INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO

RIO GRANDE DO SUL

CAMPUS CANOAS

CURSO TÉCNICO EM INFORMÁTICA INTEGRADO AO ENSINO MÉDIO

GABRIEL NUNES DE SIQUEIRA

SmartInv - Uma plataforma de Inventário de Informática com algoritmo gerenciador de computadores inativos

**Orientador**: Marcio Bigolin

Canoas, 26 de junho de 2017

# Proposta de Trabalho de Conclusão de Curso

## Descrição do Problema

É comum observar em várias instituições acadêmicas e empresas atuando na área da informática possuírem uma grande demanda de serviços simples que exigem uma excessiva quantidade de tempo para serem solucionados. A simples montagem de um computador inativo pode levar horas para ser efetuada, encontrar peças e equipamentos compatíveis acabam ocupando tempo útil que poderia ser aplicado em demais funções dentro da equipe ou setor.

Ao longo do ano de 2016 durante o projeto de extensão *Offboard* - Manutenção e Suporte a Serviço da Comunidade realizado no Instituto Federal Campus Canoas uma grande quantidade de componentes e equipamentos foram sendo adicionados ao estoque do projeto através de doações da comunidade, a solução para catalogar esses itens foi utilizar o GLPI (Gestion Libre de Parc Informatique), uma plataforma presente no mercado há alguns anos. A ferramenta é eficiente, porém acaba perdendo em quesito de organização e tempo quando é necessário a montagem rápida de computadores. A localização dos componentes dentro do sistema acaba sendo confusa, principalmente quando é necessário associar peças entre sí, já que o sistema não permite identificar qual peça está sendo ou já foi associada a outro componente de maneira ágil.

A utilização de sistemas de inventário é importante principalmente quando existe a necessidade de manter um controle sobre as peças em circulação, quando se trata de informática, as peças acabam sendo catalogadas com informações úteis sobre a qualidade da peça ou desempenho. Essas informações poderiam ser utilizadas para automatizar o processo de criação e montagem dos computadores dentro de um sistema de inventário para informática.

.

## Proposta de Solução

Desenvolver uma plataforma de inventário de informática para gerenciar computadores inativos, dentro de uma empresa ou instituição. Considera-se inativo todo aquele computador que ainda não está operando, apenas foi configurado. Esse sistema receberá as informações cadastradas pelos usuários e apresentará uma sugestão da melhor configuração disponível para as peças em estoque. O sistema funcionará com um algoritmo, buscando reduzir o tempo necessário em configurações de computadores inativos.

## Objetivo

### Objetivo Geral

O presente trabalho busca apresentar uma solução para inventário de informática com um algoritmo para otimizar o tempo na criação de computadores inativos.

### Objetivos Específicos

Pesquisar a utilização de inventários de informática e a importância dentro de grandes instituições;

Reunir trabalhos semelhantes e estudos sobre algoritmos de organização;

Realizar testes quanto a eficiência do algoritmo desenvolvido;

Analisar ferramentas que auxiliem o desenvolvimento da pesquisa;

Implementação da plataforma.

# Modelo de Requisitos

## Diagrama de Casos de Uso

Os seguintes caso de uso apresentam as possíveis interações do usuário dentro do sistema, existe apenas um tipo de usuário que é responsável por gerenciar todo o catálogo de peças, incluindo criar, deletar e excluir componentes. Gerenciar o funcionamento do algorítmo através de configurações prévias escolhidas pelo usuário (gerar configuração) ou montar computadores manualmente no sistema com todas as associações realizadas pelos usuários (montar computador manualmente), gerar relatórios com informações sobre quantidade de peças no sistema e computadores catalogados.

Figura 1: Casos de uso representando o sistema.



