

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 Controle de  
Estoques de Componentes com Algoritmo  
Gerenciador de Computadores**

Canoas, 17 de novembro de 2017.

GABRIEL NUNES DE SIQUEIRA

**SmartInv - Uma plataforma de Controle de  
Estoques de Componentes com Algoritmo  
Gerenciador de Computadores**

Trabalho de Conclusão de Curso  
apresentado como requisito parcial para  
obtenção do grau de Técnico em  
Informática pelo Instituto Federal de  
Educação, Ciência e Tecnologia do Rio  
Grande do Sul – Campus Canoas.

Prof. Me. Marcio Bigolin  
Orientador

Canoas, 17 de novembro de 2017

## RESUMO

O presente trabalho de conclusão visa a criação de uma plataforma gerenciadora de estoques para informática. O desenvolvimento teve embasamento através de pesquisas dentro da área de Gestão de Estoques e Administração de Recursos, utilizadas para conceituar as funcionalidades da plataforma. Ao utilizar um sistema de controle de estoque destinado a Informática, o usuário necessita ter uma visão geral dos dados cadastrados; quando necessário vincular demais componentes a um computador, é imprescindível o conhecimento em manipulação de *hardwares*. O SmartInv possibilita total manipulação dos dados cadastrados, bem como associar automaticamente os componentes disponíveis através de um algoritmo, realizando testes quanto a sua compatibilidade, possibilitando criar múltiplos computadores sem a necessidade do conhecimento aprofundado. O usuário consegue gerar relatórios demonstrando a quantidade de componentes presentes em estoque e a relação com todos os computadores existentes.

**Palavras-Chave:** Estoque. Algoritmo. Informática. Inventário.

## LISTA DE FIGURAS

Figura 1: Exemplo de identificação de uma peça vinculada ao GLPI .....	7
Figura 2: Visualização de um componente no GLPI.....	8
Figura 3: Participantes das oficinas portas abertas do Offboard .....	13
Figura 4: Diagrama de entidade-relacionamento representando o banco de dados. ....	14
Figura 5: Diagrama de Casos de Uso .....	15
Figura 6: Tela exibindo a lista com um tipo de componente (Ampliada). ....	16
Figura 7: Tela para Cadastro de componentes.....	16
Figura 8: Tela para Cadastro de Interfaces/Barramento .....	17
Figura 9: Exemplificação de um algoritmo <i>Hill Climbing</i> .....	18
Figura 10: Exemplificação de <i>shoulders</i> na execução de um algoritmo <i>Hill Climbing</i> . 18	
Figura 11: Diagrama de Atividades representando o algoritmo.....	19
Figura 12: Resultado dos testes no algoritmo.....	21
Figura 13: Tela para migração de dados entre bancos de dados .....	22
Figura 14: Tela representando o relatório de peças cadastradas .....	23
Figura 15: Tela representando o relatório com todos os computadores existentes (Ampliado). ....	23

## LISTA DE ABREVIATURAS E SIGLAS

CSS	<i>Cascading Style Sheets</i> (Folha de Estilos em Cascata)
GLPI	<i>Gestion Libre de Parc Informatique</i>
HTML	<i>HyperText Markup Language</i>
ID	Identificação
JS	<i>JavaScript</i>
MVC	<i>Model-View-Controller</i>
PHP	<i>PHP Hypertext Preprocessor</i>
PROEJA	Programa Nacional de Integração da Educação Profissional com a Educação Básica na Modalidade de Educação de Jovens e Adultos
SGBD	Sistema de Gerenciamento de Banco de Dados
TADS	Tecnólogo em Análise e Desenvolvimento de Sistemas
UML	<i>Unified Modeling Language</i> (Linguagem de Modelagem Unificado)

## SUMÁRIO

<b>RESUMO.....</b>	<b>3</b>
<b>LISTA DE FIGURAS.....</b>	<b>4</b>
<b>LISTA DE ABREVIATURAS E SIGLAS .....</b>	<b>5</b>
<b>1 INTRODUÇÃO .....</b>	<b>7</b>
<b>2 OBJETIVOS .....</b>	<b>9</b>
2.1 GERAL.....	9
2.2 ESPECÍFICOS .....	9
<b>3 METODOLOGIA.....</b>	<b>10</b>
<b>4 TRABALHOS RELACIONADOS .....</b>	<b>12</b>
4.1 CRAFTMYBOX .....	12
4.2 CHIPART .....	12
4.3 THE GLPI PROJECT.....	12
<b>5 O PROJETO <i>OFFBOARD</i> .....</b>	<b>13</b>
<b>6 DESENVOLVIMENTO.....</b>	<b>14</b>
6.1 ALGORITMO .....	17
6.2 MIGRAÇÃO DE DADOS .....	21
6.3 RELATÓRIOS .....	22
<b>7 CONCLUSÃO.....</b>	<b>24</b>
7.1 TRABALHOS FUTUROS .....	25
<b>REFERÊNCIAS .....</b>	<b>26</b>
<b>APÊNDICE A – ESPECIFICAÇÃO DE CASOS DE USO.....</b>	<b>28</b>

## 1 INTRODUÇÃO

É comum observar em várias instituições acadêmicas e empresas atuando na área da informática a existência de uma grande demanda de serviços que exigem uma excessiva quantidade de tempo para serem solucionados. A simples montagem de um computador pode levar horas para ser efetuada, seja por problemas de arranjo físico do depósito descrito por Chiavenato (2005) como “disposição física dos equipamentos, pessoas e materiais, da maneira mais adequada ao processo produtivo” ou problemas relacionados a compatibilidade entre componentes. A montagem de um computador requer componentes básicos, descrito por Morimoto (2009) como a combinação entre os seguintes componentes:

- Fonte de alimentação;
- Placa Mãe;
- Processador;
- Disco Rígido;
- Memória.

Ao longo do ano de 2016 durante a execução do projeto de extensão Offboard - Manutenção e Suporte a Serviço da Comunidade, realizado no Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul Campus Canoas, uma grande quantidade de componentes e equipamentos foram sendo adicionados ao estoque do projeto através de doações da comunidade. O acúmulo de estoques em níveis adequados é uma necessidade para o normal funcionamento do sistema produtivo (CHIAVENATO, 2005), considerando o objetivo do projeto Offboard de fomentar a descoberta e aprendizagem na área de manutenção, uma grande quantidade de peças acaba sendo reutilizada para aprendizagem ou montagem de computadores para serem doados. Deste modo, surge a necessidade da realização de inventários, descritos por Chiavenato (2005) como a verificação da existência dos materiais dentro do estoque.

Figura 1: Exemplo de identificação de uma peça vinculada ao GLPI



Fonte: Autoria própria.

