

PUC Minas - Informática

Laboratório de Projeto de Software

MANUTENBUS

Pedro Diógenes
Carlos Alberto Nassif
João Rene
Bernardo Victor

Modificado de <https://github.com/vitornere/partiuformar/wiki>

Histórico da Revisão

Data	Versão	Descrição	Autor
05/06/2020	1.0	Versão Inicial	Pedro Diógenes

Visão

Manutenbus

Versão 1.4

Histórico da Revisão

Data	Versão	Descrição	Autor
05/06/2020	1.0	Documento inteiro	Pedro Diógenes

1. Introdução

Este documento tem como intenção, analisar e definir os recursos necessários para realizar com sucesso o desenvolvimento do software **Manutenbus**. Para isso, serão necessários a elaboração de alguns artefatos como descrição de caso de uso, diagrama de sequência e planejamentos das iterações que serão

apresentados ao longo deste documento e que são necessários para satisfazer de forma completa os usuários na utilização deste software.

1.1 Finalidade

Este documento contém os fatores mais relevantes que levarão à construção do software **Manutenbus**. Esse projeto será desenvolvido pelos estudantes da PUC-MG, campus Coração Eucarístico. Serão apresentadas aqui todas as principais características, finalidades e motivações para o desenvolvimento desta aplicação, também irá descrever características gerais do sistema, os envolvidos no seu desenvolvimento assim como o público alvo.

1.2 Escopo

O projeto tem como objetivo desenvolver um sistema que permite que o cliente possa acompanhar o processo de manutenção de uma forma dinâmica e em tempo real, realizar um planejamento a partir das informações disponíveis, ser possível organizar e criar metas. Poderá acompanhar a manutenção de sua frota e categorizar quais veículos tem o melhor desempenho.

1.3 Visão Geral

Faremos uso deste nosso documento como uma forma de organizar informações necessárias para futuras atividades do projeto em desenvolvimento. Informações acerca do contexto do projeto, funcionalidades, e características serão organizados e estarão disponíveis neste documento afim de poupar o atraso no processo de desenvolvimento do projeto.

1.4 Acrônimos

CPF: Cadastro de Pessoa Física.
JS/js: JavaScript
MVC – Model View Controller

2. Posicionamento

2.1 Descrição do Problema

As empresas que gerenciam a frota de ônibus na nossa cidade tem problemas com a gestão da manutenção dos veículos da sua frota. O projeto tem como objetivo remediar isso dando autonomia aos gestores da empresa para que exerçam maior controle sobre alguns pontos específicos das manutenções requeridas.

O problema de	Falha no gerenciamento da manutenção frota de ônibus
Afeta	Mecânicos, motoristas e consequentemente usuários do transporte coletivo
Cujo impacto é	Possíveis atrasos na vida cotidiana de diversas pessoas
Uma solução seria	Uma plataforma que melhor controle a gerência de manutenção da frota de ônibus de forma a evitar atrasos e má gerência da mesma.

2.2 Sentença de Posição do Sistema

Para	Gerentes e mecânicos da empresa de ônibus
Que	Necessitam acompanhar, monitorar e gerenciar as manutenções dos ônibus da frota
O	Manutenbus auxilia os trabalhadores
Que	Necessitam maior controle sobre a frota para evitar possíveis atrasos na vida cotidiana
O Manutenbus	É um <i>software</i> que possibilita iniciar, gerenciar, e manter controle sobre manutenções da frota de ônibus
Nosso sistema	Será de grande ajuda para que os trabalhadores da empresa possam gerenciar de forma eficiente as manutenções dos ônibus

3. Descrições dos Envolvidos e Usuários

3.1 Resumo dos Envolvidos

Nome	Descrição	Responsabilidades	Interesse	Decisão
------	-----------	-------------------	-----------	---------

Empresas de ônibus	Empresas que usarão o software para gerenciar a sua frota	Gerenciar a manutenção de sua frota	Alto	Alto
Desenvolvedores e Gerentes	Alunos que implementarão e integrarão o projeto descrito	Irão produzir o sistema visionado e projetado nos documentos posteriores	Alto	Alto
Passageiros que dependem de ônibus	Pessoas que dependem do ônibus para ir ao trabalho, médico, etc		Alto	Baixo

3.2 Resumo dos Usuários

Nome	Descrição
Mecânicos	Funcionários que devem realizar a manutenção física, usam o sistema para alocar ferramentas e peças para a manutenção em que estiver trabalhando
Gerentes	Gerenciam as manutenções e os funcionários
Administradores do Sistema	Responsáveis por manter as informações no sistema

3.3 Principais Necessidades dos Usuários ou dos Envolvidos

Necessidade	Prioridade	Preocupações	Solução Proposta
-------------	------------	--------------	------------------

Iniciar manutenção	Alta	Deve ser possível iniciar a manutenção a partir do programa, escolhendo uma valeta na oficina e também funcionários para trabalhar na mesma	Armazenar dados das valetas e dos funcionários para escolha dos mesmos ao iniciar a manutenção
Listar ônibus e seus pneus	Alta	Deve ser permitido que se liste todos os ônibus da rota para que se possa verificar quais ônibus necessitam de manutenção	Armazenar informações relevantes de todos os ônibus e seus pneus para verificação dos mesmos
Notificação de necessidade de manutenção	Baixa	Deve ser enviada uma notificação quando um ônibus necessitar manutenção	Implementar um sistema de notificações que utiliza os dados do ônibus
Alocar peças e ferramentas para a manutenção	Média	O mecânico deve ser capaz de escolher peças e ferramentas que sejam necessárias para a manutenção do ônibus	Armazenar dados das ferramentas e peças, desenvolver sistema em que o mecânico pode alocar as peças e ferramentas para a manutenção que estiver realizando

Alta: Alta necessidade no sistema deve ser implementado primeiramente.

Média: Necessário ao sistema, porém pode ser implementado posteriormente.

Baixa: A falta da implementação não gera perdas ao sistema, podendo ser implementado por último.

4. Ambiente do Usuário

O sistema será implementado e testado em um servidor web, onde poderá ser acessado de qualquer navegador com acesso a internet.

5. Visão Geral do Sistema

5.1 Perspectiva do Sistema

O sistema tem como objetivo organizar a gerência de uma frota de ônibus para que a mesma possa ser realizada sem atrasos e dificuldades.

5.2 Suposições e Dependências

Para a utilização do sistema é suposto que a oficina possua conexão com internet. O usuário precisar ter um conhecimento mínimo sobre como realizar a manutenção dos ônibus.

5.3 Licenciamento e Instalação

Todas as tecnologias utilizadas durante o desenvolvimento e utilização do sistema são “**open-source license**”, ou seja código aberto.

Requisitos de Software

Versão 1.1

Histórico da Revisão

Data	Versão	Descrição	Autor
25/03/2020	1.0	Iniciação, definição e finalização do documento	Pedro Diógenes João Rene Carlos Alberto Bernardo Victor

1. Introdução

1. Requisitos Funcionais do Sistema

Esta seção tem a função de informar as características e funcionalidades do sistema **Manutenbus**.

