Instituto Federal de Educação, Ciência e Tecnologia de São Paulo

Campus São Paulo

Guilherme Oliveira de Souza Leão – SP3013243

Pedro Brenicci Freitas – SP3013154

IODWare

São Paulo

2019

Instituto Federal de Educação, Ciência e Tecnologia de São Paulo

Campus São Paulo

IODWARE

Trabalho de avaliação como requisito parcial para a conclusão da disciplina Banco de Dados II (BD2A3) do curso de Tecnologia em Análise e Desenvolvimento de Sistemas, sob orientação do Professor Me. Francisco Veríssimo Luciano.

São Paulo

2019

**Resumo**

O IODWare é uma plataforma que possui registros de diferentes peças de hardware relacionadas a composição de um computador de mesa, visando oferecer suporte à montagem do computador ao usuário.

O usuário terá acesso a uma tela onde ele poderá escolher entre diferentes tipos de componentes, alguns obrigatórios e outros opcionais. Após a definição dos componentes do computador, o programa irá indicar possíveis problemas relacionados à montagem, como incompatibilidade ou falta de ineficiência entre componentes, nesses casos, o usuário receberá um alerta indicando o problema e caberá a ele decidir continuar assim mesmo ou retornar à primeiro etapa. No final o programa indicará links de compra online dos componentes e um preço aproximado dos custos de cada componente e um total.

O objetivo deste software é facilitar e agilizar o planejamento da montagem de computadores, facilitando para pessoa com baixos conhecimentos sobre o assunto a tomarem melhores decisões e auxiliar, incluindo usuários mais avançados, a terem uma estimativa de custos e fornecedores.

**Palavras-chave:** Hardware, Computador, Peças, Usuário, Componentes.

LISTA DE FIGURAS

LISTA DE QUADROS E TABELAS

LISTA DE ABREVIAÇÕES E SIGLAS

Sumário

[1- Introdução 8](#_Toc5264923)

[1.1- Descrição do Problema 9](#_Toc5264924)

[1.2- Solução Proposta 10](#_Toc5264925)

[1.3- Objetivo 11](#_Toc5264926)

[1.3.1- Geral 12](#_Toc5264927)

[1.3.2- Específico 12](#_Toc5264928)

[1.4- Análise de Concorrência 13](#_Toc5264929)

[1.5- Estrutura do Trabalho 14](#_Toc5264930)

[2- Revisão da literatura 14](#_Toc5264931)

[2.1- Engenharia de Software 14](#_Toc5264932)

[2.2- Banco de Dados 14](#_Toc5264933)

[2.3- Análise de Requisitos 15](#_Toc5264934)

[3- Projeto Lógico 16](#_Toc5264935)

[4- Projeto Físico 16](#_Toc5264936)

# Introdução

## Descrição do Problema

Usuários de computador que são leigos na área tem problemas em escolher peças de hardware, pois não fazem ideia do que funciona bem com o que. É um grande problema pois muitas lojas tentam enganar pessoas vendendo peças ruins fingindo ser boas.

## Solução Proposta

Criar um aplicativo que mostre combinações de peças que façam sentido e trabalhem bem juntas, com opções e comentários sobre como tal peça funciona com qual.

## Objetivo

Fazer um aplicativo em Java com conexão a um banco de dados com dados das peças guardados. O cliente acessa o aplicativo e ao procurar informações sobre um item, o sistema acessa o banco de dados e exibe as informações.

### 1.3.1- Geral

### 1.3.2- Específico

## Análise de Concorrência

Existem vários sites de análise de hardware em países da Europa e América do Note, mas no Brasil não existe nenhum grande nome neste ramo. É possível fazer o negócio crescer com mais tranquilidade com a falta de concorrência.

