**ETEC ABDIAS DO NASCIMENTO**

**Técnico em Informática**

**JONATHAN FRANCISCO DA SILVA GOMES**

**MAYARA CAROLINE DA SILVA**

**MICHELE DOS SANTOS SILVA**

**NICOLAS AGUIAR DE SOUZA**

**RENILSON SANTANA SANTOS**

**TCC: HONIBUS (SISTEMA DE CONTROLE DE ÔNIBUS)**

**SÃO PAULO**

**2018**

**JONATHAN FRANCISCO DA SILVA GOMES**

**MAYARA CAROLINE DA SILVA**

**MICHELE DOS SANTOS SILVA**

**NICOLAS AGUIAR DE SOUZA**

**RENILSON SANTANA SANTOS**

**TCC: HONIBUS (SISTEMA DE CONTROLE DE ÔNIBUS)**

**SÃO PAULO**

**2018**

# Dedicatórias

**Jonathan Francisco da Silva Gomes:**

Esta dedicatória eu agradeço a todas as pessoas que me apoiaram fazendo que eu não desistisse do curso e principalmente aos professores e amigos, que a todo o momento teve paciência e sempre estando junto comigo fazendo que eu chegasse até a conclusão deste curso.

**Mayara Caroline da Silva:**

Começo essa dedicatória agradecendo a todos aqueles que dedicaram o seu tempo para nos ajudar a chegar até aqui, principalmente aos professores Hebert e Ricardo, pela paciência, pelos conselhos e por partilhar seus conhecimentos para que nós pudéssemos concluir com êxito esse curso.

**Michele dos Santos Silva:**

Primeiramente agradeço a Deus por me dar saúde, a minha mãe por esta sempre do meu lado me apoiando. Agradeço aos meus colegas pelo bom trabalho em equipe. E assim concretizo um dos meus objetivos concluindo o curso.

**Nicolas Aguiar de Souza:**

Dedico esse trabalho primeiramente a Deus e minha família, por ser minha base. Agradeço aos professores pela paciência, por nos disciplinar e partilhar seus conhecimentos, em especial aos professores Hebert e Ricardo pela persistência e sempre acreditar em nosso potencial, durante os semestres, para conclusão do curso.

**Renilson Santana Santos:**

Eu não dedico esse trabalho a ninguém em especial. Mas quero agradecer a Deus e a todas as pessoas que me apoiam e tem grandes expectativas na minha pessoa. Meu objetivo com o trabalho em si é concluir o curso.

# Resumo

A ideia surgiu enquanto estávamos em um ponto na espera de um ônibus que estava atrasado, com isso, observamos como seria útil saber que horário ele teria saído do terminal, conversando em grupo percebemos que todos os dados referentes ao fluxo dos ônibus eram armazenados em papéis. Um dia, chegando à escola idealizamos em forma de projeto como poderia ser útil ter um sistema para suprir a necessidade de diminuição de gastos de papéis, com isso, facilitando também o monitoramento em relação à grade de horas, rápido acesso aos dados e evitaria perca de informações, por conta de desgaste do papel. Começamos por uma pesquisa para poder ter uma noção do que e como eram armazenadas as informações, para que assim pudéssemos chegar o mais perto possível de um resultado satisfatório. Através de um questionário, estipulado por nós e com nossas dúvidas, um funcionário de um terminal nos ajudou a respondê-las, também nos deu um relatório que os fiscais preenchem para sabermos o que deveria conter em nosso sistema. Então, iniciamos um planejamento, descrevendo as necessidades que precisaríamos para suprir de forma que fosse possível atender a todas, com isso, se iniciou a execução do projeto.

# Abstract

Our system will be simple and easy to use, having the administrator Login, after that will have a menu for queries, transport register and control of arrival and departure.

At the moment the administrator initiates the query, it will present a field asking for the numbering of the respective transport, in this, will be shown all the information about it.

As soon as you start the registration screen, you will be asked for the name of the driver, numbering of the bus and the times that it will be in service. And once you are registered you will already be in the database, and can be edited, if you change your driver or your schedules.

When you start the flow control screen, it will display a screen with all the transports and their schedules of the respective day, and you can also choose the day that you want to consult.

# Lista de ilustrações

Figura 1 – Cronograma 2

Figura 2 – Fluxograma Processo de Consulta 3

Figura 3 – Fluxograma de Processo de Cadastro de Ônibus 4

Figura 4 – Fluxograma de Processo de Cadastro de Motorista 5

Figura 5 – Fluxograma de Consultar Fluxo do Dia 7

Figura 6 – Fluxograma de Processo de Cadastro de Ocorrência 10

Figura 7 – Visual Studio 11

Figura 8 – C# 12

Figura 9 – Adobe Photoshop 13

Figura 10 – Microsoft SQL Server 13

Figura 11 – Word 15

Figura 12 - EAP 17

Figura 13 – Diagrama de Caso de Uso 20

Figura 14 – Diagrama de Entidade e Relacionamento 21

Figura 15 – Diagrama de Classes 21

Figura 16 – Banco de Dados parte 1 26

Figura 17 – Banco de dados parte 2 59

Figura 18 – Banco de Dados parte 3 63

Figura 19 – Tela de Carregamento do Sistema 63

Figura 20 – Tela de Login do Sistema 63

Figura 21 – Tela de Consultas do Sistema 64

Figura 22 – Tela de Consultas de Ônibus do Sistema 64

Figura 23 – Tela de Consulta de Motorista do Sistema 64

Figura 24 – Tela de Consulta de Ocorrências 64

Figura 25 – Tela de Consulta de Fluxo do Sistema 65

Figura 26 – Tela de Cadastro de Motorista do Sistema 65

Figura 27 – Tela de Cadastro de Ônibus do Sistema 66

Figura 28 – Tela de Fluxo do Dia do Sistema 67

Figura 29 – Tela de Ocorrências do Sistema 67

Figura 30 – Tela de Pesquisa do Sistema 67

Figura 31 – Tela de Busca de Grade do Sistema 67

Sumário

[1 Introdução 8](#_Toc517899091)

[2 Objetivos – Gerais e Específicos 9](#_Toc517899092)

[2.1 Objetivos Gerais 9](#_Toc517899093)

[2.2 Objetivo Especifico 9](#_Toc517899094)

[3 Metodologia 10](#_Toc517899095)

[3.1 Ferramentas e Linguagens utilizadas 10](#_Toc517899096)

[3.1.1 Visual Studio 10](#_Toc517899097)

[3.1.2 Linguagem de Programação 11](#_Toc517899098)

[3.1.3 Adobe Photoshop 11](#_Toc517899099)

[3.1.4 Microsoft SQL Server 12](#_Toc517899100)

[3.1.5 Microsoft Word 12](#_Toc517899101)

[4 Pesquisa de Campo 13](#_Toc517899102)

[5 Desenvolvimento do Projeto 13](#_Toc517899103)

[5.1 Fluxograma 13](#_Toc517899104)

[6 Pesquisa de Mercado 15](#_Toc517899105)

[?? Diferencial 16](#_Toc517899106)

[9 Análise 16](#_Toc517899107)

[9.1 EAP 17](#_Toc517899108)

[9.2 Diagrama de Caso de uso 18](#_Toc517899109)

[9.3 Diagrama de Entidade e Relacionamento 19](#_Toc517899110)

[9.4 Diagrama de Classe 20](#_Toc517899111)

[9.5 Diagrama de Pacotes 21](#_Toc517899112)

[10 Requisitos Funcionais 22](#_Toc517899113)

[10.1 Banco de Dados: 22](#_Toc517899114)

[10.2 Sistema: 24](#_Toc517899115)

[ Splash 24](#_Toc517899116)

[ Tela de Login: 25](#_Toc517899117)

[ Consultas: 25](#_Toc517899118)

[ Consultar Ônibus: 26](#_Toc517899119)

[ Consultar Motorista 27](#_Toc517899120)

[ Consultar Ocorrências 27](#_Toc517899121)

[ Consultar Fluxo 28](#_Toc517899122)

[ Cadastro/Motorista 28](#_Toc517899123)

[ Cadastro Ônibus 29](#_Toc517899124)

[ Fluxo do Dia: 30](#_Toc517899125)

[ Ocorrências: 30](#_Toc517899126)

[ Pesquisa 31](#_Toc517899127)

[ Grade: 31](#_Toc517899128)

[11 Requisitos funcionais 32](#_Toc517899129)

[11.1 Requisitos não funcionais 32](#_Toc517899130)

[12 PSI – Política de Segurança da Informação 32](#_Toc517899131)

[ Objetivos 33](#_Toc517899132)

[ Aplicações da PSI 33](#_Toc517899133)

[ Princípios da PSI 33](#_Toc517899134)

[ Requisitos da PSI 34](#_Toc517899135)

[ Responsabilidades Específicas 34](#_Toc517899136)

[ Monitoramento e Auditoria do Ambiente 35](#_Toc517899137)

[ Identificação 35](#_Toc517899138)

[ Disposições finais 35](#_Toc517899139)

[13 Conclusão 36](#_Toc517899140)

[?? Anexo 37](#_Toc517899141)

[??.1 Cronograma 37](#_Toc517899142)

[??.2 Dados Coletados com a Pesquisa 37](#_Toc517899143)

[??.3 Código das Telas 39](#_Toc517899144)

[- Código Tela Pesquisa: 65](#_Toc517899145)

# 

# 1 Introdução

Temos como objetivo dinamizar e facilitar o controle de chegada e saída dos ônibus. Com isso, trazer praticidade e rapidez na administração dos horários.

Através do sistema os locais de saída e chegada dos ônibus ganhará tempo, pois deixará de ser um trabalho totalmente manual.

O sistema facilita no armazenamento dos dados, pois além de evitar possíveis perdas, é de baixo custo. Assim, empresas de transportes poderá ter um controle melhor sobre tais.

# 

# 2 Objetivos – Gerais e Específicos

## 2.1 Objetivos Gerais

* Facilitar o trabalho dos fiscais de terminais e pontos de ônibus.

## 2.2 Objetivo Especifico

* Ajudar na ecologia com a preservação do meio ambiente
* Ajudar na segurança das informações coletadas.

# 

# 3 Metodologia

A metodologia que será utilizada para a criação do software será:

* Visual Studio 2013 que será utilizado para a criação do sistema em C#.
* Adobe Photoshop CC 2017 para a criação da logo.
* SQL Serve para o desenvolvimento do Banco de Dados.

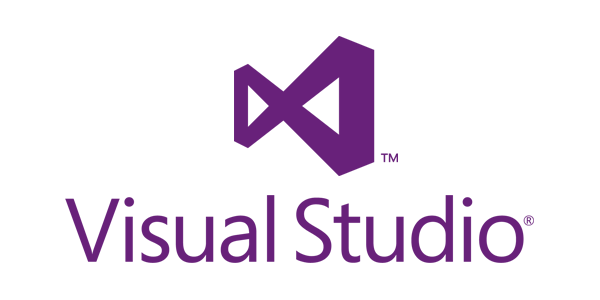
## 3.1 Ferramentas e Linguagens utilizadas

As ferramentas e linguagem utilizadas para o desenvolvimento do TCC serão:

### 3.1.1 Visual Studio

A Microsoft Visual Studio é um pacote de programas da Microsoft que contém diversas ferramentas para o desenvolvimento de softwares.

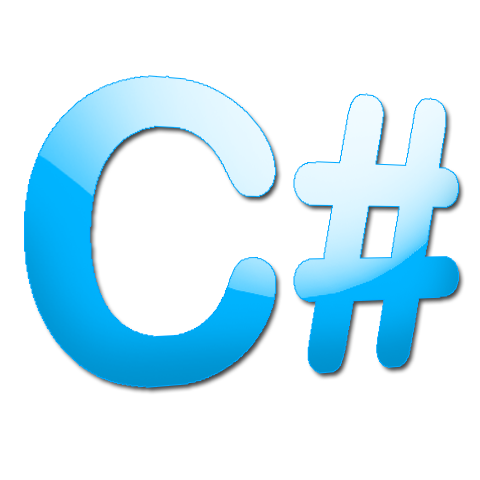
Figura 1 – Visual Studio



### 3.1.2 Linguagem de Programação

C# é uma linguagem de programação, multiparadigma, de tipagem forte, desenvolvida pela Microsoft.

Figura 2 – C#



### 3.1.3 Adobe Photoshop

Adobe Photoshop é um programa para edição de imagens e seu uso é fácil com resultados muito eficientes. Ele é líder em edição de fotos.

Figura 3 – Adobe Photoshop



### 3.1.4 Microsoft SQL Server

SQL Server é um software para o gerenciamento de banco de dados, ele é bem fácil de se utilizar.

Figura 4 – Microsoft SQL Server



## 

### 3.1.5 Microsoft Word

O Word é um editor de texto produzido pela Microsoft, suas funções variam das mais simples como escrever um texto comum, até produzir grandes documentos e formulários.

Figura 5 – Word



# 4 Pesquisa de Campo

Entrevista feita com profissional do Terminal Campo Belo, com objetivo de obter informações para melhor elaboração do TCC.