## Especificação de Casos de Uso

|  |  |
| --- | --- |
| Código e Nome do Caso de Uso | CdU01 - Gerenciar peças |
| Ator Primário: | Usuário |
| Fluxo Principal de Eventos | P1. O usuário acessa o menu de componentes.  P2. O sistema apresenta os tipos de componentes disponíveis.  P3. O usuário escolhe o tipo de componente que deseja visualizar.  P4. O sistema apresenta uma tabela com o tipo de componente solicitado.  P5. O usuário acessa a tabela com os dados existentes referente ao componente solicitado (IV01).  P6. O usuário escolhe inserir (A1), remover (A2) ou alterar (A3) componentes existentes. |
| Fluxos Alternativos | **A1. O usuário escolhe a opção inserir novo componente.**  A1.1. O sistema apresenta a tela de inserção de componentes (IV02).  A1.2. O usuário preenche os campos apresentados.  A1.3. O sistema realiza a inserção dos dados no banco.  A1.4. O usuário retorna ao menu anterior.  **A2. O usuário escolhe a opção remover um componente existente.**  A2.1. O Sistema apresenta uma mensagem de confirmação sobre remoção de componentes.  A2.2. O usuário confirma a exclusão.  A2.3. O sistema exclui os dados selecionados do banco.  A2.4. O usuário retorna ao menu anterior.  **A3. O usuário escolhe a opção de alterar um componente existente.**  A3.1. O sistema apresenta a tela de edição do componente solicitado (IV02).  A3.2. O usuário altera os dados desejados nos campos apresentados.  A3.3. O sistema realiza as alterações dentro do banco de dados.  A3.4. O usuário retorna ao menu anterior. |

|  |  |
| --- | --- |
| Código e Nome do Caso de Uso | CdU02 – Gerar relatório |
| Ator Primário: | Usuário |
| Fluxo Principal de Eventos | P1. O usuário acessa a opção de gerar relatório.  P2. O sistema apresenta as opções de relatórios disponíveis.  P3. O usuário escolhe se deseja visualizar o relatório de peças (A1), o relatório de computadores (A2) ou um gráfico com os computadores sugeridos pelo sistema (A3). |
| Fluxos Alternativos | **A1. O usuário escolhe um relatório de peças.**  A1.1. O sistema verifica dentro do banco de dados todas as peças catalogadas.  A1.2. O sistema apresenta a contagem das peças com suas respectivas identificações.  A1.3. O sistema disponibiliza a opção de imprimir.  A1.4. O usuário retorna para a tela anterior (P2).  **A2. O usuário escolhe a opção relatório de computadores.**  A2.1. O sistema verifica dentro do banco de dados todas as associações entre peças e computadores.  A2.2. O sistema apresenta o relatório com todos os computadores catalogados dentro do sistema.  A2.3. O sistema disponibiliza a opção de imprimir.  A2.4. O usuário retorna para a tela anterior (P2).  **A3. O usuário escolhe a opção gerar gráfico comparativo com os computadores sugeridos pelo sistema.**  A3.1. O sistema verifica dentro do banco de dados as tabelas com os computadores sugeridos e compara com os existentes se estão funcionando.  A3.2. O sistema gera um gráfico exibindo a porcentagem de funcionamento entre todos.  A3.3. O sistema disponibiliza a opção de imprimir.  A3.4. O usuário retorna para a tela anterior (P2). |

|  |  |
| --- | --- |
| Código e Nome do Caso de Uso | CdU03 – Montar computador manualmente |
| Ator Primário: | Usuário |
| Fluxo Principal de Eventos | P1. O usuário acessa o menu de computadores.  P2. O sistema apresenta os computadores existentes.  P3. O usuário tem a opção de gerenciar um computador existente (CdU01).  P4. O usuário escolhe o computador desejado.  P5. O sistema apresenta as telas de edição do computador e as peças vinculadas a ele.  P6. O usuário retorna ao menu anterior. |

|  |  |
| --- | --- |
| Código e Nome do Caso de Uso | CdU04 – Gerar configuração |
| Ator Primário: | Usuário |
| Fluxo Principal de Eventos | P1. O usuário acessa a página de configuração do algoritmo.  P2. O sistema procura no banco de dados a melhor configuração para criar vários computadores. (A1)  P3. O sistema apresenta uma tabela com todas as sugestões encontradas. (IV03)  P4. O usuário escolhe as sugestões que deseja remover e confirma a criação dos novos computadores. |
| Fluxos Alternativos | **A1. Não foram encontrados configurações de computadores compatíveis.**  A1.1. O sistema apresenta uma mensagem ao usuário dizendo a incompatibilidade.  A1.2. O usuário retorna a tela inicial do sistema. |

## Protótipos de Tela

Figura 2: IV01. Representa a exibição de todos os componentes em geral.

