

**UNIVERSIDADE FEDERAL DE ALAGOAS-UFAL
CAMPUS A. C. SIMÕES
CIÊNCIA DA COMPUTAÇÃO**

MYRON DAVID LUCENA CAMPOS PEIXOTO

**UMA PROPOSTA DE SOFTWARE PARA APOIO LOGÍSTICO A OFICIAIS DE
JUSTIÇA NO CUMPRIMENTO DE DILIGENCIAS.**

**MACEIÓ
2022**

Myron David Lucena Campos Peixoto

Uma Proposta de Software para Apoio Logístico a Oficiais de Justiça no Cumprimento de Diligencias.

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal de Alagoas - UFAL, Campus A. C. Simões.

Orientador: Prof. Dr. Erick de Andrade Barboza

Maceió
2022

Catálogo na Fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 - 1767

P379p Peixoto, Myron David Lucena Campos.

Uma proposta de *software* para apoio logístico a oficiais de justiça no cumprimento de diligências / Myron David Lucena Campos
Peixoto. – 2022.

47 f. : il.

Orientador: Erick de Andrade Barboza.

Monografia (Trabalho de conclusão de curso em Ciência da Computação) – Universidade Federal de Alagoas, Instituto de Computação. Maceió, 2022.

Bibliografia: f. 46-47.

1. Oficial de justiça. 2. Diligência. 3. Poder judiciário. 4. Roteamento. 5. *Software*. I. Título.

CDU: 004.4:34

Ao meu avô.

AGRADECIMENTOS

Agradeço a Deus, por tudo que tem me proporcionado. Agradeço ao Professor Erick, que nobremente aceitou me orientar. Aos meus familiares, que me suportam desde que nasci. Aos meus amigos, em especial aos membros do grupo do Toco e derivados. E por fim, agradeço a todos aqueles que de alguma forma colaboraram para esta realização.

Eu me deitei e dormi; acordei, porque o SENHOR me sustentou.

Salmos 3, 5

RESUMO

Oficiais de justiça são funcionários públicos que exercem funções de grande importância para o poder judiciário brasileiro. Sabendo disto, devido a alta demanda ocasionada por um grande número de processos judiciais que surgem todos os dias nas diversas comarcas espalhadas pelo Brasil, esses profissionais acabam sendo sobrecarregados. Neste caso a utilização da tecnologia da informação como forma de apoio acaba sendo um fator decisivo para que estes profissionais desempenhem seu trabalho de forma eficiente. No entanto, atualmente apesar do judiciário brasileiro adotar o processo judicial eletrônico como padrão, e fornecer uma ferramenta para controle de diligências, não há nenhum apoio computacional que ajude os oficiais logisticamente no cumprimento do seu dever. Todo planejamento de deslocamento aos endereços das diligências é realizado pelo oficial de justiça, planejamento este que toma tempo que poderia ser utilizado para empenho de outras tarefas. Neste trabalho apresentamos uma proposta de software que automatiza este planejamento de rotas, fornecendo sugestões de forma a maximizar a eficiência do oficial de justiça.

Palavras-chave: Oficial de Justiça. Diligência. Judiciário. Rotas. Roteamento. Software.

ABSTRACT

Judicial officers are civil servants who perform functions of great importance for the Brazilian judiciary. Knowing this, due to the high demand caused by a large number of lawsuits that arise every day in the various districts spread throughout Brazil, these professionals end up being overloaded. In this case, the use of information technology as a form of support ends up being a decisive factor for these professionals to perform their work efficiently. However, despite the Brazilian judiciary currently adopting the electronic judicial process as a standard, and providing a tool to control the diligence, there is no computing support that helps the officers logistically in the performance of their duty. All the planning of the displacement to the addresses of the proceedings is done by the bailiff, planning that takes time that could be used for other tasks. In this paper we present a software proposal that automates this route planning, providing suggestions in order to maximize the efficiency of the process server.

Keywords: Judicial Officers. Judiciary. Routes. Routing. Software.

LISTA DE FIGURAS

Figura 1 – Representação do grafo G	20
Figura 2 – Representação matricial do grafo G	21
Figura 3 – Representação do Grafo G como uma lista de adjacências	21
Figura 4 – Caminho destacado no grafo G	22
Figura 5 – Diferença entre BFS e DFS	23
Figura 6 – Painel do oficial de justiça, PJE 2.x	29
Figura 7 – Tela inicial da aplicação	32
Figura 8 – Tela de cadastro de endereços	33
Figura 9 – Seletor de endereço de partida	33
Figura 10 – Seletor de endereço de destino	33
Figura 11 – Seletor de tempo para realização das diligências	34
Figura 12 – Tela de apresentação da rota sugerida	34
Figura 13 – Opção de criação de rotas manualmente	35
Figura 14 – Tela de rotas exibindo a rota definida pelo usuário	36
Figura 15 – Exemplo de matriz de distâncias obtida através da API	37
Figura 16 – Arquitetura modular do PJE 2.1	43
Figura 17 – Arquitetura sugerida para implementação de um novo serviço	44

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Objetivos	15
1.1.1	Objetivo Geral	15
1.1.2	Objetivos Específicos	16
1.2	Organização do documento	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	O oficial de justiça	17
2.2	Diligências	17
2.2.1	Citação	18
2.2.2	Notificação	18
2.2.3	Intimação	18
2.2.4	Condução Coercitiva	18
2.2.5	Prisão	18
2.2.6	Busca e Apreensão	19
2.2.7	Alvará de Soltura	19
2.3	Teoria dos Grafos	19
2.3.1	Grafo	19
2.3.2	Representações de Grafos	20
2.3.3	Caminhos	21
2.3.4	Conectividade dos Grafos	22
2.3.5	Busca em Grafos	22
2.3.5.1	DFS - Busca em Profundidade	23
2.3.5.2	BFS - Busca em Largura	24
2.3.6	Grafos como estrutura de dados	25
2.4	Roteirização	25
2.4.1	Roteirização aplicada ao trabalho do oficial de justiça	26
2.5	MVP - Produto Mínimo Viável	26
3	PROBLEMÁTICA	28
3.1	A ferramenta atual e o nosso problema foco	29
4	PROPOSTA	31
4.1	A ferramenta	31
4.1.1	Tecnologias Utilizadas	31

4.1.2	Funcionamento da aplicação para o usuário	32
4.1.3	Funcionamento do cálculo de rotas	36
4.1.3.1	Dados	36
4.1.3.2	Obtenção de informações geográficas	37
4.1.3.3	Geração da rota de diligências	37
5	CONSIDERAÇÕES SOBRE O PROTÓTIPO E POSSÍVEIS MELHO- RIAS FUTURAS	41
5.1	Coleta de endereços	41
5.2	Cálculo de rota	41
5.3	Arquitetura da aplicação	42
6	CONCLUSÃO	45
	REFERÊNCIAS	46

1 INTRODUÇÃO

O cargo de oficial de justiça é um cargo de vasta importância no funcionalismo judicial brasileiro, sendo uma profissão milenar, com registros até mesmo na Bíblia cristã (ALEXANDRIA, 2010), os oficiais de justiça são os operadores do Direito responsáveis por cumprir as decisões da justiça brasileira.

Atualmente a função do oficial de justiça pode ser vista como a realização de alguns atos judiciais, que podem ser resumidos em atos de comunicação no início do processo, como por exemplo, citações, intimações, notificações e etc. Além de atos de execução ao seu fim. Muitos desses atos podem ser chamados de diligências. Uma diligência é, de acordo com o a Lei n 13.105/15 do novo Código de Processo Civil, a realização de atos de ofício por funcionários de justiça. Na prática a diligência pode ser considerada a execução de medidas judiciais feita por oficial de justiça fora dos tribunais ou cartórios.

O controle de diligências, bem como de todo o processo judicial até pouco tempo era realizado inteiramente de forma manual com processos físicos, o que fazia com que o tempo de duração de um processo se prolongasse. Esse panorama começou a mudar com a implantação do processo eletrônico, a partir da Lei 11.419/2006, que dispõe sobre a informatização do processo judicial (FONSECA et al., 2018). Desde então, grandes avanços de tecnologia da informação ocorreram na justiça brasileira, culminando na Resolução Nº 185 de 18/12/2013 do CNJ, que instituiu o PJe como sistema nacional de processamento de informações e prática de atos processuais.

De acordo com um estudo encomendado pelo CNJ e realizado pela FGV no ano de 2018, mais de quatro milhões de processos tramitam em pelo menos 2,2 mil órgãos julgadores (OTONI, 2018). Comprovadamente a utilização do PJe trouxe ganhos significativos de eficiência, efetividade e redução de custos, uniformizando e automatizando os trâmites processuais (FALCÃO et al., 2018).

Apesar da melhoria significativa trazida pelo processo judicial eletrônico, existe uma grande demanda de oficiais de justiça para dar conta das inúmeras diligências que surgem diariamente nos tribunais espalhados pelo Brasil. No sistema atual do PJe, já existe uma solução para o controle de diligências, permitindo que os oficiais de justiça consultem, visualizem os autos e registrem seus resultados. Esta solução cumpre o papel a que se propõe, que é de fazer o controle dos mandados. No entanto, quando pensamos no estado atual da tecnologia de desenvolvimento de sistemas, podemos perceber que existem espaços para melhorias e novas

funcionalidades que já são consolidadas em outros sistemas de informação de outras áreas.

O oficial de justiça que utiliza o PJe atualmente já tem na prática uma boa melhora na produtividade em comparativo com o passado onde tudo ocorria manualmente nos cartórios e secretarias das comarcas pelo Brasil, pois pode trabalhar de qualquer lugar, desde que tenha acesso a internet e um computador disponível, no entanto grande parte do serviço externo, o cumprimento das diligências em si, é todo pensando de forma manual pelo próprio oficial, que faz suas próprias rotas e trajetos, verifica qual diligência é mais urgente, quais diligências cumprir em um mesmo trajeto, etc. Muitas dessas tarefas podem ser auxiliadas através de ferramentas de software, que trazem notáveis ganhos de produtividade.

Um exemplo de software bem difundido no meio corporativo é o chamado sistema de roteirização, que é muito utilizado pela indústria de transportes. Nada mais é que uma ferramenta que auxilia na automatização e no planejamento de rotas (JUNIOR et al., 2013). Outro tipo de sistema bastante consolidado no meio da Tecnologia da Informação, são os softwares de help desk, que servem para registrar e automatizar solicitações dos clientes de uma empresa. De uma forma geral, a solução utilizada no PJe para os oficiais de justiça se assemelha bastante com um software de help desk, no entanto muitos recursos já existentes em outros softwares, como o caso de funcionalidades de roteamento, poderiam ser incorporados a este sistema, proporcionando ganhos de produtividade dos servidores e ajudando-os a conseguir suprir a enorme demanda existente no judiciário brasileiro.

