

**UNIVERSIDADE FEDERAL DO PARANÁ**

**AUGUSTO HENRIQUE MACIEL SILVA**

**SISTEMA DE APOIO PARA TOMADA DE DECISÃO EM LOGÍSTICA FLORESTAL**

**Curitiba, 2020**

Augusto Henrique Maciel Silva

## SISTEMA DE APOIO PARA TOMADA DE DECISÃO EM LOGÍSTICA FLORESTAL

Trabalho de Conclusão de Curso apresentado ao Programa de Educação Continuada em Ciências Agrárias do Setor de Ciências Agrárias da Universidade Federal do Paraná como requisito parcial à obtenção do título de Especialista MBA em Gestão Florestal.

Professor Orientador: Dr. Julio Eduardo Arce

Curitiba, 2020

## RESUMO

O transporte florestal é um dos principais componentes de custos da madeira nas indústrias e é diretamente influenciado pelas rotas de transporte estabelecidas entre dois pontos. A determinação de rotas adequadas de transporte de madeira é fundamental para a entrega da matéria-prima na indústria com o menor custo, melhorando os resultados financeiros das empresas de base florestal. Entretanto, a logística florestal é bastante dinâmica e a melhor rota entre dois pontos pode ser mutável de acordo com variações nas características das rotas. Para auxiliar na tomada de decisão quanto a rota mais adequada entre dois pontos, foi desenvolvido um sistema de otimização de rotas em Excel utilizando Visual Basic for Applications (VBA), que informa ao operador três rotas possíveis entre dois pontos, e qual é a mais adequada para determinada situação de acordo com o *input* selecionado. O sistema se mostrou eficaz e pode ser implementado no dia a dia de uma operação, desde que sejam realizadas algumas ampliações nas suas capacidades de geração de rotas dentro de uma malha viária.

Palavras-chave: transporte florestal, otimização, logística

## **ABSTRACT**

The forest transport is one of the wood main cost components of industries, and it is directly affected by transport routes established between two points. The determination of suitable wood transport routes is fundamental to deliver the raw material with less cost, improving the financial results of forest companies. However, forest logistics is very dynamic, and the best route between two points can be changeable according variations in its characteristics. To support the decision making related the most adequate rout between two points, it was developed an optimized system in Excel utilizing Visual Basic for Applications (VBA), which informs the operator three possible routes between two points, and which one is most suitable to a specific situation accordingly with selected input. The system proved to be effective, and can be implemented in the day-to-day operation, as long as some expansions are made in the capacity of generates routes within a road network.

Keywords: forest transport, optimization, logistics

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>5</b>
<b>2 REVISÃO DE LITERATURA .....</b>	<b>6</b>
2.1 SETOR FLORESTAL .....	6
2.2 LOGÍSTICA FLORESTAL .....	6
2.2.1 CUSTOS .....	7
2.3 SISTEMAS DE OTIMIZAÇÃO .....	8
<b>3 METODOLOGIA .....</b>	<b>9</b>
<b>4 RESULTADOS E DISCUSSÃO .....</b>	<b>12</b>
<b>5 CONCLUSÕES .....</b>	<b>27</b>
<b>REFERÊNCIAS.....</b>	<b>28</b>
<b>APÊNDICE.....</b>	<b>30</b>

## 1 INTRODUÇÃO

O transporte de produto florestal *in natura* é bastante dependente do modal rodoviário, tornando-o uma variável de grande importância no custo final da madeira posta fábrica para as indústrias. As componentes do custo logístico estão relacionadas a distância de transporte, tipo de estrada, custos com manutenção de estradas, e investimentos em obras de arte no trecho.

A definição dos custos de transporte florestal para cada área é uma tarefa bastante complexa e apresenta algumas variáveis que se alteram com certa facilidade e em um horizonte de curto prazo, principalmente aquelas que envolvem os custos com frota e os rendimentos operacionais que são relacionados com as características das estradas.

Conforme supracitado, a qualidade das estradas pode se alterar em um curto prazo, por exemplo, uma intensa chuva pode deteriorar determinada rota de transporte e gerar uma dúvida na tomada de decisão, se serão executados reparos na via atual ou se será utilizada uma rota alternativa.

Neste contexto, além da definição dos custos de logística serem complexos para serem calculados, também é mutável, e demanda uma quantidade de tempo considerável para a definição da rota mais adequada e com o melhor retorno financeiro para a operação. Desta forma, a utilização de um sistema otimizado para auxílio na tomada de decisão fornece rapidez e assertividade ao gestor da operação de logística, garantindo um menor custo no transporte florestal, ou uma escolha mais adequada para determinada situação pontual.

O objetivo deste trabalho é o desenvolvimento e teste de um sistema no Excel que gere rotas de transporte através do indicativo do ponto inicial e do ponto final dentro de uma malha viária cadastrada previamente. Além de gerar as rotas, o sistema deve ser capaz de informar custos, tempo de deslocamento e comprimento da rota conforme requisitado pelo usuário.

## 2 REVISÃO DE LITERATURA

### 2.1 SETOR FLORESTAL

O setor florestal tem se mostrado importante para a economia brasileira, tendo uma participação no ano de 2018 de 1,3% do Produto Interno Bruto (PIB) e 6,9% do PIB Industrial do país, participando com aproximadamente R\$ 86,6 bilhões, tendo demonstrado um crescimento de aproximadamente 12,3% ao ano no saldo da balança comercial, entre os anos de 2012 e 2018 (IBÁ, 2019)

No ano de 2018 foram produzidos aproximadamente 131 milhões de metros cúbicos de madeira em tora e cerca de 51,2 milhões metros cúbicos de madeira para lenha no Brasil (IBGE, 2020). O Brasil tem cerca de 9,85 milhões de hectares de florestas plantadas, tendo sua maior concentração em eucalipto, cerca de 75,2%, 20,6% em pinus, e o restante em outros gêneros florestais (NITAHARA, 2018).

O maior produtor de madeira em tora do Brasil no ano de 2018 foi o estado do Paraná, com uma produção de aproximadamente 32,3 milhões de metros cúbicos, seguido do estado de São Paulo, com uma produção de 21,2 milhões de metros cúbicos, e ocupando a terceira posição o estado do Mato Grosso do Sul, com uma produção de aproximadamente 18 milhões de metros cúbicos (IBGE, 2018).

A principal utilização da madeira de florestas plantadas no Brasil é a indústria de papel e celulose, em segundo outras indústrias de transformação, entre elas painéis reconstituídos e serrados, posteriormente a indústria de carvão vegetal, o setor de lenhas, e por fim o setor de produtos não madeireiros, como resinas e outros extrativos (IBGE, 2018).

### 2.2 LOGÍSTICA FLORESTAL

A logística florestal é primordial para a movimentação desta grande quantidade de madeira que é produzida no Brasil. De acordo com Machado et al. (2011) o transporte é responsável por movimentar a madeira e seus derivados entre o ponto A e o ponto B, sendo dividido em três métodos de transporte florestal: fluvial, rodoviário e ferroviário.

O processo de logística florestal engloba as atividades de carregamento da madeira na beira do talhão da propriedade, transporte desta madeira até a unidade

fábrica, pátio intermediário, ou até mesmo ao consumidor, nos casos de transporte de madeira em toras as quais não sofrerão transformação (PIANA, 2016).

No Brasil, aproximadamente 60% da carga que é movimentada tem seu transporte realizado através do modal rodoviário (CBIE, 2020). Estima-se que cerca de 85% da madeira é transportada através do modal rodoviário (STEIN; CARASCHI; GUERREIRO, 2013). A FIBRIA no ano de 2016, nas unidades de Três Lagoas, Jacaré e Aracruz transportou cerca de 81,2% da madeira via modal rodoviário, 17,4% via modal ferroviário, e 1,4% via modal fluvial.

### 2.2.1 CUSTOS

De acordo com Malinovski e Fenner (1986, p.68) o custo do transporte florestal é em média 40% do custo da madeira posta fábrica (apud Berger et al., 2003, p.54). No ano de 2015, o transporte florestal representou aproximadamente 42% do custo da madeira para a International Paper nas fábricas de Luiz Antônio/SP e Mogi Guaçu/SP, sendo que os demais custos foram de silvicultura (29%), colheita (19%) e administrativo (10%), e por fim, o transporte teve participação de aproximadamente 4,5% no custo de produção do papel (PIANA, 2016).

A madeira é um produto com baixo valor específico, o volume transportado é bastante alto em comparação ao valor da carga, tornando o custo do transporte florestal mais alto em comparação ao transporte de outros produtos (SEIXAS, 1992). Além do baixo valor específico, os implementos utilizados para o transporte florestal são bem específicos e não possibilitam a sua utilização para o transporte de outros produtos, tornando os custos maiores, já que não é possível diluir os custos de capital transportando outros produtos (MACHADO et al., 2011).

Os custos envolvidos no custo final do transporte florestal podem ser divididos em três categorias, os fixos, os indiretos e os variáveis. Os custos fixos são aqueles que envolvem o custo de capital, salário do motorista, depreciação do veículo, licenciamento e seguro. Os custos indiretos são as despesas administrativas e os custos variáveis são compostos por custos com peças de reposição, combustível, lubrificantes, lavagem e pneus (SAVI; CANEPPELE; OLIVEIRA, 2012).

Além dos custos supracitados, na logística florestal há os custos com construção e manutenção das estradas, sendo que a qualidade das estradas

também influencia no rendimento operacional, e conseqüentemente nos custos da operação (LOPES; MACHADO; SOUZA, 2011).

Um dos indicadores utilizados para medir a produção de um modo de transporte é o TKU (tonelada por quilômetro útil), geralmente expressa por reais por tonelada por quilômetro R\$/t.km (ANTA, 2020). No trabalho realizado por Alves et al. (2013) para calcular o custo de transporte florestal, o formato utilizado para quantificar o custo foi R\$/m<sup>3</sup>.km, seguindo a mesma lógica do TKU, mas utilizando m<sup>3</sup> como variável de referência.

### 2.3 SISTEMAS DE OTIMIZAÇÃO

As técnicas de otimização são fundamentais para prestar apoio na tomada de decisão no setor florestal e já são amplamente utilizadas no manejo, na colheita, na logística e na gestão florestal (RODRIGUEZ, 2014). A otimização envolve técnicas de programação matemáticas, entre elas a programação linear, programação inteira, programação dinâmica, simulação e PERT/COM (MACHADO et al., 2011).

As planilhas eletrônicas, entre elas a do Excel permitem a construção de diversas estruturas matemáticas, inclusive utilizando linguagens de programação VBA, possibilitando a criação de um ambiente para simulações e análises de cenários e que são adaptados para a determinada situação, porém de fácil alteração, permitindo uma flexibilidade a quem opera a planilha (RODRIGUEZ, 2014).

Berger et al. (2003) utilizaram técnicas de programação linear para realizar um estudo de otimização do planejamento de transporte florestal, no qual os resultados indicaram a possibilidade de um aumento de 22,7% da quantidade de madeira posta no pátio e uma redução de 18,33% no custo final da matéria-prima.

Em um trabalho proposto por Martins (2011), foi utilizado o Excel e programação em Visual Basic for Applications (VBA) para simular e otimizar o planejamento florestal de médio prazo, o qual se mostrou uma ferramenta bastante útil para a tomada de decisão, visto que propiciou a entrega de informações com qualidade aos gestores responsáveis pelas tomadas de decisão.