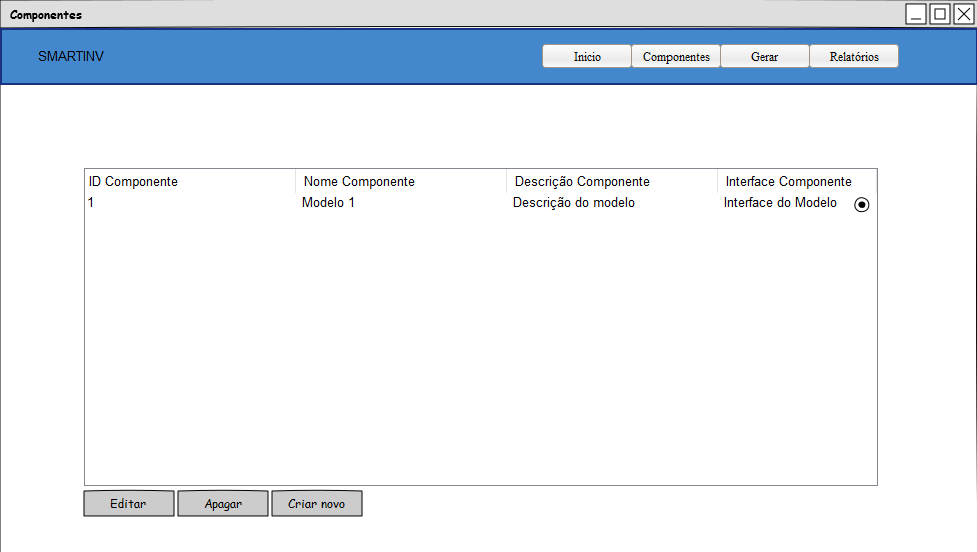


Figura 3: IV02. Representa a criação ou edição de um componente.