O objetivo de nossa entrevista foi coletar mais informações sobre o nosso tema para iniciar o desenvolvimento de nosso sistema baseado nas necessidades do cliente. Com a entrevista conseguimos coletar quais dados são os mais importantes para o sistema.

## 4.1 Pesquisa de Mercado

Não encontramos informações sobre programas que servem para obtenção de informações de transportes e armazenamento sobre tais. Por isso, tivemos a ideia de criar algo que ajude, já que esse trabalho é totalmente manual.

# 5 Desenvolvimento do Projeto

Com a falta de informalização nos terminais de pontos de ônibus. Surgiu nossa ideia de desenvolver um software para gerenciar o fluxo dos terminais e pontos finais de ônibus. O Sistema contará com o cadastro de motoristas, ônibus, consultas de fluxo diário e consultas em geral das informações guardadas em nosso banco de dados.

## 5.1 Fluxograma

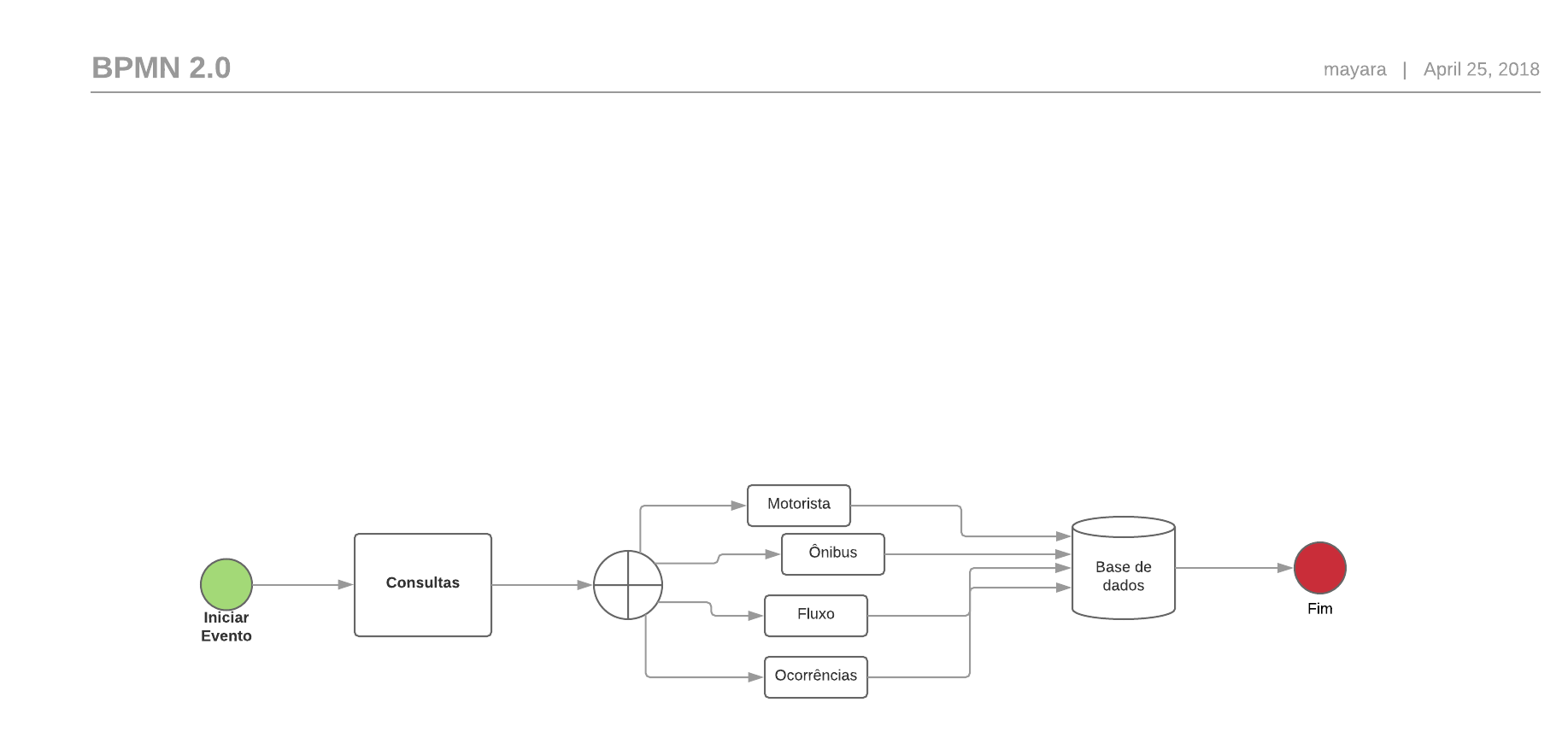
Figura 2 – Fluxograma Processo de Consulta

Figura 3 – Fluxograma de Processo de Cadastro de Ônibus

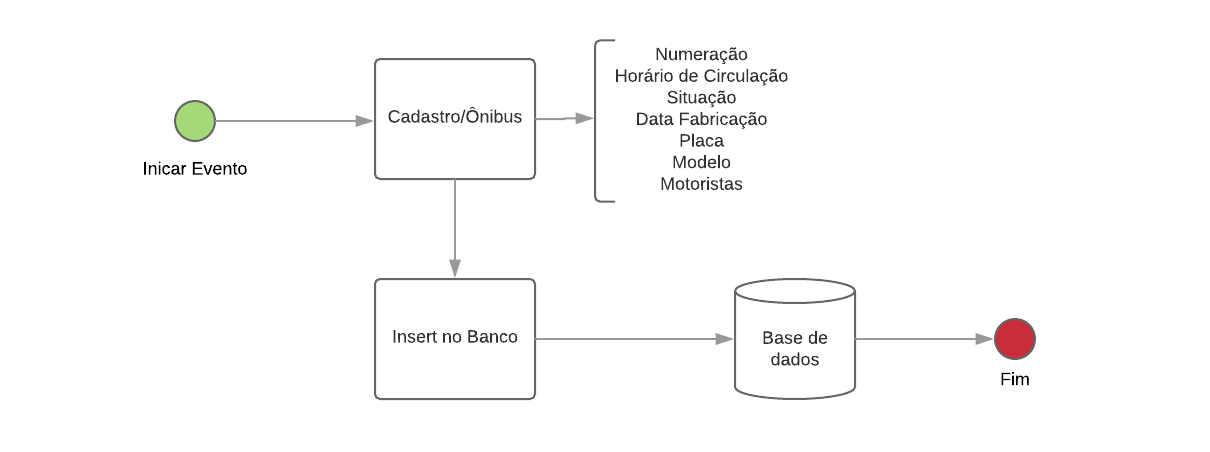


Figura 4 – Fluxograma de Processo de Cadastro de Motorista

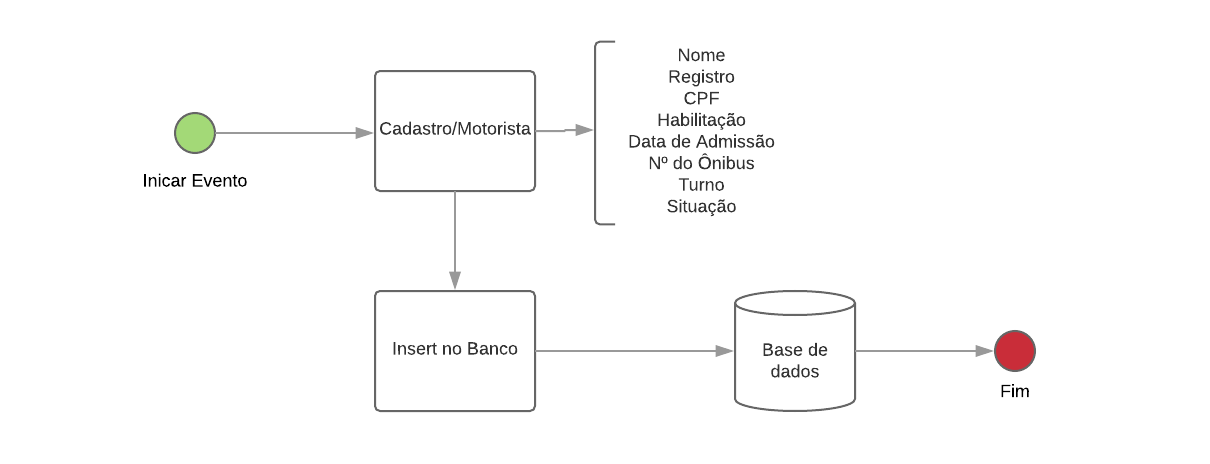


Figura 5 – Fluxograma de Consultar Fluxo do Dia

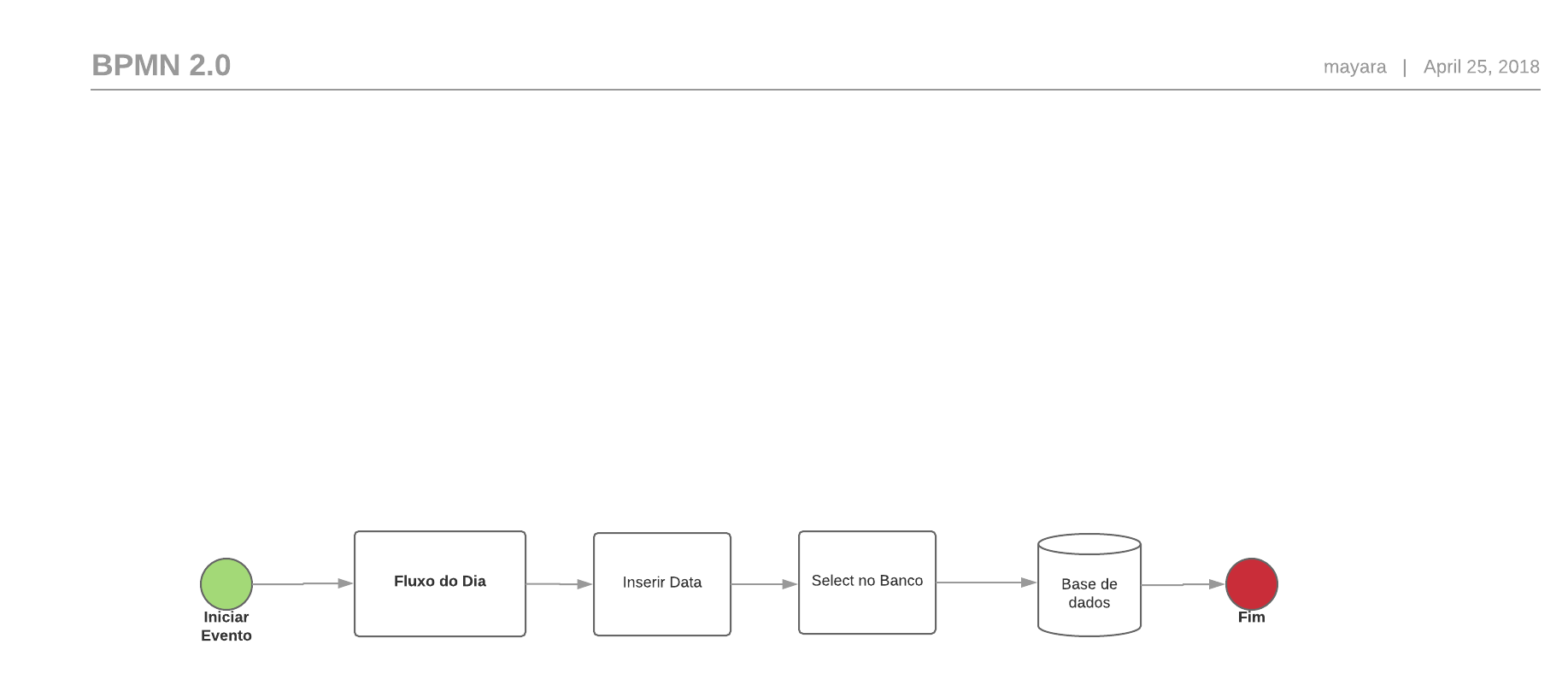
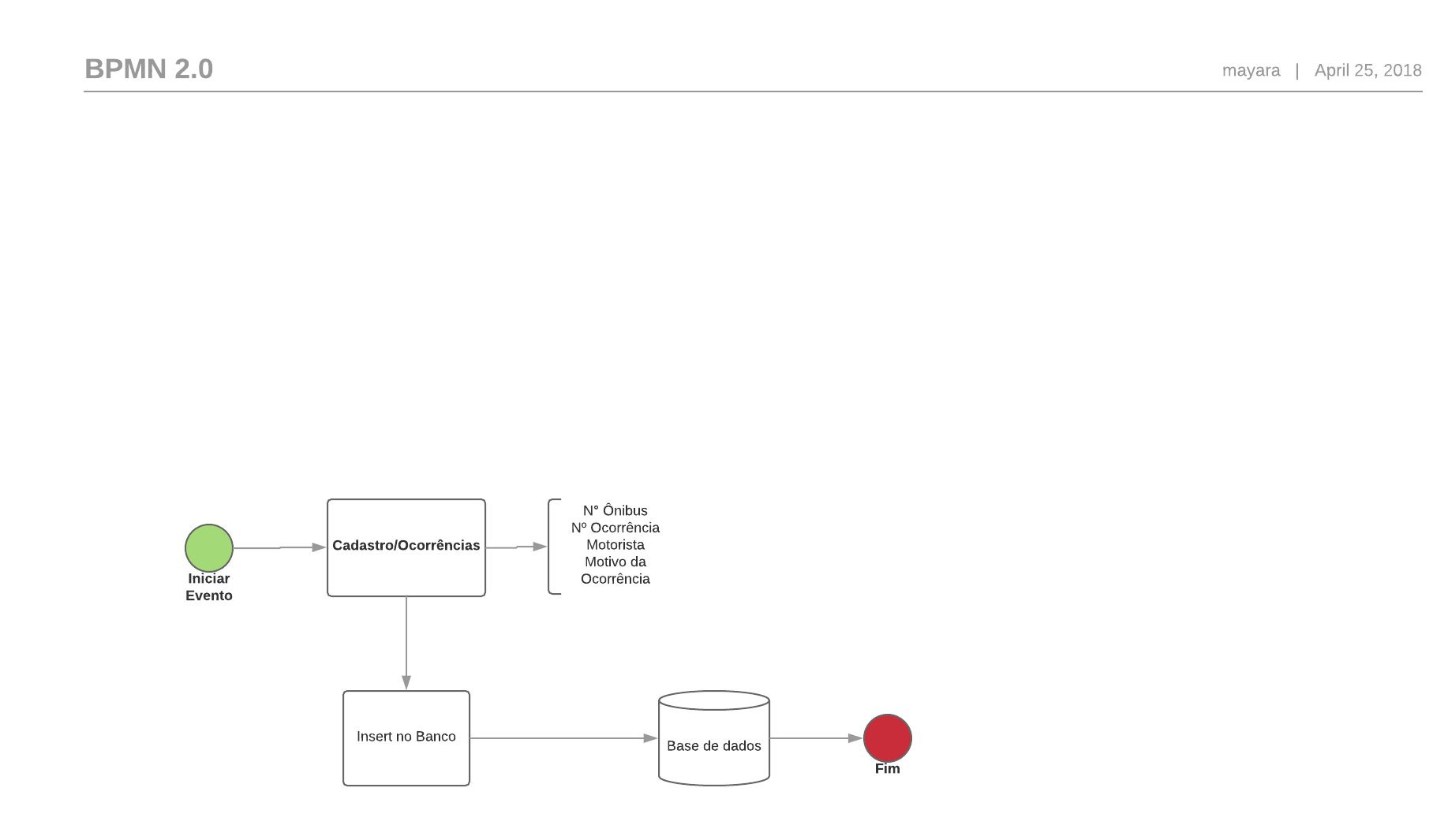


