



Paulo Filipe
Martins Fournier

Otimização de rotas

Planeamento em viaturas pesadas

Trabalho de Projeto submetida como requisito
parcial para obtenção do grau de **Mestrado em**
Engenharia de Software

Júri

Presidente (Prof. Cláudio Sapateiro)

Orientador (Prof. Nuno Pina Gonçalves)

Vogal (Prof. Miguel Guevara Lopez)

Dezembro de 2022

A felicidade é feita de pequenos momentos especiais.

Agradecimentos

Não poderia deixar de agradecer a todos aqueles que diretamente ou indiretamente me ajudaram e apoiaram durante a elaboração deste projeto.

Começo por agradecer à Escola Superior de Tecnologia de Setúbal, e a todos os docentes que tive o privilégio de conhecer ao longo destes 2 anos do curso de mestrado em Engenharia de Software, com especial foco ao meu professor e orientador Nuno Pina Gonçalves, um ser humano excepcional que sempre me acompanhou e apoiou nesta aventura que se adivinhava difícil e cheia de desafios.

Queria também agradecer ao responsável desta unidade curricular, professor Cláudio Sapateiro que sempre defendeu o interesse do aluno na elaboração e aprovação desta proposta de projeto.

À empresa TRACKiT Consulting, LDA. que se demonstrou sempre disponível para dar o seu feedback e disponibilizar casos reais dos seus clientes para testar e propor melhorias no projeto.

E por fim, à minha esposa Inês Fournier por me ter apoiado em todos os níveis pois, sem a sua compreensão e ajuda, seria impossível conseguir conciliar a minha vida pessoal e laboral com este percurso académico.

Resumo

Atualmente a internet não é vista apenas como uma ferramenta, cada vez mais é parte integrante em diversas áreas de negócio e está enraizada em qualquer empresa que se diz moderna e atual. Com o avanço das tecnologias de comunicação e com o acesso à internet em qualquer parte do mundo, permite que várias empresas estejam constantemente conectadas podendo assim consultar e agir a qualquer instante e em qualquer lugar.

Com esta premissa em mente e com um aumento dos serviços de entregas de mercadorias nacionais e internacionais que se verificou no passado recente, nasceu a ideia de desenvolver um projeto que ajuda as empresas a otimizar a sua atividade de transporte e logística com especial foco em viaturas pesadas. Este tipo de viaturas têm outro tipo de características quando comparadas com viaturas ligeiras, apresentado assim outros desafios para as empresas como garantir que a viatura possa descarregar no local indicado e que as rotas são calculadas de acordo com essas mesmas restrições.

O projeto iniciou com o levantamento do estado da arte atual, tanto nacional como internacional, e após essa análise foram estudadas diferentes alternativas de integração com ferramentas externas onde se integrou com a HERE Technologies para obter a matriz de distâncias e tempos entre todos os locais de carga e descarga, sempre tendo em conta as restrições das viaturas pesadas, bem como a integração da biblioteca da Google AI, nomeadamente a OR-Tools, que trouxe uma grande mais valia ao projeto. O desenvolvimento ainda contemplou a implementação de novas funcionalidades e módulos, trazendo vários desafios que foram ultrapassados com diferentes estratégias aqui descritas.

O objetivo inicial e que foi alcançado consistiu em desenvolver um protótipo funcional que permitisse a um utilizador gerar e definir todos os dados necessários para alimentar o otimizador, como viaturas, armazéns e serviços, e depois otimizar e consultar os resultados obtidos do mesmo através de um *dashboard* completo e intuitivo e que contém toda essa informação resumida e pronta a trabalhar.

O resultado final foi satisfatório e foi conseguido demonstrar que uma ferramenta deste género consegue otimizar os tempos de trabalho dos gestores de tráfego, responsáveis por gerir toda esta atividade, bem como minimizar os custos do transporte ao reduzir as distâncias percorridas e a utilização desnecessária de viaturas.

Palavras-chave: Otimização, Viaturas Pesadas, Serviços, Rotas, Mapas.

Abstract

Currently, the internet is not only seen as a tool, but also increasingly an integral part of several areas of business and is rooted in any company that claims to be modern and current. With the advancement of communication technologies and access to the internet anywhere in the world, it allows several companies to be constantly connected, thus being able to consult and act at any time and in any place.

With this premise in mind and with an increase in national and international goods delivery services that occurred in the recent past, the idea was born to develop a project that helps companies to optimize their transport and logistics activity with a special focus on heavy vehicles, since they have different characteristics when compared to light vehicles, thus presenting other challenges for companies such as ensuring that the vehicle can unload at the indicated location and that the routes are calculated according to these same restrictions.

The project began with a survey of the current state of the art, both nationally and internationally, and after this analysis, different alternatives for integration with external tools were studied, where it was integrated with HERE Technologies to obtain the matrix of distances and times between all locations of loading and unloading, always taking into account the restrictions of heavy vehicles, as well as the integration of the Google AI library, namely OR-Tools, which brought great added value to the project. The development also included the implementation of new features and modules, bringing several challenges that were overcome with different strategies described in the report.