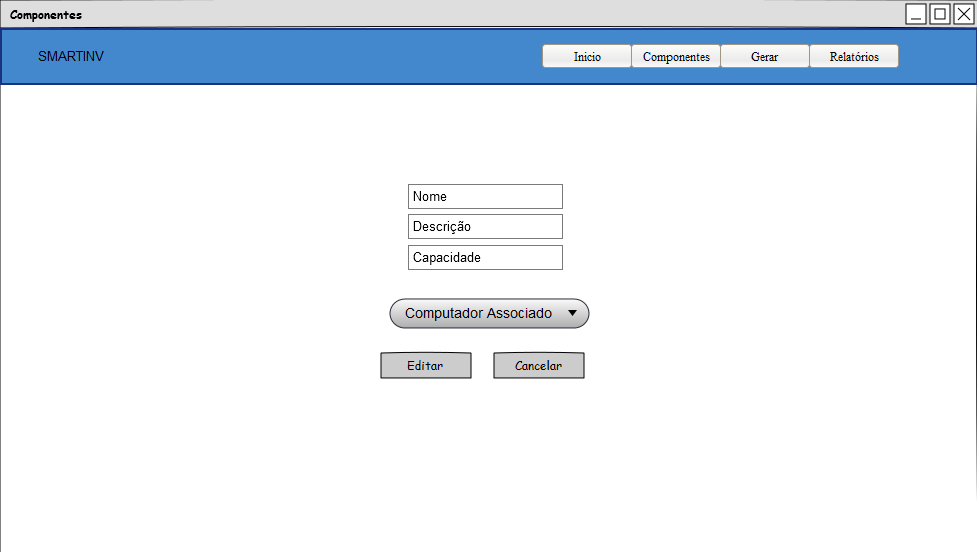
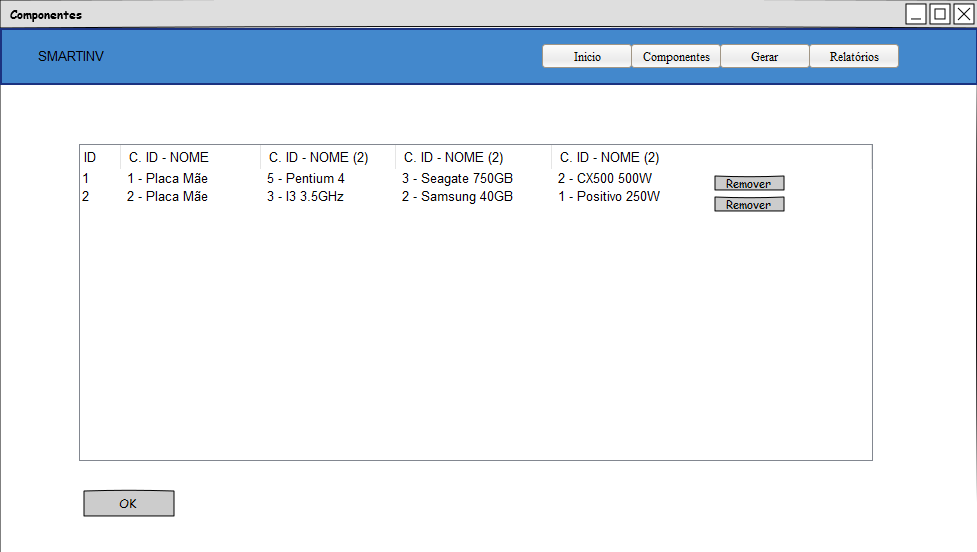


Figura 4: IV03. Representa as sugestões apresentada pelo algorítmo.