Figura 6 – Fluxograma de Processo de Cadastro de Ocorrência



# ?? Diferencial

O controle de transportes deve ser utilizado para controlar o fluxo de tais, como seus horários e seus motoristas. Com isso, caso haja algum problema com os ônibus pode ser acrescentado no cadastro do transporte, e seu motorista deve informar o administrador do sistema sobre tal problema para o transporte ser encaminhado para uma avaliação.

As principais vantagens que podemos destacar é a praticidade para armazenamento e já que esse trabalho é feito manualmente, podendo ser perdido, o sistema oferece uma segurança maior.

# 9 Análise

O controle de transportes é necessário para a empresa, pois tem como função manter a empresa informada de modo constante com seriedade sobre seus transportes.

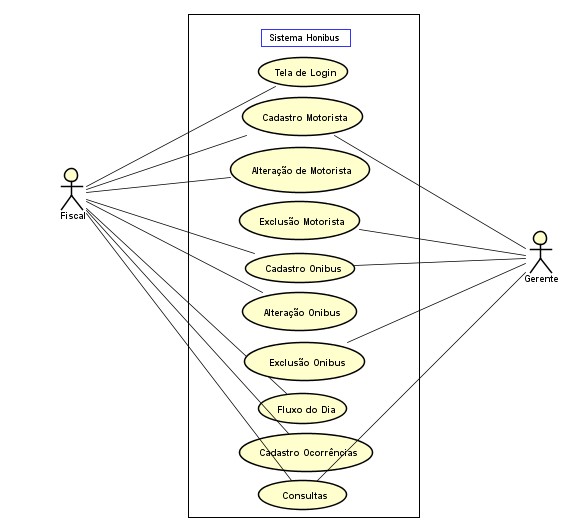
É necessário constatar a obrigação de ter um controle eficaz, facilitando o desenvolvimento da empresa e que a mesma cumpra suas funções de atender com rapidez e eficiência.

## 9.1 EAP

Figura 12 - EAP

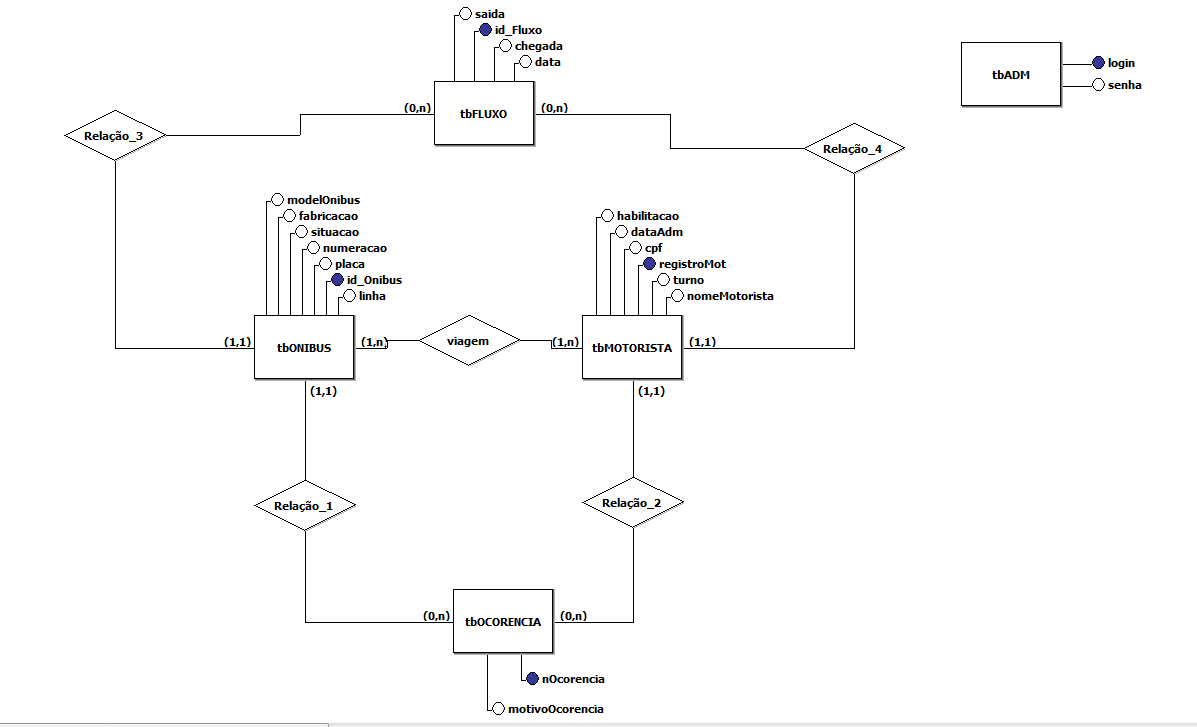
## 9.2 Diagrama de Caso de uso

Figura 13 – Diagrama de Caso de Uso



## 9.3 Diagrama de Entidade e Relacionamento

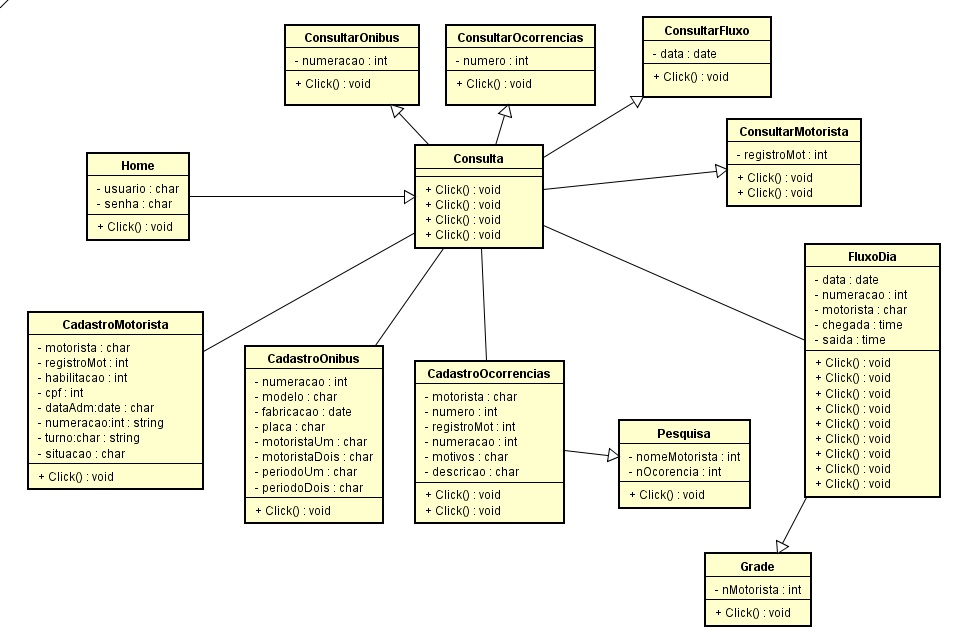
Figura 14 – Diagrama de Entidade e Relacionamento



## 

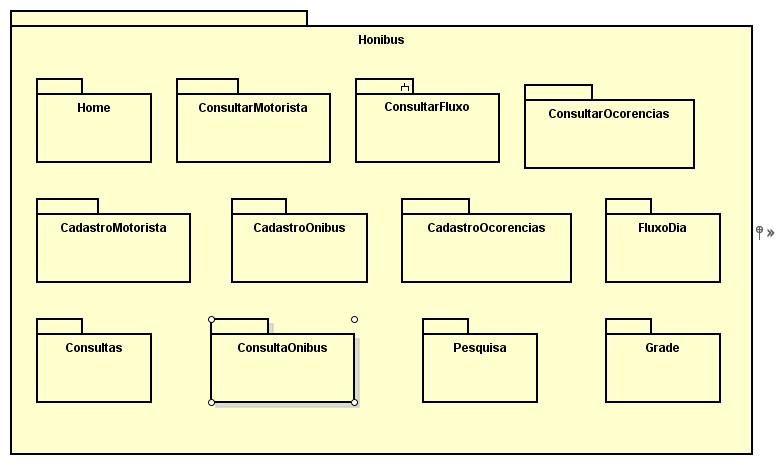
## 9.4 Diagrama de Classe

Figura 15 – Diagrama de Classes



## 9.5 Diagrama de Pacotes

Figura ?? – Diagrama de Pacotes



# 10 Requisitos Funcionais

### 10.1 Banco de Dados:

Figura 16 – Banco de Dados parte 1

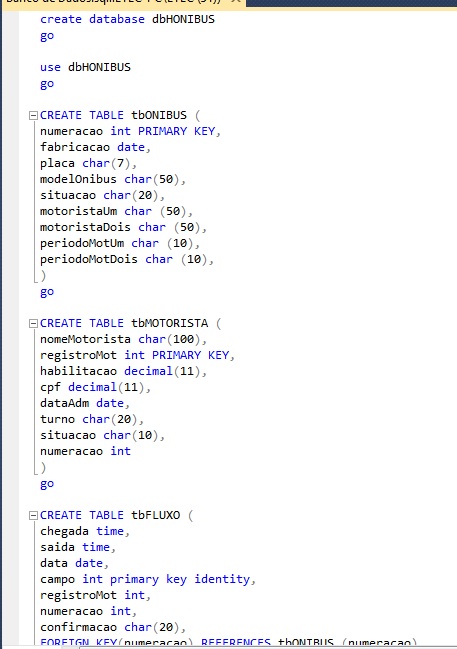
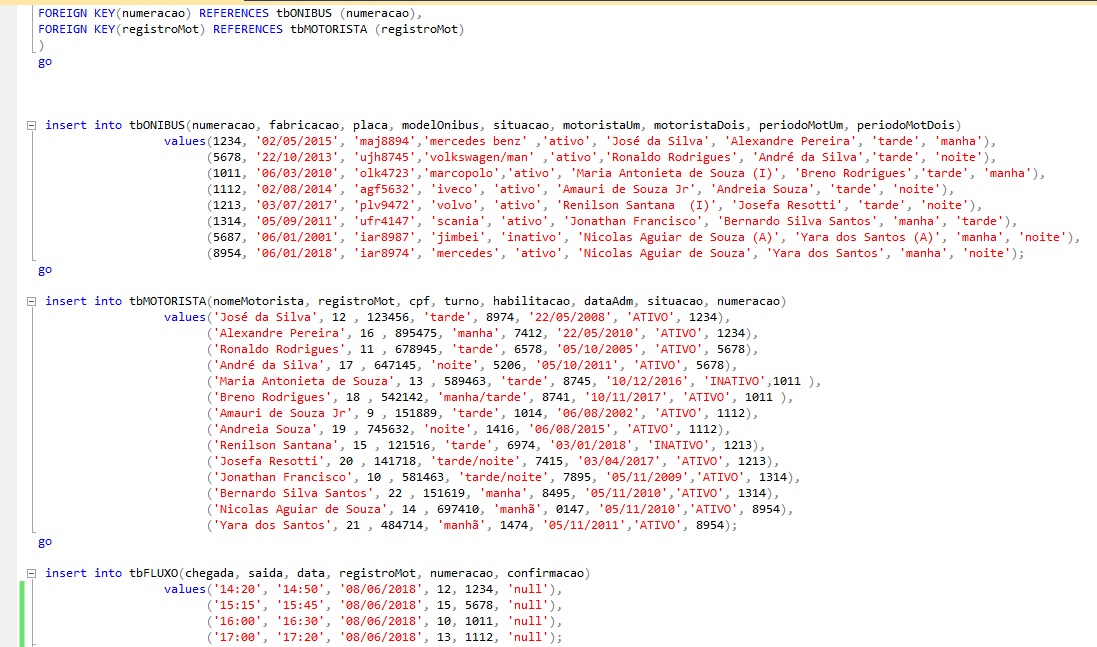
****