Os itens inventariados necessitavam ser catalogados (Figura 1) dentro de um sistema online, a solução foi utilizar o GLPI (Gestion Libre de Parc Informatique), uma plataforma de código aberto presente no mercado há aproximadamente quinze anos (GLPI, 2017). A ferramenta é eficiente, porém deixa a desejar tratando-se de organização e tempo quando é necessária a montagem rápida de computadores, demonstrando-se assim incapaz de solucionar a alta demanda pelo estoque.

A utilização de sistemas de catálogo é importante principalmente quando existe a necessidade de manter um controle sobre as peças em circulação, possibilitando identificar discrepâncias entre o estoque catalogado e o estoque físico. (CHIAVENATO, 2005). 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.

Conforme a Figura 2, é possível observar que as informações características do componente ainda não estão completas, ao se tratar de um processador de computador é possível ainda informar a *socket* ou também conhecido como espaço disponível para que seja conectado o processador facilitando trocas (MORIMOTO, 2009), número de interfaces na placa e seus respectivos nomes, tipo de memória na placa e demais informações relevantes. Com essas informações catalogadas seria possível realizar a associação automática entre as peças dentro do estoque se necessário, o usuário poderia acessar uma lista com todos os processadores disponíveis para aquela placa mãe específica, todos os discos rígidos que a placa mãe tenha capacidade de suportar entre outras possíveis combinações, agilizando o processo de criação de um computador.

Figura 2: Visualização de um componente no GLPI

The screenshot displays the GLPI interface for a component. At the top, there is a header bar with a 'Lista' link on the left and the title 'Placa mãe - 1 - M2N-E SLI - ID 1' on the right. Below this, a form titled 'Placa mãe - ID 1' contains several input fields: 'Nome' with the value '1 - M2N-E SLI', 'Fabricante' with a dropdown menu showing 'Asus' and a plus icon, and 'Chipset' with the value 'null'. To the right of these fields is a 'Comentários' section with a large text area. At the bottom right of the form is an orange 'Salvar' button.

Fonte: Autoria própria.

A proposta visa desenvolver uma plataforma de inventário de informática para gerenciar computadores em estoque, dentro de uma empresa ou instituição. Esse sistema receberá as informações cadastradas pelos usuários e apresentará uma sugestão da melhor configuração disponível para as peças em estoque. O SmartInv funcionará com um algoritmo, buscando reduzir o tempo necessário em configurações de computadores.



## **2 OBJETIVOS**

### **2.1 GERAL**

Apresentar uma solução para inventário de informática com um algoritmo para otimizar o tempo na criação de computadores dentro de um estoque.

### **2.2 ESPECÍFICOS**

- Pesquisar a utilização de inventários de informática e a importância dentro de grandes instituições;
- Reunir trabalhos semelhantes e estudos sobre algoritmos;
- Analisar ferramentas que auxiliem a solução da proposta;
- Implementar a plataforma;
- Realizar testes quanto a eficiência do algoritmo desenvolvido;
- Migrar os dados do GLPI.

### 3 METODOLOGIA

O presente trabalho se dará por pesquisa científica, descrita por Gerhardt e Silveira (2009) como “[...] o resultado de um inquérito ou exame minucioso, realizado com o objetivo de resolver um problema, recorrendo a procedimentos científicos” e servirá para por em prática o conhecimento adquirido.

O desenvolvimento da pesquisa ocorreu através de consultas a livros e artigos acadêmicos na área de gestão de estoques e informática, utilizando o livro Administração de materiais: uma abordagem introdutória do autor Idalberto Chiavenato (2005) objetivando embasar o conteúdo relacionado a gestão de estoques foi possível conceituar e obter conhecimentos sobre a área.

Visando abordar questões na área da informática, foram consultados livros como *Artificial Intelligence: A Modern Approach* do autor americano Russel Norvig buscando conhecimentos sobre algoritmos e uma introdução a *softwares* inteligentes, junto com o Guia do Hardware de Carlos Morimoto (2007) abordando questões acerca de componentes de computadores e seus detalhes. Como base para a pesquisa, foram utilizados livros como Engenharia de Software (SOMMERVILLE, 2011) e Sistema de Banco de Dados (ELSMARI e NAVATHE, 2005) para contextualizar e implementar o desenvolvimento padronizado do projeto. Todos os livros e artigos foram obtidos através de consultas na internet ou na biblioteca física do *campus*.

Os requisitos foram analisados através da participação do autor como bolsista no projeto *Offboard* realizado em 2016, onde era possível identificar problemas os seguintes problemas:

- Conhecer o que existe em estoque precisamente;
- Maximizar o atendimento da demanda pela disponibilidade do material em estoque.

Após a identificação dos problemas, promoveu-se o levantamento dos requisitos da plataforma, modelando de acordo com os padrões da linguagem UML (*Unified Modeling Language*), descrita por Sommerville (2011) como a linguagem gráfica utilizada no desenvolvimento orientado a objetos.

A modelagem inicial se deu pela diagramação entidade-relacionamento, descrita por Elmasri e Navathe (2004) como um modelo de dados conceitual de alto nível, utilizado em aplicações de bancos de dados. Com os diagramas entidade-relacionamento finalizados, a criação dos casos de uso responsáveis por identificar possíveis atores ou usuários do sistema e suas interações com as funcionalidades (SOMMERVILLE, 2011) teve início.

Os diagramas de casos de uso foram convertidos para diagramas de atividades, descritos por Larman (2004) como a exibição de ações ou atividades em sequência,

úteis para modelar algoritmos complexos, fluxos de trabalho e de dados. Através do diagrama de atividades foi possível obter detalhes sobre o desenvolvimento da aplicação.

A construção da plataforma foi realizada com o auxílio de *frameworks*, descritos por Booch (1998, p. 162, tradução nossa) como um conjunto de classes que provém um determinado serviço ou finalidade, podendo auxiliar o usuário no desenvolvimento. O principal *framework* utilizado foi o Enyalius<sup>1</sup>, desenvolvido para a criação de formulários e páginas.

O Enyalius é usado também para conexão com o banco de dados e padronização da estrutura das páginas através da combinação de outras ferramentas como o *bootstrap*, conhecido por ser um conjunto de ferramentas para desenvolvimento da interface visual e todos os componentes visualizados pelo usuário (BOOTSTRAP, 2017).

A plataforma necessitou da utilização de um Sistema de Gerenciamento de Banco de Dados (SGBD), fornecido pela instituição e utilizado neste trabalho de conclusão, o PostgreSQL foi escolhido por ser totalmente gratuito, por se adaptar a diversos sistemas sem apresentar problemas (THE POSTGRESQL DEVELOPMENT GROUP, 2017), demonstra-se confiável para manter os dados do SmartInv.