Tendo em vista a lacuna existente na ferramenta atual embutida no PJe, o que é proposto neste trabalho é uma melhoria para esta ferramenta, adicionando uma funcionalidade que auxilie o oficial de justiça não só no controle das diligências, mas que o possibilite um melhor desempenho, de maneira a automatizar tarefas que demandam tempo do oficial, como o caso do roteamento de diligências, que é feito manualmente pelos oficiais. E desta forma evitar o desgaste do servidor, potencializando o foco no que realmente importa, que é o cumprimento da diligência de maneira eficaz.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Este trabalho tem como objetivo a proposta de uma funcionalidade de software que forneça sugestões automáticas de rotas de diligências para oficiais de justiça e além disso, que permita que o oficial crie suas próprias rotas de forma mais eficiente.

1.1.2 Objetivos Específicos

- Implementar a funcionalidade proposta na forma de um MVP (Produto Viável Mínimo), de maneira a atestar sua viabilidade.
- Fornecer ao usuário no protótipo implementado, rotas que englobem maior quantidade de diligências no tempo escolhido pelo usuário.
- Permitir que o usuário crie suas próprias rotas.
- Fornecer uma visualização do percurso em mapa.
- Elencar possíveis melhorias após a implementação do MVP.

1.2 ORGANIZAÇÃO DO DOCUMENTO

O presente trabalho está organizado em 6 capítulos contando com esta introdução. O capítulo 2 aborda a fundamentação teórica, trazendo uma breve explicação sobre o oficial de justiça e suas atribuições, bem como uma revisão de alguns conceitos da teoria dos grafos e roteirização. O capítulo 3 diz respeito a problemática que a proposta deste trabalho busca resolver ou minimizar. No capítulo 4 a ferramenta é apresentada e seu funcionamento explicado. O capítulo 5 traz considerações surgidas com o desenvolvimento da ferramenta e algumas propostas de melhoria. Ao final, o capítulo 6 apresenta as considerações finais deste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Para que melhor possamos compreender a problemática que gira em torno da proposta deste trabalho, é necessário entendermos alguns conceitos do direito brasileiro, bem como uma fundamentação teórica a respeito de técnicas e outros temas tratados que abrangem a área da computação.

2.1 O OFICIAL DE JUSTIÇA

No Brasil, a função do oficial de justiça é executar ordens judiciais, realizando atos de comunicação ou de execução. De acordo com o Código do Processo Civil, de 2015, em seu artigo 149:

São auxiliares da Justiça, além de outros cujas atribuições sejam determinadas pelas normas de organização judiciária, o escrivão, o chefe de secretaria, o oficial de justiça, o perito, o depositário, o administrador, o intérprete, o tradutor, o mediador, o conciliador judicial, o partidor, o distribuidor, o contabilista e o regulador de avarias.

Ainda no Código do Processo Civil, o artigo 154 define as incumbências do oficial de justiça:

Incumbe ao oficial de justiça:

I - Fazer pessoalmente citações, prisões, penhoras, arrestos e demais diligências próprias do seu ofício, sempre que possível na presença de 2 (duas) testemunhas, certificando no mandado o ocorrido, com menção ao lugar, ao dia e à hora;

II - Executar as ordens do juiz a que estiver subordinado;

III - entregar o mandado em cartório após seu cumprimento;

IV - Auxiliar o juiz na manutenção da ordem;

V - Efetuar avaliações, quando for o caso;

VI - Certificar, em mandado, proposta de autocomposição apresentada por qualquer das partes, na ocasião de realização de ato de comunicação que lhe couber.

Parágrafo único. Certificada a proposta de autocomposição prevista no inciso VI, o juiz ordenará a intimação da parte contrária para manifestar-se, no prazo de 5 (cinco) dias, sem prejuízo do andamento regular do processo, entendendo-se o silêncio como recusa.

2.2 DILIGÊNCIAS

Em sua maioria, o cumprimento de diligências envolve o deslocamento do oficial de justiça até o endereço da parte envolvida no processo. No entanto hoje em dia outras formas de comunicação também são válidas, como por exemplo a ligação ou mensagens instantâneas por aplicativo (SILVA, 2016).

Como vimos no artigo 154 citado acima, o oficial de justiça é encarregado de realizar vários tipos de diligências, elencaremos as principais de acordo com (ANUNCIACÃO, 2015).

2.2.1 Citação

A finalidade da citação é avisar ao réu de que existe ação penal contra ele e que o mesmo deverá vir em juízo para apresentar sua defesa. A citação pode ser por mandado, onde o oficial de justiça entra em contato com o réu, ou por edital, que ocorre justamente quando o oficial não consegue encontrar o indivíduo citado.

2.2.2 Notificação

A notificação nada mais é que o ato de dar ciência de alguma coisa a alguém. Ocorre sempre que exista algum acontecimento processual, onde serão notificados: o acusado, os advogados, o Ministério Público, testemunhas ou qualquer outro envolvido no processo.

2.2.3 Intimação

A intimação é semelhante a notificação, sendo por definição dar conhecimento as partes envolvidas num processo de algum ato já realizado ou a realizar. Normalmente a intimação serve para informar sobre algum ocorrido, enquanto na notificação, o objetivo geralmente é informar sobre algo que o envolvido tenha que fazer.

Assim como a citação, a intimação também pode ser por mandado, porém há também a possibilidade de a intimação ocorrer por despacho. Na prática, ambas as formas resultam em uma diligência, que é nosso foco.

2.2.4 Condução Coercitiva

A condução coercitiva ocorre quando uma testemunha que tenha sido devidamente intimada não compareça a juízo e não forneça justificativa. Dessa forma, para o cumprimento dessa diligência o oficial de justiça intimará novamente a parte e a conduzirá a presença do juiz. A condução pode ou não ter a ajuda da força policial.

2.2.5 Prisão

A prisão é o ato executado mediante o mandado de prisão. O mandado de prisão necessita da assinatura de uma autoridade ordenante, normalmente o juiz, além de descrever quem deverá ser preso, também deve descrever com clareza o fato delituoso. O mandado pode ser cumprido

tanto pelo oficial de justiça quanto pela polícia. Como qualquer outra diligência, o oficial de justiça certificará o dia, hora e lugar a qual o ato foi realizado.

2.2.6 Busca e Apreensão

A busca e apreensão é o procedimento que ocorre para se evitar que provas de algum crime desapareçam. Para isso, apreendem-se os instrumentos e todos os objetos que possuam relação com o delito. O ato de busca e apreensão assim como o ato de prisão ocorre mediante mandado de busca e apreensão.

2.2.7 Alvará de Soltura

Quando um preso tem sua soltura determinada por um juiz, o oficial de justiça é encarregado de dar o cumprimento ao alvará, determinando sua liberdade imediata, obviamente caso ele não possua outros processos que o mantenham preso.

2.3 TEORIA DOS GRAFOS

Apesar de existirem vários softwares que realizam roteirização de trajetos em mapas de forma simples, é importante saber os conceitos básicos que são comuns ao funcionamento de todos eles, sendo assim, apresentaremos uma breve revisão sobre a teoria dos grafos focando em conceitos que serão utilizados em nossa proposta. Os conceitos apresentados nesta sessão são baseados em (BONDY et al., 1976).

2.3.1 Grafo

Definição: Um grafo G é definido como um par (V, E) , em que V é um conjunto arbitrário e E é um conjunto de $V \times V$. Os elementos de V são chamados vértices e os de E são chamados de arestas.

- Uma aresta $e \in E$ será denotada simplesmente pelo par de vértices $e = (v, u)$ ou simplesmente vu . Diremos que a aresta e incide em v e que v é um extremo de e . Caso vu seja uma aresta, diremos que os vértices v e u são vizinhos ou adjacentes.
- Utilizaremos $|V| = n$ e $|E| = m$ para denotar o número de vértices e de arestas, respectivamente.
- Cada aresta pode ter um peso atribuído, que pode variar a depender da forma como será representada.

Exemplo:

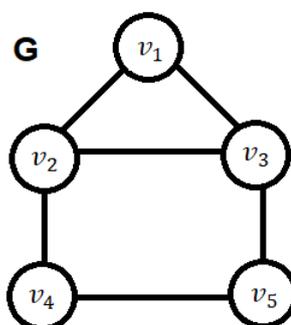
$$G = (V, E)$$

$$V = v_1, v_2, v_3, v_4, v_5$$

$$E = v_1v_2, v_1v_3, v_2v_3, v_2v_4, v_3v_5, v_4v_5$$

- Um grafo pode ser visualizado através de uma representação geométrica na qual seus vértices correspondem a pontos distintos do plano em posições arbitrárias.

Figura 1 – Representação do grafo G



Fonte: Autoria Própria.

2.3.2 Representações de Grafos

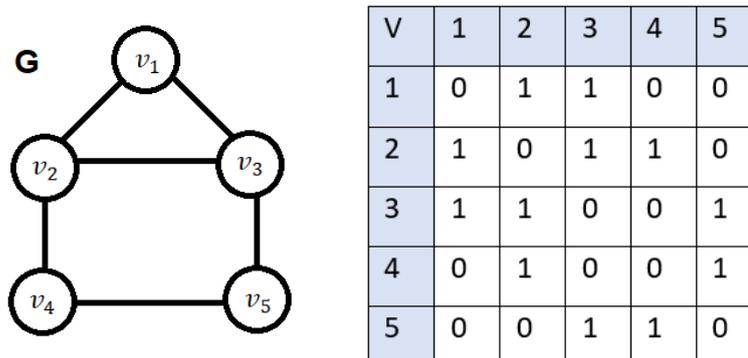
Existem várias formas de se representar um grafo computacionalmente, no entanto se destacam as matrizes e listas de adjacência.

Uma matriz de adjacências é uma matriz $n \times n$, onde n é o número de vértices do grafo. Os elementos da matriz são marcados com 0 ou 1. Quando um elemento da linha i é adjacente a um elemento da coluna j , ou seja, é uma aresta, marca-se o elemento com o valor 1, caso contrário, marca-se com o valor 0.

Exemplo:

Para o grafo G, teremos a seguinte matriz:

Figura 2 – Representação matricial do grafo G



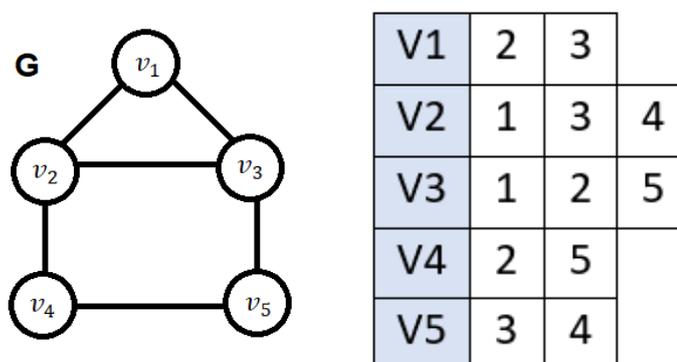
Fonte: Autoria Própria.