Figura 17 – Banco de dados parte 2



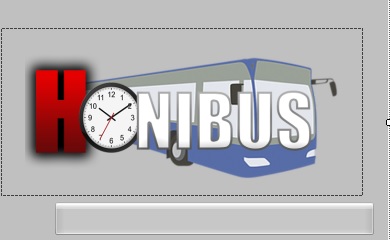
Figura 18 – Banco de Dados parte 3



### 10.2 Sistema:

### Splash

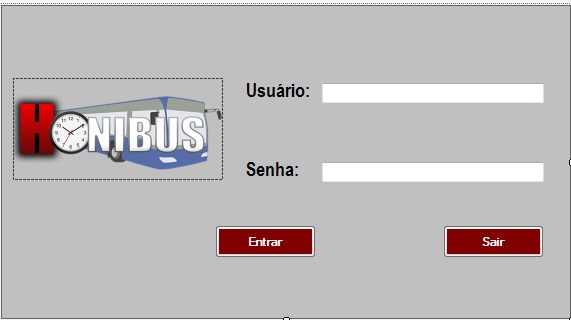
Figura 19 – Tela de Carregamento do Sistema



**Explicação da Tela:** Está é a Tela Splash, onde aparece carregando nosso sistema.

### Tela de Login:

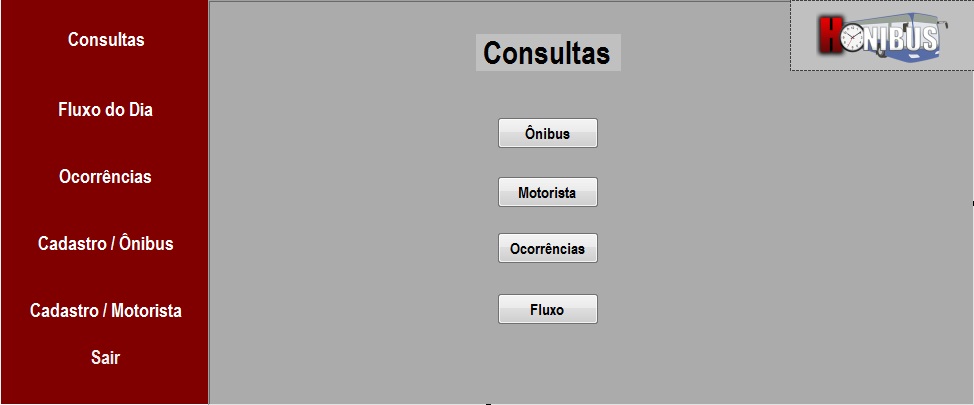
Figura 20 – Tela de Login do Sistema

****

**Explicação da Tela:** Esta é a tela de Login, onde no primeiro text box (Usuário) ele coletará o e-mail do usuário e no segundo (Senha) coletará a senha do usuário. Clicando no botão Entrar o sistema fará uma verificação no banco de dados se o E-mail e a Senha são validos, e no Botão sair fechará o sistema.

### Consultas:

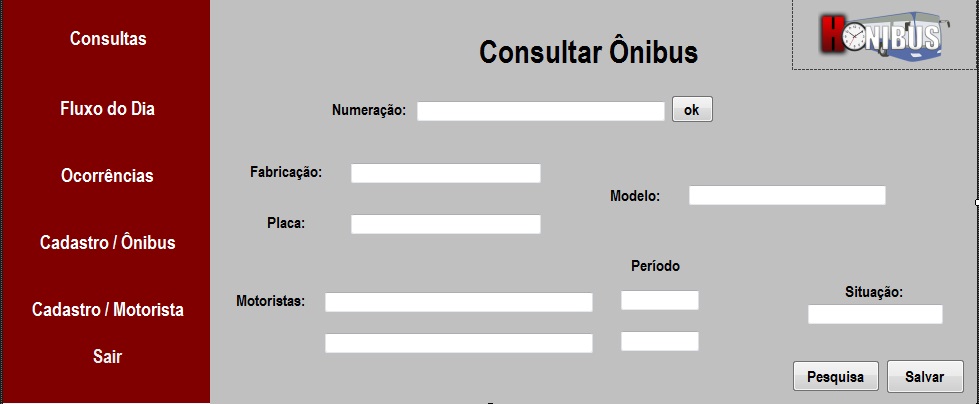
Figura 21 – Tela de Consultas do Sistema



**Explicação da Tela:** Está é a tela de consultas, do lado esquerdo da tela aparece o menu completo do sistema e no lado direito as opções de consultas.

### Consultar Ônibus:

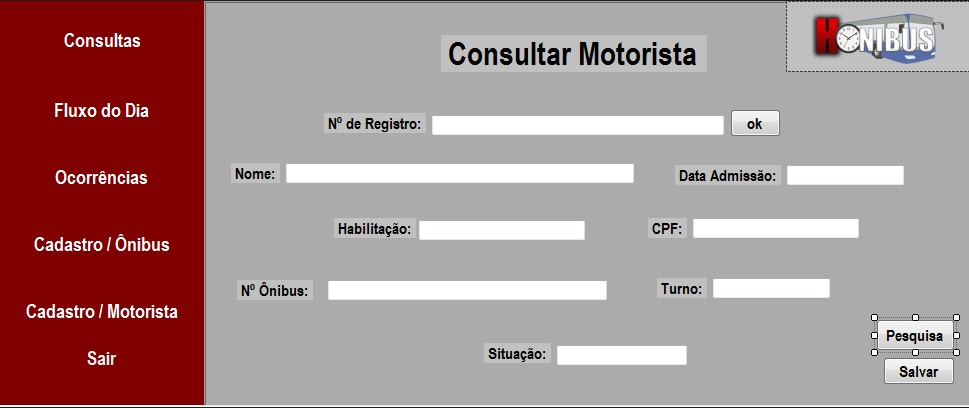
Figura 22 – Tela de Consultas de Ônibus do Sistema

****

**Explicação da Tela:** Na tela de Consulta Ônibus, no primeiro text box (Numeração) o usuário deverá digitar a numeração do ônibus desejado e clicar no botão OK, feito isso todos os campos serão preenchidos com os dados do respectivo ônibus e o único campo em que poderá ser alterado é o da Situação podendo coletar “Ativo” ou “Inativo”. Clicando no botão Salvar ele fará uma alteração no banco de dados.

### Consultar Motorista

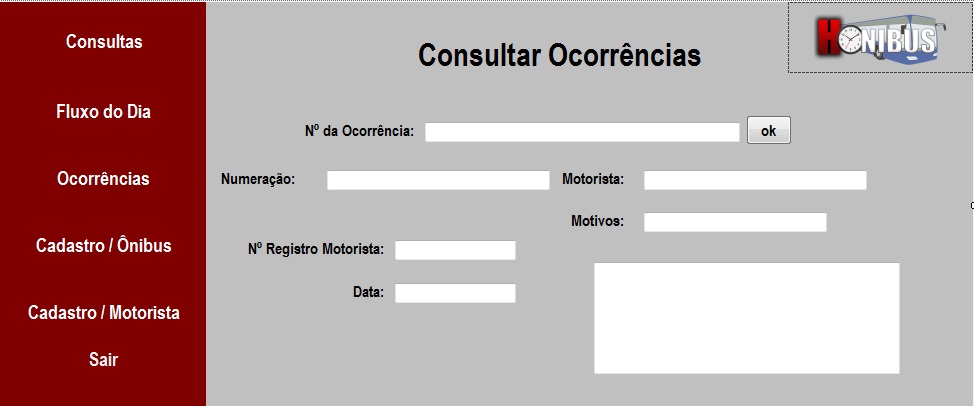
Figura 23 – Tela de Consulta de Motorista do Sistema



**Explicação da Tela:** A Tela de Consultar Motorista é muito semelhante a de Consulta de Ônibus. Nela o usuário digitará o número do registo do motorista no primeiro campo e logo em seguida clicará no botão OK, feito isso todos os campos preencherá automaticamente e só poderá altera o campo de Situação para “Ativo” ou “Inativo”, clicando no botão Salvar ele alterará a situação no banco de dados.

### Consultar Ocorrências

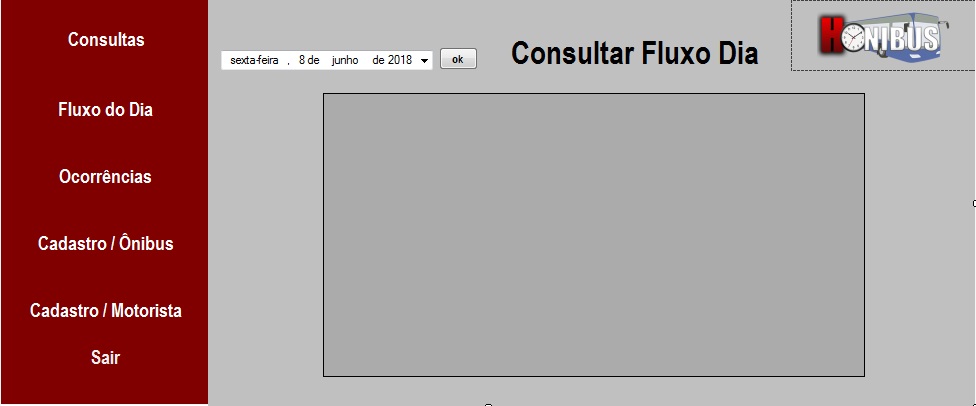
Figura 24 – Tela de Consulta de Ocorrências

****

**Explicação da Tela:** Semelhante com as telas anteriores, na Tela de Consultar Ocorrências o usuário digitará a numeração da ocorrência e clicando no botão ok preencheram todos os campos automaticamente podendo visualizar todos os dados da respectiva Ocorrência.

### Consultar Fluxo

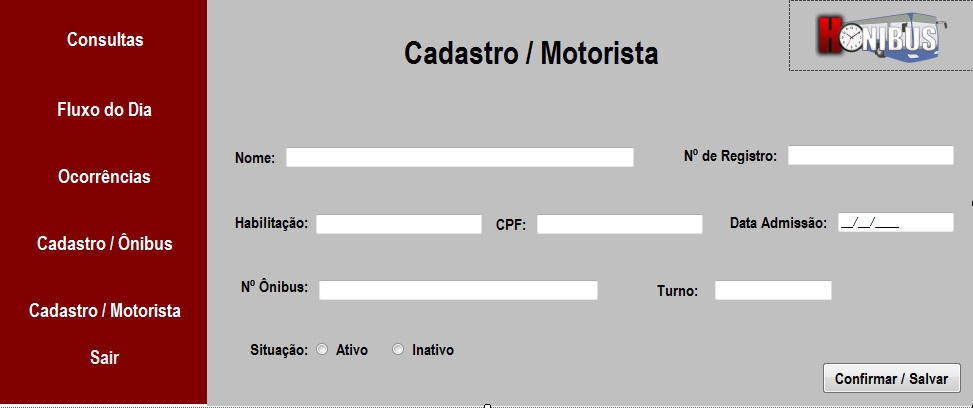
Figura 25 – Tela de Consulta de Fluxo do Sistema

****

**Explicação da Tela:** Na tela Consultar Fluxo Dia o usuário selecionará a data do dia em que ele gostaria de consultar no calendário e feito isso é só clicar no botão ok que preencherá uma tabela com todas as informações do fluxo do dia desejado.