## 1.5- Estrutura do Trabalho

# 2- Revisão da literatura

## 2.1- Engenharia de Software

## 2.2- Banco de Dados

## 2.3- Análise de Requisitos

|  |  |
| --- | --- |
| F1 Definição da composição do computador | Oculto( ) |
| **Descrição :**   * O usuário deverá selecionar os componentes que ele deseja a partir de uma lista de opções, onde o software deverá notifica-lo em caso de incompatibilidade ou ineficiência devido a uma combinação não recomendada. * O usuário deverá pode definir para que finalidade o computador será utilizado, e o software fará recomendações de acordo com a finalidade escolhida e o custo benefício. | |

|  |  |
| --- | --- |
| F2 Validação do projeto de computador | Oculto( ) |
| **Descrição :**   * A falta de componentes mínimos ou incompatibilidade entre eles deverá ser acusado pelo programa. * Caso a placa seja on-board, o programa não deverá acusar a falta de componentes por falta de uma placa de vídeo. * Não configura falta de componentes se o projeto incluir um SSD sem um HD, vice-versa ou ambos, mas a ausência de ambos deverá acusar falta de componentes. * O software deverá exibir fornecedores que oferecem o componente desejado e o preço. Com base no fornecedor e o preço, o software deverá ser capaz de gerar um preço final. * O programa poderá criar uma lista de compras em documento txt apontando os componentes e fornecedores em potencial, se o usuário assim desejar. | |

|  |  |
| --- | --- |
| F3 Banco de dados | Oculto( ) |
| **Descrição :**   * O banco de dados deve ser relacional. * Deverá ser atualizável para eventuais mudanças tecnológicas. * O banco de dados deverá armazenar registos de: placas mãe, HDs, Placas de vídeos, Processadores, Coolers entre outros componentes relevantes para a montagem de um computador. * O banco de dados deverá ser modelado em SQL. | |

|  |  |
| --- | --- |
| F4 Programação | Oculto( ) |
| **Descrição :**   * A programação deverá ser feita em Java. * A programação deverá estar de acordo com princípios GRASP * A programação deverá estar de acordo com princípios SOLID | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requisitos Não Funcionais** | | | | |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| NF1 Iteração usuário software | O programa deverá dar suporte tanto para usuários avançados como leigos, logo deverá ser de fácil entendimento e tentar ser o mais simples e claro possível nas descrições técnicas da placa. | Usabilidade | (x) | ( ) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requisitos Não Funcionais** | | | | |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| NF2 Orientação ao usuário | O programa também deverá fornecer indicações para que tipos de componentes são recomendados para as atividades pretendidas pelo usuário (como se uma placa de vídeo é recomendada para a montagem de um pc gamer). | Usabilidade | (x) | ( ) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requisitos Não Funcionais** | | | | |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| NF3 Confiabilidade e consistência dos dados | O software deverá ser capaz de armazenar informações relativas a sua funcionalidade, sendo altamente aberto a atualizações, a fim de manter os dados atualizados em relação aos componentes de hardware disponíveis no mercado e seus fornecedores. | Manutenabilidade | (x) | (x) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requisitos Não Funcionais** | | | | |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| NF4 Forma de cesso ao software | O software deverá ser utilizado off-line, tornando-o altamente disponível, porém o uso de versões antigas pode comprometer a confiabilidade. | Disponibilidade | ( ) | ( ) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requisitos Não Funcionais** | | | | |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| NF5 Qualidade dos dados | O banco de dados dessa aplicação exige atualizações contínuas para que os dados reflitam a realidade, e assim, garantir que o suporte seja benéfico. | Confiabilidade | ( ) | ( ) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requisitos Não Funcionais** | | | | |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| NF6 Compatibilidade com diferentes dispositivos | O software deverá ser possível de ser utilizado em diferentes sistemas operacionais, para auxiliar nisso, deverá ser usada uma linguagem de programação multiplataforma. | Portabilidade | (x) | (x) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requisitos Não Funcionais** | | | | |
| **Nome** | **Restrição** | **Categoria** | **Desejável** | **Permanente** |
| NF7 Reaplicação de recursos desenvolvidos | O banco de dados e a programação deverão ser reutilizáveis com poucas modificações para possível desenvolvimento de futuros softwares que exijam registros sobre hardware. | Reusabilidade | (x) | ( ) |

# 

# 3- Projeto Lógico

# 4- Projeto Físico