1.1 Tabela de Requisitos

Requisito	Descrição	Prioridade
RF01 – CRUD Ônibus	O sistema deve permitir que o usuário realize cadastro, recuperação, atualização e remoção dos dados de um ônibus.	Alta
RF02 – CRUD Pneus	O sistema deve permitir que o usuário realize cadastro, recuperação, atualização e remoção dos dados dos pneus.	Média
RF03 – CRUD Peças	O sistema deve permitir que o usuário realize o cadastro, recuperação, atualização e remoção dos dados das peças dos ônibus.	Alta

RF04 – CRUD Funcionário	O sistema deve permitir que o usuário realize o cadastro, recuperação, atualização e remoção dos dados dos funcionários.	Alta
RF05 – CRUD Ferramentas	O sistema deve permitir que o usuário realize o cadastro, recuperação, atualização e remoção dos dados das ferramentas na oficina.	Alta
RF06 – Notificação de necessidade de manutenção	O sistema deve enviar ao usuário uma notificação quando qualquer um dos ônibus necessitar manutenção. Isso ocorre quando qualquer ônibus atinge 10.000 km rodados	Baixa
RF07 – Manutenção	O sistema deve permitir que o gerente a manutenção de um ônibus, alocando-a em uma valeta. A manutenção deverá estar disponível para o mecânico na sua página inicial.	Alta
RF08 – Alocar funcionários	O sistema deve permitir ao gerente alocar mecânicos adicionais à uma manutenção	Alta
RF09 – Alocar peças e ferramentas	O sistema deve permitir o mecânico selecionar peças e ferramentas que serão utilizadas na manutenção que estiver ativa.	Média

2. Requisitos Não Funcionais do Sistema

RNF01 – Tecnologia BD – O sistema deve utilizar o banco de dados MySQL.

RNF02 – Aplicação – O sistema deve ser uma aplicação web e deve ser possível rodar em qualquer computador com acesso a internet.

RNF03 – Tempo de resposta – O sistema deve ter um tempo de resposta menor do que 10 segundos para todas suas funções.

RNF04 – Interface – O sistema deve ter uma interface fácil e intuitiva de se utilizar.

RNF05 - Treinamento - Usuários deverão conseguir operar o sistema após uma semana de treinamento.

3. Regras de Negócio

RN01 - O sistema deverá atender às leis brasileiras, etc.

RN02 - O padrão de quilometragem para o sistema atribuir um ônibus para manutenção é de 10.000 km, ou 6 meses desde a última manutenção.

Especificação de Caso de Uso

Versão 1.0

Histórico da Revisão

Data	Versão	Descrição	Autor
25/03/2020	1.0	Especificação do caso de Uso	Pedro Diógenes, João Rene, Carlos Alberto, Bernardo Victor

UC01 – <<CRUD>> (genérico)	
Descrição	<p>Este caso de uso é uma generalização dos demais casos de uso que envolvem CRUDs. O “elemento” em questão, neste detalhamento, é substituído em cada caso de uso ao qual se aplica esta generalização.</p> <p>Generalização de - <<CRUD>> <i>ônibus</i>, <<CRUD>> <i>pneus</i>, <<CRUD>> <i>peças</i>, <<CRUD>> <i>funcionários</i>, <<CRUD>> <i>ferramentas</i>.</p>
Atores	Gerente da frota
Prioridade	Alta
Requisitos associados	RF01, RF02, RF03, RF04, RF05
Fluxo Principal	<ol style="list-style-type: none">1. O gerente (ator) deseja manter os dados de um elemento, e então inicia o caso de uso2. O sistema exibe uma tela com as opções: Pesquisar, Alterar, Incluir, e Excluir<ol style="list-style-type: none">a. Consultar: ver variante Consultar (2.1)b. Alterar: ver variante Alterar (2.2)c. Incluir: ver variante Incluir (2.3)d. Excluir: ver variante Excluir (2.4)3. O sistema valida os dados da operação realizada4. O sistema atualiza a base de dados com a operação realizada.
Fluxos Alternativos (variantes)	<p>Fluxo 2.1 - Consultar:</p> <ol style="list-style-type: none">1. O sistema exibe uma tela listando todos os elementos em questão, ordenados pelo seu atributo primário

	<ol style="list-style-type: none"> 2. O usuário seleciona um dos itens disponíveis 3. O sistema apresenta os dados do item selecionado 4. O sistema encerra o caso de uso <p>Fluxo 2.2 - Alterar:</p> <ol style="list-style-type: none"> 1. O sistema chama a variante 2.1 (Consultar) 2. O usuário altera os dados apresentados do item selecionado 3. Continua no passo 3 do fluxo principal <p>Fluxo 2.3 - Incluir:</p> <ol style="list-style-type: none"> 1. O sistema exibe uma tela para inserção dos dados relevantes ao elemento em questão 2. O usuário entra com os dados requisitados 3. O sistema define a data de inclusão como a data atual 4. Continua no passo 3 do fluxo principal <p>Fluxo 2.4 - Excluir:</p> <ol style="list-style-type: none"> 1. O sistema chama a variante 2.1 (Consultar) 2. O usuário confirma a exclusão do item em questão 3. Continua no passo 4 do fluxo principal
--	--

UC02 – MANUTENÇÃO	
Descrição	Esse caso de uso está relacionado a manutenção dos ônibus, agendar e gerenciar manutenções, assim como controlar desgaste dos pneus, alertar sobre possíveis problemas, atribuir funcionários e controlar número de peças
Atores	Gerente da frota, Mecânico
Prioridade	Alta
Requisitos associados	RF07, RF08
Fluxo Principal	<ol style="list-style-type: none"> 1. O gerente de frota quer fazer o controle da manutenção. 2. O sistema exibe uma tela com as seguintes opções: <ol style="list-style-type: none"> a. Agendamentos: ver variante Agendamento (2.1) b. Desgaste ônibus: ver variante Onibus (2.2) c. Desgaste pneus: ver variante Pneus (2.3) d. Controle de peças: ver variante Pecas (2.4) e. Funcionários alocados: ver variante Funcionários (2.5)

	<ol style="list-style-type: none"> 3. O sistema valida os dados da operação realizada 4. O sistema atualiza a base de dados com a operação realizada.
Fluxos Alternativos (variantes)	<p>Fluxo 2.1 - Agendamento:</p> <ol style="list-style-type: none"> 1. O sistema abre uma tela com os agendamentos de manutenção previstas 2. O Usuário seleciona um dos agendamentos previsto 3. O Usuário deve aprovar o agendamento ou descartar e escolher uma nova data 4. O sistema encerra o caso de uso <p>Fluxo 2.2 - Ônibus:</p> <ol style="list-style-type: none"> 1. O sistema abre uma tela com os ônibus da empresa 2. O usuário seleciona um dos ônibus disponíveis 3. O sistema apresenta os dados do item selecionado 4. O sistema permite agendar manutenção para o ônibus. 5. O sistema encerra o caso de uso <p>Fluxo 2.3 - Pneus:</p> <ol style="list-style-type: none"> 1. O sistema abre uma tela com todos pneus, quilometragem, os ônibus que estão alocados e a posição. 2. O usuário seleciona um dos ônibus disponíveis 3. O sistema permite agendar troca da posição de pneus, ou troca de veículos. 4. O sistema encerra o caso de uso <p>Fluxo 2.4 - Peças e ferramentas:</p> <ol style="list-style-type: none"> 1. O sistema abre uma tela com todos as peças e ferramentas disponíveis e quantidades na oficina. 2. O usuário seleciona uma peça ou ferramenta. 3. O sistema permite alocar a peça ou ferramenta para a manutenção 4. O sistema encerra o caso de uso <p>Fluxo 2.5 - Funcionários:</p> <ol style="list-style-type: none"> 1. O sistema abre uma tela com todos os funcionários da oficina.