### Cadastro/Motorista

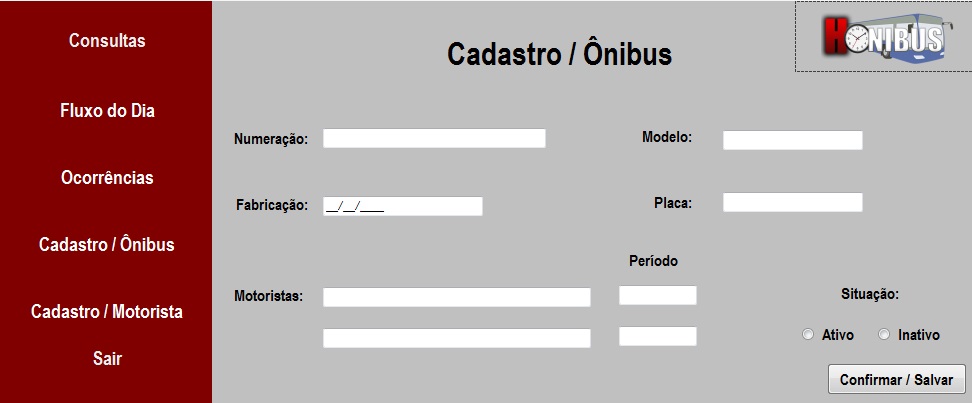
Figura 26 – Tela de Cadastro de Motorista do Sistema

****

**Explicação da Tela:** Na Tela de Cadastro de Motorista o usuário deve simplesmente preencher todos os campos com os dados do motorista e clicar no botão “Confirmar / Salvar” que irá fazer um include no banco de dados com todas essas informações.

### Cadastro Ônibus

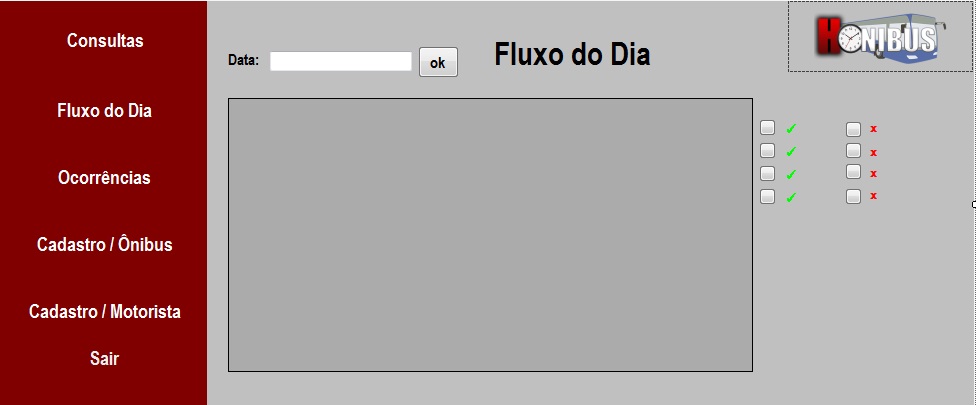
Figura 27 – Tela de Cadastro de Ônibus do Sistema

****

**Explicação da Tela:** Assim como na tela anterior, nessa tela o usuário preencherá todos os campos com as informações do Ônibus e clicará no botão “Conformar / Salvar” que todos os dados serão incluídos no banco de dados.

### Fluxo do Dia:

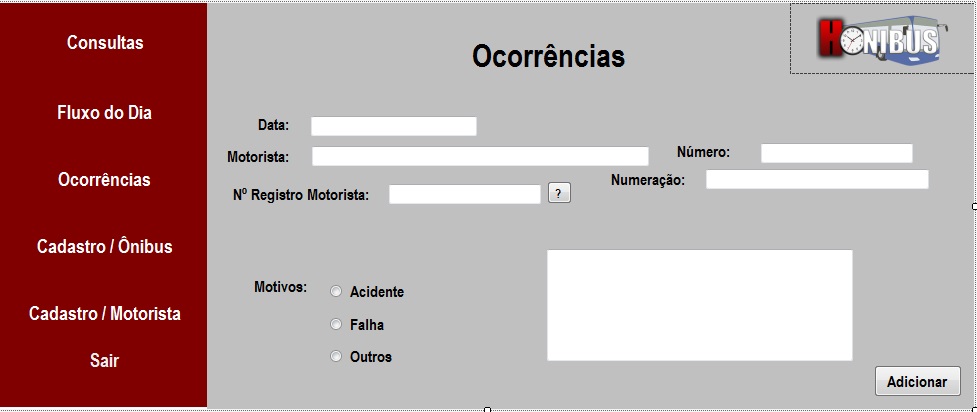
Figura 28 – Tela de Fluxo do Dia do Sistema

****

**Explicação da Tela:** Está é a Tela do Fluxo do Dia, nela irá aparecer a data do dia atual e uma tabela com todo o fluxo diário (nome do Motorista, Ônibus, Horário de sair e chegada, etc.), quando tudo ocorrer de acordo com a tabela o usuário simplesmente clicara no botão do visto confirmando que está tudo certo, e caso ocorra um imprevisto como um atraso por exemplo ele clicará no botão do x vermelho, que irá redirecionar para a tela de ocorrências onde ele especificará o problema.

### **Ocorrências**:

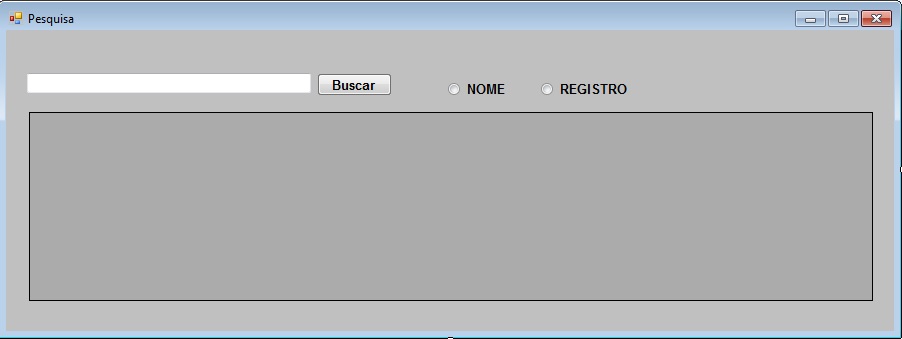
Figura 29 – Tela de Ocorrências do Sistema

****

**Explicação da Tela:** Esta é a tela de Ocorrência, ela preencherá tudo automaticamente, o usuário só precisará clicar no botão “?” e escolher o nome do motorista, feito isso ele escolherá o motivo da ocorrência e descrever o que ouve. Clicando o botão “Adicionar” a ocorrência será incluída no banco de dados.

### Pesquisa

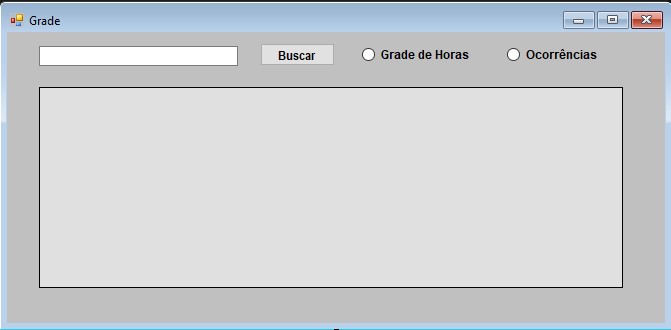
Figura 30 – Tela de Pesquisa do Sistema

****

**Explicação da Tela:** Está é a tela de pesquisa de motorista. Clicando no botão “?” da tela anterior ela abrirá, ela é uma tela simples, o usuário só precisa preencher o campo com o nome ou o número do registo do mesmo e especificar o que digitou. Clicando no botão “Buscar” irá aparecer as informações do motorista, então ele só deverá clicar.

### Grade:

Figura 31 – Tela de Busca de Grade do Sistema

****

**Explicação da Tela:** Está é a tela grade, no text box o usuário deve preencher a numeração do registro do motorista e especificar se ele quer consultar a grade de horário ou ocorrências do mesmo, e clicando no botão “Buscar aparecera uma tabela com as informações especificadas.

# 11 Requisitos funcionais

# 11.1 Requisitos não funcionais

Nossos requisitos não funcionais será ter apenas um login, que será passado apenas a pessoas autorizadas, tendo como responsabilidade de tais não o repassar a outros.

# 12 PSI – Política de Segurança da Informação

Documentos de Diretrizes e Normas Administrativas

# Objetivos

Designar normas para que permita aos usuários do Sistema Honibus seguirem regras de ética relacionadas à segurança da informação de forma adequada, mantendo a integridade e necessidade dos colaboradores.

Estabelecer a descrição de procedimentos de segurança da informação, como incluir controles e processos.

Garantir as informações do Honibus quanto à:

1. **Integridade:** manter a informação original, visando sua proteção, contra qualquer uso indevido.
2. **Confidencialidade:** garantia de que o acesso seja apenas a pessoas autorizadas.
3. **Disponibilidade:** manter o acesso a usuários autorizados sempre que necessário.

# Aplicações da PSI

Os colaboradores devem prosseguir com as normas aqui estabelecidas, e tais se aplicam à informação em qualquer meio ou suporte.

Essas regras deixam explícitas a cada colaborador autorizado, que tem acesso ao sistema, que serão monitorados conforme a lei. É dever de cada funcionário se manter atualizado em relação à PSI.

# Princípios da PSI

Quaisquer informações recebida pelos colaboradores pertence a instituição Honibus. Todas as normas devem ser explicitas e formalizadas com a equipe.

O sistema deve ser utilizado para atividades profissionais. O Honibus poderá registrar o uso, garantindo a disponibilidade e segurança das informações.

# Requisitos da PSI

Para que o sistema seja utilizado de forma adequada, todos os colaboradores devem ser comunicados, a fim de que sua politica seja cumprida. Um comitê deve ser organizado, visando às necessidades de cada colaborador.

# Responsabilidades Específicas

1. **Diretor**

Diretor deve ser responsável por toda a gestão da companhia, guiando seus supervisores para que executem com êxito a politica estabelecida.

1. **Supervisor**

Todas as instituições que faz a utilização do Honibus como sistema principal, deve ter um supervisor para que direcione os administradores do Sistema, visando cumprir a politica e ética da instituição.

1. **Administradores**

As companhias colaboradoras devem ter administradores para que façam a gestão do Sistema, conforme as normas estabelecidas na PSI.

Os administradores têm acesso à programação dos horários, podendo organizar conforme mandado pelo supervisor, através do Banco de Dados.

1. **Fiscais**

Fiscais são necessários para o cumprimento dos horários, descrições de ocorrências, caso ocorra. Eles devem confirmar os horários diretos no sistema.

# Monitoramento e Auditoria do Ambiente

Para garantia as normas estabelecidas na PSI, o Honibus poderá:

* Fazer-se públicas informações contidas no sistema, no caso de haver exigência judicial ou solicitação do Comitê de Segurança da Informação.
* Realização de inspeção física nas máquinas dos administradores.

# Identificação

O sistema contém usuário e senha para que o acesso seja privado, sendo utilizado apenas por funcionários autorizados, evitando que pessoas não autorizadas acessem.

O uso indevido do login é considerado crime perante o Código Penal Brasileiro (Art. 307 – Falsa Identidade).

A norma deve ser seguida por todos os colaboradores, com responsabilidade.

# Disposições finais

A ética e segurança devem ser conhecidas como parte fundamental da instituição. Isto é, qualquer uso indevido de segurança será entendido como alguém agindo contra a ética geral regido na companhia.

# 13 Conclusão

Neste trabalho abordamos o assunto referente ao fluxo de ônibus. Toda empresa de transportes necessita de um meio para armazenar informações, os horários de funcionamento de seus transportes e seus motoristas. Com isso, constatamos que é necessária uma gestão de transportes bem planejada e eficiente, utilizando as ferramentas certas para trazer resultados positivos, como diminuir tempo trabalhando manualmente.

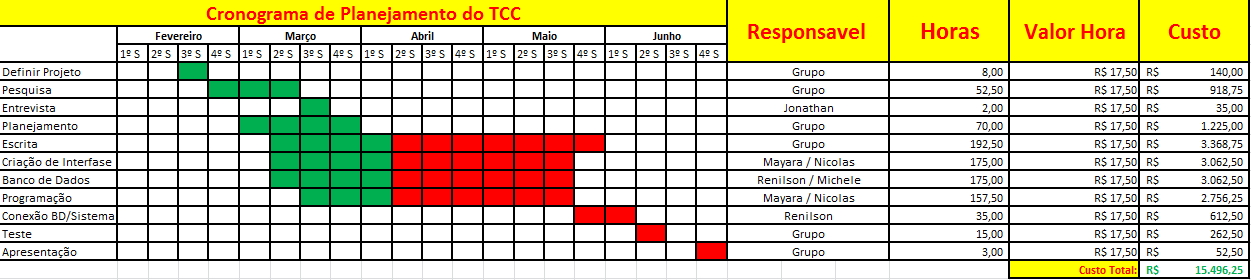
Efetuamos algumas as metas que nós tínhamos proposto, porém pelo curto tempo não foi possível realizar algumas, como, autenticação de dois fatores, contagem de passageiros e ter logins diferentes para cada função contida no sistema.