---

<sup>1</sup> Pode ser encontrado no seguinte link: <[gitlab.com/enyalius](https://gitlab.com/enyalius)>

## 4 TRABALHOS RELACIONADOS

O desenvolvimento do SmartInv se dará a partir da combinação de características a serem observadas em três plataformas distintas, são elas: CraftMyBox, Loja Chipart e GLPI. Atendendo os objetivos específicos do trabalho, a análise de sistemas que realizem combinações com restrições é essencial para a implementação da plataforma.

### 4.1 CRAFTMYBOX

*CraftMyBox* é um banco de dados com mais de mil peças catalogadas destinada para usuários sem conhecimentos aprofundados em informática montarem seus computadores a partir de combinações de peças. Através de um algoritmo responsável por realizar mais de quarenta validações, o usuário é informado das possíveis incompatibilidades ao associar determinadas peças entre si (CRAFTMYBOX, 2017).

O *CraftMyBox* apresenta-se como uma plataforma dedicada para usuários buscarem preços acessíveis na montagem de um computador, a medida que o usuário escolhe os componentes o algoritmo calcula o preço total e informa a loja com o menor preço.

### 4.2 CHIPART

A Chipart é uma loja virtual especializada na montagem de computadores específicos para jogadores desde 2009 (CHIPART, 2017), a loja é especializada em conteúdo gamer, os produtos variam desde monitores customizados até gabinetes temáticos (caixa que envolve o computador fisicamente). A loja possui um sistema que permite um usuário inexperiente montar o próprio computador combinando peças individualmente mostrando os preços, permitindo o utilizador controlar seus gastos.

### 4.3 THE GLPI PROJECT

A plataforma GLPI está presente no mercado de inventários de informática desde 2002 e conta com ferramentas para administração e controle do estoque físico ou lógico. (GLPI DEVELOPMENT TEAM, 2017)

As principais funcionalidades da plataforma são:

- O controle de inventário preciso de todos os itens em estoque;
- O histórico das ações realizadas dentro do sistema.

O GLPI consiste em uma plataforma com controle de estoque e gerenciamento de chamados técnicos para solucionar problemas dentro de uma empresa ou setor. Através do GLPI o técnico consegue ter um controle sobre todos os computadores ativos dentro do sistema e efetuar possíveis reparos quando necessário.

## 5 O PROJETO OFFBOARD

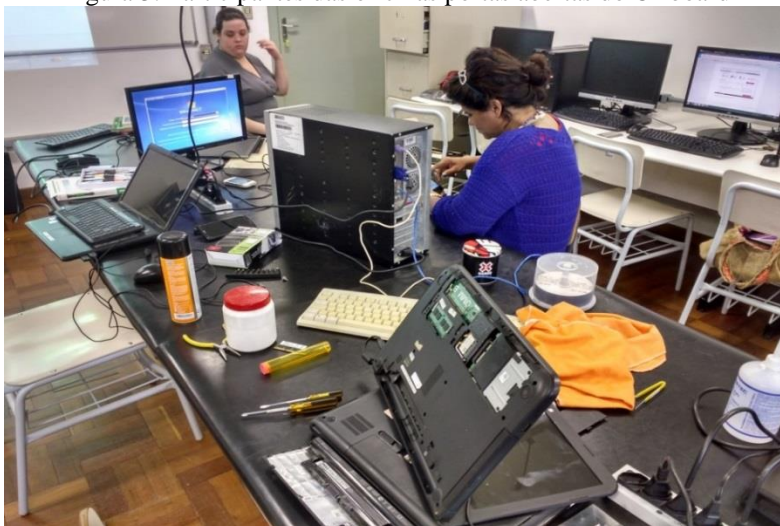
Iniciado em 2016, o projeto de extensão Offboard – Manutenção e suporte a serviço da comunidade visa atender a população do IFRS Canoas seja externa ou interna provendo serviços relacionados a manutenção de computadores.

O projeto recebe doações de equipamentos de informática através da comunidade e por meio de testes específicos em cada componente são separados e descartados corretamente caso apresentem defeito. Os componentes que não apresentarem defeitos são catalogados dentro da plataforma GLPI e armazenados no estoque do projeto.

Com as peças em catálogo os participantes do projeto podem através da montagem de um computador descobrir compatibilidades entre componentes específicos e preparar máquinas para serem doadas para a comunidade.

Todo o conhecimento obtido e explorado pelos participantes durante a realização do projeto pode ser compartilhado, de acordo com Drago, Ribeiro e Silva (2011) o conhecimento compartilhado gera produtos, processos e serviços. As experiências vivenciadas pelos colaboradores precisam ser transmitidas, pois o compartilhamento é vital para o sucesso.

Figura 3: Participantes das oficinas portas abertas do Offboard



Fonte: Autoria própria.

Os problemas apresentados neste trabalho de conclusão foram resultados da observação *in loco*, técnica de levantamento de requisitos descrita por Leite (2000) como a participação do analista dentro da área a ser analisada, a fim de promover bons resultados. A observação ocorreu através do autor atuando como participante dentro do projeto *Offboard* durante o período compreendido entre 2016 à 2017.

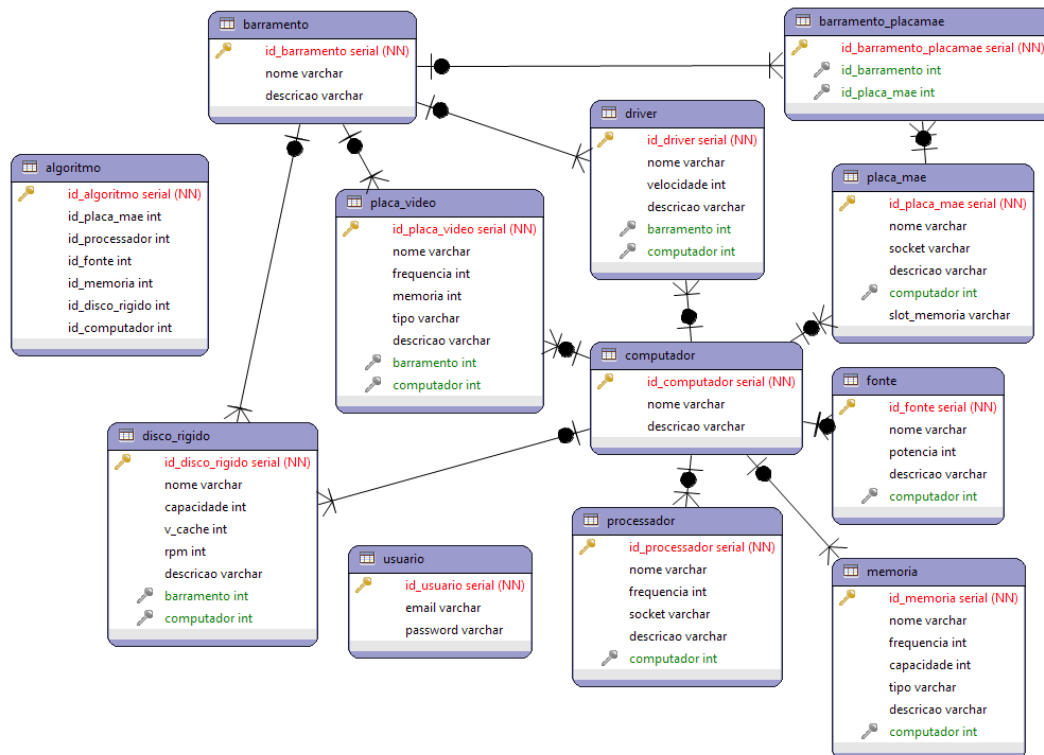
## 6 DESENVOLVIMENTO

A plataforma teve desenvolvimento embasada no conceito de estoques definido por Chiavenato (2005, p.67) como:

A composição de materiais [...] que não é utilizada em determinado momento na empresa, mas que precisa existir em função de futuras necessidades. Assim, o estoque constitui todo o sortimento de materiais que a empresa possui e utiliza no processo de produção de seus produtos/serviços.

Com os conceitos definidos, foi possível definir os requisitos através da análise dos objetivos específicos, identificando as necessidades do usuário ao utilizar um sistema de estoque, os resultados foram transformados em um modelo entidade-relacionamento (Figura 4) visando iniciar a implementação do banco de dados do SmartInv.

Figura 4: Diagrama de entidade-relacionamento representando o banco de dados.

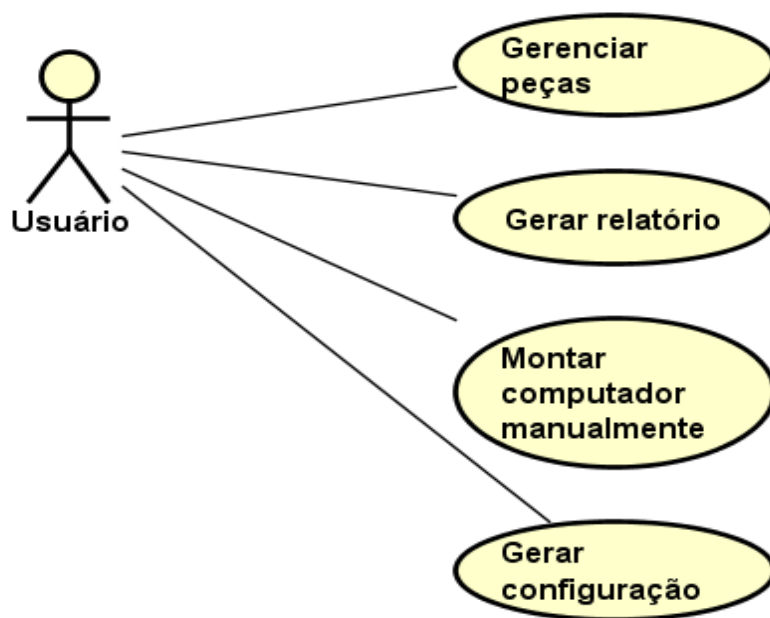


Fonte: Autoria própria.

Todos os componentes do SmartInv se relacionam a uma entidade denominada computador. O registro informando a qual computador aquele componente pertence é armazenado no próprio componente, permitindo existir múltiplos semelhantes vinculados ao mesmo computador.

Após a modelagem do diagrama entidade relacionamento ser finalizada, foi possível aplicar os demais conceitos da UML, através da diagramação dos casos de uso, foi possível identificar as necessidades do ator. Existe apenas um tipo de usuário que é responsável por gerenciar todo o catálogo de peças, incluindo criar, deletar e excluir componentes.

Figura 5: Diagrama de Casos de Uso



Fonte: Autoria própria.

É possível gerenciar o funcionamento do algoritmo 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 sobre o total de peças e computadores catalogados.

Após a modelagem ser finalizada, foi escolhido utilizar o PostgreSQL como o sistema gerenciador de banco de dados, visto que o servidor disponibilizado pela instituição utiliza deste.

Visando obter uma melhor manipulação do SmartInv e automatização na manipulação dos dados, optou-se por utilizar o *framework* Enyalius em conjunto com o padrão *model-view-controller* (MVC), separando a apresentação e a interação dos dados do sistema, estruturando em três elementos (controlador, visualizador e modelo) que interagem entre si (SOMMERVILLE, 2011).

Figura 6: Tela exibindo a lista com um tipo de componente (Ampliada).

Processador					
ID	Nome	Frequência	Socket	Descrição	Computador
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
1	Intel Core i7-6700K	4000	FCLGA1151		
2	Intel Core i3	3100	FCLGA1151		1
3	Intel i3 7100K	3900	FCLGA1151	70° OVERCLOCK	2
4	Intel Core i3	2400	FCLGA1151		
5	Intel Core i5	2200	FCLGA1151	Alcança 90 graus em	

Fonte: Autoria própria.

Com a utilização do Enyalius é possível configurar tabelas para exibição, edição, inserção e remoção dos dados através dos controladores de cada componente. A Figura 6 mostra todos os componentes do tipo Processador listados em uma tabela com suas propriedades específicas (ID, Nome, Frequência, e etc.).

Na Figura 7, podemos observar o cadastro de um componente e as informações requeridas pelo sistema para controlar as peças, na parte inferior esquerda temos um botão para limpar os campos a serem preenchidos e na parte direita um botão para submeter as alterações. A tela de cadastro é padronizada para todos os componentes da plataforma, adaptada para cada componente específico e suas características.

Figura 7: Tela para Cadastro de componentes

SmartInv

[Home](#)
[Componentes](#)
[Algoritmo](#)
[Relatório](#)
[Perfil](#)

Placa Mãe

Nome

ASUS P4SD

Socket

LGA775

Slot Memória RAM

DDR2

Descrição

Corrosão nas trilhas BIOS

Computador

2 - OFFBOARD 02

Limpar

Enviar

© Gabriel Nunes de Siqueira - SMARTINV, 2017.

Fonte: Autoria própria.

Foi implementada uma tela (Figura 8) para o usuário poder cadastrar novos barramentos, doravante denominados interfaces. Cada componente exige uma interface de comunicação com a placa mãe do computador, fator a ser levado em consideração para o funcionamento do algoritmo.



Figura 8: Tela para Cadastro de Interfaces/Barramento

Fonte: Autoria própria.

Para montar um computador, o usuário necessita criar este na respectiva página, em sequência pode acessar as tabelas com os componentes que deseja associar, vinculando individualmente cada peça desejada.

