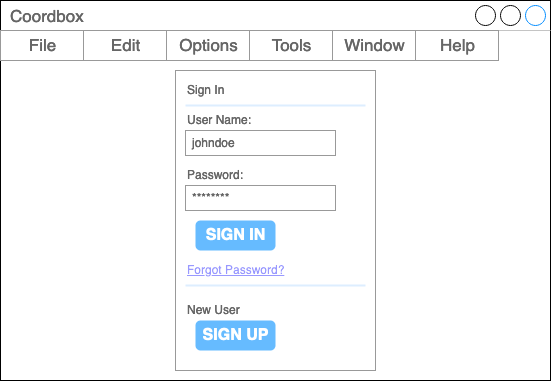
O código ficara hospedado do Github.

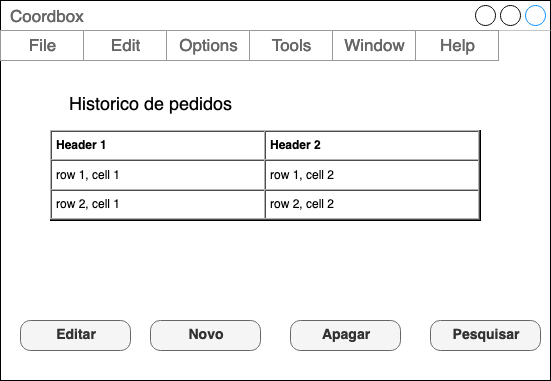
O projeto será gerenciado usando a ferramenta Trello

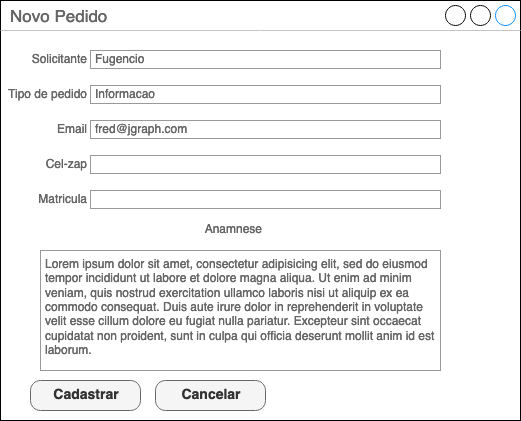
### 4.4 Sistemas e componentes externos utilizados

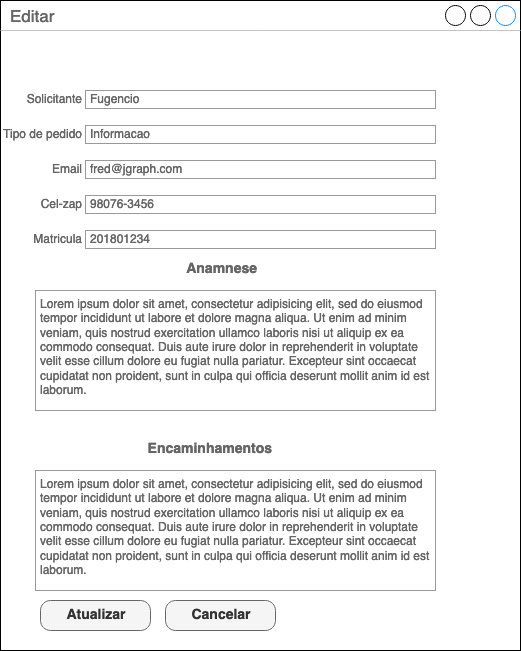
Será utilizado, como complemento do aplicativo, a API do Google Drive / Dropbox para o compartilhamento e centralização de informações além de backup.

## 5 - Esboço da interface com o usuário (IHC)









## 6 - Bibliografia Utilizada

SOMMERVILLE, Ian. Engenharia de Software 8. ed. Tradução Selma Shin Melnikoff; Reginaldo Arakaki; Edilson de Andrade Barbosa. São Paulo: Persson, 2007.

VERGILIO, S. R. Arquitetura de Software. Disponível em: <http://www.inf.ufpr.br/andrey/ci163/IntroduzArquiteturaAl.pdf>. Acesso em: 04 out. 2017.