A finalização deste projeto nos foi importante para o nosso desenvolvimento pessoal e profissional, pois nos permitiu aprofundar nossos conhecimentos, compreendendo melhor a importância do controle dos terminais de ônibus. Contudo, este trabalho nos permitiu aprimorar nossas competências de percepção, organização e comunicação da informação.

# ?? Anexo

## ??.1 Cronograma

Figura 1 – Cronograma



## ??.2 Dados Coletados com a Pesquisa

1. Quais tipos de dados é preciso armazenar para administrar o fluxo dos ônibus?

* Controle de horários chegada/saída.
* Ocorrências.
* Dados dos ônibus.
* Dados dos motoristas.

1. Facilitaria não guardar essas informações em papeis e sim através de um programa?

* Evitaria perdas.
* Facilitaria consultas.
* Economizaria com os gastos de papeis diários.
* Evitaria alteração nos horários, pois o programa não permitiria alteração por ficais, sendo um acesso apenas para o Gerente.

1. Já aconteceu de perder alguma das informações por ser guardada em papeis?

* Sim, pois é difícil organizar muitos papeis, porque até guardar os papeis alguns são perdidos.
* Em época de chuva, houve uma ocorrência e os dados foram perdidos, porque o papel que continha os dados foi molhado.

1. Cada motorista tem um determinado ônibus ou qualquer motorista que faz aquele percurso pode ir com qualquer ônibus?

* Cada motorista tem um ônibus especifico, tendo ele também substitutos, caso não possa trabalhar no dia.

1. Os ônibus quebram com frequência? O terminal tem controle dos ônibus que estão em manutenção?

* Ônibus quebrados são mandados para garagem, onde tem pessoas para fazer a manutenção.
* Frota disponibiliza outro ônibus, caso aquele não seja consertado no dia.

1. Com que frequência os ônibus são mandados para manutenção?

* Depende do tempo do ônibus em circulação e também da forma como é cuidado.

1. Em média quanto tempo um ônibus fica em circulação?

* 10 anos, após isso é vendido para países do continente da África.

1. O sistema facilitaria consultas do ônibus e quais motoristas estão responsáveis por eles?

* Sim, pois facilitaria o controle dos ônibus e dos motoristas, evitando confusões em casos de ocorrências.

1. Onde são guardados os documentos do fluxo de ônibus? Caso sejam guardados, facilitaria ter um banco de dados desses documentos?

* Fica numa sala de arquivos, o que leva um tempo para poder encontrar algum dado.
* Facilitaria, pois seria mais rápido.

1. Você teria alguma sugestão para incrementar em nosso sistema?

* Controle de passageiros.

- Não haverá controle de passageiros, pois não faz parte da estrutura do sistema, porque seria difícil incrementar por enquanto, mas podendo ser adicionado futuramente.

## ??.3 Código das Telas

**- Código Tela Splash:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Honibus

{

public partial class Splash : Form

{

public Splash()

{

InitializeComponent();

}

private void timer1\_Tick(object sender, EventArgs e)

{

if(carrega.Value < 100){

carrega.Value = carrega.Value + 2;

}

else{

timer1.Enabled = false;

this.Visible = false;

Home login = new Home();

login.ShowDialog();

}

}

private void Splash\_Load(object sender, EventArgs e)

{

}

}

}

**- Código Program.cs:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Honibus

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Splash());

}

}

}

**- Código Tela Login:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Honibus

{

public partial class Home : Form

{

SqlConnection sqlConn = null;

private string strConn = @"Data Source=ETEC-PC;Initial Catalog=dbHONIBUS;Integrated Security=True";

private string \_Sql = string.Empty;

public Home()

{

InitializeComponent();

}

public void logar()

{

sqlConn = new SqlConnection(strConn);

string usuario, senha;

try

{

usuario = textBox1.Text;

senha = textBox2.Text;

\_Sql = "SELECT COUNT (usuario) FROM tbADM WHERE usuario = @usuario AND senha = @senha ";

SqlCommand comando = new SqlCommand(\_Sql,sqlConn);

comando.Parameters.Add("@usuario", SqlDbType.VarChar).Value = usuario;

comando.Parameters.Add("@senha", SqlDbType.VarChar).Value = senha;

sqlConn.Open();

int v = (int)comando.ExecuteScalar();

if (v > 0)

{

this.Visible = false;

Consultas Form2 = new Consultas();// feito para deixar tela anterior invísivel

Consultas newForm3 = new Consultas();// abrindo a nova tela

newForm3.ShowDialog();

}

else

{

MessageBox.Show("Usuario ou senha incorretos");

}

}

catch (SqlException erro)

{

MessageBox.Show(erro + "No banco");

}

}

private void button2\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void button1\_Click(object sender, EventArgs e)

{

logar();

}

}

}

**- Código Tela Consultas:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Honibus

{

public partial class Consultas : Form

{

public Consultas()

{

InitializeComponent();

}

private void Form3\_Load(object sender, EventArgs e)

{

}

private void fluxoDoDiaToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

Consultas Form2 = new Consultas();

Fluxo newForm4 = new Fluxo();

newForm4.ShowDialog();

}

private void consultasToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

Consultas Form2 = new Consultas();

Consultas newForm3 = new Consultas();

newForm3.ShowDialog();

}

private void ocorrênciasToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

Consultas Form2 = new Consultas();

Ocorrencias newForm5 = new Ocorrencias();

newForm5.ShowDialog();

}

private void cadastroÔnibusToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

Consultas Form2 = new Consultas();

CadastroOni newForm6 = new CadastroOni();

newForm6.ShowDialog();

}

private void cadastroToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

Consultas Form2 = new Consultas();

CadastroMoto newForm7 = new CadastroMoto();

newForm7.ShowDialog();

}

private void sairToolStripMenuItem\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void button1\_Click(object sender, EventArgs e)

{

this.Visible = false;

Consultas Form2 = new Consultas();

ConsultarOni newForm8 = new ConsultarOni();

newForm8.ShowDialog();

}

private void button5\_Click(object sender, EventArgs e)

{

this.Visible = false;

Consultas Form2 = new Consultas();

ConsultarMoto newForm10 = new ConsultarMoto();

newForm10.ShowDialog();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Visible = false;

Consultas Form2 = new Consultas();

ConsultarOco newForm12 = new ConsultarOco();

newForm12.ShowDialog();

}

private void button3\_Click(object sender, EventArgs e)

{

this.Visible = false;

Consultas Form2 = new Consultas();

ConsultarFluxo newForm14 = new ConsultarFluxo();

newForm14.ShowDialog();

}

}

}

**- Código Tela Consultar Ônibus:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Honibus

{

public partial class ConsultarOni : Form

{

SqlConnection sqlConn = new SqlConnection(@"Data Source=ETEC-PC;Initial Catalog=dbHONIBUS;Integrated Security=True");

public ConsultarOni()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

string sql = "SELECT \* FROM tbONIBUS WHERE numeracao='" + numeracao.Text + "' ";

SqlCommand comando = new SqlCommand(sql, sqlConn);

sqlConn.Open();

SqlDataReader ler = comando.ExecuteReader();

if (ler.Read() == true)

{

fabricacao.Text = ler["fabricacao"].ToString();

placa.Text = ler["placa"].ToString();

modelo.Text = ler["modelOnibus"].ToString();

situacao.Text = ler["situacao"].ToString();

motorista1.Text = ler["motoristaUm"].ToString();

motorista2.Text = ler["motoristaDois"].ToString();

periodo1.Text = ler["periodoMotUm"].ToString();

periodo2.Text = ler["periodoMotDois"].ToString();

}

else

{

MessageBox.Show("Registro não Cadastrado");

}

sqlConn.Close();

}

private void consultasToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

ConsultarOni Form2 = new ConsultarOni();

Consultas newForm3 = new Consultas();

newForm3.ShowDialog();

}

private void fluxoDoDiaToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

ConsultarOni Form2 = new ConsultarOni();

Fluxo newForm4 = new Fluxo();

newForm4.ShowDialog();

}

private void ocorrênciasToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

ConsultarOni Form2 = new ConsultarOni();

Ocorrencias newForm5 = new Ocorrencias();

newForm5.ShowDialog();

}

private void cadastroÔnibusToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

ConsultarOni Form2 = new ConsultarOni();

CadastroOni newForm6 = new CadastroOni();

newForm6.ShowDialog();

}

private void cadastroToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

ConsultarOni Form2 = new ConsultarOni();

CadastroMoto newForm7 = new CadastroMoto();

newForm7.ShowDialog();

}

private void sairToolStripMenuItem\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void button2\_Click(object sender, EventArgs e)

{

string strSQL = "UPDATE tbONIBUS SET situacao='" + situacao.Text + "'";

SqlCommand comando = new SqlCommand(strSQL, sqlConn);

try

{

sqlConn.Open();

comando.ExecuteNonQuery();

MessageBox.Show("Salvo com Sucesso!");

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message);

}

finally

{

sqlConn.Close();

}

}

private void ConsultarOni\_Load(object sender, EventArgs e)

{

}

private void button3\_Click(object sender, EventArgs e)

{

var form = new Grade2(numeracao.Text);

form.Show();

}

}

}

**- Código Tela Consultar Motorista:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Honibus

{

public partial class ConsultarMoto : Form

{

SqlConnection conx = new SqlConnection(@"Data Source=ETEC-PC;Initial Catalog=dbHONIBUS;Integrated Security=True");

private string \_Sql = string.Empty;

public ConsultarMoto()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

string sql = "SELECT \* FROM tbMOTORISTA WHERE registroMot='" + registroMot.Text + "' ";

SqlCommand comando = new SqlCommand(sql, conx);

conx.Open();

SqlDataReader ler = comando.ExecuteReader();

if (ler.Read() == true)

{

nome.Text = ler["nomeMotorista"].ToString();

dataAdm.Text = ler["dataAdm"].ToString();

habilitacao.Text = ler["habilitacao"].ToString();

cpf.Text = ler["cpf"].ToString();

turno.Text = ler["turno"].ToString();

situacao.Text = ler["situacao"].ToString();

numeracao.Text = ler["numeracao"].ToString();

}

else

{

MessageBox.Show("Registro não Cadastrado");

}

conx.Close();

}

private void consultasToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

ConsultarMoto Form2 = new ConsultarMoto();

Consultas newForm3 = new Consultas();

newForm3.ShowDialog();

}

private void fluxoDoDiaToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

ConsultarMoto Form2 = new ConsultarMoto();

Fluxo newForm4 = new Fluxo();

newForm4.ShowDialog();

}

private void ocorrênciasToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

ConsultarMoto Form2 = new ConsultarMoto();

Ocorrencias newForm5 = new Ocorrencias();

newForm5.ShowDialog();

}

private void cadastroÔnibusToolStripMenuItem\_Click(object sender, EventArgs

{

this.Visible = false;

ConsultarMoto Form2 = new ConsultarMoto();

CadastroOni newForm6 = new CadastroOni();

newForm6.ShowDialog();

}

private void cadastroToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

ConsultarMoto Form2 = new ConsultarMoto();

CadastroMoto newForm7 = new CadastroMoto();

newForm7.ShowDialog();

}

private void sairToolStripMenuItem\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void button2\_Click(object sender, EventArgs e)

{

string strSQL = "UPDATE tbMOTORISTA SET situacao='" + situacao.Text + '";

SqlCommand comando = new SqlCommand(strSQL, conx);

try

{

conx.Open();

comando.ExecuteNonQuery();

MessageBox.Show("Salvo com Sucesso!");

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message);

}

finally

{

conx.Close();

}

}

private void ConsultarMoto\_Load(object sender, EventArgs e)

{

}

private void button3\_Click(object sender, EventArgs e)

{

var form = new Grade(registroMot.Text);

form.Show();

}

}

}

**- Código Tela Consultar Ocorrências:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Honibus

{

public partial class ConsultarOco : Form

{

SqlConnection sqlConn = new SqlConnection(@"Data Source=ETEC-PC;Initial Catalog=dbHONIBUS;Integrated Security=True");

private string \_Sql = string.Empty;

public ConsultarOco()

{

InitializeComponent();

}

private void consultasToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

ConsultarOco Form2 = new ConsultarOco();

Consultas newForm3 = new Consultas();

newForm3.ShowDialog();

}

private void fluxoDoDiaToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

ConsultarOco Form2 = new ConsultarOco();

Fluxo newForm4 = new Fluxo();

newForm4.ShowDialog();

}

private void ocorrênciasToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

ConsultarOco Form2 = new ConsultarOco();

Ocorrencias newForm5 = new Ocorrencias();

newForm5.ShowDialog();

}

private void cadastroÔnibusToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

ConsultarOco Form2 = new ConsultarOco();

CadastroOni newForm6 = new CadastroOni();

newForm6.ShowDialog();

}

private void cadastroToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

ConsultarOco Form2 = new ConsultarOco();