De modo semelhante a matriz, uma lista de adjacências é um conjunto de n listas, onde n é a quantidade de vértices. Cada lista do conjunto é denominada lista de adjacências do vértice v , e os elementos pertencentes a esta lista são os vértices adjacentes ao vértice v .

Exemplo:

Para o mesmo grafo G, temos a seguinte lista:

Figura 3 – Representação do Grafo G como uma lista de adjacências



Fonte: Autoria Própria.

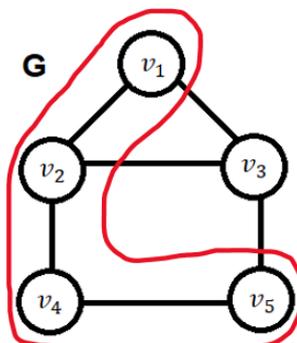
2.3.3 Caminhos

Um caminho é uma sequência de vértices $v_1, v_2, v_3, \dots, v_k$ onde cada par (v_j, v_{j+1}) com $1 \leq j \leq k - 1$, são vértices adjacentes, isto é, formam aresta, sendo $k - 1$ o comprimento do caminho.

Exemplo de caminho:

$$v_1 \rightarrow v_2 \rightarrow v_4 \rightarrow v_5$$

Figura 4 – Caminho destacado no grafo G



Fonte: Autoria Própria.

Já um ciclo é um caminho no qual o último vértice é igual ao primeiro. Considerando o grafo G acima, o seguinte caminho seria um ciclo:

$$v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_1$$

Um caminho ou ciclo que contenha cada vértice do grafo uma única vez é chamado hamiltoniano. Já um ciclo que contenha cada aresta do grafo exatamente uma vez é chamado euleriano.

Chamamos de distância entre dois vértices o comprimento do menor caminho entre eles.

2.3.4 Conectividade dos Grafos

Chamamos de “grafo conexo” o grafo em que para cada par de vértices exista um caminho. Do contrário o grafo é desconexo. Um grafo que não possuir arestas é chamado de totalmente desconexo.

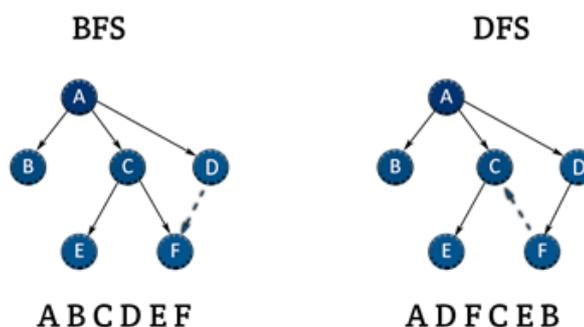
2.3.5 Busca em Grafos

Um algoritmo de busca em um grafo é um algoritmo que visita todos os vértices de um grafo percorrendo-os através de suas arestas, de um vértice para seu vértice adjacente. Os algoritmos de busca mais conhecidos são o de busca em largura, conhecido como BFS (*breadth first search*) e o de busca em profundidade, conhecido como DFS (*depth first search*).

A diferença óbvia entre eles é descrita pelos seus respectivos nomes, na busca em largura, dado um grafo e um vértice inicial, são explorados seus vértices vizinhos e para cada um

desse, são explorados seus vizinhos. Já na busca em profundidade a partir do vértice inicial são explorados os vértices imediatamente adjacentes a ele até que não exista mais adjacentes, no caso de um árvore, até que o vértice não possua nó filho. (OPEN4TECH, 2019)

Figura 5 – Diferença entre BFS e DFS



Fonte: Disponível em: <<https://open4tech.com/bfs-vs-dfs/>>. Acesso em: 11 de nov. de 2022.

2.3.5.1 DFS - Busca em Profundidade

O algoritmo de busca em profundidade pode ter como entrada apenas um grafo e escolher aleatoriamente um vértice para ser o nó inicial. Neste trabalho, adotaremos a entrada como sendo um grafo e um vértice inicial já selecionado. Iniciando do nó raiz recebido na entrada, o algoritmo percorrerá cada caminho o mais longe possível antes de passar para o outro caminho, isto é, visitar o nó vizinho. (FEOFILOFF, 2019b)

Existem diferentes formas de se implementar este algoritmo, a forma apresentada neste trabalho utiliza uma fila e uma lista e pode ser vista no pseudocódigo baseado abaixo:

```
DFS(grafo , no_inicial){
    // cria uma pilha de nos que serao visitados
    pilha = pilha vazia

    // cria uma lista para guardar os nos ja visitados
    visitados = lista vazia

    // adiciona o no inicial na pilha
    pilha.push(no_inicial)

    // executa enquanto a pilha nao estiver vazia
    while (pilha nao estiver vazia){
        // desempilha o no
```

```

no = pilha.pop()

// se o no nao foi visitado ,
// visita-o e adiciona na lista de visitados
if (no nao esta em visitados){
    visitados.add(no)

    // adiciona os nos vizinhos a pilha para serem visitados
    foreach (vizinho in no.visinhas)
        pilha.push(vizinho)
    }
}
}

```

2.3.5.2 BFS - Busca em Largura

A entrada do algoritmo de busca em largura é semelhante ao da DFS, logo também utilizaremos como entrada um grafo e um vértice para ser o nó inicial. Iniciando no nó raiz recebido na entrada, o algoritmo visitará todos nós próximos a este antes de visitar os mais distantes e para cada vizinho visitado, será feito o mesmo processo. (FEOFILOFF, 2019a)

A forma como implementamos é bem semelhante a implementação da busca em profundidade vista acima, no entanto utiliza uma pilha no lugar da fila, como pode ser visto no pseudocódigo abaixo:

```

BFS(grafo , no_inicial){
    // cria uma fila de nos que serao visitados
    fila = fila vazia

    // cria uma lista para guardar os nos ja visitados
    visitados = lista vazia

    // adiciona o no inicial na fila
    fila.add(no_inicial)

    // executa enquanto a fila nao estiver vazia
    while (fila nao for vazia){
        // desenfilera o no
        no = fila.dequeue()
    }
}

```

```

        // se o no nao foi visitado ,
// visita-o e adiciona na lista de visitados
if (no nao esta em visitados){
    visitados.add(no)

    // adiciona os nos vizinhos a fila para serem visitados
    foreach (vizinho in no.visinhas)
        fila.add(vizinho)
    }
}
}

```

Os pseudocódigos exibidos foram baseados nos algoritmos apresentados por (ARAKAKI, 2011).

2.3.6 Grafos como estrutura de dados

Como visto anteriormente, podemos representar grafos computacionalmente com matrizes e listas. Dessa forma podemos utilizar grafos como estrutura de dados para representar coisas do cotidiano de maneira abstrata.

Um exemplo comum é visualizar dados geográficos na forma de um grafo, onde cada localização é representada por um vértice e os caminhos entre essas localizações são representadas por arestas. A distância entre os caminhos é atribuída como o peso em cada aresta. Este é só um simples exemplo, no entanto existem diversas outras aplicações.

2.4 ROTEIRIZAÇÃO

Em grande parte da literatura, o termo roteirização aparece relacionado ao processo de roteamento de veículos, onde designa o processo de determinação de roteiros ou sequências de paradas para determinado veículo. De acordo com (CUNHA, 2000) este processo tem por objetivo a visita de um conjunto de endereços geográficos pré-determinados que necessitam de algum tipo de atendimento.

Segundo NOVAES apud (OLIVEIRA, 2019), os processos de roteirização podem ser descritos como a decisão de quais clientes serão visitados e a sequência que essas visitas ocorrerão.

O processo de roteamento pode ser complexo e envolver diversas restrições a serem consideradas, como a literatura trata desse processo voltado a frotas de veículos e produtos,

normalmente são consideradas algumas questões como o tamanho da frota, tipo de frota, garagem, natureza da demanda, capacidade dos veículos, tempo máximo de rotas, custos, entre outras, como visto em (Enomoto e Lima, 2007) apud (OLIVEIRA, 2019).

O planejamento de rotas com a utilização de métodos de roteirização faz com que seja possível um melhor aproveitamento dos recursos disponíveis, onde de acordo com (JUNIOR et al., 2013), é possível ter um maior controle das rotas, reduzir a sobreposição de visitas, além de possibilitar rotas mais rentáveis, dessa forma possibilitando que a operação logística seja mais eficiente, reduzindo custos e melhorando o nível do serviço prestado.

2.4.1 Roteirização aplicada ao trabalho do oficial de justiça

Apesar do contexto do nosso trabalho não ser relacionado diretamente com veículos ou frotas, de certa forma o trabalho desempenhado pelos oficiais de justiça se encaixa naturalmente como um processo de roteirização.

O servidor do judiciário brasileiro recebe uma lista de diligências as quais possuem um prazo para cumprimento, cada diligência pode possuir um ou mais endereços os quais o oficial deve se deslocar para realizá-las. Este processo logístico atualmente é feito por inteiro unicamente pelo próprio oficial de justiça, que conta apenas com sua expertise para planejar quais diligências cumprirá em determinado dia de trabalho.

Algumas restrições de roteamento são exclusivas aos oficiais de justiça, por exemplo, o prazo para cumprimento e a urgência de determinada diligência, além disso o oficial é encarregado do próprio deslocamento, tendo que escolher se utilizará carro, moto etc.

Em muitos casos, existem diligências com endereços em locais distantes que possivelmente passam por locais de outras diligências, no entanto selecionar uma lista de diligências para cumprir que se encaixem em uma rota inteligente não é uma tarefa das mais simples. Com o que vimos, fica claro que o trabalho do oficial poderia ser potencializado com a utilização de processos de roteirização automatizados, mesmo que de formas simplistas, já que atualmente não existe nenhum outro apoio logístico para estes profissionais.

2.5 MVP - PRODUTO MÍNIMO VIÁVEL

O termo MVP (Minimum Viable Product) tem sido bastante utilizado atualmente com toda a popularização da cultura de startups e é definido por (RIES, 2009) como uma versão de um novo produto que permite que o desenvolvedor ou time envolvido valide e adquira conhecimento

sobre o cliente com o menor esforço possível. Ries ainda afirma que é um processo iterativo de geração de ideias, prototipagem, apresentação, coleta de dados, análise e aprendizado.

De acordo com (Blank, 2010) apud (LENARDUZZI; TAIBI, 2016), o produto mínimo viável possui apenas as funcionalidades que permitem que o produto seja implantado.

Alguns autores como (Munch, 2016) apud (LENARDUZZI; TAIBI, 2016) afirma que o MVP é um objeto experimental que permite o teste empírico de hipóteses de valor. Ou também pode ser visto como um produto simples, de baixa qualidade, um protótipo inicial, como dito por (Jawanda 2015) apud (LENARDUZZI; TAIBI, 2016).