## 6.1 ALGORITMO

O conceito de algoritmos definido por Cormen (2002, p. 3, grifos do autor) afirma que:

Informalmente, um **algoritmo** é qualquer procedimento computacional bem definido que toma algum valor ou conjunto de valores como **entrada** e produz algum valor ou conjunto de valores como **saída**. [...] Também podemos visualizar um algoritmo como uma ferramenta para resolver um **problema computacional**.

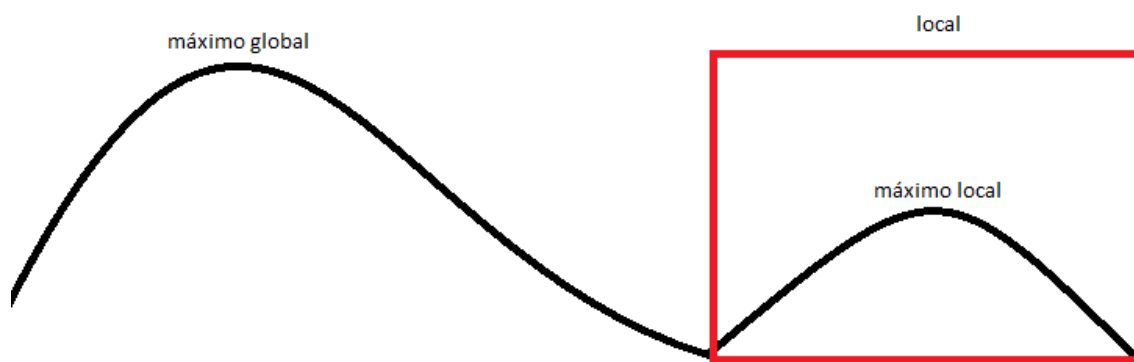
Visando solucionar o problema identificado na questão de montagem de computadores, alguns estudos foram realizados sobre um tipo de algoritmo em particular, denominado *Hill Climbing* definido por Norvig e Russel (2010, tradução nossa) como um *looping* de forma ascendente que percorre os valores em uma região denominada local, ao atingir o valor máximo da região, ele interrompe o fluxo de continuidade.

Conforme a

Figura 9, a área destacada em vermelho denominada de local, corresponde a busca atual do algoritmo, que tende sempre a subir incrementando seus resultados através da busca pelo maior valor. Isto pode gerar um problema futuro, ao atingir um ponto máximo local, já que o algoritmo não verifica seus vizinhos imediatamente próximos, estagnando em um resultado não desejado.

Norvig e Russel (2010, tradução nossa) afirmam que a aplicação ideal para um algoritmo *Hill Climbing* se baseia em uma situação onde não há múltiplos valores máximos locais, caso seja necessário o uso, a solução mais rápida a se utilizar é reiniciar o algoritmo a cada impasse.

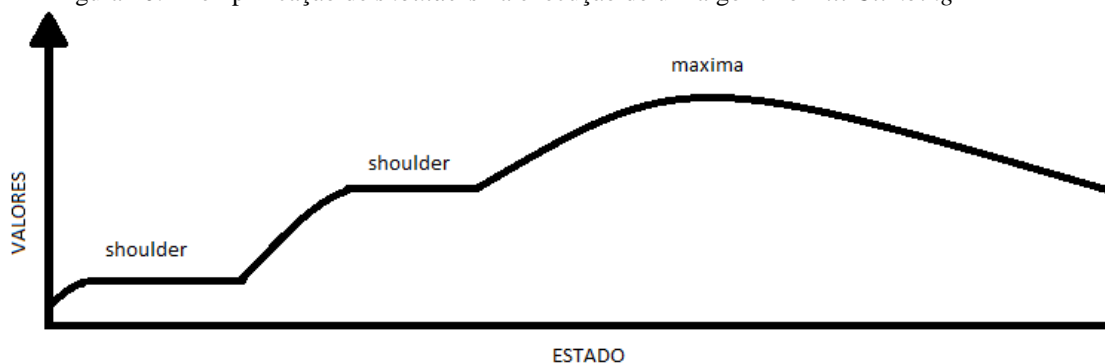
Figura 9: Exemplificação de um algoritmo *Hill Climbing*



Fonte: Autoria própria.

Outro problema recorrente de um algoritmo *Hill Climbing* é sua incapacidade de prosseguir ao chegar em regiões denominadas *shoulders* (ombros) ou regiões planas (Figura 10), como o processo de consulta consiste em verificar qual o ponto máximo consultando o valor anterior, o atual e o próximo, acaba ficando sem avanços dentro da execução ao atingir uma sequência de valores iguais.

Figura 10: Exemplificação de *shoulders* na execução de um algoritmo *Hill Climbing*



Fonte: Autoria própria.

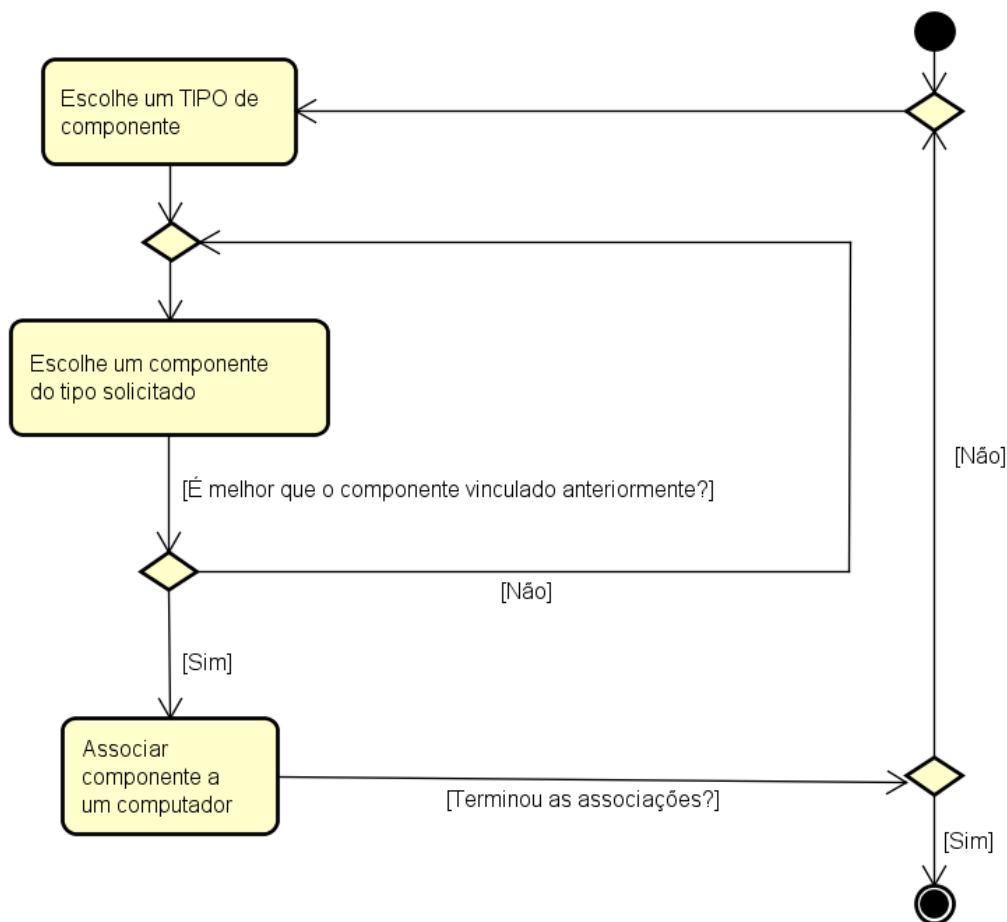
Através da pesquisa realizada, o algoritmo utilizado no SmartInv baseia-se em princípios utilizados no *Hill Climbing*, com exceções e regras adaptados para a utilização em um estoque.