### 3 METODOLOGIA

Este trabalho foi desenvolvido em duas etapas, a primeira etapa consistiu na elaboração de uma planilha no Excel utilizando suas diversas funcionalidades, principalmente as macros através da linguagem de programação VBA. A segunda etapa consistiu no teste da planilha desenvolvida em uma situação hipotética de tomada de decisão na gestão da logística florestal.

Para a execução da primeira etapa foi desenvolvido uma aba principal onde foram alocadas figuras geométricas com textos, as quais são utilizadas para acionar as caixas de formulários, onde são colocados os *inputs* e conseqüentemente são informados os *outputs* do sistema.

Os formulários foram criados no próprio Excel, através do VBA, utilizando todos os recursos possíveis, tais como *Labels*, *TextBoxs*, *Buttons* e *ListBoxs*. O código dos formulários foram adequados para o recebimento dos *inputs* do usuário e alguns casos para a execução de alguns cálculos preliminares do sistema.

Após o desenvolvimento dos formulários, foram desenvolvidas as macros principais e secundárias. Estas macros recebem os *inputs* dos formulários e executam os cálculos para determinar as rotas, bem como suas características. Cada macro foi desenvolvida para executar uma atividade distinta com o objetivo final de determinar as rotas existentes e as rotas mais adequadas para determinada situação.

Após a criação das macros, formulários e montagem final da planilha, esta foi testada com variáveis hipotéticas, testando todas as funções apresentadas *a priori*. A malha viária utilizada para os testes pode ser observada na figura 1, bem como as suas características, nas Tabelas 1 e 2.

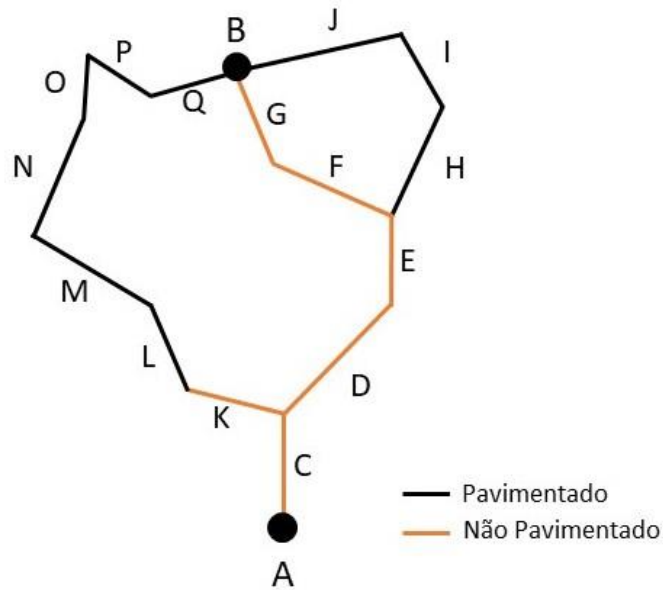


Figura 1 – Malha viária utilizada para validação do sistema

Trecho	Distância (km)		Tempo de trajeto (minutos)	TKU (R\$/t.km)
	Pavimentada	Não Pavimentada		
A	-	-	-	-
B	-	-	-	-
C	-	17	40,8	0,32
D	-	25	60,0	0,31
E	-	14	33,6	0,27
F	-	19	45,6	0,31
G	-	13	31,2	0,30
H	20	-	21,8	0,31
I	15	-	16,4	0,32
J	24	-	26,2	0,28
K	-	15	36,0	0,27
L	13	-	14,2	0,28
M	25	-	27,3	0,32
N	20	-	21,8	0,28
O	12	-	13,1	0,27
P	12	-	13,1	0,29
Q	11	-	12,0	0,32

Tabela 1 – Característica dos trechos da malha viária

Trecho	Conexão 1	Conexão 2	Conexão 3	Conexão 4	Conexão 5
A	C	-	-	-	-
B	G	Q	J	-	-
C	A	D	K	-	-
D	C	K	E	-	-
E	D	F	H	-	-
F	E	H	G	-	-
G	B	F	-	-	-
H	E	I	F	-	-
I	H	J	-	-	-
J	I	B	-	-	-
K	C	D	L	-	-
L	K	M	-	-	-
M	L	N	-	-	-
N	M	O	-	-	-
O	N	P	-	-	-
P	O	Q	-	-	-
Q	P	B	J	-	-

Tabela 2 – Trechos e suas conexões na malha viária

Para a realização do teste, primeiramente foram inseridos os trechos um a um dentro do banco de dados, utilizando um dos formulários desenvolvidos. Posteriormente alguns destes trechos foram deletados, para verificar a funcionalidade do formulário “Deletar Rotas”, e reinseridos ao banco de dados. E por fim, foi requisitado ao sistema quais são as possíveis rotas entro o trecho A e B através do formulário “Verificar Trechos”, e qual destas é a de menor distância, de menor custo e de menor tempo de deslocamento.

## 4 RESULTADOS E DISCUSSÃO

Os contatos do usuário com a planilha são efetuados através de uma única aba, denominada de Main (Figura 2), a qual apresenta os botões que permitem cadastrar novos trechos, deletar trechos e verificar as rotas possíveis entre dois pontos requeridos.

Abaixo de cada botão (retângulos com textos centralizados) há uma série de explicações e dicas para a melhor operação da planilha e obtenção das informações. Este sistema simplificado permite que usuários com pouca familiaridade com o Excel utilizem a ferramenta, principalmente por não criar uma interação direta com as células de cálculo e com as fórmulas, evitando que haja interferência externa ao sistema. Neste contexto, a única interferência realizada pelos usuários são os *inputs*.

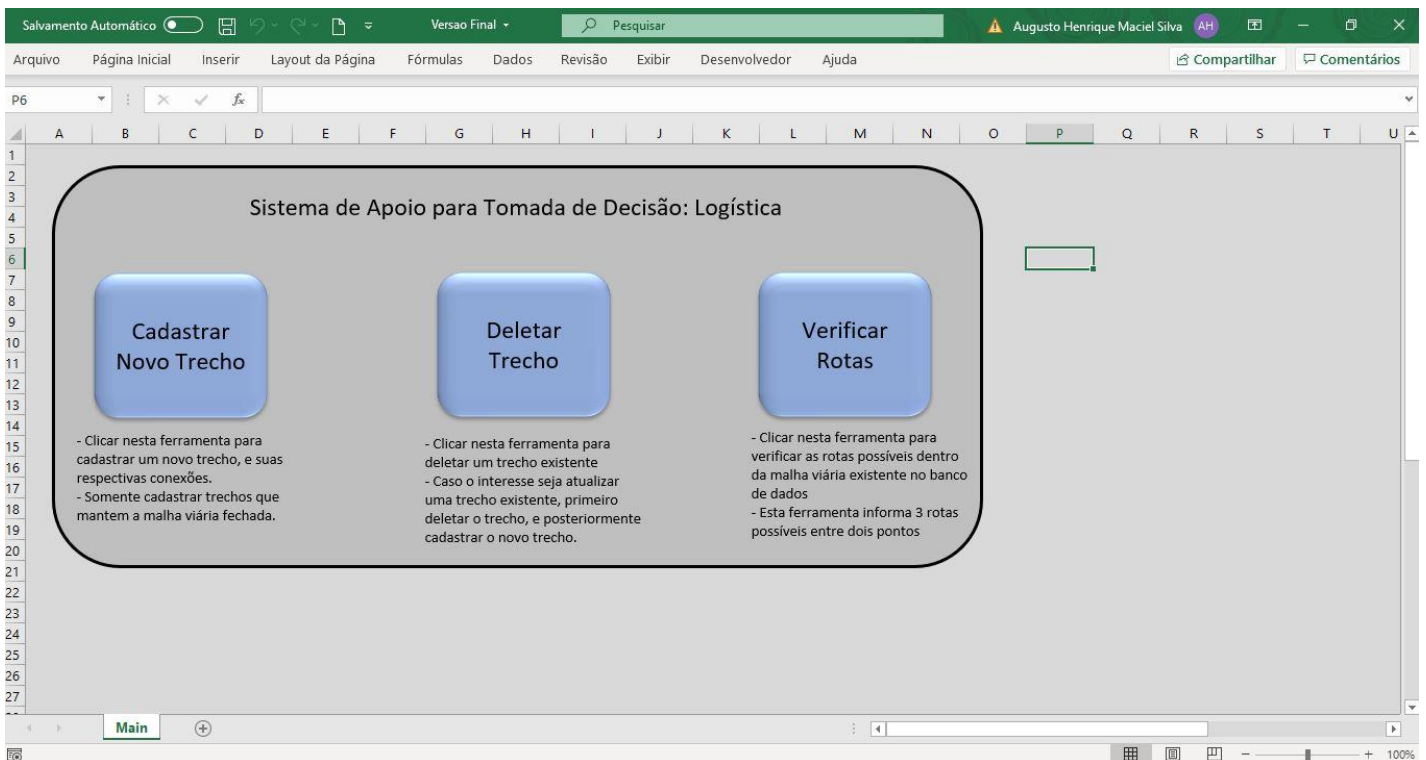


Figura 2 – Aba principal do sistema

Para o desenvolvimento da macro do primeiro botão (Cadastrar Novo Trecho), a lógica utilizada foi a seguinte: a macro armazena as informações do formulário, verifica linha a linha dentro do banco de dados, assim que encontra a primeira linha vazia a macro aloca as informações nesta linha, colocando cada informação em uma coluna pré-determinada, conforme a Figura 3.

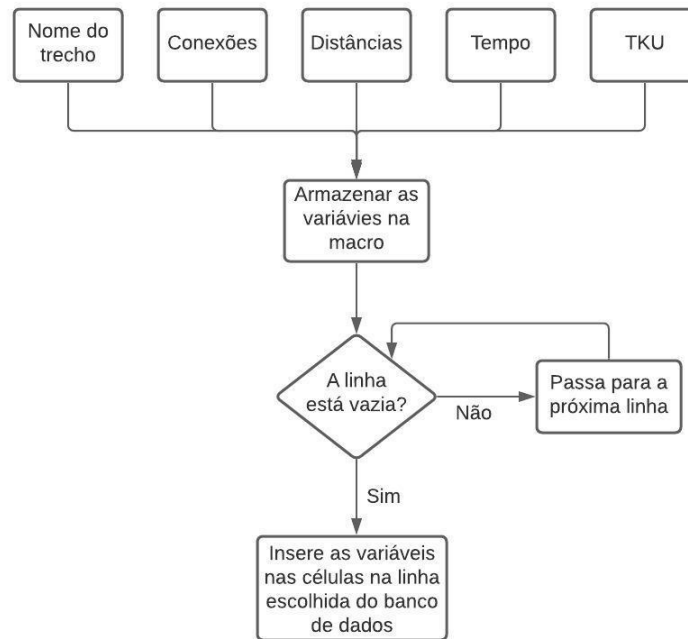


Figura 3 – Lógica da macro para cadastro das rotas

Ao clicar no botão “Cadastrar Novo Trecho” surge um formulário requisitando o nome do trecho (Figura 4), as suas respectivas conexões, a distância deste trecho em via pavimentada e via não pavimentada, o tempo estimado para transitar neste trecho em minutos e o custo em reais por tonelada por quilômetro. Importante ressaltar que ao inserir os números, a separação entre números inteiros e decimais deve ser realizada através do ponto final e não da vírgula.