Apesar das diferentes definições, grande parte delas sintetizam que o propósito geral de um MVP é servir de validação para uma ideia ou produto, sem que haja grande esforço produtivo gasto inicialmente, isso possibilita que sejam avaliadas novas funcionalidades de um software, que caso validadas, poderão ser implementadas de forma mais profissional no futuro.

3 PROBLEMÁTICA

O processo judicial eletrônico trouxe ganhos significativos para o funcionamento do judiciário brasileiro, no entanto, a alta carga de processos diários é um problema que continua demandando muito dos servidores, sejam eles do judiciário federal ou estadual.

Se tratando especialmente dos oficiais de justiça, não é incomum se deparar com notícias sobre reivindicações a respeito da sobrecarga de trabalho por parte dos sindicatos do setor. De acordo com o Sindicato dos Oficiais de Justiça de Alagoas, o sindicato tem recebido inúmeras reclamações de diversas comarcas sobre a sobrecarga de trabalho (SINDOJUS/AL, 2022).

Já em nota publicada pelo Sindicato dos Servidores do Poder Judiciário do Rio de Janeiro em setembro deste ano, o sindicato reivindica a convocação oficiais de justiça, de modo a minimizar a carência de servidores, a mesma nota elenca a situação de algumas centrais de mandados, para exemplificar, apenas na cidade de Santa Cruz, é dito:

Esta Central deveria contar com 21 (vinte e um) Oficiais de Justiça, mas conta com apenas 11 (onze) e a ajuda esporádica de 3 Oficiais, em auxílio provisório. Aliás, o auxílio de colegas de outras comarcas vem sendo uma alternativa, mas, por certo, não resolve o problema, a uma porque estes colegas desfalcam a sua própria serventia ao auxiliar outra central; a duas, porque são servidores deslocados de sua lotação de origem, de ofício, para regiões que não conhecem, as quais possuem características e riscos próprios, mostrando-se esta alternativa apenas um frágil paliativo. Os Oficiais recebem, em média, 300 ou mais mandados por mês, o que é humanamente impossível de se cumprir, pois são mandados de todo tipo, desde citação/intimação até mandados de prisão, passando por avaliações e conduções, que exigem longos preparativos e estrutura, demandando um enorme tempo para cumprimento. (SINDJUSTIÇA-RJ, 2022)

Como podemos ver pelos exemplos acima, fica evidente o problema: há uma grande quantidade de processos e uma quantidade limitada de servidores que não conseguem suprir a demanda, o que causa atrasos e conseqüentemente o número de processos nunca fica menor. Obviamente que não é trivial solucionar um problema desta natureza, pois envolve muitas variáveis como por exemplo o estado, questões orçamentárias, administrativas etc. Porém, de maneira a simplificar nosso entendimento, iremos visualizar basicamente duas opções: (1) aumentar a quantidade de oficiais de justiça nas comarcas e/ou (2) ajudar os oficiais a serem mais eficientes no cumprimento de diligência, aumentando dessa forma a capacidade individual dos servidores.

Neste trabalho, abordaremos a segunda opção descrita acima, dando foco no auxílio aos oficiais de justiça no diz respeito ao cumprimento de diligências, de maneira a tornar esta atividade mais eficiente, reduzindo o tempo de cumprimento e conseqüentemente aumentando a produtividade dos servidores.

3.1 A FERRAMENTA ATUAL E O NOSSO PROBLEMA FOCO

Atualmente a ferramenta utilizada pelos oficiais de justiça está embutida no sistema PJe e chama-se “Painel do oficial de justiça”.¹ A funcionalidade principal deste painel é listar os processos distribuídos para o usuário logado no sistema, que no caso, é o próprio oficial de justiça. A imagem abaixo mostra as seguintes funcionalidades listadas:

Figura 6 – Painel do oficial de justiça, PJE 2.x

The screenshot displays the 'Painel do oficial de justiça' interface. At the top, there is a search filter section with the following fields: 'Número do processo', 'Nome do destinatário', 'Tipo de mandado', 'Grupo Oficial de Justiça', 'Jurisdição', 'Órgão julgador', and 'Próxima audiência'. Below these filters are buttons for 'PESQUISAR' and 'LIMPAR'. A red circle with the number '1' highlights the 'PESQUISAR' button. Below the filters is a table titled 'Painel do Oficial de Justiça' with columns: 'Ato de comunicação', 'Próxima audiência', 'Endereço(s)', and 'Anexos'. The table contains three rows of data. Red circles with numbers 2, 3, 4, and 5 highlight specific features in the first row: 2. Print button, 3. Register button, 4. Show autos button, and 5. Urgency indicator.

Fonte: Autoria Própria.

1. Filtros de busca para expedientes
2. Botão para consultar e imprimir o mandado da diligência
3. Botão para registrar o resultado da diligência
4. Botão para exibir os autos do processo
5. Indicativo de urgência para cumprimento

Na listagem de processos são exibidas algumas informações do processo, o destinatário e o endereço para cumprimento da diligência.

Um único oficial de justiça pode possuir diversos expedientes para cumprimento, na figura acima por exemplo, o usuário do sistema possui 61 expedientes. Cada expediente normalmente

¹ Informações disponíveis em (TRF1, 2019)

está associado com destinatários e endereços diferentes, o que faz com que o oficial tenha que se deslocar e realizar vários trajetos diferentes para cumprir as diligências. Uma única diligência pode inclusive envolver mais de um endereço.

Todos os trajetos para a realização dos expedientes são pensados unicamente pelo oficial, onde ele deve levar em conta as distâncias dos destinatários, custos do trajeto, o tempo do percurso, a quantidade de expedientes que ele pode cumprir num mesmo trajeto, dentre outras variáveis. Fazer todo esse “cálculo” mentalmente não é uma tarefa simples e exige bastante expertise por parte do servidor. Além disso, organizar um trajeto para realizar expedientes toma um certo tempo que poderia ser utilizado para outras tarefas.

4 PROPOSTA

Dada a problemática descrita, a ideia deste trabalho é propor uma nova funcionalidade para a ferramenta de controle de diligências utilizada pelos oficiais de justiça no sistema PJe: a possibilidade de criação de roteiros para o cumprimento de expedientes, de maneira a calcular de forma automatizada o melhor trajeto e as diligências que deverão ser cumpridas neste trajeto.

A ideia de melhor trajeto utilizada neste trabalho é específica para as restrições existentes para o cumprimento de diligências pelos oficiais, em nosso caso o melhor trajeto não é necessariamente o que possua menor custo ou menor distância, mas sim o que se adequa a necessidade do oficial em cumprir uma maior quantidade de diligências em menor tempo.

Nossa ideia inicial era de desenvolver tal funcionalidade de forma integrada a ferramenta já existente no PJe, no entanto, o código do sistema não é aberto e como é um sistema de suma importância para o judiciário brasileiro, contendo dados sensíveis e pessoais, é de se imaginar a severa burocracia existente para ter acesso ao mesmo. Sendo assim optamos por desenvolver um protótipo simples, um MVP(Produto Mínimo Viável) como forma de validar nossa proposta.

4.1 A FERRAMENTA

Como mencionado anteriormente, a aplicação descrita neste trabalho tem como objetivo validar a funcionalidade de geração e sugestão de rotas para cumprimento de diligências pelos oficiais de justiça.

4.1.1 Tecnologias Utilizadas

Para o desenvolvimento do protótipo, optamos por utilizar a plataforma de desenvolvimento .NET¹, utilizando a linguagem Csharp e o framework Windows Forms² para criação de aplicações desktop em conjunto com a IDE Visual Studio 2022³. A escolha desse conjunto se deu pela familiaridade do autor com a linguagem e respectivos frameworks, além da facilidade de implementação que a plataforma fornece para criação interfaces gráficas desktop.

A aplicação também utilizou um banco de dados simples feito com SQLite⁴, além disso foi utilizada a biblioteca QuikGraph⁵, para lidar com grafos e foram realizadas consultas a API

¹ Disponível em <<https://dotnet.microsoft.com>>

² Disponível em <<https://learn.microsoft.com/pt-br/dotnet/desktop/winforms/?view=netdesktop-6.0>>

³ Disponível em <<https://visualstudio.microsoft.com>>

⁴ Disponível em <<https://www.sqlite.org>>

⁵ Disponível em <<https://github.com/KeRNeLith/QuikGraph>>

do Google Maps⁶ para obter informações de percursos entre localizações.

4.1.2 Funcionamento da aplicação para o usuário

O funcionamento da aplicação é simples e sua interface segue o padrão gráfico do framework Windows Forms. A disposição dos itens na tela segue o mesmo formato já utilizado no Pje, onde algumas opções aparecem no topo da tela e logo abaixo há uma listagem com as diligências que o oficial necessita cumprir, como mostra a figura 7:

Figura 7 – Tela inicial da aplicação

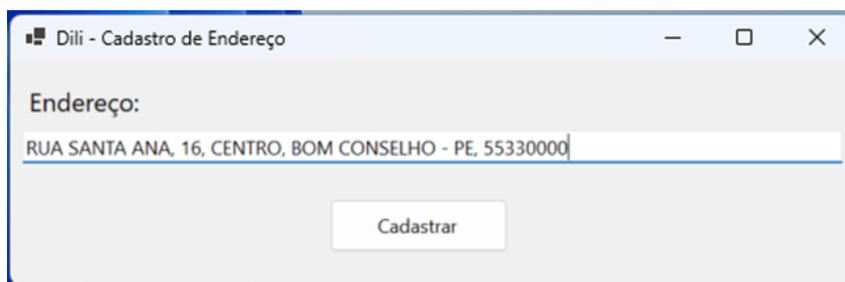
ID	Número do Processo	Destinatário	Endereço
4	0000749-92.2019.8.17.2300	JOBEL CORREIA DE MELO	Zona Rural, s/n, Sítio Limeira, Centro, BOM CONSELHO - PE - CEP: 55305-000
5	0000042-22.2022.8.17.2300	VALERIO DUARTE & CIA LTDA	PRACA DR FRANCISCO PEREIRA LOPES, 9, CENTRO, TEREZINHA - PE - CEP: 55305-000
6	0000232-78.2016.8.17.0300	BANCO DO NORDESTE DO BRASIL SA	AV ENGENHEIRO DOMINGOS FERREIRA, - de 2203 a 3041 - lado ímpar, BOA VIAGEM,
7	0000251-79.2019.8.17.0300	LEONARDO DOS SANTOS SOUZA	R TENENTE NILTON PRADO, 38, CENTRO, BOM CONSELHO - PE - CEP: 55330-000
8	0001591-67.2022.8.17.2300	MARLUCE RODRIGUES DE DEUS	SITIO BOA SORTE, S/N, ZONA RURAL, BOM CONSELHO - PE - CEP: 55330-000
9	0001052-10.2010.8.17.0300	JOSE ABILIO FERRO NETO	, 20, BOM CONSELHO - PE - CEP: 55305-000
10	0000355-80.2022.8.17.2300	MARCELO HENRIQUE DE OLIVEIRA MONTEIRO	RUA LOTUS, 19, 2 RUA AO LADO DA ETE, NOVO JARDIM, BOM CONSELHO - PE - CEP:
11	0000656-66.2018.8.17.2300	CARLOS MARCELO TENORIO DE ALMEIDA	Fazenda Boa Esperança, 1105, Zona Rural, BOM CONSELHO - PE - CEP: 55330-000
12	0000016-92.2020.8.17.2300	MARIA ROSANGELA SANTOS VIANA	Rua Um, 80, Nossa Senhora de Fátima, BOM CONSELHO - PE - CEP: 55305-000
13	0000156-92.2021.8.17.2300	MARCOS VENICIUS PADILHA TENORIO	BARAO DO RIO BRANCO, 216, CASA, CENTRO, BOM CONSELHO - PE - CEP: 55305-000
14	0002366-19.2021.8.17.2300	Caroline Marques Bezerra da Silva	Rua São Francisco, 28, São Rafael, BOM CONSELHO - PE - CEP: 55305-000
15	0000329-82.2022.8.17.2300	MARCELINO DOS SANTOS TENORIO	Sítio Lagoa do Pau Ferro, S/N, Zona Rural, BOM CONSELHO - PE - CEP: 55305-000
16	0000665-28.2018.8.17.2300	WELLINGTON SANTANA DA SILVA	SÍTIO QUEIMADA GRANDE, S/N, ZONA RURAL, BOM CONSELHO - PE - CEP: 55305-000
17	0002131-52.2021.8.17.2300	Bruna Izabelly Marques Vieira	Rua Alda Teixeira, CENTRO, BOM CONSELHO - PE - CEP: 55305-000
18	0000550-13.2006.8.17.0300	EUNICE GOMES CARDOSO - ME	, 25, BOM CONSELHO - PE - CEP: 55305-000
19	0001613-28.2022.8.17.2300	ERIVONE CLAUDIO TAVARES DA SILVA	Sítio Água Branca, S/N, Casa, Zona Rural, BOM CONSELHO - PE - CEP: 55330-000
20	0000284-54.2017.8.17.2300	ADILSON CARLOS DA SILVA	R JOSE BONIFACIO, 48, E-mail adilson.acshotmail.com, CENTRO, BOM CONSELHO - PE
21	0001814-20.2022.8.17.2300	JUIZO DA COMARCA DE BOM CONSELHO - PE	PRAÇA DOM PEDRO II, 34, CENTRO, BOM CONSELHO - PE - CEP: 55305-000
22	0000558-24.2005.8.17.0300	JOSE EVARISTO DA SILVA	, BOM CONSELHO - PE - CEP: 55305-000

Fonte: Autoria Própria.

Logo no canto superior esquerdo, há um botão chamado “Cadastrar novo endereço”, ao clicar neste botão o usuário é direcionado a uma tela em que poderá cadastrar um novo endereço, como exemplificado na imagem abaixo:

⁶ Disponível em <<https://developers.google.com/maps>>

Figura 8 – Tela de cadastro de endereços



Dili - Cadastro de Endereço

Endereço:

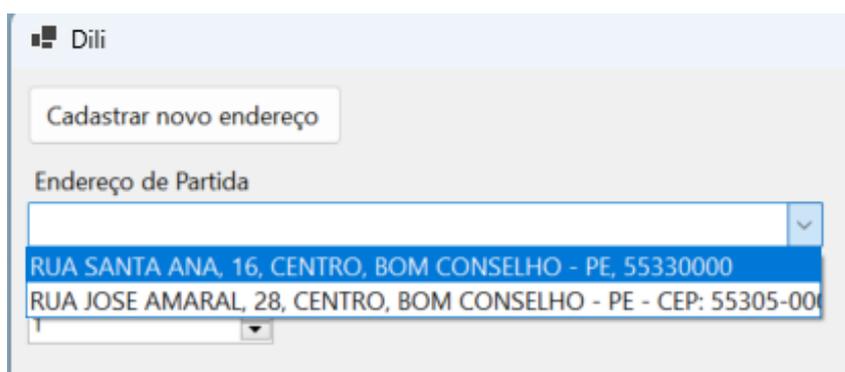
RUA SANTA ANA, 16, CENTRO, BOM CONSELHO - PE, 55330000

Cadastrar

Fonte: Autoria Própria.

Este endereço deverá aparecer como uma opção selecionável para o endereço de partida e o endereço de destino.

Figura 9 – Seletor de endereço de partida



Dili

Cadastrar novo endereço

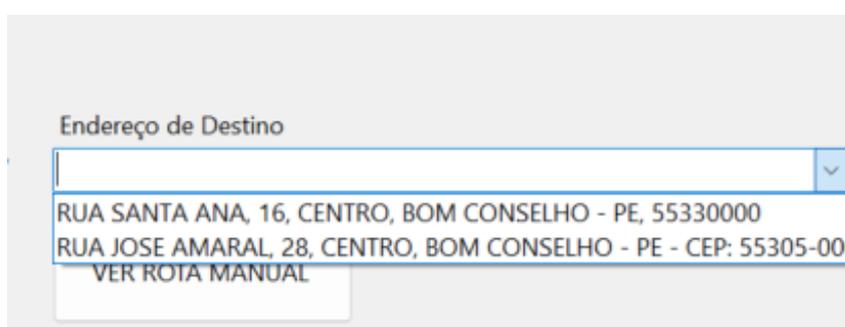
Endereço de Partida

RUA SANTA ANA, 16, CENTRO, BOM CONSELHO - PE, 55330000

RUA JOSE AMARAL, 28, CENTRO, BOM CONSELHO - PE - CEP: 55305-000

Fonte: Autoria Própria.

Figura 10 – Seletor de endereço de destino



Endereço de Destino

RUA SANTA ANA, 16, CENTRO, BOM CONSELHO - PE, 55330000

RUA JOSE AMARAL, 28, CENTRO, BOM CONSELHO - PE - CEP: 55305-000

VER ROTA MANUAL

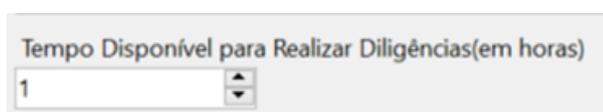
Fonte: Autoria Própria.

Como já visto acima, existem duas caixas de seleção em que o usuário pode escolher o endereço de partida e o endereço de destino.

Há também um checkbox onde pode-se escolher se o endereço de destino é igual o de partida, pois normalmente quando um oficial de justiça sai para cumprir diligências ele volta ao endereço de partida, que pode ser o fórum ou sua própria residência. Essa distinção é importante pois pode modificar a forma como a rota é calculada.

Após o usuário escolher o endereço de partida e o endereço de destino, logo abaixo a um seletor onde pode-se escolher o tempo disponível em horas para cumprir as diligências, isto é, o tempo total que o percurso da rota deverá ser baseado.

Figura 11 – Seletor de tempo para realização das diligências

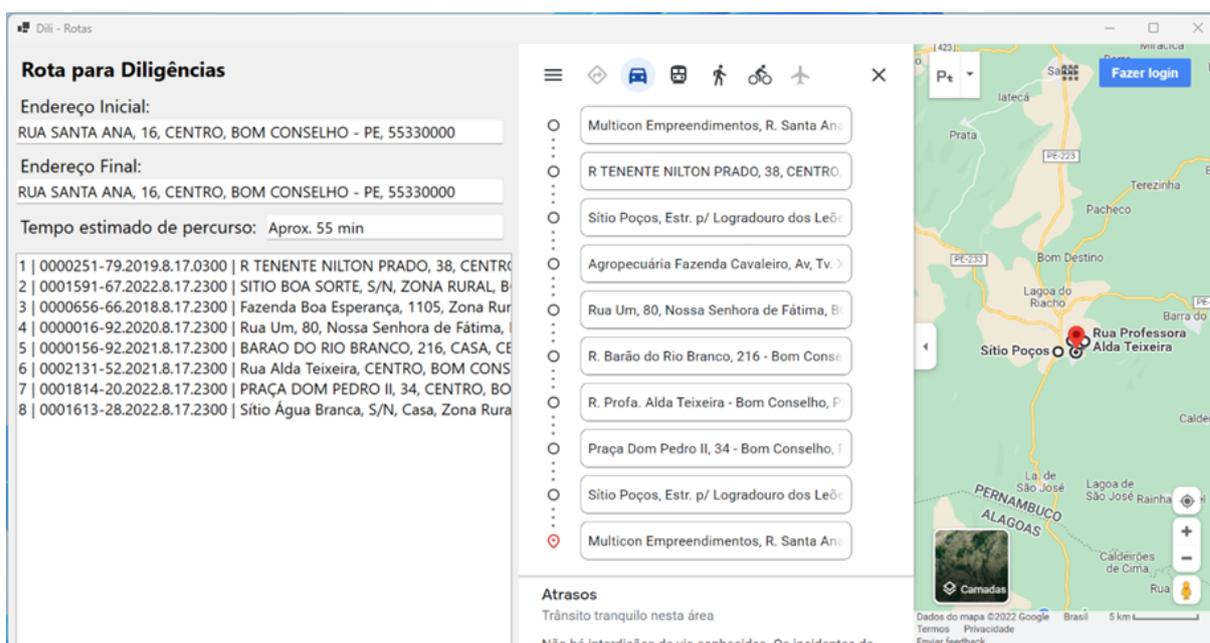


Fonte: Autoria Prória.

Com todas as opções devidamente preenchidas, existem dois botões os quais o usuário pode clicar, o botão “VER ROTA SUGERIDA” e o botão “VER ROTA MANUAL”.

Ao clicar no botão “VER ROTA SUGERIDA”, o sistema calculará automaticamente uma rota e o usuário será apresentado a uma tela contendo a lista de diligências desta rota, e a ordem de visita (o roteiro a ser seguido).

Figura 12 – Tela de apresentação da rota sugerida



Rota para Diligências

Endereço Inicial:
RUA SANTA ANA, 16, CENTRO, BOM CONSELHO - PE, 55330000

Endereço Final:
RUA SANTA ANA, 16, CENTRO, BOM CONSELHO - PE, 55330000

Tempo estimado de percurso: Aprox. 55 min

1		0000251-79.2019.8.17.0300		R TENENTE NILTON PRADO, 38, CENTR
2		0001591-67.2022.8.17.2300		SITIO BOA SORTE, S/N, ZONA RURAL, B
3		0000656-66.2018.8.17.2300		Fazenda Boa Esperança, 1105, Zona Rur
4		0000016-92.2020.8.17.2300		Rua Um, 80, Nossa Senhora de Fátima, B
5		0000156-92.2021.8.17.2300		BARAO DO RIO BRANCO, 216, CASA, CE
6		0002131-52.2021.8.17.2300		Rua Alda Teixeira, CENTRO, BOM CONS
7		0001814-20.2022.8.17.2300		PRAÇA DOM PEDRO II, 34, CENTRO, BO
8		0001613-28.2022.8.17.2300		Sítio Água Branca, S/N, Casa, Zona Rura

Atrasos
Trânsito tranquilo nesta área
Não há interdições de via conhecidas. Os incidentes de

Fonte: Autoria Prória.