CadastroMoto newForm7 = new CadastroMoto();

newForm7.ShowDialog();

}

private void sairToolStripMenuItem\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void button1\_Click(object sender, EventArgs e)

{

string sql = "SELECT \* FROM tbOCORENCIA WHERE nOcorencia='" + nOcorrencia.Text + "' ";

SqlCommand comando = new SqlCommand(sql, sqlConn);

sqlConn.Open();

SqlDataReader ler = comando.ExecuteReader();

if (ler.Read() == true)

{

numeracao.Text = ler["numeracao"].ToString();

motivo.Text = ler["motivoOcorencia"].ToString();

motorista.Text = ler["motorista"].ToString();

registroMot.Text = ler["registroMot"].ToString();

descricao.Text = ler["descricaoMotivo"].ToString();

data.Text = ler["data"].ToString();

}

else

{

MessageBox.Show("Registro não Cadastrado");

}

sqlConn.Close();

}

private void ConsultarOco\_Load(object sender, EventArgs e)

{

}

}

}

**- Código Tela Consultar Fluxo:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using Honibus.dbHONIBUSDataSetTableAdapters;

namespace Honibus

{

public partial class ConsultarFluxo : Form

{

SqlConnection sqlConn = new SqlConnection(@"Data Source=ETEC-PC;Initial Catalog=dbHONIBUS;Integrated Security=True");

private string \_Sql = string.Empty;

public ConsultarFluxo()

{

InitializeComponent();

}

private void consultasToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

ConsultarFluxo Form2 = new ConsultarFluxo();

Consultas newForm3 = new Consultas();

newForm3.ShowDialog();

}

private void fluxoDoDiaToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

ConsultarFluxo Form2 = new ConsultarFluxo();

Fluxo newForm4 = new Fluxo();

newForm4.ShowDialog();

}

private void ocorrênciasToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

ConsultarFluxo Form2 = new ConsultarFluxo();

Ocorrencias newForm5 = new Ocorrencias();

newForm5.ShowDialog();

}

private void cadastroÔnibusToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

ConsultarFluxo Form2 = new ConsultarFluxo();

CadastroOni newForm6 = new CadastroOni();

newForm6.ShowDialog();

}

private void cadastroToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

ConsultarFluxo Form2 = new ConsultarFluxo();

CadastroMoto newForm7 = new CadastroMoto();

newForm7.ShowDialog();

}

private void sairToolStripMenuItem\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void ConsultarFluxo\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dbHONIBUSDataSet.tbFLUXO' table. You can move, or remove it, as needed.

this.tbFLUXOTableAdapter.Fill(this.dbHONIBUSDataSet.tbFLUXO);

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

}

private void dateTimePicker1\_ValueChanged(object sender, EventArgs e)

{

}

private void button1\_Click\_1(object sender, EventArgs e)

{

var data = dateTimePicker\_pesquisar.Value.Date;

sqlConn.Open();

SqlCommand sqlCommand = sqlConn.CreateCommand();

sqlCommand.CommandText = ("select chegada, saida, data, registroMot, numeracao from tbFLUXO WHERE convert(date, data) = @data ");

SqlParameter parameterdata = new SqlParameter()

{

ParameterName = "@data",

SqlDbType = SqlDbType.DateTime,

Direction = ParameterDirection.Input,

Value = data

};

sqlCommand.Parameters.Add(parameterdata);

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand);

DataTable dataTable = new DataTable();

sqlDataAdapter.Fill(dataTable);

sqlCommand.ExecuteNonQuery();

dataGridView1.DataSource = dataTable;

sqlConn.Close();

}

}

}

**- Código Tela Cadastro/Motorista:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Honibus

{

public partial class CadastroMoto : Form

{

SqlConnection sqlConn = null;

private string strConn = @"Data Source=ETEC-PC;Initial Catalog=dbHONIBUS;Integrated Security=True";

private string \_Sql = string.Empty;

public CadastroMoto()

{

InitializeComponent();

}

private void sairToolStripMenuItem\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void Form7\_Load(object sender, EventArgs e)

{

Ativo.Checked = true;

Random nAleatorio = new Random();

textBox2.Text = Convert.ToString(nAleatorio.Next(0, 100));

}

private void consultasToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

CadastroMoto Form2 = new CadastroMoto();

Consultas newForm3 = new Consultas();

newForm3.ShowDialog();

}

private void fluxoDoDiaToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

CadastroMoto Form2 = new CadastroMoto();

Fluxo newForm4 = new Fluxo();

newForm4.ShowDialog();

}

private void ocorrênciasToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

CadastroMoto Form2 = new CadastroMoto();

Ocorrencias newForm5 = new Ocorrencias();

newForm5.ShowDialog();

}

private void cadastroÔnibusToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

CadastroMoto Form2 = new CadastroMoto();

CadastroOni newForm6 = new CadastroOni();

newForm6.ShowDialog();

}

private void cadastroToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

CadastroMoto Form2 = new CadastroMoto();

CadastroMoto newForm7 = new CadastroMoto();

newForm7.ShowDialog();

}

private void button1\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "" || textBox2.Text == "" || textBox2.Text == "" || textBox3.Text == "" || textBox4.Text == "" || maskedTextBox1.Text == "" || textBox8.Text == "")

{

MessageBox.Show("Preencha Campo(os) Vazio(os)");

}

else

{

\_Sql = "INSERT INTO tbMOTORISTA (nomeMotorista, registroMot, habilitacao, cpf, dataAdm, turno) VALUES (@nomeMotorista, @registroMot, @habilitacao, @cpf, @dataAdm, @turno) ";

sqlConn = new SqlConnection(strConn);

SqlCommand comando = new SqlCommand(\_Sql, sqlConn);

comando.Parameters.Add("@nomeMotorista", SqlDbType.VarChar).Value = textBox1.Text;

comando.Parameters.Add("@registroMot", SqlDbType.Int).Value = Convert.ToInt32(textBox2.Text);

comando.Parameters.Add("@habilitacao", SqlDbType.Decimal).Value = Convert.ToInt32(textBox3.Text);

comando.Parameters.Add("@cpf", SqlDbType.Decimal).Value = Convert.ToInt32(textBox4.Text);

comando.Parameters.Add("@dataAdm", SqlDbType.Date).Value = maskedTextBox1.Text;

comando.Parameters.Add("@turno", SqlDbType.VarChar).Value = textBox8.Text;

if (Ativo.Checked)

{

comando.Parameters.Add("@situacao", SqlDbType.VarChar, 5).Value = "Ativo";

}

else

{

comando.Parameters.Add("@situacao", SqlDbType.VarChar, 7).Value = "Inativo";

}

try

{

sqlConn.Open();

comando.ExecuteNonQuery();

MessageBox.Show("Cadastro efetuado com Sucesso");

}

catch (Exception)

{

MessageBox.Show("Cadastro já existente ou Informações Incorretas");

}

finally

{

sqlConn.Close();

}

}

}

}

}

**- Código Tela Cadastro/Ônibus:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Honibus

{

public partial class CadastroOni : Form

{

SqlConnection sqlConn = null;

private string strConn = @"Data Source=ETEC-PC;Initial Catalog=dbHONIBUS;Integrated Security=True";

private string \_Sql = string.Empty;

public CadastroOni()

{

InitializeComponent();

}

private void Form6\_Load(object sender, EventArgs e)

{

Ativo.Checked = true;

}

private void sairToolStripMenuItem\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void consultasToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

CadastroOni Form2 = new CadastroOni();

Consultas newForm3 = new Consultas();

newForm3.ShowDialog();

}

private void fluxoDoDiaToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

CadastroOni Form2 = new CadastroOni();

Fluxo newForm4 = new Fluxo();

newForm4.ShowDialog();

}

private void ocorrênciasToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

CadastroOni Form2 = new CadastroOni();

Ocorrencias newForm5 = new Ocorrencias();

newForm5.ShowDialog();

}

private void cadastroÔnibusToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

CadastroOni Form2 = new CadastroOni();

CadastroOni newForm6 = new CadastroOni();

newForm6.ShowDialog();

}

private void cadastroToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

CadastroOni Form2 = new CadastroOni();

CadastroMoto newForm7 = new CadastroMoto();

newForm7.ShowDialog();

}

private void button1\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "" || textBox9.Text == "" || maskedTextBox1.Text == "" || textBox8.Text == "" || textBox4.Text == "" || textBox5.Text == "" || textBox6.Text == "" || textBox7.Text == "")

{

MessageBox.Show("Preencha Campo(os) Vazio(os)");

}

else

{

\_Sql = "INSERT INTO tbONIBUS (numeracao, fabricacao, placa, modelOnibus, situacao, motoristaUm, motoristaDois, periodoMotUm, periodoMotDois) VALUES (@numeracao, @fabricacao, @placa, @modelOnibus, @situacao, @motoristaUm, @motoristaDois, @periodoMotUm, @periodoMotDois) ";

sqlConn = new SqlConnection(strConn);

SqlCommand comando = new SqlCommand(\_Sql, sqlConn);

comando.Parameters.Add("@numeracao", SqlDbType.VarChar).Value = textBox1.Text;

comando.Parameters.Add("@fabricacao", SqlDbType.Date).Value = maskedTextBox1.Text;

comando.Parameters.Add("@placa", SqlDbType.VarChar).Value = textBox8.Text;

comando.Parameters.Add("@modelOnibus", SqlDbType.VarChar).Value = textBox9.Text;

comando.Parameters.Add("@motoristaUm", SqlDbType.VarChar).Value = textBox4.Text;

comando.Parameters.Add("@motoristaDois", SqlDbType.VarChar).Value = textBox5.Text;

comando.Parameters.Add("@periodoMotUm", SqlDbType.VarChar).Value = textBox6.Text;

comando.Parameters.Add("@periodoMotDois", SqlDbType.VarChar).Value = textBox7.Text;

if (Ativo.Checked)

{

comando.Parameters.Add("@situacao", SqlDbType.VarChar, 5).Value = "Ativo";

}

else

{

comando.Parameters.Add("@situacao", SqlDbType.VarChar, 7).Value = "Inativo";

}

try

{

sqlConn.Open();

comando.ExecuteNonQuery();

MessageBox.Show("Cadastro efetuado com Sucesso");

}

catch (Exception)

{

MessageBox.Show("Cadastro já existente ou Informações Incorretas");

}

finally

{

sqlConn.Close();

}

}

}

}

}

**- Código Tela Fluxo do Dia:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using Honibus.dbHONIBUSDataSetTableAdapters;

namespace Honibus

{

public partial class Fluxo : Form

{

private SqlConnection sqlConn;

private SqlCommand comando;

private string \_Sql = string.Empty;

public Fluxo()

{

InitializeComponent();

sqlConn = new SqlConnection("Data Source=ETEC-PC;Initial Catalog=dbHONIBUS;Integrated Security=True");

}

private void sairToolStripMenuItem\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void fluxoDoDiaToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

Fluxo Form2 = new Fluxo();

Fluxo newForm4 = new Fluxo();

newForm4.ShowDialog();

}

private void ocorrênciasToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

Fluxo Form2 = new Fluxo();

Ocorrencias newForm5 = new Ocorrencias();

newForm5.ShowDialog();

}

private void consultasToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

Fluxo Form2 = new Fluxo();

Consultas newForm3 = new Consultas();

newForm3.ShowDialog();

}

private void cadastroÔnibusToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

Fluxo Form2 = new Fluxo();

CadastroOni newForm6 = new CadastroOni();

newForm6.ShowDialog();

}

private void cadastroToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

Fluxo Form2 = new Fluxo();

CadastroMoto newForm7 = new CadastroMoto();

newForm7.ShowDialog();

}

private void Fluxo\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dbHONIBUSDataSet.tbFLUXO' table. You can move, or remove it, as needed.

this.tbFLUXOTableAdapter.Fill(this.dbHONIBUSDataSet.tbFLUXO);

data.Text = DateTime.Now.ToString();

}

private void button1\_Click(object sender, EventArgs e)

{

string comando = "SELECT chegada, saida, data, registroMot, numeracao, confirmacao FROM tbFLUXO WHERE data='" + data.Text + "' ";

DataTable dttbFLUXO = new DataTable();

try

{

sqlConn.Open();

if (sqlConn.State == ConnectionState.Open)

{

SqlDataAdapter Adp = new SqlDataAdapter(comando, sqlConn);

Adp.Fill(dttbFLUXO);

dataFluxo.DataSource = dttbFLUXO;

}

else

{

MessageBox.Show("Falha na conexão");

}

}

catch

{

MessageBox.Show("ERRO NA LEITURA");

}

finally

{

sqlConn.Close();

}

}

private void button2\_Click\_1(object sender, EventArgs e)

{

button2.Enabled = false;

if (button2.Enabled == false)

{

button9.Enabled = false;

}

string strSQL = "update tbFLUXO set confirmacao='ok' where campo=1";

SqlCommand comando = new SqlCommand(strSQL, sqlConn);

try

{

sqlConn.Open();

comando.ExecuteNonQuery();