Todas as rotas citadas pertencentes a malha viária foram inseridas no banco de dados através da ferramenta citada anteriormente sem qualquer falha de funcionamento da planilha. Além disso, esta atividade foi realizada de forma bastante ágil.

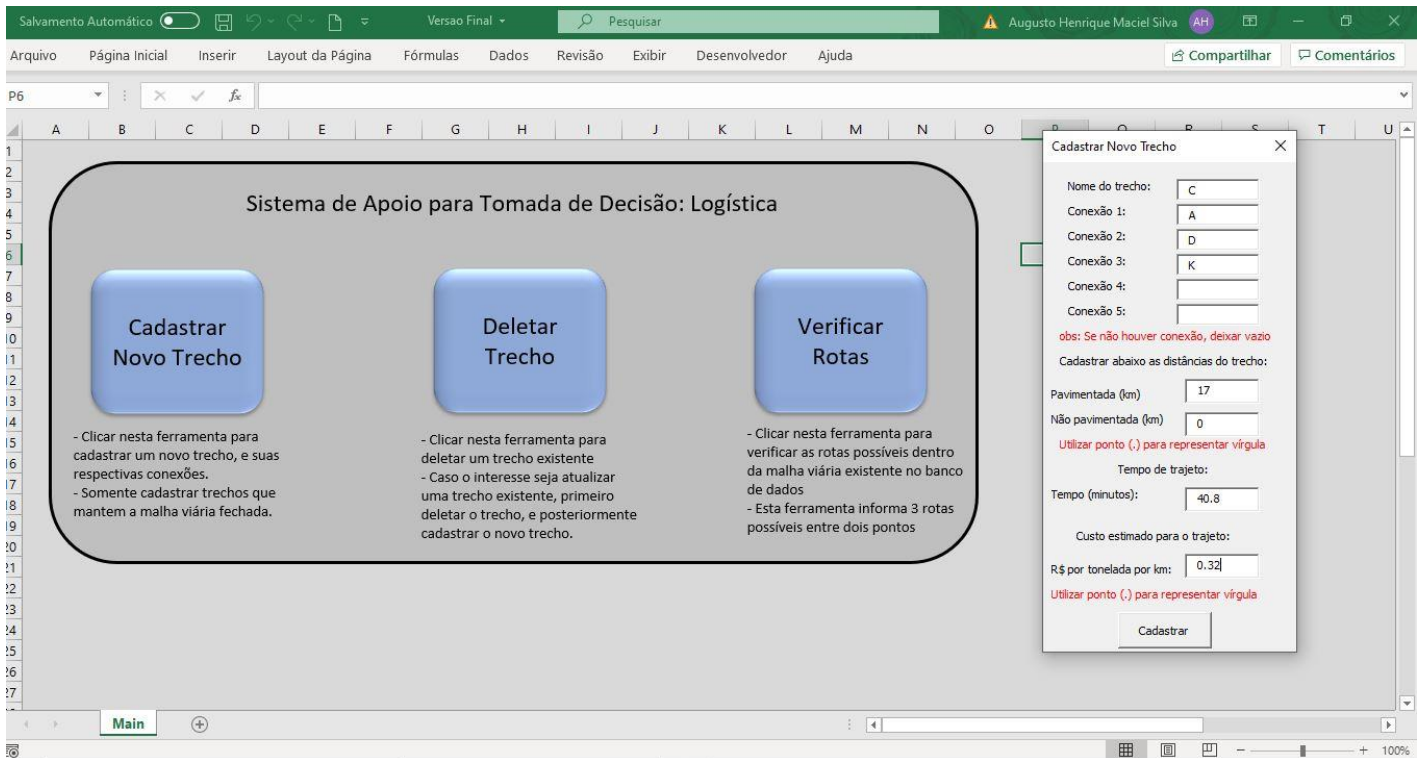


Figura 4 – Ferramenta Cadastrar Novo Trecho

O segundo botão (Deletar Trecho) é responsável por deletar um dos trechos. É apresentado ao usuário o nome de todos os trechos existentes no banco de dados através de uma caixa de listagem, onde é possível selecionar o trecho indesejado, e ao clicar no botão do formulário, o trecho é deletado.

A lógica desta macro é bastante simples, armazena o nome do trecho a ser eliminado e aciona uma macro secundária. A macro secundária procura o trecho dentro do banco de dados, ao localizá-lo deleta todas as informações pertencentes a este trecho dentro desta linha no banco de dados. A lógica utilizada pode ser observada na figura 5.

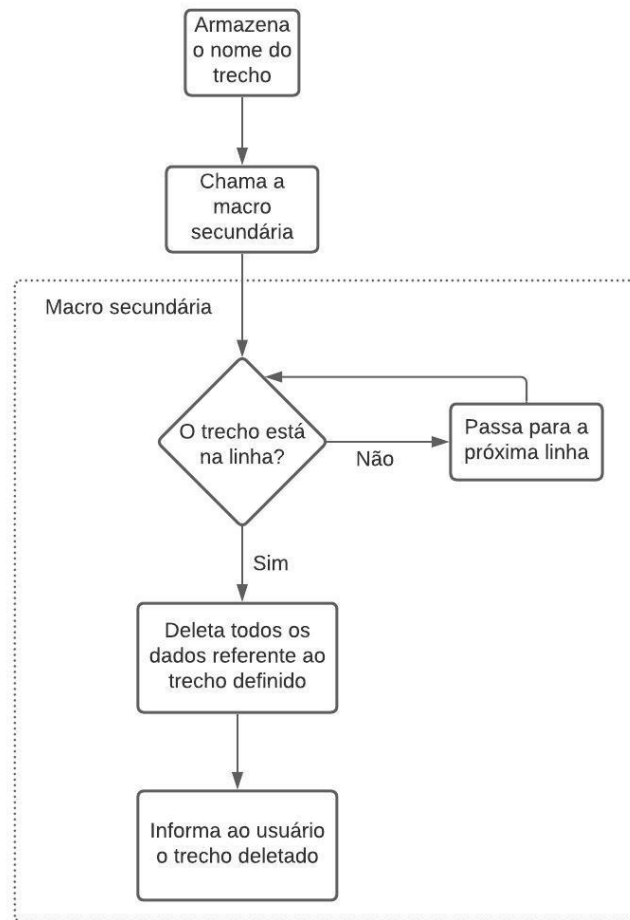


Figura 5 – Lógica utilizada para deletar um trecho do banco de dados

A ferramenta para deletar trechos (Figura 6) funcionou perfeitamente e se mostrou de fácil utilização por parte do usuário, já que é uma seleção simples de um determinado trecho seguido da confirmação de deletar o trecho através de um botão no formulário.

Esta ferramenta além de ser utilizada para deletar trechos existentes, também pode ser utilizada para verificar quais trechos já estão cadastrados na malha viária do banco de dados.

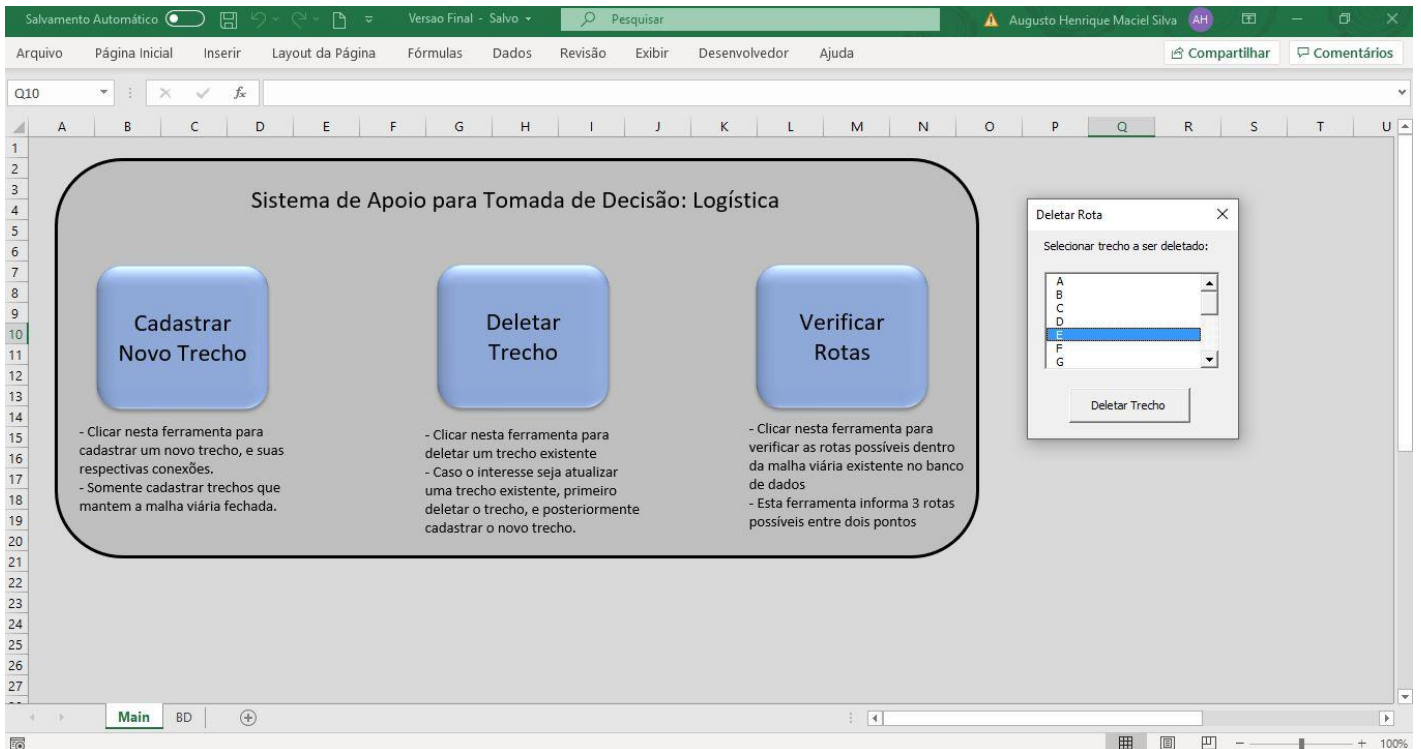


Figura 6 – Ferramenta Deletar Rota

A função para verificar as rotas possíveis dentro da malha viária foi desenvolvida utilizando diversas macros secundárias. Primeiramente foi utilizado um formulário para requisitar ao usuário qual é o ponto inicial e o ponto final da rota a ser estabelecida. O resultado das três rotas possíveis é informado ao usuário no mesmo formulário. Com as rotas identificadas, o usuário pode verificar qual das rotas é a de menor distância, qual é a mais rápida e qual é a de menor custo.

A lógica do formulário (Figura 7) é armazenar as informações do ponto inicial e do ponto final, inserir estas informações em uma célula específica da aba de apoio da planilha, chamar a macro secundária que cria as rotas através das conexões existentes entre os trechos. A macro secundária retorna as três rotas criadas, e estas são mostradas ao usuário no próprio formulário.

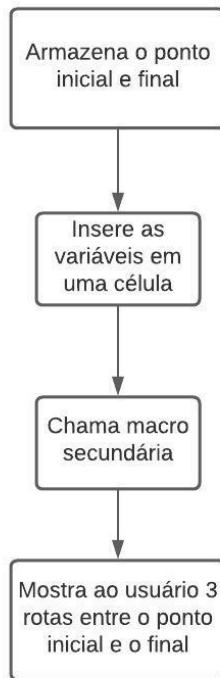


Figura 7 – Lógica formulário do cálculo das rotas

O formulário para verificar os trechos apareceu normalmente quando requisitado através do clique no botão “Verificar Rotas” da aba Main no Excel, conforme a figura 8.