O algoritmo inicia com um *looping* carregando todos os componentes do sistema que não possuem vínculos com computadores já criados, a partir do resultado, cada componente é submetido a testes para forçar um filtro durante as associações.

É realizada uma verificação se o componente possui vínculo com um computador existente, se ele possui um valor superior ao anterior, se é o único disponível para aquele computador ou se a interface ou soquete é compatível com a placa mãe associada inicialmente. A cada *looping* o componente que apresentar uma característica com o valor mais alto entre todos do mesmo tipo é identificado e isolado dentro de uma lista, sendo o valor máximo atual, caso não exista ocorrência de um valor superior, ele é atribuído ao computador.

Quando um componente tem valores idênticos a outro, o algoritmo prossegue para o próximo da lista, sem realizar alterações. Durante a implementação, foi priorizado a capacidade do algoritmo de avançar sempre em busca do maior resultado, não estagnando em pontos específicos. O diagrama de atividades (Figura 11) mostra com detalhes o processo de montagem pelo algoritmo já considerando a compatibilidade entre os componentes.

Figura 11: Diagrama de Atividades representando o algoritmo



Fonte: Autoria própria.

Na execução do algoritmo, todos os números de identificação das peças associadas a um computador são armazenados em uma lista temporária, desse modo, quando uma nova peça é comparada com a anterior e apresenta um resultado melhor, o algoritmo substitui a ID correspondente na lista temporária, possibilitando a peça anteriormente associada ser utilizada novamente. Os detalhes quanto a especificação podem ser observados no Apêndice A – Especificação de Casos de Uso.

Com os resultados obtidos após a execução do algoritmo o usuário é redirecionado para uma tela de configuração, onde pode escolher se deseja manter ou não aquela configuração gerada. Caso o usuário escolha manter a configuração, um novo computador é criado dentro do sistema com as peças diretamente vinculadas.

Para validação do algoritmo e sua eficiência, alguns testes foram realizados; através da consulta em sites de vendas, ocorreu a seleção de dois computadores distintos para serem utilizados como comparação.

O primeiro computador denominado A possui características de ultima geração, componentes caros e sofisticados destinados para a utilização em jogos. O segundo computador, doravante denominado B possui características simples, voltado para o uso doméstico em pequenas aplicações.

As peças de cada um foram isoladas e cadastradas dentro do SmartInv, o objetivo do primeiro teste era executar o algoritmo e resultar na associação das respectivas peças de cada computador semelhante a original.

Computador A<sup>2</sup>:

- Placa Mãe: H110M DDR4;
- Processador: Intel Core i5-7400, 3.0ghz;
- Memória: 8Gb 2133Mhz DDR4;
- Armazenamento: HD 1TB Sata III 3.5";
- Fonte: Corsair VS400 80 Plus White.

Computador B<sup>3</sup>:

- Processador: J3060;
- Placa Mãe: Marca genérica não identificada;
- Memória: 4GB DDR3;
- Armazenamento: HD 1TB Sata III 3.5";
- Fonte: Genérica 200W.

Ao executar o algoritmo com os dados cadastrados, ele imediatamente realizou a associação dos componentes com os respectivos computadores sem encontrar problemas quanto a compatibilidade. Todos os componentes pertencentes ao computador A, identificados pela ID 1 (um) foram devidamente agrupados em um único computador, o mesmo vale para o computador B, identificado pela ID 2 (dois). A exceção ocorreu no disco rígido, pois ambos os discos apresentavam as mesmas características, não causando mudanças no resultado final, como demonstrado na Figura 12.

---

<sup>2</sup> PICHAU. Disponível em <<http://bit.ly/2z7Pu40>>. Acesso em: 09 nov. 2017

<sup>3</sup> PICHAU. Disponível em <<http://bit.ly/2iYXIWN>>. Acesso em: 09 nov. 2017

Figura 12: Resultado dos testes no algoritmo



The image shows a web interface for 'SmartInv'. At the top, there is a dark header with the text 'SmartInv' in white. Below the header, the interface is divided into two sections, each representing a computer's configuration. Section 1, titled '1. Computador', lists: Placa Mãe: 1, Processador: 1, Memória: 1, Disco Rígido: 2, and Fonte: 1. Section 2, titled '2. Computador', lists: Placa Mãe: 2, Processador: 2, Memória: 2, Disco Rígido: 1, and Fonte: 2. Below these sections is a button labeled 'Prosseguir'. At the bottom, there is a copyright notice: '© Gabriel Nunes de Siqueira - SMARTINV, 2017.'

SmartInv

1. Computador

- Placa Mãe: 1
- Processador: 1
- Memória: 1
- Disco Rígido: 2
- Fonte: 1

2. Computador

- Placa Mãe: 2
- Processador: 2
- Memória: 2
- Disco Rígido: 1
- Fonte: 2

Prosseguir

© Gabriel Nunes de Siqueira - SMARTINV, 2017.

Fonte: Autoria própria.

## 6.2 MIGRAÇÃO DE DADOS

Ao decorrer do desenvolvimento do SmartInv, surgiu a necessidade de realizar testes quanto a eficiência do algoritmo proposto, em razão disto, foi pensado aproveitar os dados já existentes dentro da plataforma GLPI do projeto *Offboard*. O estoque do projeto possui aproximadamente 170 peças cadastradas em estoque (OFFBOARD, 2017).

Visto que a plataforma GLPI utiliza o SGBD MySQL e apresenta uma estrutura diferenciada em sua composição de tabelas e colunas, foi necessário desenvolver um código PHP para realizar a migração entre os bancos, garantindo a integridade dos dados, o código carrega todas as tabelas existentes em ambos os bancos de dados com suas respectivas colunas e armazena em listas.