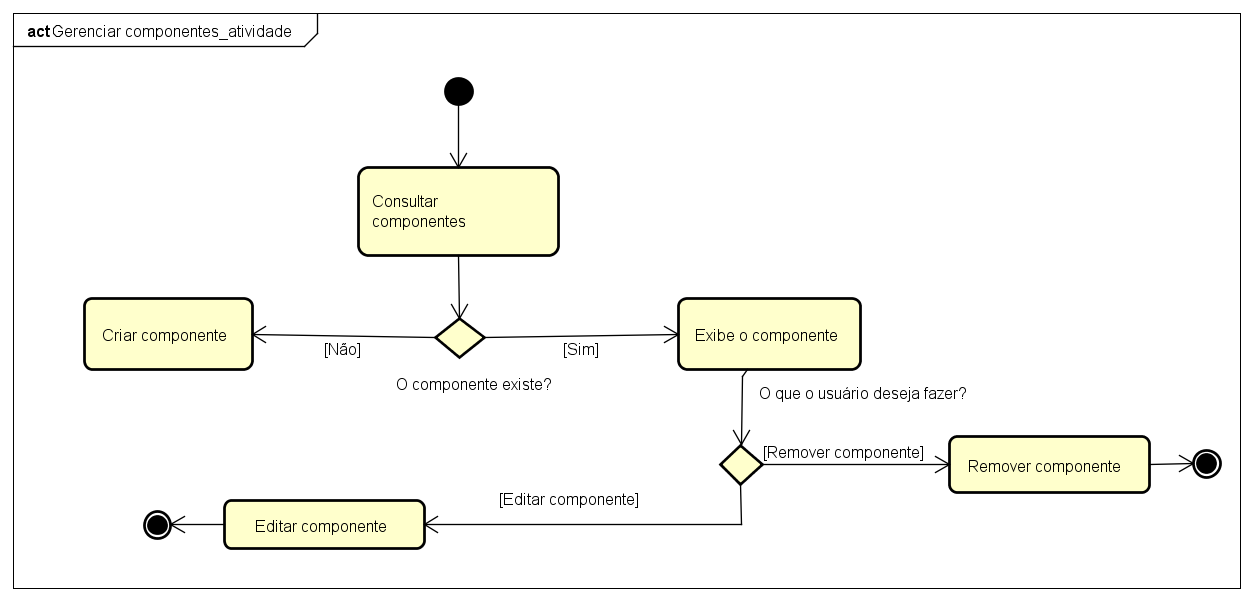


# Modelagem do Sistema

## Diagramas de Atividade

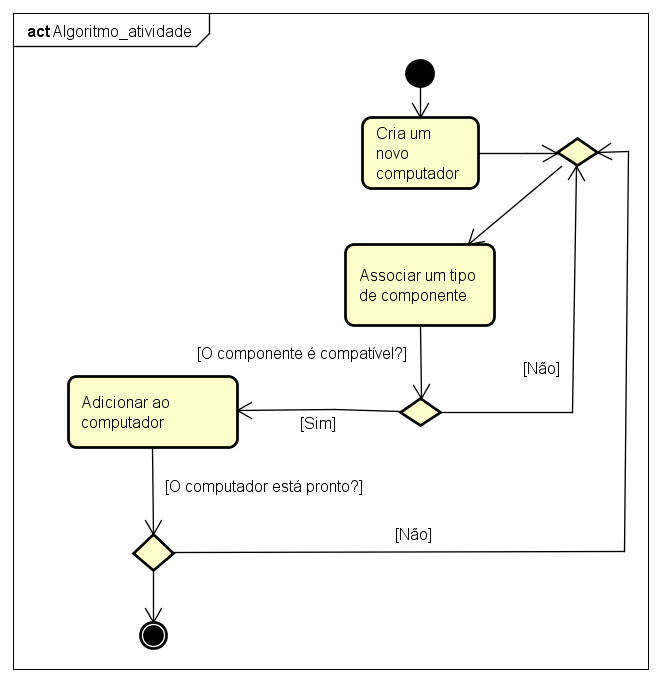
O diagrama abaixo representa a interação do usuário ao criar, editar ou remover um componente do sistema, primeiramente ele será levado a uma tela para consultar os componentes já existentes por meio de uma tabela e então poderá decidir o que deseja fazer.

Figura 5: Representa a interação do usuário ao gerenciar componentes.



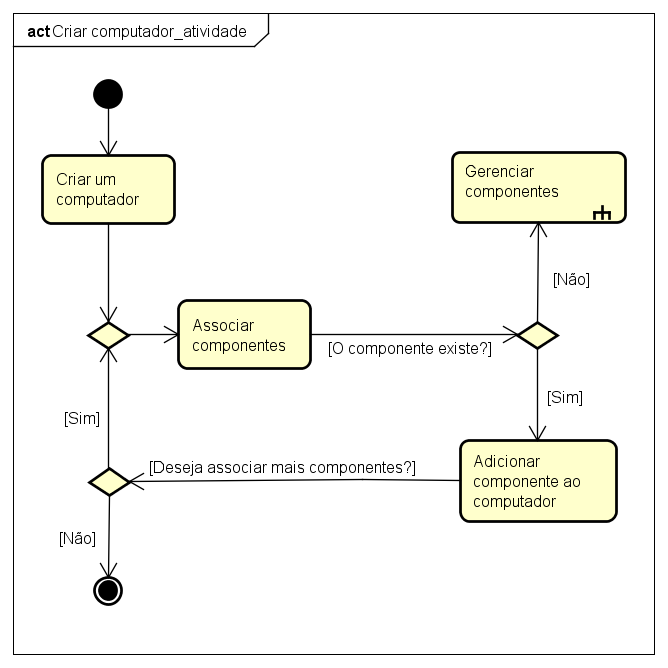
O diagrama abaixo refere-se ao algoritmo que irá operar no sistema, será acionado por um usuário com configurações prévias. Após a inicialização, o algoritmo criará um computador e começará a associar peças ao mesmo através de testes de compatibilidade, se o computador estiver com todas as peças necessárias associadas a ele, o algoritmo encerrará a atividade.