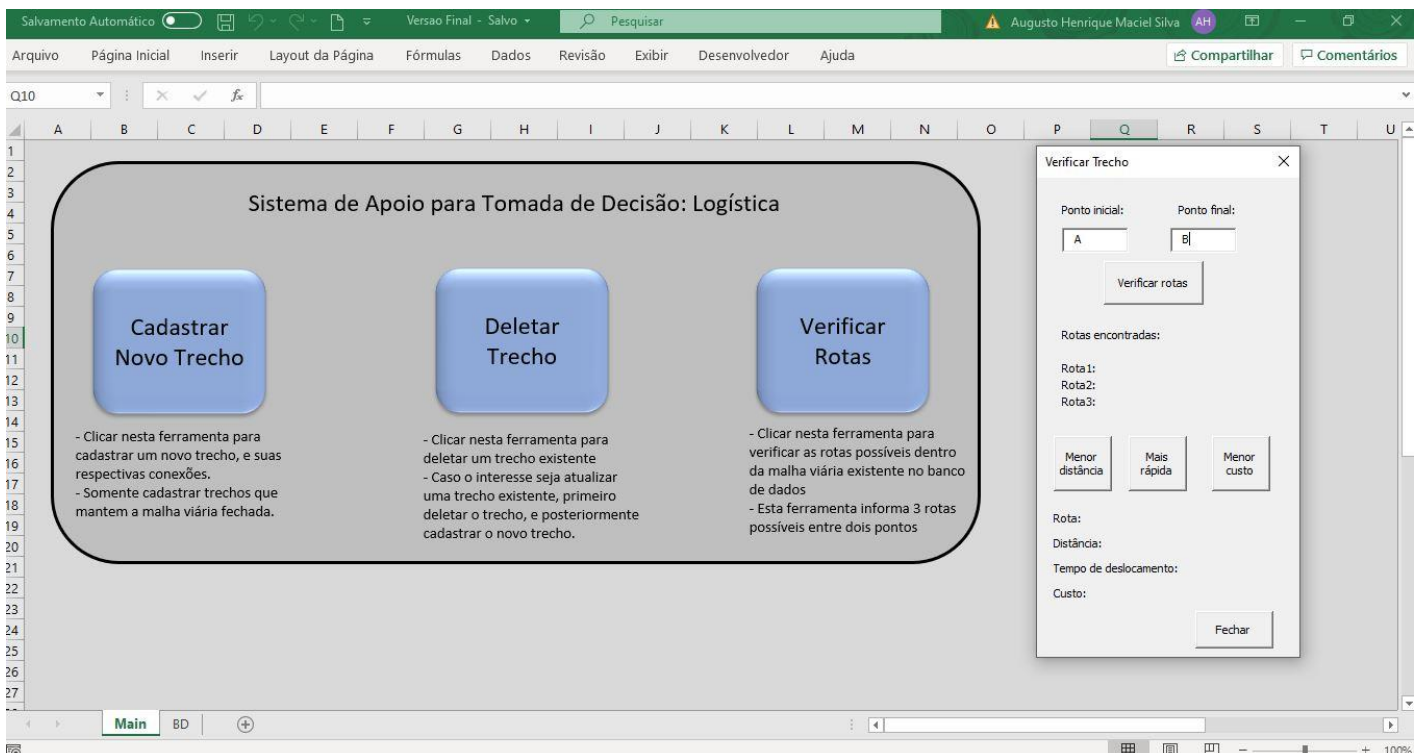


Figura 8 – Formulário Verificar Rotas

Ao ser requisitado o cálculo das rotas possíveis entre os dois pontos, o formulário aciona uma macro secundária a qual é dividida em quatro partes, e propicia a criação de três rotas distintas entre os pontos.

A primeira macro acionada é a macro principal deste sistema, que tem como objetivo chamar as outras três macros, uma de cada vez, e alocar as informações em três linhas distintas dentro da aba secundária da planilha.

A primeira função chamada é responsável por gerar a primeira rota entre o ponto inicial e o ponto final. Esta rota é definida através da composição entre as conexões entre o ponto inicial e o ponto final, evitando que um mesmo trecho seja utilizado duas vezes durante a composição da rota. A lógica envolvida na definição da primeira rota pode ser observada na figura 9.

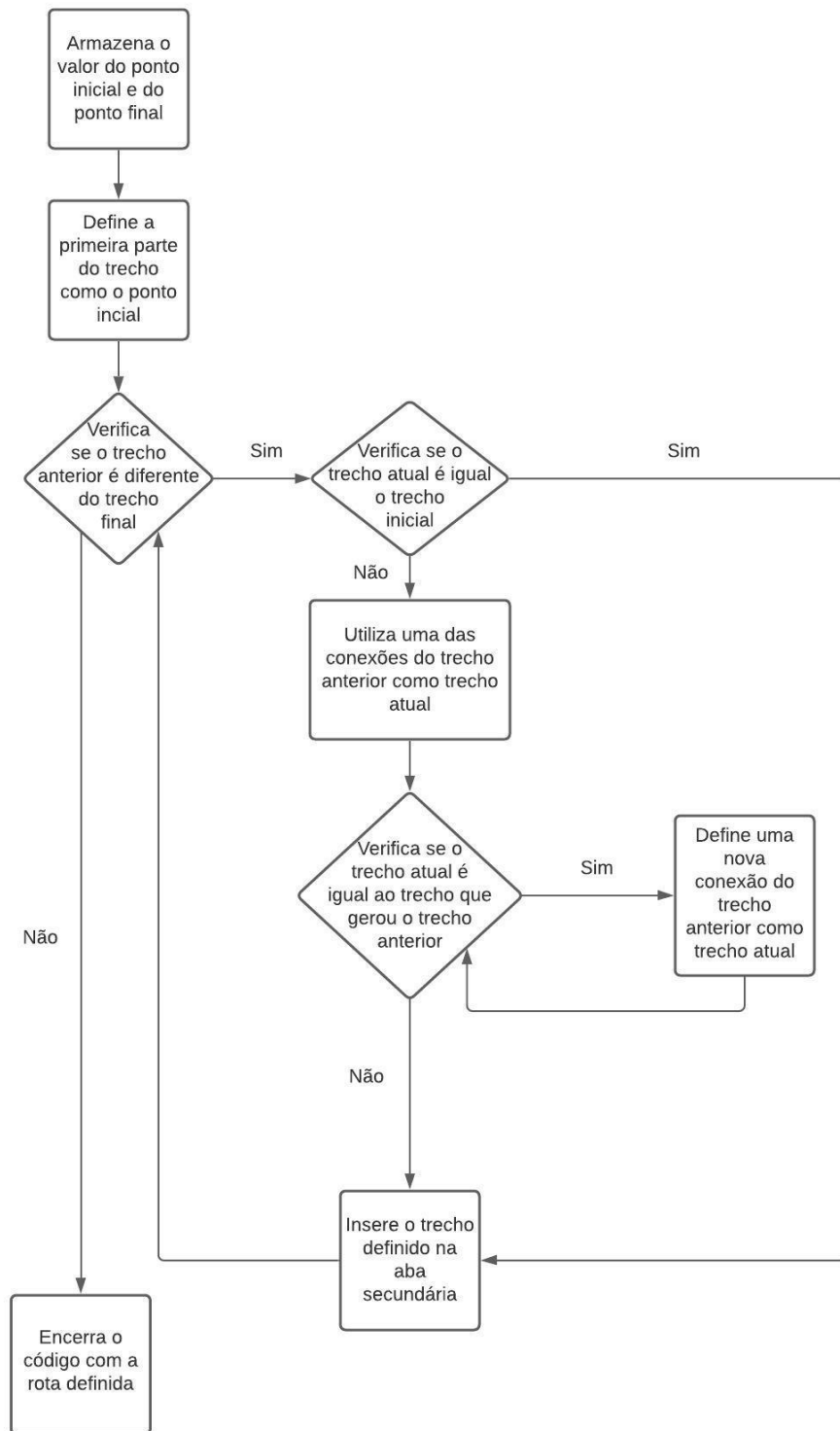


Figura 9 – Definição da primeira rota entre o ponto inicial e final

A função que define a segunda rota é similar à anterior, a diferença consiste em verificar se o trecho definido é igual ao da rota anterior, caso seja, verifica se é possível utilizar outro trecho no lugar, em casos afirmativos o trecho é alterado, e em

caso negativo é mantido o mesmo, e o código passa para o cálculo do próximo trecho, conforme a figura 10.

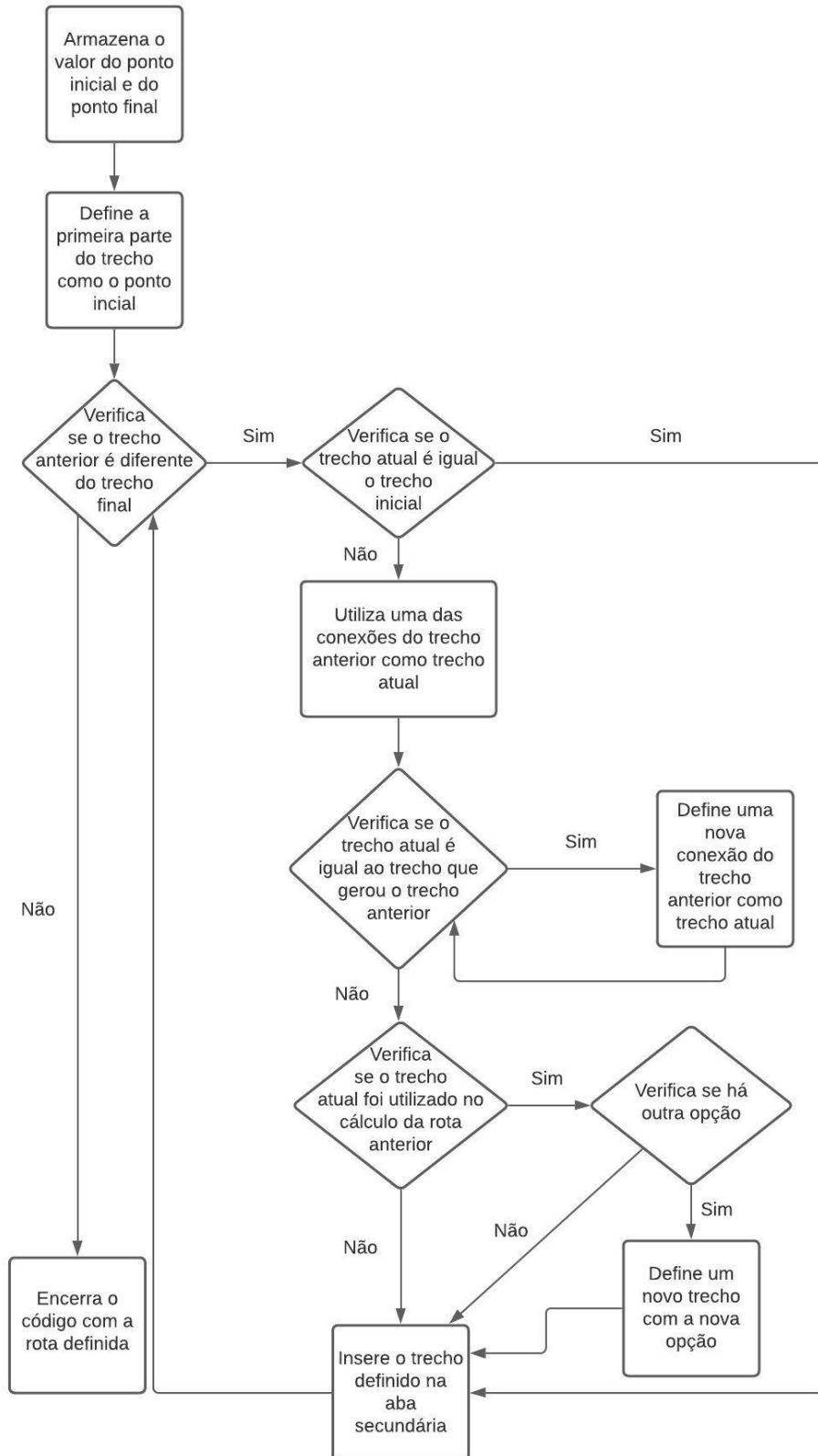


Figura 10 – Definição da segunda rota entre o ponto inicial e final

A terceira rota é definida através de uma função similar as anteriores, entretanto, nesta função é incrementado uma verificação a qual tem como objetivo analisar se o trecho calculado é similar as rotas 1 e 2, e caso seja, verificar se essa é a única opção, não sendo, o código define um novo trecho.