A migração ocorre manualmente, o usuário escolhe a tabela do banco de dados pertencente ao GLPI que deseja migrar e escolhe as colunas em sequência; após as colunas da tabela serem definidas, o usuário escolhe a tabela pertencente ao SmartInv que receberá os dados (Figura 13). O código monta as inserções no banco de dados e o usuário pode executar a inserção em seguida.

Figura 13: Tela para migração de dados entre bancos de dados

glpi\_deviceharddrives
disco\_rigido
Enviar

id
id\_disco\_rigido
Enviar

```

INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(id, designation, rpm, cache, capacity_default, comment)

INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(1, 1 - 32049H2, 5400, -1, 20491, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(2, 2 - WD400 Cassino, -1, -1, 40000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(3, 3 - SV0411N, -1, -1, 40000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(4, 4 - SV0221N, -1, -1, 20000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(5, 5 - ST340015A, -1, -1, 40000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(6, 6 - SP0411N, -1, -1, 40000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(7, 7 - WD400, -1, -1, 40000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(8, 8 - D33019, -1, -1, 40000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(9, 9 - WD400, -1, -1, 40000, );
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(10, 10 - HD161HJ, -1, -1, 160000, );
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(11, 11 - HD161HJ, -1, -1, 160000, );
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(12, 12 - WD8007D, -1, -1, 80000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(13, 13 - MHV2080BH, null, null, 80000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(14, 14 - HM5007I, null, null, 500000, );
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(15, 15 - WD800, null, null, 80000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(16, 16 - HDD2190, , , 40000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(18, 18 - ST3200822A, 1, 1, 200000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(21, 21 - ST3200822A5, 1, 1, 200000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(22, 22 - ST340016A, 0, 0, 40000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(23, 23 - 8T380011A, 0, 0, 80000, );
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(25, 25 - MK1246GSX, , , 120000, Não testado);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(26, 26 - MK2035GSS, , , 200000, Não testado);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(27, 27 - HM321HI, , , 320000, Não testado);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(28, 28 - WD400, , , 40000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(29, ID - 29, -1, -1, 20000, NÃO TESTADA);
);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(30, ID - 30, , , 20000, NÃO TESTADA);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(31, ID - 31, , , 40000, NÃO TESTADA);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(32, 32 - SP0411N, -1, -1, 40000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(33, 33 - HD161GJ, 7200, 8, 160000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(34, 34 - ST38410A, -1, -1, 8400, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(35, 35 - SP2001H, -1, -1, 20000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(36, 36 - NAR61HA0, -1, -1, 40000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(37, 37 - SP0411N, 7200, 2, 40000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(38, 38 - SP0411N, -1, -1, 40000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(39, 39 - STM3160215A, -1, -1, 160000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(40, 40 - SP0411N, -1, -1, 40000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(41, 41 - VAM517J0, -1, -1, 20000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(42, 42 - SP0411N, -1, -1, 40000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(43, 43 - D33019, -1, -1, 40000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(44, 44 - ST320014A, -1, -1, 20000, OK);
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(45, 45 - D33019, -1, -1, 40000, );
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(46, 46 - MPE3084AE, -1, -1, 20000, );
INSERT INTO disco_rigido(id_disco_rigido, nome, v_cache, v_cache, capacidade, descricao) VALUES(47, 47 - SP1644N, -1, -1, 160000, );

```

Fonte: Autoria própria.

## 6.3 RELATÓRIOS

Para um bom funcionamento do estoque e controle de inventário, a plataforma possui um recurso utilizado para gerar relatórios acerca do número de peças ou computadores cadastrados, existem dois tipos de relatórios a serem gerados; um controle sobre a quantidade de peças cadastradas no estoque Figura 14 e uma relação com todos os computadores e suas respectivas peças associadas.

Figura 14: Tela representando o relatório de peças cadastradas

SmartInv					Home Componentes Algoritmo Relatório Perfil				
					O total de Placas Mãe: 5 O total de Processadores: 6 O total de Memórias RAM: 4 O total de Disco Rígidos: 5 O total de Placas de Vídeo: 1 O total de Fontes de Alimentação: 2 O total de Drivers: 1 A soma total de componentes em estoque é: 24				
ID	Nome	Slot	Socket	Computador					
3	Teste	DDR4	123	Não possui computador					
1	H110M-HG4	DDR2	FCLGA1151	Vinculado					
2	asdasd	DASD	213	Vinculado					
10	Teste	DDR2	FCLGA1151	Não possui computador					
4	Placa mãe 775		LGA775	Vinculado					

ID	Nome	Socket	Frequência	Computador
3	Intel i3 7100K	FCLGA1151	3900	Vinculado
1	Intel Core i7-6700K	FCLGA1151	4000	Não possui computador
2	Intel Core i3	FCLGA1151	3100	Vinculado
5	Intel Core i5	FCLGA1151	2200	Não possui computador
4	Intel Core i3	FCLGA1151	2400	Não possui computador
13	Processador Teste	213	250	Não possui computador

ID	Nome	Frequência	Capacidade	Tipo	Computador
1	CMK8GX4M2A2400C14	2400	8000	DDR4	Não possui computador
2	Apresentação 2	1000	8000	DDR2	Não possui computador
3	Teste	500	6000	DDR2	Não possui computador
5	Memória	600	4000	DDR2	Vinculado

ID	Nome	Capacidade	Cache	RPM	Computador
2	MH-V2000BH	80000	8	5400	Não possui computador
1	ST1000DM010	1000000	64	7200	Não possui computador
6	AA	1000	32	8000	Não possui computador
3	WD400	40000	2	5400	Não possui computador
4	WD800	80000	2	5400	Não possui computador

ID	Nome	Frequência	Memória	Tipo	Computador
1	GEFORCE GTX 1080	10000	8000	GDDR5X	Vinculado

ID	Nome	Potência	Computador
1	Teste DDR4	600	Não possui computador
2	teste	500	Não possui computador

Fonte: Autoria própria.

Ao acessar a página, todos os dados existentes no banco são carregados e formatados para serem impressos, assim o usuário consegue manter o controle sobre a circulação de itens em estoque.

Figura 15: Tela representando o relatório com todos os computadores existentes (Ampliado).

SmartInv		
1. Computador 1 <ul style="list-style-type: none"> <li>Placa Mãe: 1 - H110M-HG4</li> <li>Processador: 2 - Intel Core i3</li> <li>Memória: 5 - Memória</li> <li>Disco Rígido: -</li> <li>Fonte: -</li> <li>Drivers: 1 - iHAS122-14</li> </ul>		
2. Computador 2 <ul style="list-style-type: none"> <li>Placa Mãe: 4 - Placa mãe 775</li> <li>Processador: 3 - Intel i3 7100K</li> <li>Memória: 1 - CMK8GX4M2A2400C14</li> <li>Disco Rígido: 4 - WD800</li> <li>Fonte: -</li> <li>Drivers: -</li> </ul>		
© Gabriel Nunes de Siqueira - SMARTINV, 2017.		