	2. O sistema permite realocar o funcionário para outro trabalho. 3. O sistema encerra o caso de uso
--	--

UC03 – Notificação de necessidade de Manutenção	
Descrição	Seguindo a regra de negócio 02, a manutenção deve ocorrer quando o ônibus atingir a quilometragem de 10.000km. Então quando esta meta de quilometragem for atingida, o sistema irá notificar o usuário para que ele agende a manutenção imediata!
Atores	Gerente da frota
Prioridade	Alta
Requisitos associados	RF06
Fluxo Principal	<ol style="list-style-type: none"> 1. Sistema constantemente realiza leitura das quilometragens. 2. Sistema notifica o usuário caso algum ônibus tenha atingido 10.000km de quilometragem. 3. Usuário seleciona opção de expandir notificação. 4. Sistema exibe detalhes dos ônibus que requerem manutenção. 5. Usuário seleciona um ônibus. 6. Sistema redireciona usuário para página de marcar manutenção com os dados do ônibus escolhido carregados.
Fluxos Alternativos (variantes)	Não se aplica

UC03 – Notificação de necessidade de Manutenção	
Descrição	Seguindo a regra de negócio 02, a manutenção deve ocorrer quando o ônibus atingir a quilometragem de 10.000km. Então quando esta meta de quilometragem for atingida, o sistema irá notificar o usuário para que ele agende a manutenção imediata!
Atores	Gerente da frota

Prioridade	Alta
Requisitos associados	RF06
Fluxo Principal	<ol style="list-style-type: none"> 1. Sistema constantemente realiza leitura das quilometragens. 2. Sistema notifica o usuário caso algum ônibus tenha atingido 10.000km de quilometragem. 3. Sistema redireciona usuário para página de marcar manutenção com os dados do ônibus escolhido carregados.
Fluxos Alternativos (variantes)	Não se aplica

Plano de Desenvolvimento de Software

ManutenBus

Versão 1.0

Histórico da Revisão

Data	Versão	Descrição	Autor
01/03/2020	1.0	Criação do documento. Preenchimento dos itens de introdução, finalidade, escopo, referências, visão geral, visão geral do projeto, organização do projeto, e alguns itens do processo de gerenciamento	Irene, Pedro, Bernardo, Carlos

1. Introdução

Este documento apresenta o plano de desenvolvimento de *software* para manutenção da frota de ônibus da empresa Paranaíba Transportes LTDA.

1.1 Finalidade

Este documento tem por finalidade definir aspectos em comuns em planos de desenvolvimento de *software* para organizar o desenvolvimento do software a fim de que todos os envolvidos no projeto tenham conhecimento de como se trabalhar.

1.2 Definições, Acrônimos e Abreviações

CPF: Cadastro de Pessoa Física.
JS/js: JavaScript

1.3 Referências

Manuais de manutenção dos ônibus

1.4 Visão Geral

Este documento está dividido em três itens importantes para um desenvolvimento de *software*:

- Visão geral do projeto;
- Organização do projeto;

- Processo de gerenciamento

2. Visão Geral do Projeto

Esta seção descreve uma visão geral do projeto, tais como escopo, restrições e entregáveis.

2.1 Finalidade, Escopo do produto e Objetivos do Projeto

O projeto tem como objetivo desenvolver um sistema que permite que o cliente possa, acompanhar o processo de manutenção de uma forma dinâmica e em tempo real, realizar um planejamento a partir das informações disponíveis, ser possível organizar e criar metas. Poderá acompanhar a manutenção de sua frota e categorizar quais veículos tem o melhor desempenho.

2.2 Suposições e Restrições

O desenvolvimento do projeto possui restrição de prazo que tem uma data de entrega determinada: 09 de junho de 2020.

2.3 Produtos Liberados do Projeto

O projeto possui como entregáveis os seguintes artefatos:

- Documento de Visão;
- Glossário;
- Modelo de Domínio;
- Diagrama de Classes;
- Diagrama de Casos de Uso;
- Especificação de Casos de Uso;
- Código Fonte
- Software entregue

3. Organização do Projeto

Esta seção descreve como desenvolvimento do projeto será organizado a partir dos recursos disponíveis.

3.1 Estrutura Organizacional

A equipe de trabalho do projeto foi planejada de maneira com que todos da equipe trabalhem de maneira igualitária, ou seja, todos poderão em algum

momento exercer o papel do outro, menos o testador de acordo com a tabela abaixo:

3.2 Papéis e Responsabilidades

<u>Pessoa</u>	<u>Papel</u>
<u>Bernardo</u>	<u>Desenvolvedor.</u>
<u>Carlos</u>	<u>Desenvolvedor.</u>
<u>Pedro</u>	<u>Gerente de Projeto, Desenvolvedor.</u>
<u>João</u>	<u>Tester/QA</u>

4. Processo de Gerenciamento

Nesta seção estão descritos alguns tópicos acerca do processo de gerenciamento do projeto, tais como estimativas de prazo, planos de fase, planejamento de *Releases* e recursos do projeto.

4.1 Estimativas do Projeto

Prazo|96 dias -----|-----|---- Data inicial:|01/03 Data final:| 09/06

4.2 Plano do Projeto

O projeto está dividido em sete iterações. Cada iteração passa pelas quatro fases do processo de desenvolvimento de *software* utilizado (*UP - Agile Unified Process*): Concepção, Elaboração, Construção e Transição.

4.2.1 Plano de Fase

Iteração	Objetivos	Data de início	Data de Fim
01	Termo de Abertura do projeto	01/03	14/03
02	Especificação do projeto	15/03	28/03
03	Refinar Artefatos da fase de Concepção, elaborar artefatos da fase de elaboração e implementar	29/03	11/04

	incremento de software e desenvolvimento de testes.		
04	Refinar Artefatos da fase de Concepção, elaborar artefatos da fase de elaboração e implementar incremento de software e desenvolvimento de testes.	12/04	25/04
05	Refinar Artefatos da fase de Concepção, elaborar artefatos da fase de elaboração e implementar incremento de software e desenvolvimento de testes.	26/04	09/05
06	Refinar Artefatos da fase de Concepção, elaborar artefatos da fase de elaboração e implementar incremento de software e desenvolvimento de testes.	10/05	23/05
07	Refinar Artefatos da fase de Concepção, elaborar artefatos da fase de elaboração e implementar incremento de software e desenvolvimento de testes.	24/05	06/06

4.2.2 Releases

O projeto foi planejado em duas *Releases*, sendo a primeira trazendo a funcionalidade principal do planejamento do fluxo e a segunda com as funcionalidades de planejar grade horária e estimar IRA.

Release	Características	Data
01	TAP	04/03
02	Especificações.	24/03
03	Camada de apresentação	14/04
04	Protótipo	12/05
05	Entrega Final	09/06

4.2.3 Programação do Projeto

Marcos com pontos de controle do projeto por parte da interessada Ada Love e Laços:

Descrição	Data
Entrega do TAP	04/03
Entrega Especificações	24/03
Entrega Camada de apresentação	14/04
Entrega Protótipo	12/05
Entrega Final	09/06

4.2.4 Recursos do Projeto

O Recurso Humano do projeto possui uma equipe com quatro membros:

- Bernardo Victor
- Carlos Alberto
- João Rene
- Pedro Diógenes

4.2.5 Riscos

Classificação de probabilidade (Alta, Média, Baixa)	Classificação de impacto (Alto, Médio, Baixo)	Descrição do Risco	Estratégia de Diminuição e/ou Plano de Contingência	Ação a ser tomada
Baixa	Alto	Membro da equipe sair do projeto	Integração e motivação da equipe	Redesignar atividades do membro desistente aos outros integrantes da equipe

Média	Alto	Atraso no desenvolvimento das atividades	Deixar um <i>buffer</i> de tempo em cada iteração para caso ocorra atrasos	Realocar atividades considerando o <i>buffer</i> de tempo no cronograma
Baixa	Médio	Mudança no Escopo	Deixar um <i>buffer</i> de tempo em cada iteração para caso ocorra mudanças	
Baixa	Alto	Suspensão do calendário acadêmico		Avaliar as mudanças ocorridas para realocar as atividades dessas mudanças nas próximas iterações.