A lógica de programação que executa o cálculo da terceira rota entre dois pontos dentro da malha viária pode ser observada na figura 11.



Após as três rotas estarem definidas e informadas ao usuário no mesmo formulário em que elas foram requisitadas, o usuário tem a possibilidade de requisitar qual destas rotas é a de menor distância, qual é a mais rápida e qual é a de menor custo.

Ao selecionar qualquer uma das opções, o usuário irá requisitar o cálculo dos três itens de forma simultânea, entretanto, será aplicada uma condição em que as informações serão mostradas apenas do trecho que apresenta o melhor resultado para o cenário determinado, conforme a figura 12.



Figura 12 – Definição da melhor rota de acordo com o *input*

Ao requisitar ao formulário as rotas possíveis para deslocar-se do ponto A até o ponto B, o sistema informou que as rotas adequadas seriam “ACDEFGB”, “ACKLMNOPQB” e “ACDEHIJB”, conforme a figura 13. Importante salientar que o resultado foi equivalente a representação visual das rotas, já apresentado na figura 1 deste trabalho.

Neste contexto, também é importante ressaltar que o sistema em Excel criou esta rota através de interações que verificam o trecho que está sendo analisado e suas respectivas conexões. Por exemplo, em momento algum é informado ao programa que entre o ponto A e o ponto B existe o trecho E, ele é informado que o ponto A tem uma conexão com o trecho C, que por sua vez apresenta uma conexão com o trecho D e que este apresenta uma conexão com o trecho E e assim por diante até completar a rota.

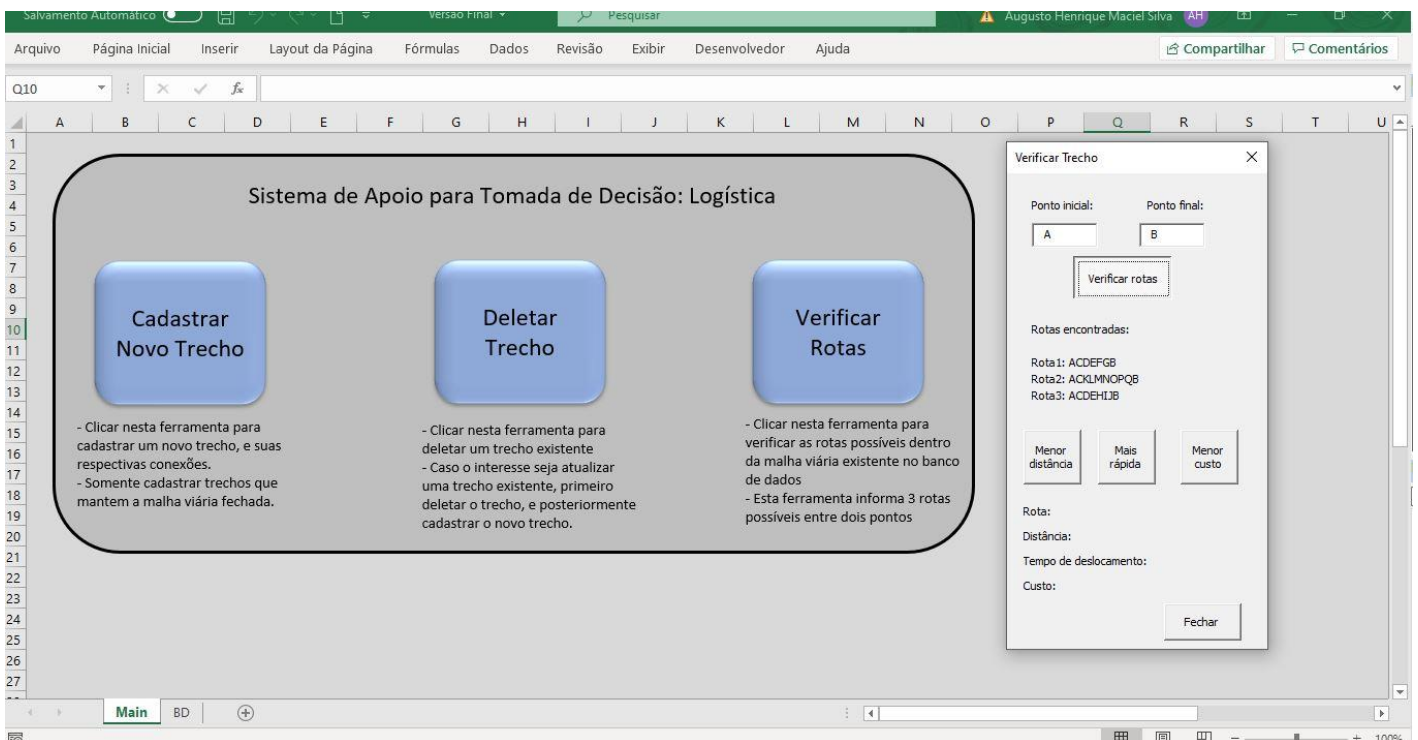


Figura 13 – Formulário Verificar Rotas

Até este ponto o sistema se mostrou confiável e gerou as informações de forma bastante ágil e assertiva. O sistema informou as três rotas possíveis entre os dois pontos em uma fração de segundos, demonstrando agilidade na obtenção das informações. Após a definição das rotas, foi requisitado ao sistema qual rota é a de menor distância, posteriormente qual é a mais rápida e por fim a de menor custo.

O *software* informou que a rota de menor distância é formada “ACDEFGB”, com uma distância de 88 quilômetros, um tempo de deslocamento de aproximadamente 211,2 minutos e um custo de R\$ 26,76 por tonelada transportada, conforme a Figura 14.

A rota definida como mais rápida é a rota “ACKLMNOPQB”, que apresenta um tempo de deslocamento de 178,3 minutos, uma distância de 125 quilômetros, e um custo aproximado de R\$ 36,97 por tonelada transportada.

Por fim, a rota de menor custo é “ACDEFGB”, a qual também é a de menor distância entre o ponto A e o ponto B. Com as premissas de cada rota informada previamente ao cadastrá-las no banco de dados, a rota “ACDEHIJB” não se classificou em primeiro em nos três cenários supracitados.

**Sistema de Apoio para Tomada de Decisão: Logística**

**Cadastrar Novo Trecho**

- Clicar nesta ferramenta para cadastrar um novo trecho, e suas respectivas conexões.
- Somente cadastrar trechos que mantem a malha viária fechada.

**Deletar Trecho**

- Clicar nesta ferramenta para deletar um trecho existente
- Caso o interesse seja atualizar uma trecho existente, primeiro deletar o trecho, e posteriormente cadastrar o novo trecho.

**Verificar Rotas**

- Clicar nesta ferramenta para verificar as rotas possíveis dentro da malha viária existente no banco de dados
- Esta ferramenta informa 3 rotas possíveis entre dois pontos

**Verificar Trecho**

Ponto inicial: A      Ponto final: B

Verificar rotas

Rotas encontradas:

Rota 1: ACDEFGB  
Rota 2: ACKLMNOPQB  
Rota 3: ACDEHIJB

Menor distância      Mais rápida      Menor custo

Rota: ACDEFGB  
Distância: 88 km  
Tempo de deslocamento: 211.2 minutos  
Custo: 26.76 R\$/t

Fechar

Figura 14 – Rota de menor distância entre o ponto A e o ponto B

Após os testes utilizando a malha viária hipotética criada, o sistema foi assertivo e gerou as informações de forma rápida, as quais em uma situação real seriam extremamente importantes para a tomada de decisão durante o dia a dia de uma operação de logística florestal.

Caso as variáveis e a malha viária utilizadas fossem reais, a decisão de transportar pela rota “ACDEFGB” representaria uma economia de aproximadamente 28% em relação ao transporte pela rota “ACKLMNOPQB”, mesmo a segunda sendo quase 16% mais rápida que a primeira.

Esta primeira versão para um sistema de otimização não apresentou erros durante a sua utilização, entretanto, é necessário utilizá-la de forma exaustiva para a

identificação de erros. Além disso, a ferramenta é limitada apresentação de 3 rotas dentro de uma malha viária fechada. Para ampliar o número de rotas calculadas é necessário melhorar significativamente o código base, o que tornaria o sistema bem mais atrativo.

## 5 CONCLUSÕES

O sistema desenvolvido para determinar as rotas em uma malha viária fechada se demonstrou bastante útil, já que foi capaz de determinar as rotas com assertividade e informar características importantes das rotas de transporte que são fundamentais para a tomada de decisão nas instâncias da operação.

A criação desta ferramenta dentro do Excel demonstra a capacidade deste *software* para a resolução de problemas, e principalmente, a sua versatilidade, pois é possível adaptar este sistema para as necessidades específicas de cada operação individualmente.

Por fim, este sistema demonstra ter potencial para a utilização na tomada de decisão de logística florestal, desde que haja melhorias pontuais, principalmente na ampliação da capacidade do banco de dados e da geração de rotas, permitindo a inserção de malhas viárias mais complexas.

## REFERÊNCIAS

ALVES, Rafael Tonetto; FIEDLER, Nilton Cesar; SILVA, Elizabeth Neire; LOPES, Eduardo da Silva; CARMO, Flávio Cipriano de Assis. Análise técnica e de custos do transporte de madeira com diferentes composições veiculares. **Revista Árvore**, Viçosa. v.37, n.5, p.897-904, 2013.

ANTA, Agência Nacional de Transportes Aquaviários. **TKU 2018**: TKU da navegação interior, de cabotagem e longo curso em vias interiores – 2018. Brasília, 2020.

BERGER, Ricardo; TIMOFEICZYK, Romano; CAMIERI, Celso; LACOWICZ, José Sawinski; BRASIL, Alexandre Anders. Minimização de custos de transporte florestal com a utilização da programação linear. **Revista Floresta**, Curitiba, v.30, n.1, p.53-62, jun 2003.