Como pode ser visto acima, a tela de rotas exibe o roteiro do percurso, o tempo aproximado e o endereço inicial e final escolhidos. Além disso, ao lado é exibido o mapa do Google Maps com o percurso já marcado.

A forma como o sistema cria estas rotas será explicada nas próximas sessões.

A aplicação permite que o usuário crie rotas manuais, ou seja, rotas em que ele próprio escolha os destinos. Para isso, o usuário deverá selecionar as diligências na listagem e clicar no botão “VER ROTA MANUAL”, como exemplificado na imagem abaixo:

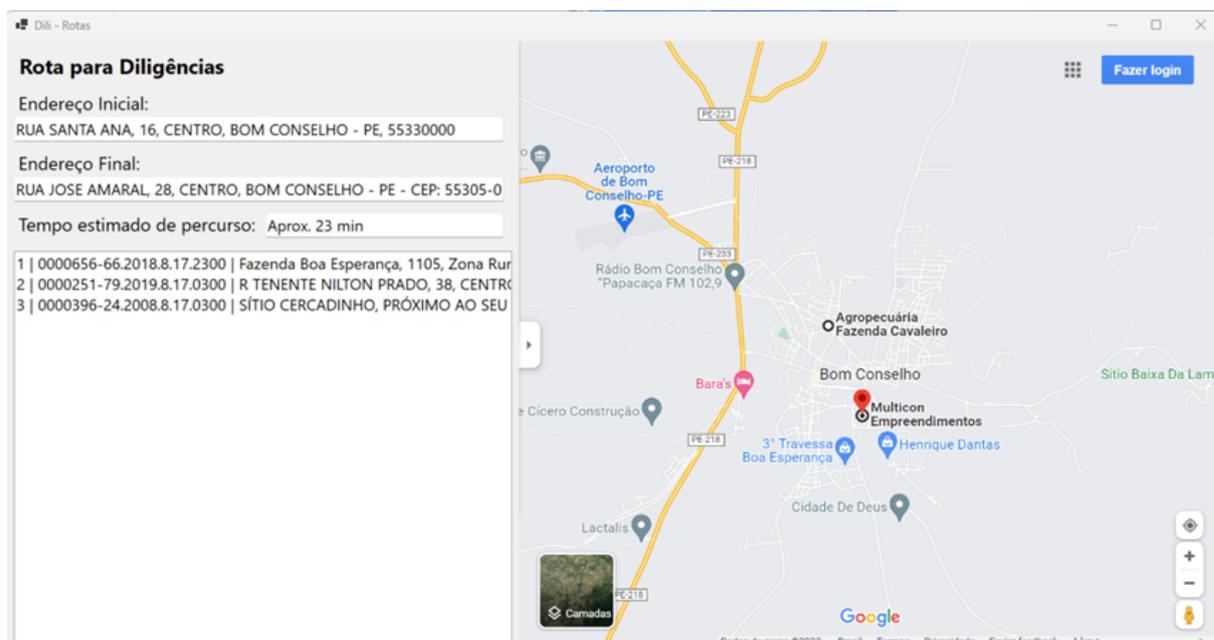
Figura 13 – Opção de criação de rotas manualmente

ID	Número do Processo	Destinatário	Endereço
1	0000402-88.2021.8.17.2300	LOURIVAL ANTONIO CALADO	MAJOR JOSE PEDRO, 3, CENTRO, TEREZINHA - PE - CEP: 55305-000
2	0000402-88.2021.8.17.2300	LOURIVAL ANTONIO CALADO	MAJOR JOSE PEDRO, 3, CENTRO, TEREZINHA - PE - CEP: 55305-000
3	0000396-24.2008.8.17.0300	ERIVALDO DA SILVA TAVARES	SÍTIO CERCADINHO, PRÓXIMO AO SEU TOLOLÔ, RAINHA IZABEL, ZONA RURAL, BOM CONSELHO - PE - CEP: 55330-000
4	0000749-92.2019.8.17.2300	JOBEL CORREIA DE MELO	Zona Rural, s/n, Sítio Limeira, Centro, BOM CONSELHO - PE - CEP: 55305-000
5	0000042-22.2022.8.17.2300	VALERIO DUARTE & CIA LTDA	PRACA DR FRANCISCO PEREIRA LOPES, 9, CENTRO, TEREZINHA - PE - CEP: 55305-000
6	0000232-78.2016.8.17.0300	BANCO DO NORDESTE DO BRASIL SA	AV ENGENHEIRO DOMINGOS FERREIRA, - de 2203 a 3041 - lado ímpar, BOA VIAGEM, BOM CONSELHO - PE - CEP: 55330-000
7	0000251-79.2019.8.17.0300	LEONARDO DOS SANTOS SOUZA	R TENENTE NILTON PRADO, 38, CENTRO, BOM CONSELHO - PE - CEP: 55330-000
8	0001591-67.2022.8.17.2300	MARLUCE RODRIGUES DE DEUS	SÍTIO BOA SORTE, S/N, ZONA RURAL, BOM CONSELHO - PE - CEP: 55330-000
9	0001052-10.2010.8.17.0300	JOSE ABILIO FERRO NETO	, 20, BOM CONSELHO - PE - CEP: 55305-000
10	0000355-80.2022.8.17.2300	MARCELO HENRIQUE DE OLIVEIRA MONTEIRO	RUA LOTUS, 19, 2 RUA AO LADO DA ETE, NOVO JARDIM, BOM CONSELHO - PE - CEP: 55330-000
11	0000656-66.2018.8.17.2300	CARLOS MARCELO TENORIO DE ALMEIDA	Fazenda Boa Esperança, 1105, Zona Rural, BOM CONSELHO - PE - CEP: 55330-000
12	0000016-92.2020.8.17.2300	MARIA ROSANGELA SANTOS VIANA	Rua Um, 80, Nossa Senhora de Fátima, BOM CONSELHO - PE - CEP: 55305-000
13	0000156-92.2021.8.17.2300	MARCOS VENICIUS PADILHA TENORIO	BARAO DO RIO BRANCO, 216, CASA, CENTRO, BOM CONSELHO - PE - CEP: 55305-000
14	0002366-19.2021.8.17.2300	Caroline Marques Bezerra da Silva	Rua São Francisco, 28, São Rafael, BOM CONSELHO - PE - CEP: 55305-000
15	0000329-82.2022.8.17.2300	MARCELINO DOS SANTOS TENORIO	Sítio Lagoa do Pau Ferro, S/N, Zona Rural, BOM CONSELHO - PE - CEP: 55305-000
16	0000665-28.2018.8.17.2300	WELLINGTON SANTANA DA SILVA	SÍTIO QUEIMADA GRANDE, S/N, ZONA RURAL, BOM CONSELHO - PE - CEP: 55305-000
17	0002131-52.2021.8.17.2300	Bruna Izabelly Marques Vieira	Rua Alda Teixeira, CENTRO, BOM CONSELHO - PE - CEP: 55305-000
18	0000550-13.2006.8.17.0300	EUNICE GOMES CARDOSO - ME	, 25, BOM CONSELHO - PE - CEP: 55305-000
19	0001613-28.2022.8.17.2300	ERIVONE CLAUDIO TAVARES DA SILVA	Sítio Água Branca, S/N, Casa, Zona Rural, BOM CONSELHO - PE - CEP: 55330-000

Fonte: Autoria Própria.

Ao selecionar as diligências desejadas e clicar no botão “VER ROTA MANUAL” o usuário será apresentado a mesma tela de rota vista anteriormente, no entanto com os endereços que ele próprio selecionou.

Figura 14 – Tela de rotas exibindo a rota definida pelo usuário



Fonte: Autoria Prória.

4.1.3 Funcionamento do cálculo de rotas

O cálculo de rotas requer algumas etapas importantes, nesta sessão descreveremos como é feita a seleção automática de rotas bem como detalhes técnicos de como os dados geográficos são adquiridos.

4.1.3.1 Dados

Para que a aplicação pudesse ser testada, eram necessários dados de endereços, dessa forma, coletamos aproximadamente 45 endereços aleatórios do município de Bom Conselho, no estado de Pernambuco. Os endereços são em localidades diversas, abrangendo tanto a zona urbana quanto a zona rural.

Os dados foram obtidos utilizando web scraping⁷ através da ferramenta Power Automate⁸, que gerou um arquivo csv com os mesmos. Após isto, foi utilizado um script para cadastrar esses dados em um banco de dados local.

⁷ Web scraping é o processo de coleta de dados estruturados da web de maneira automatizada. (KENNY, 2021)

⁸ Disponível em <<https://powerautomate.microsoft.com/pt-br>>

4.1.3.2 Obtenção de informações geográficas

Com os dados das diligências coletados, se fez necessário adquirir informações geográficas dos endereços, em específico, as distâncias e tempo de percurso entre cada um. Para isso, foi utilizada a API do Google Maps. Esta API fornece um recurso em que se pode consultar uma lista de endereços e é retornada uma matriz de distâncias. No entanto, há uma limitação na quantidade de endereços consultados, devido a quantidade de elementos de retorno, já que uma matriz de n endereços possui tamanho $n \times n$, em nosso caso tínhamos 48 endereços, o que dá um total de 2304 elementos na matriz. Com este tamanho, a API não permite consulta.

Figura 15 – Exemplo de matriz de distâncias obtida através da API

	BC	GA	PI	MA	AR
BC	1 m	46.8 km	39.0 km	174 km	79.3 km
GA	47.7 km	1 m	83.9 km	179 km	124 km
PI	39.8 km	84.0 km	1 m	136 km	44.8 km
MA	172 km	170 km	135 km	1 m	129 km
AR	79.4 km	124 km	45.2 km	128 km	1 m

Fonte: Autoria Própria.

Para contornar este problema, foi optado por realizar consultas individuais, isto é, cada consulta recebe um endereço origem e um endereço destino e retorna os dados de distância e percurso entre eles. Utilizando a biblioteca QuikGraph, foi modelado cada endereço cadastrado no banco de dados local como um nó em um grafo e criadas arestas entre todos os endereços. Para cada aresta, realizamos uma consulta a API para obter os custos (distância e tempo de percurso) e após isto, salvamos a informação da aresta e seus respectivos custos no banco de dados local.

Este procedimento se mostrou importante devido ao fato de que se demanda um certo tempo para realizar todas as requisições, além de que essas requisições à api geram um certo custo computacional e financeiro, já que a API do Google Maps é paga a partir de uma certa quantidade de requisições.