4.2.6 Ambiente:

Versão do NodeJs: 12.16.1

Versão do JS: 1.8.5 ECMA Script 5

Versão do Sequelize: v5

4.2.7 Iterações

1ª Iteração

###Data de início: 01/03/2020 ###Data de fim: 14/03/2020

Artefatos Produzidos

Plano de Iteração

Termo de Abertura do Projeto

Fim da iteração

2ª Iteração

###Data de início: 15/03/2020 ###Data de fim: 28/03/2020

Artefatos Produzidos

Plano de Iteração

Termo de Abertura do Projeto

Diagrama de Classes

Diagrama de Casos de Uso

Modelos de banco de dados

Fim da iteração

3ª Iteração

###Data de início: 29/03/2020 ###Data de fim: 11/04/2020

Artefatos Produzidos

Plano de Iteração

Termo de Abertura do Projeto

Diagrama de Classes

Diagrama de Casos de Uso

Modelos de banco de dado

Criação casos de testes

4ª Iteração

###Data de início: 12/04/2020 ###Data de fim: 25/04/2020

Artefatos Produzidos

Plano de Iteração

Termo de Abertura do Projeto

Diagrama de Classes

Diagrama de Casos de Uso

Modelos de banco de dados

Criação casos de testes

5ª Iteração

###Data de início: 26/04/2020 ###Data de fim: 09/05/2020

Artefatos Produzidos

Plano de Iteração

Termo de Abertura do Projeto

Diagrama de Classes

Diagrama de Casos de Uso

Modelos de banco de dados

Criação casos de testes

Protótipo

6ª Iteração

###Data de início: 10/05/2020 ###Data de fim: 23/05/2020

Artefatos Produzidos

Plano de Iteração

Termo de Abertura do Projeto

Diagrama de Classes

Diagrama de Casos de Uso

Modelos de banco de dados

Criação casos de testes

Protótipo

Entrega Final

Entrega Final

Artefatos Produzidos

Plano de Iteração

Termo de Abertura do Projeto

Diagrama de Classes

Diagrama de Casos de Uso

Modelos de banco de dados

Criação casos de testes

Software

Manutenbus

Documento de Arquitetura de Software

Versão 3

Histórico da Revisão

Histórico de Revisões			
Data	Autor	Descrição	Versão
[27/04/2020]	[Pedro Diógenes]	Versão inicial	[1]
[02/05/2020]	[Pedro Diógenes, Carlos Alberto, Bernardo Victor, João Rene]	Atualizado até 3.3.1	[2.1]
[03/05/2020]	[Pedro Diógenes, Carlos Alberto, Bernardo Victor, João Rene]	Atualizado por inteiro	[2.2]
[08/06/2020]	[Pedro Diógenes, Carlos Alberto, Bernardo Victor, João Rene]	Versão final, atualizado diagrama de componentes	3

1 Apresentação

O projeto tem como objetivo desenvolver um sistema que permite que o cliente possa, acompanhar o processo de manutenção de uma forma dinâmica e em tempo real, realizar um planejamento a partir das informações disponíveis, ser possível organizar e criar metas. Poderá acompanhar a manutenção de sua frota e categorizar quais veículos tem o melhor desempenho

1.1 Problema

As empresas que gerenciam a frota de ônibus na nossa cidade tem problemas com a gestão da manutenção dos veículos da sua frota. O projeto tem como objetivo remediar isso dando autonomia aos gestores da empresa para que exerçam maior controle sobre alguns pontos específicos das manutenções requeridas.

1.2 Objetivos do trabalho

O objetivo deste trabalho é apresentar a descrição arquitetural do sistema, onde ficará claro quais são as restrições arquiteturais do sistema, o modelo tanto físico quanto lógico da arquitetura do projeto, e ainda serão determinadas as especificações dos casos de uso do projeto.

1.3 Definições e Abreviações

Exemplo:

MVC – Model View Controller

JS - JavaScript

2 Requisitos da Arquitetura

2.1 Restrições Arquiteturais

As restrições impostas ao projeto que afetam sua arquitetura são:

- O software deverá ser desenvolvido em JavaScript

2.2 Mecanismos Arquiteturais

Análise	Design	Implementação
Persistência	<i>Banco de dados relacional com implementação a partir do backend</i>	MySQL com Sequelize
Front end	<i>Framework para desenvolvimento front-end em JS</i>	REACTJS
Back end	<i>Camada responsável por implementar as regras de negócio do sistema</i>	node.js
Integração	<i>Não utilizado</i>	
Log do sistema	<i>Implementação dos recursos de log do componente de persistência.</i>	node.js logging
Teste de Software	<i>Teste unitário</i>	Jasmine
Deploy	<i>Não utilizado</i>	
Fluxo de trabalho	<i>Suporte à circulação de documentos e outros itens de trabalho.</i>	LucidChart, Trello