CBIE, Centro Brasileiro de Infraestrutura. **Como é feito o transporte de cargas no Brasil**. Abr 2020. Disponível em < <https://cbie.com.br/artigos/como-e-feito-o-transporte-de-cargas-no-brasil/>>. Acesso em: 5 nov 2020.

IBÁ, Indústria Brasileira de Árvores. **Relatório 2019**. São Paulo, 2019.

IBGE, Instituto Brasileiro de Geografia e Estatística. **Produção da Extração Vegetal e da Silvicultura - PEVS**: Principais resultados, 2020.

IBGE, Instituto Brasileiro de Geografia e Estatística. **Produção da Extração Vegetal e da Silvicultura – 2018**, Rio de Janeiro, v.33, p.1-8, 2018

LOPES, Eduardo da Silva; MACHADO, Carlos Cardoso; SOUZA, Amaury Paulo. Custos de estradas em florestas plantadas. **Revista da Madeira**, Curitiba, n.129, 2011.

MACHADO, Carlos Cardoso; LOPES, Eduardo da Silva; BIRRO, Mauro Henrique Batisa. MACHADO, Raiane Ribeiro. **Transporte Rodoviário Florestal**. 2.ed. Viçosa: Editora UFV, 2011.

MARTINS, Lorena Casagrande. **Sistema de simulação para planejamento florestal em médio prazo**. 2011, 50f. Trabalho de Conclusão de Curso (Especialização em Gestão Florestal) – Universidade Federal do Paraná, Curitiba, 2011.

NITAHARA, Akemi. IBGE: Brasil tem 9,85 milhões de hectares de florestas plantadas. Set 2018. Disponível em < <https://agenciabrasil.ebc.com.br/geral/noticia/2018-09/ibge-brasil-tem-985-milhoes-de-hectares-de-florestas-plantadas>>. Acesso em: 5 de nov 2020.

PIANA, Bruno Mariani. **Melhoria de Processos na Logística Florestal**: Projeto Lean Six Sigma Black Belt, mar 2016.

RODRIGUEZ, Luiz Carlos Estraviz. **Técnicas de otimização matemática para a gestão de florestas plantadas**. 1 ed. Piracicaba, ago 2014.

SAVI, Antonio Francisco; CANEPPELE, Fernando de Lima; OLIVEIRA, Marcos Ricardo Garcia de Oliveira. Custeio de diferentes tipos de transporte rodoviário no setor madeireiro de Itapeva. **Floresta e Ambiente**, v.19, n.2, p.147-154, 2012.

SEIXAS, F. **Uma metodologia de seleção e dimensionamento da frota de veículos rodoviários para o transporte principal de madeira**. 1992, 106f. Tese (Doutorado em Engenharia de Transportes) – Universidade de São Carlos, São Carlos, 1992.

STEIN, Gustavo Henrique; CARASCHI, José Cádio; GUERREIRO, Érico Daniel Ricardi. Análise operacional de carregamento florestal. **Revista da Madeira**, Curitiba, n.135, p.60-65, 2013.

## APÊNDICE

### 1 CADASTRAR TRECHO

```
Private Sub ToggleButton1_Click()
```

```
Dim trecho As String
Dim conex1 As String
Dim conex2 As String
Dim conex3 As String
Dim conex4 As String
Dim conex5 As String
Dim controle As Boolean
Dim x As Integer
Dim y As Integer
Dim range1 As range
Dim pavimentado As Double
Dim terra As Double
Dim minutos As Double
Dim tku As Double
```

```
trecho = TextBox1.Value
conex1 = TextBox2.Value
conex2 = TextBox3.Value
conex3 = TextBox4.Value
conex4 = TextBox5.Value
conex5 = TextBox6.Value
pavimentado = TextBox7.Value
terra = TextBox8.Value
minutos = TextBox9.Value
tku = TextBox10.Value
```

```
UserForm1.Hide
```

```
Sheets("BD").Select
```

```
Set range1 = range("B4:B20")
```

```
x = 4
```

```
y = 2
```

```
While controle = False
```

```
    If Cells(x, y).Value = "" Then
```

```
        Cells(x, y) = trecho
        Cells(x, y + 1) = conex1
        Cells(x, y + 2) = conex2
        Cells(x, y + 3) = conex3
        Cells(x, y + 4) = conex4
        Cells(x, y + 5) = conex5
        Cells(x, y + 7) = pavimentado
        Cells(x, y + 8) = terra
        Cells(x, y + 9) = minutos
        Cells(x, y + 10) = tku
```

```
        controle = True
```

```
    Else
```

```
        x = x + 1
```

```
        controle = False
```

```
    End If
```

```
Wend
```

```
Sheets("Main").Select
```

```
End Sub
```

## 2 DELETAR TRECHO

### 2.1 CÓDIGO DO FORMULÁRIO

```
Private Sub ToggleButton1_Click()  
    Dim z As String  
    z = ListBox1.Value  
    Call deletar(z)  
End Sub
```

### 2.2 CÓDIGO DA FUNÇÃO DELETAR

```
Sub deletar(x)  
    Dim trecho As String  
    Dim banco As range  
    Dim cell As range  
  
    trecho = x  
  
    Sheets("BD").Select  
  
    Set banco = range("B4:B20")  
  
    For Each cell In banco  
        If cell.Value = trecho Then  
            cell.ClearContents  
            cell.Offset(0, 1).ClearContents  
            cell.Offset(0, 2).ClearContents  
            cell.Offset(0, 3).ClearContents  
            cell.Offset(0, 4).ClearContents  
            cell.Offset(0, 5).ClearContents  
            cell.Offset(0, 7).ClearContents  
            cell.Offset(0, 8).ClearContents  
            cell.Offset(0, 9).ClearContents  
            cell.Offset(0, 10).ClearContents  
  
            End If  
  
        Next  
        Sheets("Main").Select  
  
        UserForm3.Hide  
        MsgBox ("Trecho " & trecho & " deletado")  
  
    End Sub
```

### 3 VERIFICAR ROTAS

#### 3.1 CÓDIGO DO FORMULÁRIO

```

Private Sub ToggleButton1_Click()

    Dim inicial As String
    Dim final As String
    Dim rota1 As String

    inicial = TextBox1.Value
    final = TextBox2.Value

    Sheets("BD").Select

    range("J24").Value = inicial
    range("K24").Value = final

    Call linhas

    rota1 = range("C32").Value
    rota2 = range("C33").Value
    rota3 = range("C34").Value

    Sheets("Main").Select

    Me.Label4.Caption = ("Rota1: " & rota1)
    Me.Label5.Caption = ("Rota2: " & rota2)
    Me.Label6.Caption = ("Rota3: " & rota3)

End Sub

Private Sub ToggleButton2_Click()

    UserForm2.Hide

End Sub

Private Sub ToggleButton3_Click()

    Dim nome As String
    Dim distanciafinal As Double
    Dim tempo As String
    Dim custo As String
    Dim distancia1 As Double
    Dim distancia2 As Double
    Dim distancia3 As Double
    Dim rota As String

    Call distancia
    Call tempo
    Call custo

    Sheets("BD").Select

    distancia1 = range("D32").Value

    distancia2 = range("D33").Value

    distancia3 = range("D34").Value

    If distancia1 < distancia2 And distancia1 < distancia3 Then

        distanciafinal = distancia1
        tempo = range("D32").Offset(0, 1).Value
        custo = range("D32").Offset(0, 2).Value
        rota = range("D32").Offset(0, -1).Value
    
```

```

End If

If distancia2 < distancia1 And distancia2 < distancia3 Then

    distanciafinal = distancia2
    tempo = range("D33").Offset(0, 1).Value
    custo = range("D33").Offset(0, 2).Value
    rota = range("D33").Offset(0, -1).Value

End If

If distancia3 < distancia2 And distancia3 < distancia1 Then

    distanciafinal = distancia3
    tempo = range("D34").Offset(0, 1).Value
    custo = range("D34").Offset(0, 2).Value
    rota = range("D34").Offset(0, -1).Value

End If

Me.Label7.Caption = ("Rota: " & rota)
Me.Label8.Caption = ("Distância: " & distanciafinal & " km")
Me.Label9.Caption = ("Tempo de deslocamento: " & tempo & " minutos")
Me.Label10.Caption = ("Custo: " & custo & " R$/t")

End Sub

Private Sub ToggleButton4_Click()

    Dim nome As String
    Dim tempofinal As Double
    Dim distancia As String
    Dim custo As String
    Dim tempo1 As Double
    Dim tempo2 As Double
    Dim tempo3 As Double
    Dim rota As String

    Call distancia
    Call tempo
    Call custo

    Sheets("BD").Select

    tempo1 = range("E32").Value

    tempo2 = range("E33").Value

    tempo3 = range("E34").Value

    If tempo1 < tempo2 And tempo1 < tempo3 Then

        tempofinal = tempo1
        distancia = range("E32").Offset(0, -1).Value
        custo = range("E32").Offset(0, 1).Value
        rota = range("E32").Offset(0, -2).Value

    End If

    If tempo2 < tempo1 And tempo2 < tempo3 Then

        tempofinal = tempo2
        distancia = range("E33").Offset(0, -1).Value
        custo = range("E33").Offset(0, 1).Value
        rota = range("E33").Offset(0, -2).Value

    End If

```

```

If tempo3 < tempo2 And tempo3 < tempo1 Then

    tempofinal = tempo3
    distancia = range("E34").Offset(0, -1).Value
    custo = range("E34").Offset(0, 1).Value
    rota = range("E34").Offset(0, -2).Value

End If

Me.Label7.Caption = ("Rota: " & rota)
Me.Label8.Caption = ("Distância: " & distancia & " km")
Me.Label9.Caption = ("Tempo de deslocamento: " & tempofinal & " minutos")
Me.Label10.Caption = ("Custo: " & custo & " R$/t")

End Sub

Private Sub ToggleButton5_Click()

    Dim nome As String
    Dim custofinal As Double
    Dim distancia As String
    Dim tempo As String
    Dim custo1 As Double
    Dim custo2 As Double
    Dim custo3 As Double
    Dim rota As String

    Call distancia
    Call tempo
    Call custo

    Sheets("BD").Select

    custo1 = range("F32").Value
    custo2 = range("F33").Value
    custo3 = range("F34").Value

    If custo1 < custo2 And custo1 < custo3 Then

        custofinal = custo1
        distancia = range("F32").Offset(0, -2).Value
        tempo = range("F32").Offset(0, -1).Value
        rota = range("F32").Offset(0, -3).Value

    End If

    If custo2 < custo1 And custo2 < custo3 Then

        custofinal = custo2
        distancia = range("F33").Offset(0, -2).Value
        tempo = range("F33").Offset(0, -1).Value
        rota = range("F33").Offset(0, -3).Value

    End If

    If custo3 < custo2 And custo3 < custo1 Then

        custofinal = custo3
        distancia = range("F34").Offset(0, -2).Value
        tempo = range("F34").Offset(0, -1).Value
        rota = range("F34").Offset(0, -3).Value

    End If

    Me.Label7.Caption = ("Rota: " & rota)

```

```
Me.Label8.Caption = ("Distância: " & distancia & " km")  
Me.Label9.Caption = ("Tempo de deslocamento: " & tempo & " minutos")  
Me.Label10.Caption = ("Custo: " & custofinal & " R$/t")
```

```
End Sub
```