Figura 6: O diagrama representa o funcionamento do algoritmo dentro do sistema.



O diagrama abaixo representa a interação do usuário ao criar um computador no sistema manualmente, primeiramente ele criará o objeto computador e depois começará a adicionar peças na medida que achar necessário. Ao terminar de adicionar peças, o computador será criado no sistema e poderá ser visualizado.

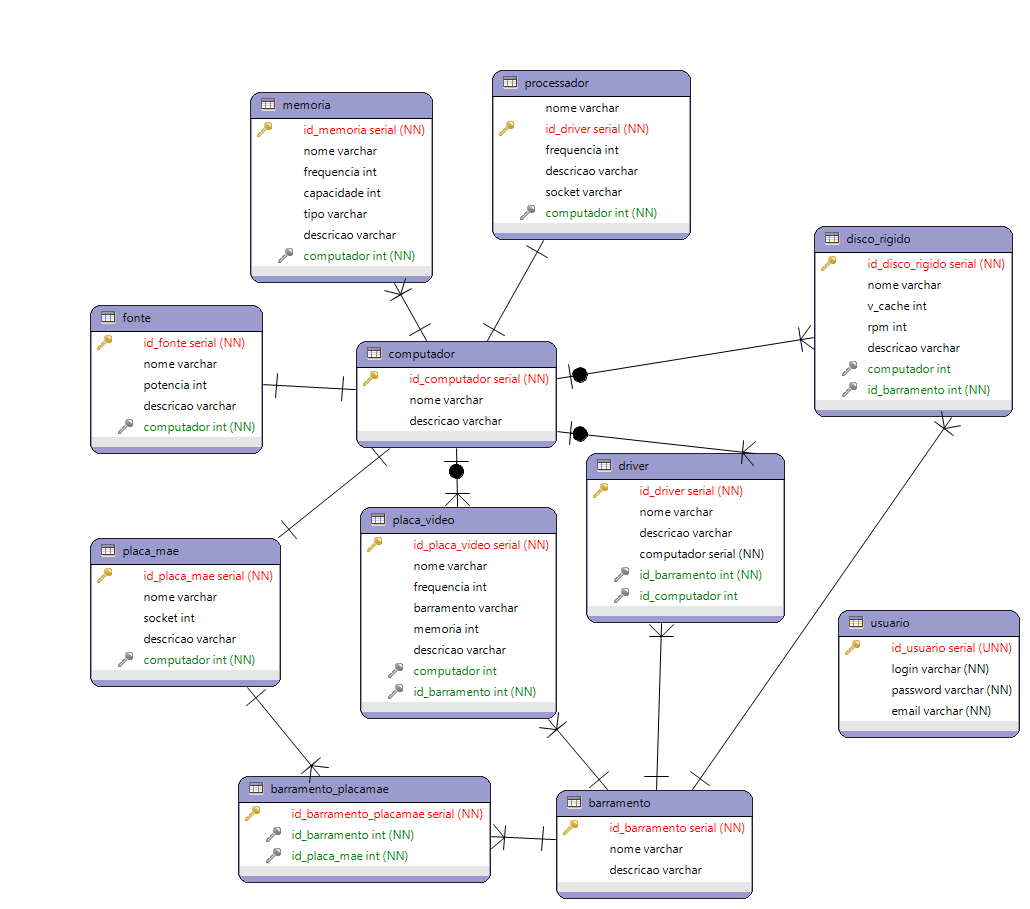
Figura 7: Diagrama representando a criação de novos computadores no sistema.



## Modelagem do Banco de Dados

O diagrama abaixo representa a interação entre as tabelas no banco de dados, é importante destacar os componentes obrigatórios no computador, que são a fonte, placa mãe, memória e processador. Os demais componentes são opcionais na associação aos computadores, o banco de dados ainda permite múltipla associações entre componentes, um computador pode ter várias peças do mesmo tipo.

Figura 8: Diagrama ERMaster representando o banco de dados do sistema.