3 Modelagem e projeto arquitetural

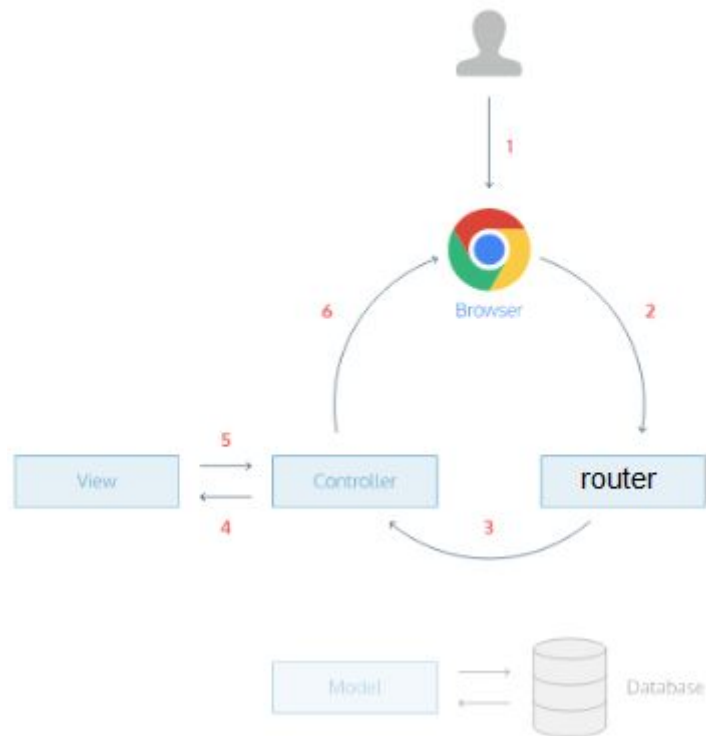


Figura 1 - Visão Geral da Solução

3.1.1 Visão Lógica

3.1.2 Modelo de Classes

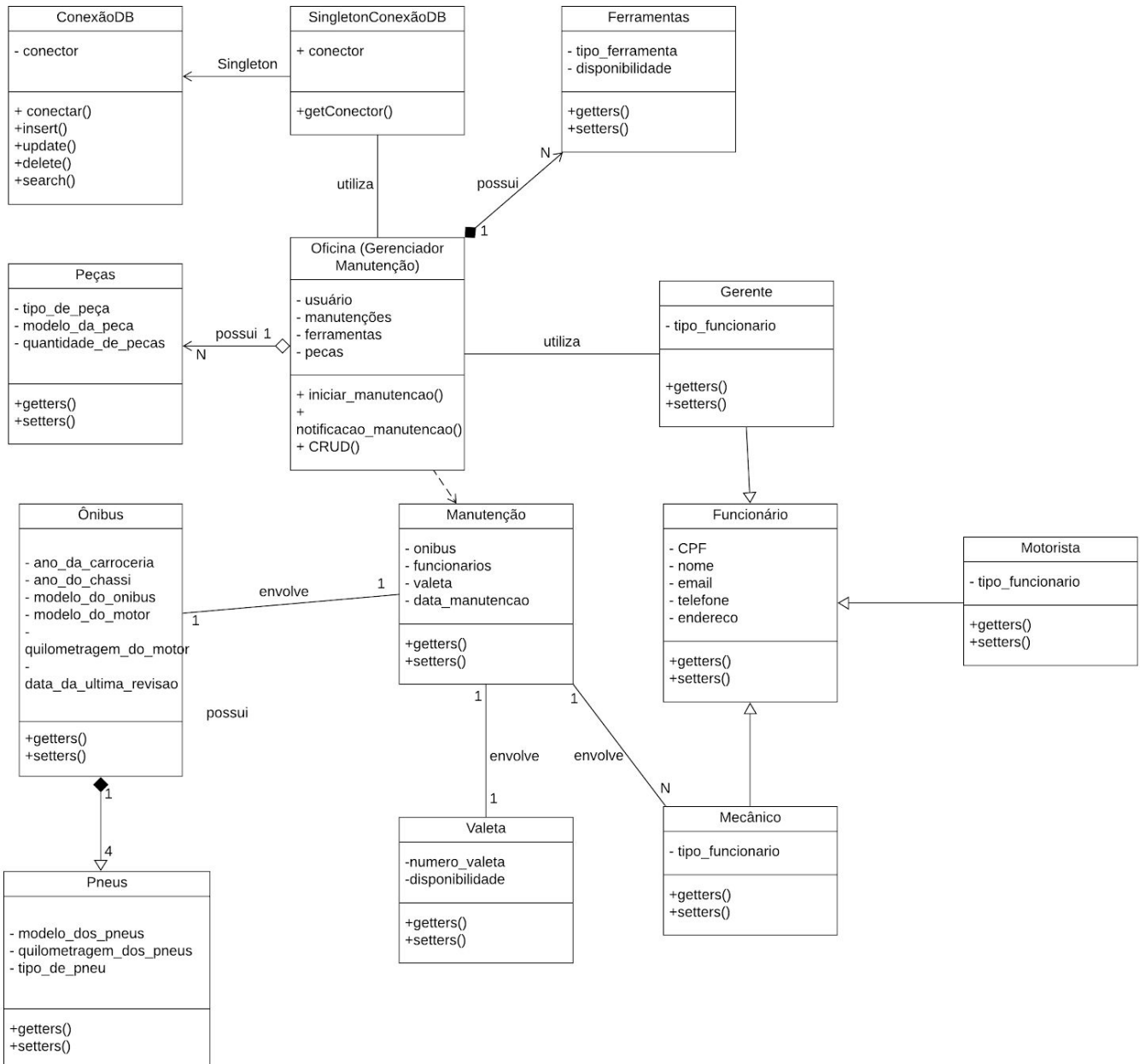


Figura 2 – Diagrama de classes

3.1.3 Modelo de componentes

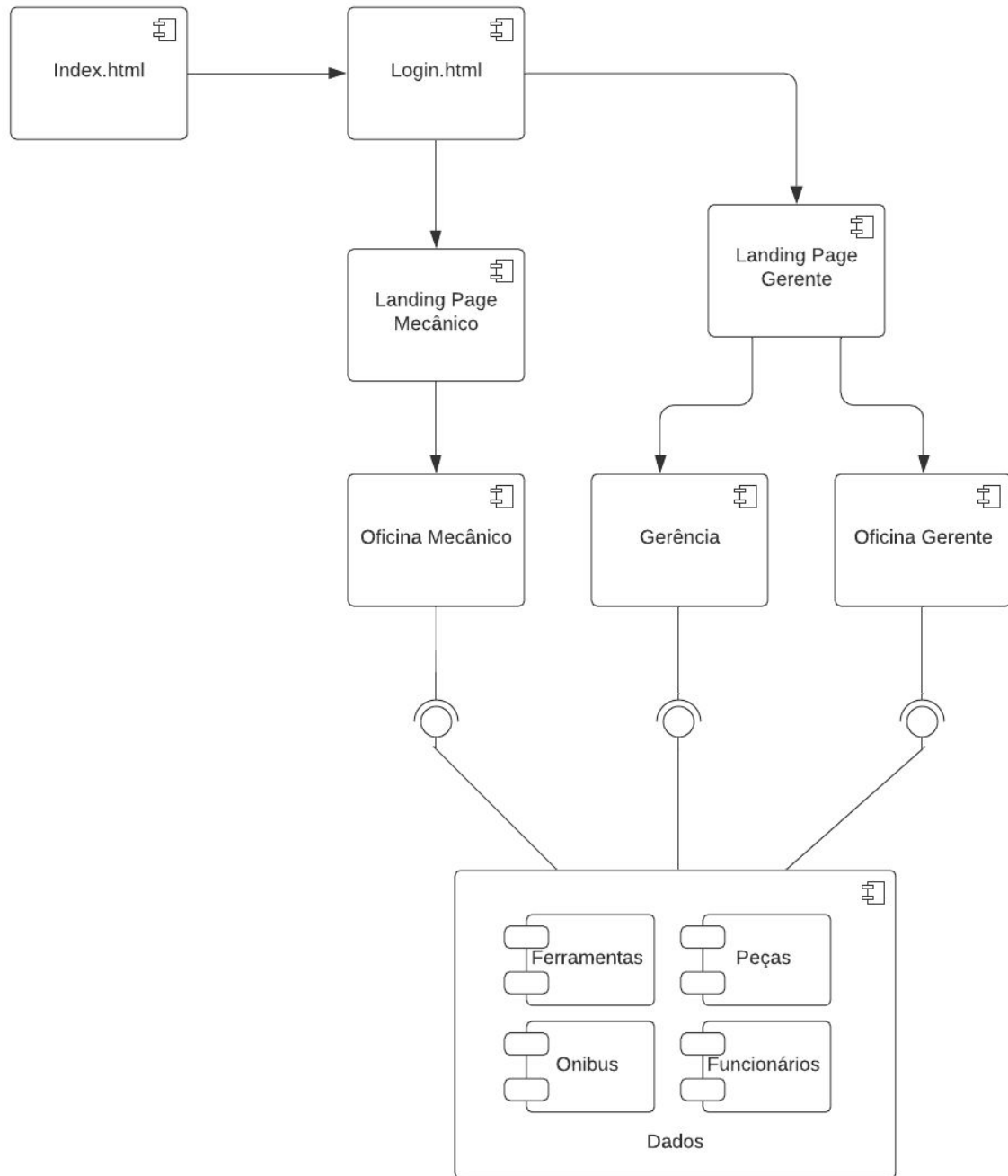


Figura 3 – Diagrama de Componentes

3.2 Visão Física

3.2.1 Modelo de implantação – Diagrama de Nodos

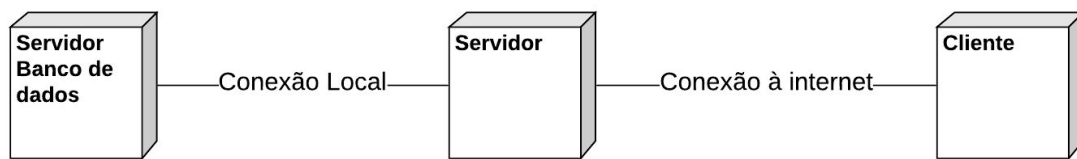
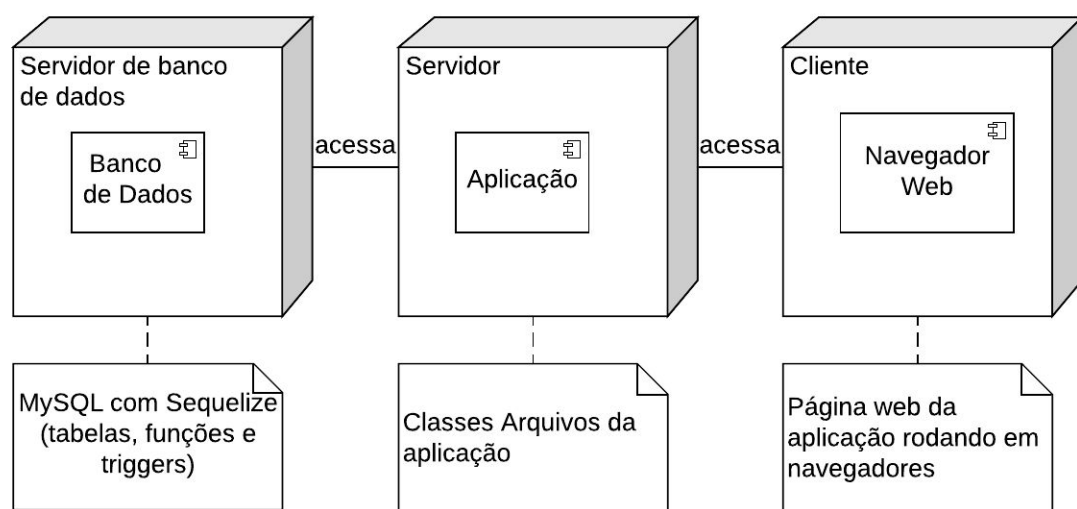


Figura 4 – Diagrama de Implantação

Conforme diagrama apresentado na Figura 4, as entidades participantes da solução são:

- **Servidor**- Servidor conectado localmente à um BD que trata as requisições do cliente