4.1.3.3 Geração da rota de diligências

Com o grafo montado com suas arestas e respectivos custos, foi possível calcularmos rotas de acordo com os critérios escolhidos. Em nosso caso consideramos que o oficial deseja realizar a maior quantidade de diligências possível em determinado período, não nos atentamos

aqui aos gastos com combustível ou outros tipos de restrições, no entanto nada impede que tais demandas sejam implementadas em uma nova versão do nosso protótipo.

Normalmente o oficial parte de um endereço inicial, cumpre as diligências e retorna ao mesmo endereço, dessa forma ao modelarmos esse problema utilizando grafos, como cada endereço é visto como um nó, o percurso que o oficial de justiça faz nada mais é que um ciclo em um grafo. Sendo assim, nossa abordagem foi utilizar um algoritmo que encontrasse os ciclos para um dado nó, este nó era nada mais que o endereço de partida.

Sabemos que em um grafo conexo como o nosso, existem inúmeros ciclos, sendo assim utilizamos o tempo de percurso como um limitador, em outras palavras, obtemos todos os ciclos para um nó de origem cujo custo (tempo de percurso) seja menor ou igual que o determinado pelo usuário. O algoritmo utilizado foi baseado em (ULRICHB, 2011), com pequenas adaptações. O pseudocódigo da função a qual nos baseamos pode ser visto abaixo.

```
# cria uma lista de listas para armazenar os ciclos
# cada ciclo e uma lista
ciclos = [[]]
encontra_ciclos_com_limitacao_de_custos(grafo, origem, limite_custo,
ciclo = []){
  # percorre os nos do grafo
  foreach(no in grafo){
    aresta = (origem, no)
    # verifica se existe a aresta no grafo
    if(aresta esta no grafo){
      # verifica se o no esta no ciclo atual
      if(no esta em ciclo){
        # caso esteja, adiciona o ciclo atual na lista de ciclos
        # apenas se a limitacao de custo for atendida
        if(custo_do_ciclo(ciclo) + aresta.custo <= limite_custo)
          ciclos.add(ciclo);
      }
      else if(custo_do_ciclo(ciclo) + aresta.custo <= limite_custo){
        # caso contrario, adiciona o no no ciclo atual
        # tambem respeitando a limitacao
        ciclo.add(no);
        # chama a funcao recursivamente para o no atual
        encontra_ciclos_com_limitacao_de_custos(grafo, no, limite_custo,
          ciclo = [])
      }
    }
  }
}
```

```

    }
  }
}

```

A partir daí, selecionamos o ciclo que possua maior quantidade de nós e caso existam ciclos com quantidades iguais, selecionamos o que tiver menor tempo de percurso. Ao final deste processo, obtemos a rota que será sugerida ao usuário.

No caso de o endereço de destino ser diferente do endereço de origem, o procedimento é semelhante, no entanto o algoritmo muda um pouco, pois aí não buscamos por ciclos e sim por caminhos no grafo. No entanto o critério é semelhante, ou seja, buscamos pelos caminhos com maior quantidade de nós e menor tempo de percurso. O algoritmo utilizado foi baseado em (GUPTA, 2022) e seu pseudocódigo pode ser visto abaixo:

```

# cria uma lista de listas para armazenar os caminhos
# cada caminho e uma lista
caminhos = [[]]

encontra_caminhos(grafo , origem , destino , caminho=[],
limite_custo){
  # adiciona o no de origem no caminho
  # respeitando o limite de custo
  if(custo_do_caminho(caminho + no) <= limite_custo)
    caminho.add(origem)

# verifica se a origem e igual ao destino
if origem == destino{
  # se sim, adiciona o caminho a lista de caminhos
  if(custo_do_caminho(caminho) <= limite_custo)
    caminhos.add(caminho)
}
else{
# caso contrario , percorre os nos adjacentes do no origem
foreach (no in origem.adjacentes){

# verifica se o no adjacente ja esta no caminho
if (no nao esta no caminho){
  # caso nao esteja , a funcao e chamada recursivamente
  # para o no adjacente
  encontra_caminhos(grafo , no , caminho)
}
}
}
}

```

```
}  
}  
}  
}
```

Após a rota ter sido gerada, utilizamos novamente a API do Google Maps para exibição dos endereços das diligências escolhidas, conforme é apresentado na figura 12.

5 CONSIDERAÇÕES SOBRE O PROTÓTIPO E POSSÍVEIS MELHORIAS FUTURAS

De forma inerente ao processo iterativo de criação de um produto mínimo viável, seu desenvolvimento traz à tona alguns questionamentos que geram aprendizado para garantir melhorias em futuras possíveis implementações.

Não diferentemente, o desenvolvimento de nosso protótipo nos possibilitou enxergar pontos em que há uma necessidade de atenção para que o MVP venha se tornar um produto real e robusto. Nesta sessão listaremos algumas considerações técnicas a respeito da implementação feita neste trabalho.

5.1 COLETA DE ENDEREÇOS

Sem dúvida um dos problemas de destaque que surgiu no início da implementação foi a coleta dos endereços das diligências. Não há um padrão de registro no sistema do PJe e o endereço aparentemente é apenas uma string, em muitos casos o endereço é registrado incompleto, o que impossibilita a realização de uma busca precisa em qualquer API de geolocalização.

Uma alternativa para caso não seja possível a padronização de um formato para os endereços seria investir em formas de tratamento dos dados e uma pré busca, de maneira a converter as strings de endereço na forma de latitude e longitude ou no caso específico do Google Maps, adquirindo o Place ID para cada endereço.

Outra situação que necessita cuidado, é o manejo das requisições para obter os dados de distância entre as localizações, pois dependendo da quantidade de endereços, o número de requisições pode ficar muito alto, o que faz com que este processo leve um certo tempo para ser completado. Algo que pode ajudar é armazenar esses dados em uma base local e fazer consultas a API apenas quando não existirem cadastros localmente. Além disso, como o procedimento de consulta pode demorar, caso a ideia seja implementada de forma concreta para uso real, é interessante que essas consultas ocorram em um horário em que os oficiais não utilizem o sistema, por exemplo, na madrugada.

5.2 CÁLCULO DE ROTA

Como pode ser visto na seção 4.1.3, o a forma como as rotas automáticas são geradas foi implementada de maneira simples, considerando apenas o tempo de percurso entre um endereço e outro, no entanto nada impede que aprimoramentos sejam feitos em versões futuras, como a

utilização de algoritmos mais sofisticados¹ e levando em consideração algumas restrições a mais.

Atualmente os oficiais de justiça podem cumprir diligências por telefone ou o utilizando aplicativos de mensagens instantâneas como o WhatsApp. Em algumas diligências, o mandado expedido possui o contato telefônico da parte a ser visitada. Levar em conta as diligências em que não é preciso deslocamento por parte do oficial traria um bom ganho de tempo, no entanto não há padrão no cadastro de telefones no sistema e essa informação aparece normalmente como texto no documento do mandado, logo para ser adquirida de forma automática pelo sistema, se exigiria um grau de complexidade maior do que o proposto neste trabalho, no entanto é algo computacionalmente possível e que pode ser explorado futuramente.

Também pode ser proveitoso levar em conta as questões de custo, não somente se baseando no tempo, mas também na forma em como será o deslocamento do oficial (carro, moto etc.), além disso pode-se verificar questões de urgência das diligências, bem como preferências do usuário para criar uma espécie de filtro que influencie na geração da rota.

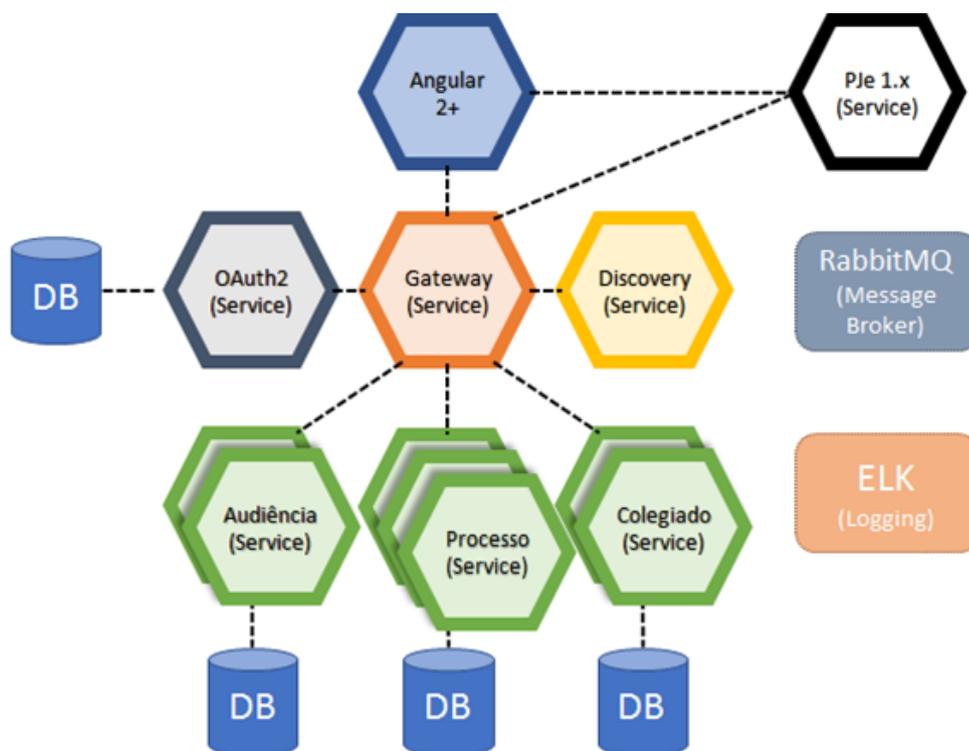
5.3 ARQUITETURA DA APLICAÇÃO

Por ser apenas um protótipo com o objetivo de prova de conceito, a aplicação implementada neste trabalho não possui arquitetura sofisticada, divisão de módulos, nem formas de integração. No entanto nossa ideia é que a funcionalidade que implementamos possa ser integrada em conjunto com o PJe atual, o que de acordo com a documentação disponibilizada pelo CNJ, é plenamente possível.

De acordo com o manual para desenvolvedores do PJe, a partir da versão 2.1 o PJe passou a utilizar uma arquitetura modular baseada em micro serviços que permite adição de novos módulos, desde que sigam o padrão de comunicação estabelecido. (TRF1, 2019)

¹ Alguns algoritmos mais sofisticados e eficientes podem ser vistos em (BIRMELÉ et al., 2013), (CORREIA, 2019) e (LIMA, 2018)

Figura 16 – Arquitetura modular do PJE 2.1



Fonte: Disponível em:

<<https://docs.pje.jus.br/manuais-de-inicializacao/referencia-para-o-desenvolvedor>>. Acesso em: 01 de dez. de 2022.