### 3.2 MACRO PRINCIPAL

```
Sub linhas()
```

```
    range("J27:BA29").Select  
    Selection.ClearContents
```

```
    Dim linha As range  
    Dim cell As range  
    Dim contar As Integer
```

```
    Set linha = range("J27:J29")
```

```
    contar = 27
```

```
    For Each cell In linha
```

```
        If contar = 27 Then
```

```
            On Error Resume Next  
            Call trechos(contar)  
        End If
```

```
        If contar = 28 Then
```

```
            On Error Resume Next  
            Call trechos2(contar)
```

```
        End If
```

```
        If contar = 29 Then
```

```
            On Error Resume Next  
            Call trechos3(contar)
```

```
        End If
```

```
        contar = contar + 1
```

```
    Next
```

```
End Sub
```

### 3.3 PRIMEIRA ROTA

Sub trechos(x)

Dim range1 As range

Dim range2 As range

Dim banco As range

Dim inicio As String

Dim fim As String

Dim n As Integer

Dim n1 As Integer

Dim linha2 As range

Dim cell As range

Dim j As Integer

Dim controle As Boolean

Dim v As Integer

Dim letra1 As String

Dim letra2 As String

Dim letra3 As String

Set range1 = range("J24")

Set range2 = range("K24")

Set banco = range("B4:H20")

inicio = range1.Value

fim = range2.Value

Cells(x, 10).Value = inicio

n1 = 10

n = 10

j = 2

While Cells(x, n - 1) <> fim

Set linha2 = range(Cells(x, n1), Cells(x, n - 1))

If Cells(x, n) = inicio Then

n = n + 1

Else

Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 1).Value, banco, j, 0)

```

For Each cell In linha2

    If Cells(x, n).Value = cell Then

        controle = False

        Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 1).Value, banco, j + 1, 0)

        If Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 2).Value, banco, 2, 0) Then

            letra1 = Cells(x, n).Value
            letra2 = Cells(x, n - 1).Value
            letra3 = Cells(x, n - 2).Value

            v = 2

            While controle = False

                If WorksheetFunction.VLookup(letra2, banco, v, 0) = letra1 Or
                WorksheetFunction.VLookup(letra2, banco, v, 0) = letra3 Then

                    controle = False
                    v = v + 1

                Else

                    Cells(x, n).Value = WorksheetFunction.VLookup(letra2, banco, v, 0)
                    controle = True

                End If

            Wend

            ElseIf Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 2).Value, banco, 3, 0)

Then

            letra1 = Cells(x, n).Value
            letra2 = Cells(x, n - 1).Value
            letra3 = Cells(x, n - 2).Value

            v = 2

            While controle = False

                If WorksheetFunction.VLookup(letra2, banco, v, 0) = letra1 Or
                WorksheetFunction.VLookup(letra2, banco, v, 0) = letra3 Then

```

```

        controle = False
        v = v + 1

    Else
        Cells(x, n).Value = WorksheetFunction.VLookup(letra2, banco, v, 0)
        controle = True

    End If

Wend

Elseif Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 2).Value, banco, 4, 0)
Then

    letra1 = Cells(x, n).Value
    letra2 = Cells(x, n - 1).Value
    letra3 = Cells(x, n - 2).Value

    v = 2

    While controle = False

        If WorksheetFunction.VLookup(letra2, banco, v, 0) = letra1 Or
WorksheetFunction.VLookup(letra2, banco, v, 0) = letra3 Then

            controle = False
            v = v + 1

        Else
            Cells(x, n).Value = WorksheetFunction.VLookup(letra2, banco, v, 0)
            controle = True

        End If

    Wend

Elseif Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 2).Value, banco, 5, 0)
Then

    letra1 = Cells(x, n).Value
    letra2 = Cells(x, n - 1).Value
    letra3 = Cells(x, n - 2).Value

    v = 2

```

```

While controle = False

    If WorksheetFunction.VLookup(letra2, banco, v, 0) = letra1 Or
WorksheetFunction.VLookup(letra2, banco, v, 0) = letra3 Then

        controle = False
        v = v + 1

    Else
        Cells(x, n).Value = WorksheetFunction.VLookup(letra2, banco, v, 0)
        controle = True

    End If

Wend

Elseif Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 2).Value, banco, 6, 0)
Then

    letra1 = Cells(x, n).Value
    letra2 = Cells(x, n - 1).Value
    letra3 = Cells(x, n - 2).Value

    v = 2

    While controle = False

        If WorksheetFunction.VLookup(letra2, banco, v, 0) = letra1 Or
WorksheetFunction.VLookup(letra2, banco, v, 0) = letra3 Then

            controle = False
            v = v + 1

        Else
            Cells(x, n).Value = WorksheetFunction.VLookup(letra2, banco, v, 0)
            controle = True

        End If

    Wend

End If

j = j + 1

```

```
End If

Next
j = 2
n = n + 1
End If
Wend

End Sub
```

### 3.4 SEGUNDA ROTA

```
Sub trechos2(x)

Dim range1 As range
Dim range2 As range
Dim banco As range
Dim inicio As String
Dim fim As String
Dim n As Integer
Dim n1 As Integer
Dim linha2 As range
Dim cell As range
Dim j As Integer
Dim controle As Boolean
Dim v As Integer
Dim letra1 As String
Dim letra2 As String
Dim letra3 As String
Dim acima As String
Dim nume1 As Integer
Dim nume2 As Integer
Dim q As Integer

Set range1 = range("J24")
Set range2 = range("K24")
Set banco = range("B4:H20")

inicio = range1.Value
fim = range2.Value

Cells(x, 10).Value = inicio

n1 = 10
```

n = 10

j = 2

While Cells(x, n - 1) <> fim

Set linha2 = range(Cells(x, n1), Cells(x, n - 1))

If Cells(x, n) = inicio Then

n = n + 1

Else

Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 1).Value, banco, j, 0)

acima = Cells(x - 1, n).Value

For Each cell In linha2

If cell = Cells(x, n).Value Then

controle = False

Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 1).Value, banco, j + 1, 0)

If Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 2).Value, banco, 2, 0) Then

letra1 = Cells(x, n).Value

letra2 = Cells(x, n - 1).Value

letra3 = Cells(x, n - 2).Value

v = 2

While controle = False

If WorksheetFunction.VLookup(letra2, banco, v, 0) = letra1 Or  
WorksheetFunction.VLookup(letra2, banco, v, 0) = letra3 Then

controle = False

v = v + 1

Else

```

        Cells(x, n).Value = WorksheetFunction.VLookup(letra2, banco, v, 0)
        controle = True

    End If

Wend

Elseif Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 2).Value, banco, 3, 0)
Then

    letra1 = Cells(x, n).Value
    letra2 = Cells(x, n - 1).Value
    letra3 = Cells(x, n - 2).Value

    v = 2

    While controle = False

        If WorksheetFunction.VLookup(letra2, banco, v, 0) = letra1 Or
WorksheetFunction.VLookup(letra2, banco, v, 0) = letra3 Then

            controle = False
            v = v + 1

        Else
            Cells(x, n).Value = WorksheetFunction.VLookup(letra2, banco, v, 0)
            controle = True

        End If

    Wend

Elseif Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 2).Value, banco, 4, 0)
Then

    letra1 = Cells(x, n).Value
    letra2 = Cells(x, n - 1).Value
    letra3 = Cells(x, n - 2).Value

    v = 2

    While controle = False

```

```

        If WorksheetFunction.VLookup(letra2, banco, v, 0) = letra1 Or
WorksheetFunction.VLookup(letra2, banco, v, 0) = letra3 Then

```

```

        controle = False

```

```

        v = v + 1

```

```

    Else

```

```

        Cells(x, n).Value = WorksheetFunction.VLookup(letra2, banco, v, 0)

```

```

        controle = True

```

```

    End If

```

```

Wend

```

```

Elseif Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 2).Value, banco, 5, 0)

```

Then

```

    letra1 = Cells(x, n).Value

```

```

    letra2 = Cells(x, n - 1).Value

```

```

    letra3 = Cells(x, n - 2).Value

```

```

    v = 2

```

```

    While controle = False

```

```

        If WorksheetFunction.VLookup(letra2, banco, v, 0) = letra1 Or
WorksheetFunction.VLookup(letra2, banco, v, 0) = letra3 Then

```

```

        controle = False

```

```

        v = v + 1

```

```

    Else

```

```

        Cells(x, n).Value = WorksheetFunction.VLookup(letra2, banco, v, 0)

```

```

        controle = True

```

```

    End If

```

```

Wend

```

```

Elseif Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 2).Value, banco, 6, 0)

```

Then

```

    letra1 = Cells(x, n).Value

```

```

    letra2 = Cells(x, n - 1).Value

```

```

    letra3 = Cells(x, n - 2).Value

```

```

    v = 2

```

```

While controle = False

    If WorksheetFunction.VLookup(letra2, banco, v, 0) = letra1 Or
WorksheetFunction.VLookup(letra2, banco, v, 0) = letra3 Then

        controle = False
        v = v + 1

    Else
        Cells(x, n).Value = WorksheetFunction.VLookup(letra2, banco, v, 0)
        controle = True

    End If

Wend

End If

j = j + 1

End If

Next

j = 2

If Cells(x, n) = Cells(x - 1, n) Then

    q = WorksheetFunction.VLookup(Cells(x, n - 1), banco, 7, 0)

    If q > 1 Then

        Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 1), banco, q + 1, 0)

    End If

End If

j = 2
n = n + 1

End If

Wend

End Sub

```

### 3.5 TERCEIRA ROTA

Sub trechos3(x)

```
Dim range1 As range
Dim range2 As range
Dim banco As range
Dim inicio As String
Dim fim As String
Dim n As Integer
Dim n1 As Integer
Dim linha2 As range
Dim cell As range
Dim j As Integer
Dim controle As Boolean
Dim v As Integer
Dim letra1 As String
Dim letra2 As String
Dim letra3 As String
Dim acima As String
Dim nume1 As Integer
Dim nume2 As Integer
Dim q As Integer
Dim acd As Integer
```

```
Set range1 = range("J24")
Set range2 = range("K24")
Set banco = range("B4:H20")
```

```
inicio = range1.Value
fim = range2.Value
```

```
Cells(x, 10).Value = inicio
```

```
n1 = 10
```

```
n = 10
```

```
j = 2
```

```
While Cells(x, n - 1) <> fim
```

```
Set linha2 = range(Cells(x, n1), Cells(x, n - 1))
```

```
If Cells(x, n) = inicio Then
```

```
n = n + 1
```

```
Else
```

```
Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 1).Value, banco, j, 0)
```

```
acima = Cells(x - 1, n).Value
```

```
For Each cell In linha2
```

```
  If cell = Cells(x, n).Value Then
```

```
    controle = False
```

```
    Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 1).Value, banco, j + 1, 0)
```

```
  If Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 2).Value, banco, 2, 0) Then
```

```
    letra1 = Cells(x, n).Value
```

```
    letra2 = Cells(x, n - 1).Value
```

```
    letra3 = Cells(x, n - 2).Value
```

```
    v = 2
```

```
    While controle = False
```

```
      If WorksheetFunction.VLookup(letra2, banco, v, 0) = letra1 Or  
WorksheetFunction.VLookup(letra2, banco, v, 0) = letra3 Then
```

```
        controle = False
```

```
        v = v + 1
```

```
      Else
```

```
        Cells(x, n).Value = WorksheetFunction.VLookup(letra2, banco, v, 0)
```

```
        controle = True
```

```
      End If
```

```
    Wend
```

```

Elseif Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 2).Value, banco, 3, 0)
Then

    letra1 = Cells(x, n).Value
    letra2 = Cells(x, n - 1).Value
    letra3 = Cells(x, n - 2).Value

    v = 2

    While controle = False

        If WorksheetFunction.VLookup(letra2, banco, v, 0) = letra1 Or
WorksheetFunction.VLookup(letra2, banco, v, 0) = letra3 Then

            controle = False
            v = v + 1

        Else
            Cells(x, n).Value = WorksheetFunction.VLookup(letra2, banco, v, 0)
            controle = True

        End If

    Wend

Elseif Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 2).Value, banco, 4, 0)
Then

    letra1 = Cells(x, n).Value
    letra2 = Cells(x, n - 1).Value
    letra3 = Cells(x, n - 2).Value

    v = 2

    While controle = False

        If WorksheetFunction.VLookup(letra2, banco, v, 0) = letra1 Or
WorksheetFunction.VLookup(letra2, banco, v, 0) = letra3 Then

            controle = False
            v = v + 1

        Else
            Cells(x, n).Value = WorksheetFunction.VLookup(letra2, banco, v, 0)
            controle = True