- **Cliente** - Páginas web que fazem requisições no servidor que podem alterar os dados no BD

3.2.2 Modelo de implantação – Diagrama de Componentes



3.2.3 Modelo de dados

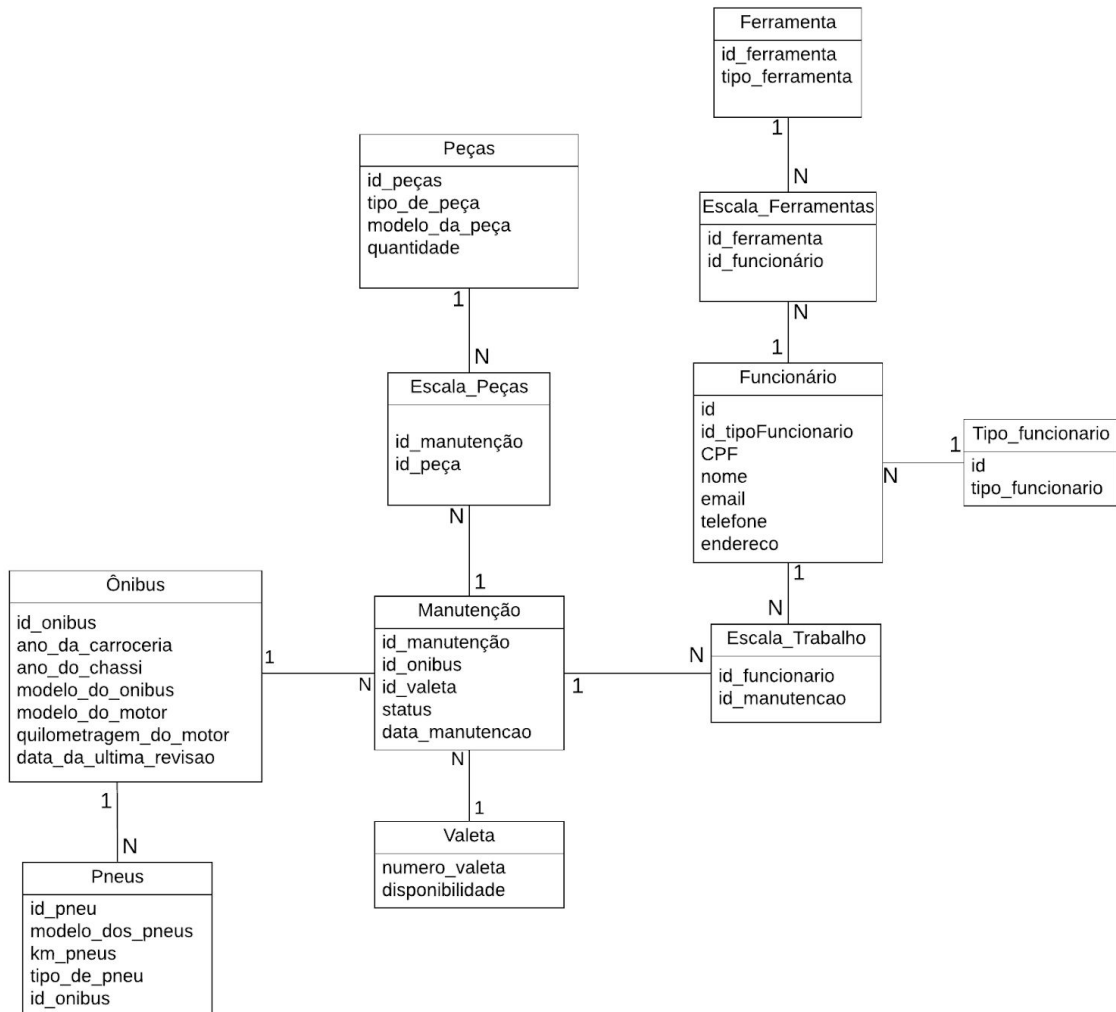


Figura 3 – Diagrama de Entidade Relacionamento (ER)

4 APÊNDICES

<https://github.com/vradoskein/Lab-Projeto-de-Sistemas>

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Graduação em Ciência da Computação

PLANEJAMENTO DE TESTES

MANUTENBUS

Participantes

Bernardo Victor Souza

Carlos Alberto Nassif

João Rene

Pedro Diógenes

Definição do Escopo

As funcionalidades do projeto cobertas por esse planejamento serão todas as funcionalidades principais do projeto (os requisitos funcionais) bem como um de seus requisitos não funcionais.