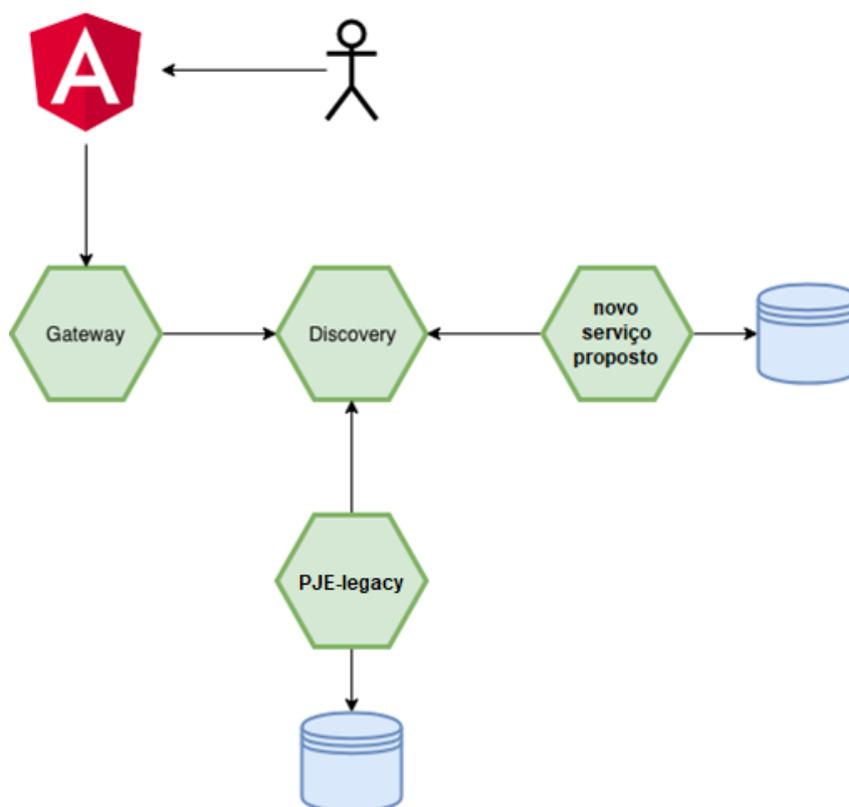
De acordo com a documentação:

Temos o Angular 2+ como porta de entrada, cliente onde o usuário terá acesso aos vários serviços do PJe. O cliente angular irá acessar a API REST através de um serviço de borda, denominado "Gateway". O Gateway nada mais é do que um serviço dentro da nossa nuvem de serviços, este conhece todos os outros e servirá de encaminhador das requisições do cliente.

Cada serviço, ao ser iniciado, se registra no "Discovery" informando que está disponível para aceitar as requisições. O Discovery também se encarrega de realizar load balancing entre os serviços registrados, desta maneira, se temos três instâncias de Processo, nesse serviço Discovery irá distribuir as requisições, de modo a não carregar uma mais que as outras.

Sendo assim, o mais indicado caso esta funcionalidade proposta fosse ser implementada seria a criação de um micro serviço específico que se integrasse a arquitetura já existente. Na própria documentação do PJe, há um modelo que detalha como adicionar novos serviços a aplicação, baseando-se neste modelo, fica a sugestão de uma possível implementação futura para esse novo serviço que implemente as funcionalidades descritas neste trabalho, como pode ser visto no diagrama arquitetural abaixo:

Figura 17 – Arquitetura sugerida para implementação de um novo serviço



Fonte: Disponível em: <<https://docs.pje.jus.br/tutoriais/criacao-de-frontend-e-backend-para-o-pje-2-1>>.

Acesso em: 02 de dez. de 2022.

6 CONCLUSÃO

No decorrer deste trabalho, descrevemos uma proposta de funcionalidade de software, bem como a implementação de um produto mínimo viável que implementa a funcionalidade descrita, além disso, também foi descrita a base teórica que serviu de apoio para o desenvolvimento da proposta.

A ferramenta permitiu atestar a viabilidade das funcionalidades propostas, além de proporcionar a visibilidade de possíveis melhorias. O cálculo automático de rotas se mostrou ter potencial para ajudar os oficiais de justiça logisticamente no cumprimento de diligências.

Não é possível mensurar a quantidade de tempo que seria economizada, ou mesmo se realmente economizaria, pois não foram feitos experimentos com a utilização real da ferramenta, no entanto, atualmente não há nada que cuide dessa questão para os oficiais, logo, é de se esperar que a funcionalidade se mostre eficaz reduzindo custos ou no mínimo sirva como uma opção para o servidor planejar suas rotas.

Como descrito no capítulo 5, o desenvolvimento da ferramenta abriu um leque de possibilidades para melhorias e inserção de novas funcionalidades baseadas no mesmo preceito de roteamento. Algumas sugestões para trabalhos futuros seriam: comparação de algoritmos para obtenção de rotas, implementação do conceito utilizando algum framework de planejamento de rotas já pronto, como é o caso do Itinero¹, GraphHopper² ou o openrouteservice³.

¹ Disponível em <<https://www.itinero.tech>>

² Disponível em <<https://www.graphhopper.com>>

³ Disponível em <<https://openrouteservice.org>>

REFERÊNCIAS

- ALEXANDRIA, A. G. d. O oficial de justiça-suas atribuições e importância para a prestação jurisdicional na área criminal. IDP/EAB, 2010.
- ANUNCIACÃO, J. O. d. A justiça bate à porta: o papel do oficial de justiça na efetividade da prestação jurisdicional. Universidade Federal do Tocantins, 2015.
- ARAKAKI, P. D. J. **Grafos - Busca em Grafos**. 2011. Disponível em: <<https://www.pucsp.br/~jarakaki/grafos/Aula6.pdf>>.
- BIRMELÉ, E.; FERREIRA, R.; GROSSI, R.; MARINO, A.; PISANTI, N.; RIZZI, R.; SACOMOTO, G. Optimal listing of cycles and st-paths in undirected graphs. In: **Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms**. USA: Society for Industrial and Applied Mathematics, 2013. (SODA '13), p. 1884–1896. ISBN 9781611972511.
- BONDY, J. A.; MURTY, U. S. R. et al. **Graph theory with applications**. [S.l.]: Macmillan London, 1976. v. 290.
- CORREIA, S. P. **Otimização de rotas para a entrega de correspondências**. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2019. Disponível em: <https://riut.utfpr.edu.br/jspui/bitstream/1/7401/1/CP_DAMAT_2019_1_05.pdf>.
- CUNHA, C. B. da. Aspectos práticos da aplicação de modelos de roteirização de veículos a problemas reais. **Transportes**, v. 8, n. 2, 2000.
- FALCÃO, J.; HARTMANN, I. A.; KELLER, C. I.; SOUZA, R. R.; MICHENER, G.; JELIHOVSCHI, A. P. G. Políticas públicas do poder judiciário: uma análise quantitativa e qualitativa do impacto da implantação do processo judicial eletrônico (pje) na produtividade dos tribunais. 2018.
- FEOFILOFF, P. **Busca em largura**. 2019. Disponível em: <https://www.ime.usp.br/~pf/algoritmos_para_grafos/aulas/bfs.html>.
- FEOFILOFF, P. **Busca em profundidade**. 2019. Disponível em: <https://www.ime.usp.br/~pf/algoritmos_para_grafos/aulas/dfs.html>.
- FONSECA, F. F.; CUNHA, D. M.; VIEIRA, E. O.; MODENA, C. M. Implicações de novas tecnologias na atividade e qualificação dos servidores: Processo judicial eletrônico e a justiça do trabalho. **Revista Brasileira de Saúde Ocupacional**, SciELO Brasil, v. 43, 2018.
- GUPTA, S. **Print all paths from a given source to a destination**. 2022. Disponível em: <<https://www.geeksforgeeks.org/find-paths-given-source-destination/>>.
- JUNIOR, C. A. de M.; NUNES, R. V.; ASSIS, C. W. C. de; FONSECA, R. de C.; ADRIANO, N. de A.; SANTOS, G. P. dos. O papel da roteirização na redução de custos logísticos e melhoria do nível de serviço em uma empresa do segmento alimentício no ceará. In: **Anais do Congresso Brasileiro de Custos-ABC**. [S.l.: s.n.], 2013.
- KENNY, C. **What Is Web Scraping?** 2021. Disponível em: <<https://www.zyte.com/learn/what-is-web-scraping/>>.

LENARDUZZI, V.; TAIBI, D. Mvp explained: A systematic mapping study on the definitions of minimal viable product. In: IEEE. **2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)**. [S.l.], 2016. p. 112–119.

LIMA, D. A. d. Aplicação do modelo do caixeiro-viajante de programação linear no roteamento de veículos de uma indústria gráfica. Universidade Federal da Paraíba, 2018. Disponível em: <<https://repositorio.ufpb.br/jspui/bitstream/123456789/13321/1/DAL09112018.pdf>>.

OLIVEIRA, G. N. de. Aplicação do problema do caixeiro viajante na distribuição de urnas eleitorais na zona eleitoral de almirante tamandaré-paraná. 2019. Disponível em: <<https://tinyurl.com/2433wktv>>.

OPEN4TECH. **BFS vs. DFS**. 2019. Disponível em: <<https://open4tech.com/bfs-vs-dfs/>>.

OTONI, L. **Processo Eletrônico (PJe) tem tramitação mais rápida no Judiciário**. 2018. <<https://www.cnj.jus.br/processo-eletronico-pje-tem-tramitacao-mais-rapida-no-judiciario/>>.

RIES, E. Minimum viable product: a guide. **Startup lessons learned**, v. 3, p. 1, 2009.

SILVA, S. **Novas atribuições do oficial de justiça no CPC/2015**. 2016. Disponível em: <<https://jus.com.br/artigos/47298/novas-atribuicoes-do-oficial-de-justica-no-cpc-2015>>.

SINDJUSTIÇA-RJ. **SIND-JUSTIÇA REQUER CONVOCAÇÃO DE OFICIAIS DE JUSTIÇA**. 2022. Disponível em: <<https://sindjustica.org.br/fala-diretor/sind-justica-requer-convocacao-de-oficiais-de-justica>>.

SINDOJUS/AL. **Sindojus/AL adota medidas contra sobrecarga de trabalho**. 2022. Disponível em: <<http://sindojus-al.org.br/portal/2022/09/sindojus-al-adota-medidas-contrasobrecarga-de-trabalho>>.

TRF1. **Manual de Processo Judicial Eletrônico**. 2019. <https://portal.trf1.jus.br/data/files/7F/44/E4/EE/1D20E610B05610E6E52809C2/PJE%202.1%20MANUAL%20OFICIAL%20DE%20JUSTI_A.pdf>.

ULRICHB. **Cycle enumeration of a directed graph with multi edges**. 2011. Disponível em: <<https://stackoverflow.com/questions/4625440/cycle-enumeration-of-a-directed-graph-with-multi-edges>>.