```

```

        End If

    Wend

Elseif Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 2).Value, banco, 5, 0)
Then

    letra1 = Cells(x, n).Value
    letra2 = Cells(x, n - 1).Value
    letra3 = Cells(x, n - 2).Value

    v = 2

    While controle = False

        If WorksheetFunction.VLookup(letra2, banco, v, 0) = letra1 Or
WorksheetFunction.VLookup(letra2, banco, v, 0) = letra3 Then

            controle = False
            v = v + 1

        Else

            Cells(x, n).Value = WorksheetFunction.VLookup(letra2, banco, v, 0)
            controle = True

        End If

    Wend

Elseif Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 2).Value, banco, 6, 0)
Then

    letra1 = Cells(x, n).Value
    letra2 = Cells(x, n - 1).Value
    letra3 = Cells(x, n - 2).Value

    v = 2

    While controle = False

        If WorksheetFunction.VLookup(letra2, banco, v, 0) = letra1 Or
WorksheetFunction.VLookup(letra2, banco, v, 0) = letra3 Then

```

```
        controle = False
        v = v + 1

    Else
        Cells(x, n).Value = WorksheetFunction.VLookup(letra2, banco, v, 0)
        controle = True

    End If

Wend

End If

j = j + 1

End If

Next

j = 2

If Cells(x, n) = Cells(x - 1, n) Then

    q = WorksheetFunction.VLookup(Cells(x, n - 1), banco, 7, 0)

    If q > 1 Then

        Cells(x, n).Value = WorksheetFunction.VLookup(Cells(x, n - 1), banco, q + 1, 0)

    End If

Else

    Dim conex1 As String
    Dim conex2 As String
    Dim conex3 As String
    Dim conex4 As String
    Dim conex5 As String
    Dim conex As Integer
    Dim conex11 As String
    Dim conex22 As String
    Dim conex33 As String
    Dim conex44 As String
    Dim conex55 As String
```

```
conex = 2
```

```
If Cells(x, n) = Cells(x - 2, n) Then
```

```
    Dim master As Boolean
```

```
    master = False
```

```
    conex1 = WorksheetFunction.VLookup(Cells(x, n - 1), banco, conex, 0)
```

```
    conex2 = WorksheetFunction.VLookup(Cells(x, n - 1), banco, conex + 1, 0)
```

```
    conex3 = WorksheetFunction.VLookup(Cells(x, n - 1), banco, conex + 2, 0)
```

```
    conex4 = WorksheetFunction.VLookup(Cells(x, n - 1), banco, conex + 3, 0)
```

```
    conex5 = WorksheetFunction.VLookup(Cells(x, n - 1), banco, conex + 4, 0)
```

```
        If conex5 <> Cells(x, n - 2) And conex5 <> Cells(x - 2, n) And conex5 <> Cells(x - 1, n)
        And conex5 <> "" And master = False Then
```

```
            conex11 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex, 0)
```

```
            conex22 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex + 1, 0)
```

```
            conex33 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex + 2, 0)
```

```
            conex44 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex + 3, 0)
```

```
            conex55 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex + 4, 0)
```

```
                If conex5 <> conex11 And conex5 <> conex22 And conex5 <> conex33 And conex5 <>
                conex44 And conex5 <> conex55 And conex5 <> "" Then
```

```
                    Cells(x, n).Value = conex5
```

```
                    master = True
```

```
                End If
```

```
            End If
```

```
        If conex4 <> Cells(x, n - 2) And conex4 <> Cells(x - 2, n) And conex4 <> Cells(x - 1, n)
        And conex4 <> "" And master = False Then
```

```
            conex11 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex, 0)
```

```
            conex22 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex + 1, 0)
```

```
            conex33 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex + 2, 0)
```

```
            conex44 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex + 3, 0)
```

```
            conex55 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex + 4, 0)
```

```
If conex4 <> conex11 And conex4 <> conex22 And conex4 <> conex33 And conex4 <>
conex44 And conex4 <> conex55 And conex4 <> "" Then
```

```
Cells(x, n).Value = conex4
```

```
master = True
```

```
End If
```

```
End If
```

```
If conex3 <> Cells(x, n - 2) And conex3 <> Cells(x - 2, n) And conex3 <> Cells(x - 1, n)
And conex3 <> "" And master = False Then
```

```
conex11 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex, 0)
```

```
conex22 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex + 1, 0)
```

```
conex33 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex + 2, 0)
```

```
conex44 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex + 3, 0)
```

```
conex55 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex + 4, 0)
```

```
If conex3 <> conex11 And conex3 <> conex22 And conex3 <> conex33 And conex3 <>
conex44 And conex3 <> conex55 And conex3 <> "" Then
```

```
Cells(x, n).Value = conex3
```

```
master = True
```

```
End If
```

```
End If
```

```
If conex2 <> Cells(x, n - 2) And conex2 <> Cells(x - 2, n) And conex2 <> Cells(x - 1, n)
And conex2 <> "" And master = False Then
```

```
conex11 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex, 0)
```

```
conex22 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex + 1, 0)
```

```
conex33 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex + 2, 0)
```

```
conex44 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex + 3, 0)
```

```
conex55 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex + 4, 0)
```

```
If conex2 <> conex11 And conex2 <> conex22 And conex2 <> conex33 And conex2 <>
conex44 And conex2 <> conex55 And conex2 <> "" Then
```

```
Cells(x, n).Value = conex2
```

```

        master = True

    End If
End If

If conex1 <> Cells(x, n - 2) And conex1 <> Cells(x - 2, n) And conex1 <> Cells(x - 1, n)
And conex1 <> "" And master = False Then

    conex11 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex, 0)
    conex22 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex + 1, 0)
    conex33 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex + 2, 0)
    conex44 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex + 3, 0)
    conex55 = WorksheetFunction.VLookup(Cells(x, n - 2), banco, conex + 4, 0)

    If conex1 <> conex11 And conex1 <> conex22 And conex1 <> conex33 And conex1 <>
conex44 And conex1 <> conex55 And conex1 <> "" Then

        Cells(x, n).Value = conex1

        master = True

    End If

End If

End If

End If

j = 2
n = n + 1
'Incrementa o n para ir para a próxima coluna no próximo while
'Só entra no while novamente se a célula que foi calculada agora for diferente de FIM

End If
'fecha o 1º if e else

Wend
'fecha o 1º while

End Sub

```

### 3.6 MACROS PARA CALCULAR DISTÂNCIA, TEMPO E CUSTO

```

Sub distancia()

    Dim distancia1 As Double
    Dim distancia2 As Double
    Dim distancia3 As Double
    Dim dist1 As Double
    Dim dist2 As Double
    Dim dist3 As Double
    Dim banco As range
    Dim i As Integer
    Dim y As Integer
    Dim cell As range

    Sheets("BD").Select
    Set banco = range("B4:N20")

    y = 10
    For i = 10 To 43

        On Error Resume Next
        dist1 = WorksheetFunction.VLookup(Cells(27, y).Value, banco, 13, 0)
        distancia1 = distancia1 + dist1
        dist1 = 0

        On Error Resume Next
        dist2 = WorksheetFunction.VLookup(Cells(28, y).Value, banco, 13, 0)
        distancia2 = distancia2 + dist2
        dist2 = 0

        On Error Resume Next
        dist3 = WorksheetFunction.VLookup(Cells(29, y).Value, banco, 13, 0)
        distancia3 = distancia3 + dist3
        dist3 = 0

        y = y + 1
    Next

    range("D32").Value = CDbI(distancia1)
    range("D33").Value = CDbI(distancia2)
    range("D34").Value = CDbI(distancia3)

End Sub

```

```
Sub tempo()  
  
    Dim tempo1 As Double  
    Dim tempo2 As Double  
    Dim tempo3 As Double  
    Dim temp1 As Double  
    Dim temp2 As Double  
    Dim temp3 As Double  
    Dim banco As range  
    Dim i As Integer  
    Dim y As Integer  
    Dim cell As range  
  
    Sheets("BD").Select  
    Set banco = range("B4:N20")  
  
    y = 10  
    For i = 10 To 43  
  
        On Error Resume Next  
        temp1 = WorksheetFunction.VLookup(Cells(27, y).Value, banco, 10, 0)  
        tempo1 = tempo1 + temp1  
        temp1 = 0  
  
        On Error Resume Next  
        temp2 = WorksheetFunction.VLookup(Cells(28, y).Value, banco, 10, 0)  
        tempo2 = tempo2 + temp2  
        temp2 = 0  
  
        On Error Resume Next  
        temp3 = WorksheetFunction.VLookup(Cells(29, y).Value, banco, 10, 0)  
        tempo3 = tempo3 + temp3  
        temp3 = 0  
  
        y = y + 1  
  
    Next  
  
    range("E32").Value = CDbI(tempo1)  
    range("E33").Value = CDbI(tempo2)  
    range("E34").Value = CDbI(tempo3)  
  
End Sub
```

```
Sub custo()  
  
    Dim custo1 As Double  
    Dim custo2 As Double  
    Dim custo3 As Double  
    Dim cust1 As Double  
    Dim cust2 As Double  
    Dim cust3 As Double  
    Dim banco As range  
    Dim i As Integer  
    Dim y As Integer  
    Dim cell As range  
  
    Sheets("BD").Select  
    Set banco = range("B4:N20")  
  
    y = 10  
    For i = 10 To 43  
  
        On Error Resume Next  
        cust1 = WorksheetFunction.VLookup(Cells(27, y).Value, banco, 12, 0)  
        custo1 = custo1 + cust1  
        cust1 = 0  
  
        On Error Resume Next  
        cust2 = WorksheetFunction.VLookup(Cells(28, y).Value, banco, 12, 0)  
        custo2 = custo2 + cust2  
        cust2 = 0  
  
        On Error Resume Next  
        cust3 = WorksheetFunction.VLookup(Cells(29, y).Value, banco, 12, 0)  
        custo3 = custo3 + cust3  
        cust3 = 0  
  
        y = y + 1  
  
    Next  
  
    range("F32").Value = CDb(custo1)  
    range("F33").Value = CDb(custo2)  
    range("F34").Value = CDb(custo3)  
  
End Sub
```