Requisitos e Planejamento de Casos de Teste

Os seguintes requisitos estão englobados neste planejamento de testes:

- **RF001** - CRUD Ônibus
- **RF002** - CRUD Pneus
- **RF003** - CRUD Peças
- **RF004** - CRUD Funcionário
- **RF005** - CRUD Valeta
- **RF006** - CRUD Ferramentas
- **RF007** - Iniciar Manutenção
- **RF008** - Notificação de necessidade de manutenção
- **RNF03** - Tempo de resposta - O sistema deve ter um tempo de resposta menor do que 10 segundos para todas suas funções

Os cenários cobertos por estes testes serão os seguintes:

1. <CRUD>:

- a. Abrir a tela de cadastro do item (nesses casos de teste, por “item”, entende-se que é o modelo a qual o CRUD se trata)
- b. Informar os dados do item (que são todos obrigatórios)
- c. Validar os dados do item
- d. Clicar em cadastrar

2. Iniciar manutenção:

- a. Abrir a tela de iniciar manutenção
- b. Escolher entre um dos ônibus que necessitem de manutenção
- c. Escolher entre um dos funcionários disponíveis
- d. Escolher entre uma das valetas disponíveis
- e. Validar as escolhas
- f. Clicar em iniciar manutenção

3. Notificação de necessidade de manutenção:

- a. Ao iniciar-se o programa, o sistema verifica a existência de ônibus que necessitem de manutenção
- b. Receber na tela os dados do ônibus que necessita de manutenção
- c. Confirmar notificação

4. Login*:

- a. O funcionário digita seus dados cadastrais
- b. O sistema valida os dados
- c. O sistema redireciona o funcionário à pagina adequada (o gerente tem acesso completo, enquanto os demais funcionários apenas tem acesso às notificações)

Os casos de teste serão planejados da seguinte maneira:

• **Testes feitos por tabela de decisão:**

	CT001	CT002
Condições		
Dados válidos de acordo com a especificação?	Sim	Não
Ações		
Validar dados	Dados Válidos	Dados Inválidos

	CT003	CT004
Condições		
Dados obrigatórios preenchidos?	Sim	Não
Ações		
Habilitar cadastro	Habilita	Não habilita

	CT005	CT006
Condições		
Dados do funcionário corretos?	Sim	Não
Ações		
Fazer login	Sim	Não

- **Testes feitos por valor limite:**

Motor			
Entrada	Valores Permitidos	Classes	Casos de Teste
Quilometragem do motor do ônibus selecionado	> 10000	> 10000 (CT007)	km = 20001
		de 0 a 10000 negativo nulo inválido (CT008)	km = 0 km = 1 km = 19999 km = 20000 km = -1 km = null km = "a"

Pneus			
Entrada	Valores Permitidos	Classes	Casos de Teste
Quilometragem dos pneus do ônibus selecionado	> 10000	> 10000 (CT009)	km = 20001
		de 0 a 10000 negativo nulo inválido (CT010)	km = 0 km = 1 km = 19999 km = 20000 km = -1 km = null km = "a"

Tempo			
Entrada	Valores Permitidos	Classes	Casos de Teste
Tempo de execução de uma função	<= 10	<= 10 (CT011)	t = 9 t = 10
		> 10 negativo nulo (CT012)	t = 11 t = -1 t = null

A partir desses cenários de testes, teremos os seguintes casos de teste:

- **CT001** - Dados dos CRUDs válidos (1.c)
 - **CT002** - Dados dos CRUDs inválidos (1.c)
 - **CT003** - Dados obrigatórios preenchidos (1.b)
 - **CT004** - Dados obrigatórios não preenchidos (1.b)
 - **CT005** - Dados do funcionário válidos (4)
 - **CT006** - Dados do funcionário inválidos (4)
 - **CT007** - Ônibus dentro das especificações (checagem por motor) (2.b, 2.e, 3.a, 3.b)
 - **CT008** - Ônibus fora das especificações (checagem por motor) (2.b, 2.e, 3.a, 3.b)
 - **CT009** - Ônibus dentro das especificações (checagem por pneus) (2.b, 2.e, 3.a, 3.b)
 - **CT010** - Ônibus fora das especificações (checagem por pneus) (2.b, 2.e, 3.a, 3.b)
 - **CT011** - Tempo de execução dentro do limite (Engloba todos)
 - **CT012** - Tempo de execução fora do limite (engloba todos)
-

Casos de teste

Caso de Teste:	CT001 - Dados dos CRUDs válidos.
Pré condições:	O gerente estar na tela do CRUD em questão, e os dados obrigatório terem sido preenchidos.
Procedimento:	<ol style="list-style-type: none">1. O gerente inicia a inserção do item2. O gerente digita os dados requeridos do item3. O gerente tenta fazer a inserção do item4. O sistema confirma que os itens obrigatórios estão preenchidos5. O sistema verifica se os dados do sistema são válidos6. O sistema faz a inserção dos dados
Resultado esperado:	O item é inserido no banco de dados do sistema
Dados de entrada:	Dados válidos para o CRUD em questão

Caso de Teste:	CT002 - Dados dos CRUDs inválidos.
Pré condições:	O gerente estar na tela do CRUD em questão, e os dados obrigatório terem sido preenchidos.
Procedimento:	<ol style="list-style-type: none">1. O gerente inicia a inserção do item2. O gerente digita os dados requeridos do item3. O gerente tenta fazer a inserção do item4. O sistema confirma que os itens obrigatórios estão preenchidos5. O sistema verifica se os dados do sistema são válidos6. O sistema exibe a mensagem "Dados inválidos"
Resultado esperado:	Mensagem de erro do sistema
Dados de entrada:	Dados inválidos para o CRUD em questão

Caso de Teste:	CT003 - Dados obrigatórios preenchidos.
Pré condições:	O gerente estar na tela do CRUD em questão
Procedimento:	<ol style="list-style-type: none"> 1. O gerente inicia a inserção do item 2. O gerente digita os dados requeridos do item 3. O gerente tenta fazer a inserção do item 4. O sistema confirma que os itens obrigatórios estão preenchidos
Resultado esperado:	O sistema procede a verificar a validade dos dados em questão
Dados de entrada:	Todos os dados obrigatórios para o CRUD em questão

Caso de Teste:	CT004 - Dados obrigatórios não preenchidos.
Pré condições:	O gerente estar na tela do CRUD em questão
Procedimento:	<ol style="list-style-type: none"> 1. O gerente inicia a inserção do item 2. O gerente digita os dados requeridos do item 3. O gerente tenta fazer a inserção do item 4. O sistema exibe uma mensagem de erro "Preencha todos os dados obrigatórios."
Resultado esperado:	Mensagem de erro do sistema
Dados de entrada:	Todos os dados obrigatórios para o CRUD em questão

Caso de Teste:	CT005 - Dados do funcionário válidos
Pré condições:	O funcionário estar na tela de Login
Procedimento:	<ol style="list-style-type: none"> 1. O funcionário preenche seus dados cadastrais na tela de login 2. O funcionário seleciona a opção Login 3. O sistema verifica se os campos obrigatórios foram preenchidos 4. O sistema verifica se o e-mail e a senha do funcionário são válidos 5. O sistema redireciona o funcionário à página referente ao seu cargo.
Resultado esperado:	Funcionário logado e redirecionado à sua página de cargo.
Dados de entrada:	Dados do login do funcionário válidos

Caso de Teste:	CT006 - Dados do funcionário inválidos
Pré condições:	O funcionário estar na tela de Login
Procedimento:	<ol style="list-style-type: none"> 1. O funcionário preenche seus dados cadastrais na tela de login 2. O funcionário seleciona a opção Login 3. O sistema verifica se os campos obrigatórios foram preenchidos 4. O sistema verifica se o e-mail e a senha do funcionário são válidos 5. O sistema exibe a mensagem "E-mail ou senha incorretos"
Resultado esperado:	Mensagem de erro do sistema
Dados de entrada:	Um login inválido