# Proposta de Solução Tecnológica Escolhida

## Tecnologias Adotadas

* PHP Hypertext Preprocessor (PHP): Linguagem de programação utilizada principalmente no desenvolvimento web e para a conexão com o banco de dados;
* HyperText Markup Language (HTML): Linguagem de marcação para criar e construir as bases das páginas web, organizando o conteúdo apresentado na plataforma;
* Cascading Style Sheets (CSS): Linguagem de estilização das páginas, usada para colorir, formatar e organizar um documento HTML;
* Structured Query Language (SQL): Linguagem estruturada utilizada em banco de dados para consultas e manipulação de banco de dados, utilizado para armazenar todos os dados do sistema;
* PostgreSQL: Banco de dados utilizado para armazenar todos os dados coletados no sistema.

## Ferramentas Adotadas

* Enyalius: Framework PHP para criação de formulários e páginas, usado também para conexão com o banco de dados e padronização das páginas (gitlab.com/enyalius);
* Composer: Ferramenta para gerenciar as dependências e bibliotecas em PHP, permite declarar as bibliotecas que serão utilizadas enquanto gerencia a instalação e atualização automaticamente (getcomposer.org);
* Smarty: Ferramenta utilizada para gerenciar templates vinculados ao PHP, separa a aplicação lógica e o conteúdo do HTML utilizado nas páginas (smarty.net);
* Bootstrap: Framework CSS/JS para customizar as páginas, aplicação de estilos e responsividade (getbootstrap.com);
* EclipsePHP: Ambiente de desenvolvimento em PHP (eclipse.org);
* ERMaster: Plugin para o Eclipse responsável por criar os diagramas do banco de dados (ermaster.sourceforge.net);
* pgAdmin III: Ambiente de desenvolvimento para SQL (pgadmin.org);
* GitLab: Repositório para gerenciar o versionamento da aplicação e armazenar todos os arquivos de maneira segura (gitlab.com);
* XAMPP: Pacote com diversos softwares, utilizado entre eles o Apache Web Server para reproduzir um servidor localmente (apachefriends.org);
* Astah: Ferramenta para modelagem e criação dos diagramas utilizados no desenvolvimento do software (astah.net).

# Cronograma

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Atividade** | **Jan** | **Fev** | **Mar** | **Abr** | **Mai** | **Jun** | **Jul** | **Ago** | **Set** | **Out** | **Nov** | **Dez** |
| Definição do problema |  |  |  |  |  |  |  |  |  |  |  |  |
| **1° Etapa – Pesquisa** | | | | | | | | | | | | |
| Estudar soluções para o problema proposto |  |  |  |  |  |  |  |  |  |  |  |  |
| Pesquisar trabalhos semelhantes |  |  |  |  |  |  |  |  |  |  |  |  |
| **2° Etapa – Modelagem** | | | | | | | | | | | | |
| Modelagem da plataforma |  |  |  |  |  |  |  |  |  |  |  |  |
| Criação de um banco de dados PostgreSQL |  |  |  |  |  |  |  |  |  |  |  |  |
| **3° Etapa – Desenvolvimento** | | | | | | | | | | | | |
| Migrar os dados da plataforma GLPI |  |  |  |  |  |  |  |  |  |  |  |  |
| Desenvolvimento do sistema |  |  |  |  |  |  |  |  |  |  |  |  |
| Elaboração de um gráfico de precisão |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementação de uma interface visual |  |  |  |  |  |  |  |  |  |  |  |  |
| **4° Etapa – Revisão** | | | | | | | | | | | | |
| Testes e correções necessárias |  |  |  |  |  |  |  |  |  |  |  |  |
| Revisão da monografia |  |  |  |  |  |  |  |  |  |  |  |  |
| **5° Etapa – Apresentação** | | | | | | | | | | | | |
| Apresentação na IFCITEC |  |  |  |  |  |  |  |  |  |  |  |  |
| Seminário de Andamento |  |  |  |  |  |  |  |  |  |  |  |  |
| Apresentação Seminário final |  |  |  |  |  |  |  |  |  |  |  |  |