MessageBox.Show("Confirmado!");

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message);

}

finally

{

sqlConn.Close();

}

}

private void button9\_Click(object sender, EventArgs e)

{

string strSQL = "update tbFLUXO set confirmacao='não confirmado' where campo=1";

SqlCommand comando = new SqlCommand(strSQL, sqlConn);

try

{

sqlConn.Open();

comando.ExecuteNonQuery();

DialogResult confirm = MessageBox.Show("Adicionar Ocorrência?", "", MessageBoxButtons.OK, MessageBoxIcon.Exclamation,MessageBoxDefaultButton.Button2);

if (confirm.ToString().ToUpper() == "OK")

{

this.Visible = false;

Fluxo Form2 = new Fluxo();

Ocorrencias newForm5 = new Ocorrencias();

newForm5.ShowDialog();

}

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message);

}

finally

{

sqlConn.Close();

}

}

private void button8\_Click(object sender, EventArgs e)

{

button8.Enabled = false;

if (button8.Enabled == false)

{

button7.Enabled = false;

}

string strSQL = "update tbFLUXO set confirmacao='ok' where campo=2";

SqlCommand comando = new SqlCommand(strSQL, sqlConn);

try

{

sqlConn.Open();

comando.ExecuteNonQuery();

MessageBox.Show("Confirmado!");

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message);

}

finally

{

sqlConn.Close();

}

}

private void button6\_Click(object sender, EventArgs e)

{

button6.Enabled = false;

if (button6.Enabled == false)

{

button5.Enabled = false;

}

string strSQL = "update tbFLUXO set confirmacao='ok' where campo=2";

SqlCommand comando = new SqlCommand(strSQL, sqlConn);

try

{

sqlConn.Open();

comando.ExecuteNonQuery();

MessageBox.Show("Confirmado!");

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message);

}

finally

{

sqlConn.Close();

}

}

private void button4\_Click(object sender, EventArgs e)

{

button4.Enabled = false;

if (button4.Enabled == false)

{

button3.Enabled = false;

}

string strSQL = "update tbFLUXO set confirmacao='ok' where campo=3";

SqlCommand comando = new SqlCommand(strSQL, sqlConn);

try

{

sqlConn.Open();

comando.ExecuteNonQuery();

MessageBox.Show("Confirmado!");

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message);

}

finally

{

sqlConn.Close();

}

}

private void button7\_Click(object sender, EventArgs e)

{

string strSQL = "update tbFLUXO set confirmacao='não confirmado' where campo=2";

SqlCommand comando = new SqlCommand(strSQL, sqlConn);

try

{

sqlConn.Open();

comando.ExecuteNonQuery();

DialogResult confirm = MessageBox.Show("Adicionar Ocorrência?", "", MessageBoxButtons.OK, MessageBoxIcon.Exclamation, MessageBoxDefaultButton.Button2);

if (confirm.ToString().ToUpper() == "OK")

{

this.Visible = false;

Fluxo Form2 = new Fluxo();

Ocorrencias newForm5 = new Ocorrencias();

newForm5.ShowDialog();

}

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message);

}

finally

{

sqlConn.Close();

}

}

private void button5\_Click(object sender, EventArgs e)

{

string strSQL = "update tbFLUXO set confirmacao='não confirmado' where campo=3";

SqlCommand comando = new SqlCommand(strSQL, sqlConn);

try

{

sqlConn.Open();

comando.ExecuteNonQuery();

DialogResult confirm = MessageBox.Show("Adicionar Ocorrência?", "", MessageBoxButtons.OK, MessageBoxIcon.Exclamation, MessageBoxDefaultButton.Button2);

if (confirm.ToString().ToUpper() == "OK")

{

this.Visible = false;

Fluxo Form2 = new Fluxo();

Ocorrencias newForm5 = new Ocorrencias();

newForm5.ShowDialog();

}

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message);

}

finally

{

sqlConn.Close();

}

}

private void button3\_Click(object sender, EventArgs e)

{

string strSQL = "update tbFLUXO set confirmacao='não confirmado' where campo=4";

SqlCommand comando = new SqlCommand(strSQL, sqlConn);

try

{

sqlConn.Open();

comando.ExecuteNonQuery();

DialogResult confirm = MessageBox.Show("Adicionar Ocorrência?", "", MessageBoxButtons.OK, MessageBoxIcon.Exclamation, MessageBoxDefaultButton.Button2);

if (confirm.ToString().ToUpper() == "OK")

{

this.Visible = false;

Fluxo Form2 = new Fluxo();

Ocorrencias newForm5 = new Ocorrencias();

newForm5.ShowDialog();

}

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message);

}

finally

{

sqlConn.Close();

}

}

}

}

**- Código Tela Ocorrências:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace Honibus

{

public partial class Ocorrencias : Form

{

SqlConnection sqlConn = null;

private string strConn = @"Data Source=ETEC-PC;Initial Catalog=dbHONIBUS;Integrated Security=True";

private string \_Sql = string.Empty;

public Ocorrencias()

{

InitializeComponent();

}

private void sairToolStripMenuItem\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void fluxoDoDiaToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

Ocorrencias Form2 = new Ocorrencias();

Fluxo newForm4 = new Fluxo();

newForm4.ShowDialog();

}

private void consultasToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

Ocorrencias Form2 = new Ocorrencias();

Consultas newForm3 = new Consultas();

newForm3.ShowDialog();

}

private void ocorrênciasToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

Ocorrencias Form2 = new Ocorrencias();

Ocorrencias newForm5 = new Ocorrencias();

newForm5.ShowDialog();

}

private void cadastroÔnibusToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

Ocorrencias Form2 = new Ocorrencias();

CadastroOni newForm6 = new CadastroOni();

newForm6.ShowDialog();

}

private void cadastroToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Visible = false;

Ocorrencias Form2 = new Ocorrencias();

CadastroMoto newForm7 = new CadastroMoto();

newForm7.ShowDialog();

}

private void button1\_Click(object sender, EventArgs e)

{

if (nome.Text == "" || textBox4.Text == "" || textBox3.Text == "")

{

MessageBox.Show("Preencha Campo(os) Vazio(os)");

}

else

{

\_Sql = "INSERT INTO tbOCORENCIA (nOcorencia, motivoOcorencia, descricaoMotivo, motorista, registroMot, numeracao, data ) VALUES (@nOcorencia, @motivoOcorencia, @descricaoMotivo, @motorista, @registroMot, @numeracao, @data ) ";

sqlConn = new SqlConnection(strConn);

SqlCommand comando = new SqlCommand(\_Sql, sqlConn);

comando.Parameters.Add("@nOcorencia", SqlDbType.Int).Value = Convert.ToInt32(textBox4.Text);

comando.Parameters.Add("@descricaoMotivo", SqlDbType.VarChar).Value = textBox3.Text;

comando.Parameters.Add("@motorista", SqlDbType.VarChar).Value = nome.Text;

comando.Parameters.Add("@registroMot", SqlDbType.VarChar).Value = Convert.ToInt32(registro1.Text);

comando.Parameters.Add("@numeracao", SqlDbType.VarChar).Value = Convert.ToInt32(numeracao.Text);

comando.Parameters.Add("@data", SqlDbType.DateTime).Value = Convert.ToDateTime(data.Text);

if (acidente.Checked)

{

comando.Parameters.Add("@motivoOcorencia", SqlDbType.VarChar, 8).Value = "Acidente";

}

else if (falha.Checked)

{

comando.Parameters.Add("@motivoOcorencia", SqlDbType.VarChar, 5).Value = "Falha";

}

else

{

comando.Parameters.Add("@motivoOcorencia", SqlDbType.VarChar, 6).Value = "Outros";

}

try

{

sqlConn.Open();

comando.ExecuteNonQuery();

MessageBox.Show("Cadastro efetuado com Sucesso");

}

catch (Exception)

{

MessageBox.Show("Numero de ocorrencia invalido ou ja cadastrado");

}

finally

{

sqlConn.Close();

}

}

}

private void Ocorrencias\_Load(object sender, EventArgs e)

{

acidente.Checked = true;

data.Text = DateTime.Now.ToString();

Random nAleatorio = new Random();

textBox4.Text = Convert.ToString(nAleatorio.Next(0, 100));

}

private void button2\_Click(object sender, EventArgs e)

{

Pesquisa form = new Pesquisa();

form.ShowDialog();

}

}

}

# - Código Tela Pesquisa:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using Honibus.dbHONIBUSDataSetTableAdapters;

namespace Honibus

{

public partial class Pesquisa : Form

{

private SqlConnection sqlConn;

private SqlDataAdapter Adp;

public Pesquisa()

{

InitializeComponent();

sqlConn = new SqlConnection("Data Source=ETEC-PC;Initial Catalog=dbHONIBUS;Integrated Security=True");

}

private void carregar\_CellClick(object sender, DataGridViewCellEventArgs e)

{

Ocorrencias escolha = new Ocorrencias();

escolha.registro1.Text = carregar[1, carregar.CurrentRow.Index].Value.ToString();

escolha.nome.Text = carregar[0, carregar.CurrentRow.Index].Value.ToString();

escolha.numeracao.Text = carregar[7, carregar.CurrentRow.Index].Value.ToString();

escolha.Show();

}

private void button1\_Click(object sender, EventArgs e)

{

if (nome.Checked == true)

{

string comando = "SELECT nomeMotorista ,registroMot ,cpf ,turno, habilitacao , dataAdm , situacao , numeracao FROM tbMOTORISTA WHERE nomeMotorista like '%" + textBox1.Text + "%'";

DataTable dttbMOTORISTA = new DataTable();

try

{

sqlConn.Open();

if (sqlConn.State == ConnectionState.Open)

{

SqlDataAdapter Adp = new SqlDataAdapter(comando, sqlConn);

Adp.Fill(dttbMOTORISTA);

carregar.DataSource = dttbMOTORISTA;

}

else

{

MessageBox.Show("Nome Incorreto");

}

}

catch

{

MessageBox.Show("ERRO NA LEITURA");

}

finally

{

sqlConn.Close();

}

}

else

{

string comando = "SELECT nomeMotorista ,registroMot ,cpf ,turno, habilitacao , dataAdm , situacao , numeracao FROM tbMOTORISTA WHERE registroMot like '%" + textBox1.Text + "%'";

DataTable dttbMOTORISTA = new DataTable();

try

{

sqlConn.Open();

if (sqlConn.State == ConnectionState.Open)

{

SqlDataAdapter Adp = new SqlDataAdapter(comando, sqlConn);

Adp.Fill(dttbMOTORISTA);

carregar.DataSource = dttbMOTORISTA;

}

else

{

MessageBox.Show("Registro Incorreto");

}

}

catch

{

MessageBox.Show("ERRO NA LEITURA");

}

finally

{

sqlConn.Close();

}

}

}

private void Pesquisa\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dbHONIBUSDataSet1.tbOCORENCIA' table. You can move, or remove it, as needed.

this.tbOCORENCIATableAdapter.Fill(this.dbHONIBUSDataSet1.tbOCORENCIA);

// TODO: This line of code loads data into the 'dbHONIBUSDataSet1.tbMOTORISTA' table. You can move, or remove it, as needed.

this.tbMOTORISTATableAdapter.Fill(this.dbHONIBUSDataSet1.tbMOTORISTA);

// TODO: This line of code loads data into the 'dbHONIBUSDataSet.tbMOTORISTA' table. You can move, or remove it, as needed.

this.tbMOTORISTATableAdapter.Fill(this.dbHONIBUSDataSet.tbMOTORISTA);

nome.Checked = true;

}

}

}

**- Código Tela Grade:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using Honibus.dbHONIBUSDataSetTableAdapters;

namespace Honibus

{

public partial class Grade : Form

{

static SqlConnection sqlConn = new SqlConnection("Data Source=ETEC-PC;Initial Catalog=dbHONIBUS;Integrated Security=True");

public Grade()

{

InitializeComponent();

}

private void Grade\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'dbHONIBUSDataSet.tbFLUXO' table. You can move, or remove it, as needed.

this.tbFLUXOTableAdapter.Fill(this.dbHONIBUSDataSet.tbFLUXO);

}

private void Buscar\_Click(object sender, EventArgs e)

{

string comando = "SELECT chegada, saida, data, registroMot, numeracao, confirmacao FROM tbFLUXO WHERE registroMot='" + textBox1.Text + "' ";

DataTable dttbFLUXO = new DataTable();

try

{

sqlConn.Open();

if (sqlConn.State == ConnectionState.Open)

{

SqlDataAdapter Adp = new SqlDataAdapter(comando, sqlConn);

Adp.Fill(dttbFLUXO);

dataGridView1.DataSource = dttbFLUXO;

}

else

{

MessageBox.Show("Falha na conexão");

}

}

catch

{

MessageBox.Show("ERRO NA LEITURA");

}

finally

{

sqlConn.Close();

}

}

public Grade(string texto)

{

InitializeComponent();

textBox1.Text = texto;

}

}

}

}