Caso de Teste:	CT007 - Ônibus dentro das especificações (checagem por motor)
Pré condições:	O gerente estar na tela de início de manutenção OU o gerente estar na tela principal
Procedimento:	<ol style="list-style-type: none"> 1. O gerente escolhe um ônibus para iniciar a manutenção <ol style="list-style-type: none"> a. Ou o gerente entra no sistema, e há uma notificação 2. O sistema retorna os dados do ônibus 3. O gerente confirma a escolha do ônibus 4. O gerente seleciona os demais dados requeridos. 5. O gerente seleciona a opção "Iniciar Manutenção" 6. O sistema verifica os dados dos itens selecionados, especialmente a km do ônibus selecionado. 7. O sistema inicia a manutenção
Resultado esperado:	Manutenção marcada para início pelo sistema
Dados de entrada:	Um ônibus válido

Caso de Teste:	CT008 - Ônibus fora das especificações (checagem por motor)
Pré condições:	O gerente estar na tela de início de manutenção OU o gerente estar na tela principal
Procedimento:	<ol style="list-style-type: none"> 1. O gerente escolhe um ônibus para iniciar a manutenção <ol style="list-style-type: none"> a. Ou o gerente entra no sistema, e há uma notificação 2. O sistema retorna os dados do ônibus 3. O gerente confirma a escolha do ônibus 4. O gerente seleciona os demais dados requeridos. 5. O gerente seleciona a opção "Iniciar Manutenção" 6. O sistema verifica os dados dos itens selecionados, especialmente a km do ônibus selecionado. 7. O sistema exibe a mensagem "O ônibus não necessita manutenção"
Resultado esperado:	Mensagem de erro do sistema
Dados de entrada:	Um ônibus inválido

Caso de Teste:	CT009 - Ônibus dentro das especificações (checagem por pneus)
Pré condições:	O gerente estar na tela de início de manutenção OU o gerente estar na tela principal
Procedimento:	<ol style="list-style-type: none"> 1. O gerente escolhe um ônibus para iniciar a manutenção <ol style="list-style-type: none"> a. Ou o gerente entra no sistema, e há uma notificação 2. O sistema retorna os dados do ônibus 3. O gerente confirma a escolha do ônibus 4. O gerente seleciona os demais dados requeridos. 5. O gerente seleciona a opção "Iniciar Manutenção" 6. O sistema verifica os dados dos itens selecionados, especialmente a km do ônibus selecionado. 7. O sistema inicia a manutenção
Resultado esperado:	Manutenção marcada para início pelo sistema
Dados de entrada:	Um ônibus válido

Caso de Teste:	CT010 - Ônibus fora das especificações (checagem por pneus)
Pré condições:	O gerente estar na tela de início de manutenção OU o gerente estar na tela principal
Procedimento:	<ol style="list-style-type: none"> 1. O gerente escolhe um ônibus para iniciar a manutenção <ol style="list-style-type: none"> a. Ou o gerente entra no sistema, e há uma notificação 2. O sistema retorna os dados do ônibus 3. O gerente confirma a escolha do ônibus 4. O gerente seleciona os demais dados requeridos. 5. O gerente seleciona a opção "Iniciar Manutenção" 6. O sistema verifica os dados dos itens selecionados, especialmente a km do ônibus selecionado. 7. O sistema exibe a mensagem "O ônibus não necessita manutenção"
Resultado esperado:	Manutenção marcada para início pelo sistema
Dados de entrada:	Um ônibus inválido

Caso de Teste:	CT011 - Tempo de execução dentro do limite
Pré condições:	Qualquer ator executar qualquer função no sistema
Procedimento:	<ol style="list-style-type: none"> 1. O ator executa qualquer função no sistema 2. O sistema mede o tempo desde a requisição até a resposta da função realizada. 3. O sistema redireciona o ator até a próxima página.
Resultado esperado:	Função executada normalmente.
Dados de entrada:	Uma função qualquer do sistema que rode normalmente dentro do tempo limite

Caso de Teste:	CT012 - Tempo de execução fora do limite
Pré condições:	Qualquer ator executar qualquer função no sistema
Procedimento:	<ol style="list-style-type: none"> 1. O ator executa qualquer função no sistema 2. O sistema mede o tempo desde a requisição até a resposta da função realizada. 3. O sistema aborta a função sendo executada 4. O sistema exibe a mensagem de erro "Timeout, tente novamente"
Resultado esperado:	Uma função qualquer do sistema que intencionalmente rode fora do tempo limite
Dados de entrada:	Tempo de resposta do sistema fora do limite

Análise de Usabilidade

Bernardo Victor de Souza e Silva

Carlos Alberto Nassif Costa Pinto

Joao Rene Teixeira Filho

Pedro Diógenes de Almeida Lara

1. Método utilizado para o teste de usabilidade

Para a verificação do nível de usabilidade dos sites analisados neste trabalho, foram utilizados os 10 princípios fundamentais de usabilidade desenvolvidos por Nielsen(2012).

Os princípios adotados por Nielsen(2012) são: Visibilidade do status do sistema, Compatibilidade entre o sistema com o mundo real, Controle do usuário e liberdade, Consistência e padrões, Ajuda os usuários a reconhecer, diagnosticar e recuperar-se de erros, Prevenção de erros, Reconhecimento em vez de memorização, Flexibilidade e eficiência de uso, Estética e design minimalista e Ajuda e documentação.

2. Análise Heurística

A análise heurística foi realizada pelos integrantes da equipe considerando os 10 princípios de Nielsen referentes à usabilidade.

Esta avaliação teve como objetivo verificar a usabilidade do projeto Manutenbus.

As notas foram atribuídas conforme a escala abaixo (C. BLASCO):

Nível 0: Não é encarado necessariamente como um problema de usabilidade.

Nível 1: Problema estético que não tem necessidade de ser corrigido, a menos que haja tempo e recurso disponível.

Nível 2: Pequeno problema com baixa prioridade na correção.

Nível 3: Problema com alta prioridade de correção.

Nível 4: Catástrofe de usabilidade, ou seja, o produto só será liberado se a correção for feita.

Os resultados desta análise foram apresentados na tabela abaixo.

Princípios	Média Aritmética	Consenso	Diferença entre Média e Consenso	Considerações, caso necessário

1. Visibilidade do status do sistema	2.25	2	0.25	Os passos a serem feitos em um processo deveriam ser especificados e mostrados para o usuário.
2. Compatibilidade do sistema com o mundo real	0.25	0	0.25	-
3. Controle do usuário e liberdade	0	0	0	-
4. Consistência e padrões	0.25	0	0.25	-
5. Ajuda os usuários a reconhecerem, diagnosticarem e recuperarem-se de erros	3.75	4	0.25	Em campos de inserção ou seleção, como login, registro, etc. adicionar recursos visuais para chamar atenção caso o usuário tente prosseguir sem realizá-las.
6. Prevenção de erros	4	4	0	Erro do usuário não solucionável no momento. Fazer botões de desfazer, confirmação em etapas definitivas do processo, como “iniciar manutenção” ou “cancelar manutenção”
7. Reconhecimento em vez de memorização	0	0	0	-
8. Flexibilidade e eficiência	2.75	3	0.25	Como a execução do

de uso				programa é por sua maior parte feita por seleção de valores pré-definidos, o usuário experiente não teria uma vantagem ao usuário comum.
9. Estética e design minimalista	0	0	0	-
10. Ajuda e documentação	4	4	0	Por ser um programa de interface simples e intuitiva, um guia de página não foi estruturado.

3. Referências

C. BLASCO, E. **ABC da Usabilidade: Análise Heurística**. Disponível em:
 <<http://www.linhadecodigo.com.br/artigo/2355/abc-da-usabilidade-analise-heuristica.aspx>> . Acesso em: 27 maio. 2020.

NIELSEN, J. **10 Heuristics for User Interface Design**. Disponível em:
 <<https://www.nngroup.com/articles/ten-usability-heuristics/>> . Acesso em: 27 abr. 2012.