Fonte: Autoria própria.

## 7 CONCLUSÃO

Durante o desenvolvimento do presente trabalho de conclusão, foi possível obter resultados significantes relacionados aos objetivos propostos no Capítulo 2. Atendendo aos objetivos, respectivamente:

- i. Através da observação *in loco* dentro do projeto Offboard, foi possível identificar que a inexistência da gestão de estoques aplicada na empresa ou projeto, como no caso do Offboard, resultaria na ineficiência dos serviços. A crescente demanda de serviços relacionados ao estoque requer um conhecimento prévio sobre a quantidade de itens disponíveis no acervo. O não funcionamento do controle de estoque pode gerar prejuízos e atrasos constantes dentro do setor.
- ii. Através da análise de estudos sobre algoritmos, é possível identificar diversos tipos para infinitas aplicações. Os algoritmos podem ser aplicados para solucionar desde pequenos problemas relacionados a situações do cotidiano; até solucionar problemas extremamente complexos que utilizam diversas técnicas e recursos, como a heurística, recursos matemáticos e inteligência artificial.
- iii. Foi possível identificar diversas ferramentas novas para a resolução do problema. Aplicando os conhecimentos adquiridos ao longo do curso em conjunto com novas tecnologias, possibilitou-se estabelecer uma base para o desenvolvimento da plataforma, a pesquisa sobre novas ferramentas se tornou essencial; as novas tecnologias corroboraram para o aperfeiçoamento do conhecimento obtido durante o curso.
- iv. A plataforma foi desenvolvida de acordo com a modelagem inicial, atendendo os padrões estabelecidos pela UML. As funcionalidades referentes ao controle de estoque demonstram atender os objetivos propostos inicialmente, o usuário tem total controle sobre os componentes a serem catalogados, podendo inserir, apagar e alterar os dados sem restrições.
- v. O algoritmo desenvolvido foi testado e demonstra bons resultados quando aplicado em um estoque com diversas peças catalogadas, a única exceção para o desempenho do algoritmo é a falha humana, ocorrendo inconsistência no preenchimento das informações relacionadas a cada componente.
- vi. A migração dos dados entre a plataforma GLPI e o SmartInv demonstrou ser eficiente, ao selecionar as tabelas o usuário tem total controle sobre os dados a serem transferidos em razão da consulta e exibição serem simultâneas evitando possíveis acidentes com informações erradas.

Todo o conhecimento ao longo do curso demonstrou-se necessário durante o desenvolvimento da plataforma, desde modelagens e padrões de engenharia de software até técnicas e linguagens de programação.



## 7.1 TRABALHOS FUTUROS

Ao decorrer do desenvolvimento da plataforma, alguns trabalhos futuros puderam ser identificados, visando complementar a implementação do SmartInv. No futuro, se pensa no aperfeiçoamento do algoritmo, refinando suas restrições e tornando-o capaz de calcular o consumo de um computador gerado, podendo associar uma fonte de alimentação adequada.

Quanto à plataforma no geral, analisa-se o desenvolvimento de uma funcionalidade para realizar chamados, onde outro usuário com diferentes níveis de acesso pode solicitar auxílio técnico em um determinado computador. Como complemento, a transformação da plataforma em um sistema *helpdesk* pode ser uma alternativa, expandindo as áreas de atuação.

Ainda visando o futuro, planeja-se o desenvolvimento de relatórios através de gráficos específicos, permitindo ao usuário ter um controle melhor sobre os dados cadastrados na plataforma.

## REFERÊNCIAS

BOOTSTRAP. Bootstrap Documentation. Disponível em: <<https://getbootstrap.com/>>. Acesso em: 09 jun. 2017.

BOOCH, Grady. **Object-oriented analysis and design**. 2°. ed. Santa Clara, California: Addison-Wesley, 1998. 543 p.

CHIAVENATO, Idalberto. **Administração de Materiais**: Uma abordagem introdutória. 3° ed. Rio de Janeiro: Elsevier, 2005. 166 p.

CHIPART. **Sobre a Chipart**. Disponível em: <<https://www.chipart.com.br/pagina/sobre-a-chipart-1>>. Acesso em: 10 set. 2017.

CORMEN, Thomas; et al. **Algoritmos**: teoria e prática. 6ª. ed. Rio de Janeiro: Elsevier, 2002. 934 p.

CRAFTMYBOX, **CraftMyBox - Sobre Nós**. Disponível em: <<http://craftmybox.com/sobre-nos>>. Acesso em: 11 set. 2017.

DRAGO, I.; SCROCH, K. A. S.; RIBEIRO, M. R.; SILVA, H. de F. N. **Metodologias que estimulam o compartilhamento de conhecimentos**: a experiência do Global Forum América Latina - GFAL. AtoZ, Curitiba, v. 1, n. 1, p. 38-49, jan./jun. 2011. Disponível em: <<http://revistas.ufpr.br/atoz/article/view/41282/25204>>. Acesso em 20/10/2017.

ELMASRI, Ramez; NAVATHE, Shamkant. **Sistema de Bancos de Dados**. 4°. ed. Porto Alegre: Pearson, 2005. 798 p.

GERHARDT, Tatiana; SILVEIRA, Denise. **Métodos de pesquisa**. 1ª. ed. Porto Alegre: Editora da UFRGS, 2009. 120 p.

GLPI Development Team. **About GLPI**. Disponível em: <<http://glpi-project.org/spip.php?rubrique26>>. Acesso em: 15 set. 2017.

LARMAN, Craig. **Applying UML and Patterns**: An Introduction to Object-Oriented Analysis and Design and Iterative Development. 3ª. ed. [S.l.]: Addison Wesley Professional, 2004. 736 p.

LEITE, Jair. **Análise e Especificação de Requisitos**. Disponível em: <<https://www.dimap.ufrn.br/~jair/ES/c4.html>>. Acesso em: 18 out. 2017.

MORIMOTO, Carlos. **Hardware**: o guia definitivo. Porto Alegre: Sul Editores, 2007. 1038 p.

RUSSEL, Stuart; NORVIG, Peter. **Artificial Intelligence**: A Modern Approach. 3ª. ed. New Jersey: Pearson Education, 2010. 1154 p.

SOMMERVILLE, Ian. **Engenharia de Software**. 9ª. ed. São Paulo: Pearson Prentice Hall, 2011. 544 p.

THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL**. Disponível em: <<http://www.postgresql.org>>. Acesso em: 11 jun. 2017.

## APÊNDICE A – ESPECIFICAÇÃO DE CASOS DE USO

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

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

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

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