The initial objective, which was achieved, consisted of developing a functional prototype that would allow a user to generate and define all the data necessary to feed the optimizer, such as vehicles, depots and services, and then optimize and consult the results obtained from the optimizer through a complete and intuitive dashboard that contains all this information summarized and ready to work.

The final result was satisfactory, and it was possible to demonstrate that a tool of this kind can optimize the working times of traffic managers, responsible for managing all this activity, as well as minimizing transport costs by reducing the distances travelled and the unnecessary use of vehicles.

Keywords: Optimization, Heavy Vehicles, Services, Routes, Maps.

Índice

Agradecimentos	iv
Resumo	v
Abstract	vi
Lista de Figuras	ix
Lista de Tabelas.....	xi
Lista de Siglas e Acrónimos	xii
Lista de Símbolos.....	xiii
Capítulo 1	1
Introdução	1
Problema	1
Metodologia	2
Objetivo	3
4.1. Cálculo da rota	3
4.2. Otimização de rotas	3
4.3. Restrições de tempo	5
4.4. Restrições de capacidade	5
4.5. Restrições de carga ou descarga.....	5
4.6. O motorista	6
Capítulo 2	7
Estado da arte.....	7
5.1. Análise para cálculos de rotas	7
5.1.1. HERE Technologies.....	7
5.1.2. Tomtom	8
5.1.3. Sygic.....	8
5.2. Análise para otimização de rotas	9
5.2.1. Widescope.....	9
5.2.2. Workwave.....	9
5.2.3. OptimoRoute	10
5.2.4. Google OR-Tools	11
5.3. Análise para gestão de motoristas.....	11
Fundamentos Teóricos	12
Análise e Especificação de Requisitos	17
7.7. Módulos.....	17

7.8.	Requisitos Funcionais	18
7.7.	Requisitos Não Funcionais	20
	Desenho	21
8.1.	Arquitetura Geral	21
8.2.	Arquitetura Física.....	22
8.3.	Processos de negócio	23
8.4.	Interface com o Utilizador	24
8.5.	Planeamento	25
	Implementação	26
7.1.	Metodologia e Ferramentas.....	26
7.2.	Tecnologias.....	27
7.2.1.	<i>Tecnologias no Frontend</i>	27
7.2.2.	<i>Tecnologias no Backend</i>	28
7.2.3.	<i>Base de Dados</i>	29
7.2.4.	<i>Integração</i>	30
	Capítulo 3	39
	Resultados	39
10.1.	Módulo Armazéns	41
10.2.	Módulo Viaturas	41
10.3.	Módulo Serviços.....	42
10.4.	Módulo Otimizador.....	43
10.5.	Análise dos resultados	47
10.5.1.	<i>Dificuldades</i>	54
	Trabalhos Futuros	55
	Conclusão	56
	Bibliografia.....	57
	ANEXO I.....	i

Lista de Figuras

Figura 1 - Grafo exemplificativo do Caixeiro Viajante	4
Figura 2 - Grafo exemplificativo com 4 rotas distintas	5
Figura 3 - Cópia de ecrã do software da widescope	9
Figura 4 - Cópia de ecrã do software da workwave	10
Figura 5 - Cópia de ecrã do software da optimoroute	11
Figura 6 - Solução aproximada à esquerda (boa) vs ótima à direita	13
Figura 7 - Diferença entre o método de cobertura em cone e cobertura circular	15
Figura 8 - Árvore de custo mínimo à esquerda e a sua aumentada à direita	16
Figura 9 - Diagrama da arquitetura geral do sistema	21
Figura 10 - Diagrama de componentes	22
Figura 11 - Processo de negócio da otimização de rotas	23
Figura 12 - Mockup de baixa definição do dashboard	24
Figura 13 - Gráfico gantt com o planeamento do projeto	25
Figura 14 – Exemplo de um quadro Kanban	27
Figura 15 - Exemplo da aplicação da D3js	28
Figura 16 - Exemplo da utilização da HERE Technologies	30
Figura 17 – Exemplo simplificado da integração "geocode"	31
Figura 18 - Exemplo simplificado da integração "calculateroute"	32
Figura 19 - Exemplo de uma matriz de distância entre 10 localizações	33
Figura 20 - Exemplo simplificado da integração "calculatematrix"	34
Figura 21 - Configurações do otimizador	35
Figura 22 - Exemplo simplificado do resultado do otimizador	38
Figura 23 - Cópia de ecrã do painel de gestão de Armazéns	41
Figura 24 - Cópia de ecrã do painel de gestão de Viaturas	42
Figura 25 - Cópia de ecrã do painel de gestão de Serviços	43
Figura 26 - Cópia de ecrã da aplicação, com foco no Otimizador	44
Figura 27 - Painéis com listagem de viaturas e serviços	44
Figura 28 - Painel de consulta das rotas otimizadas	45
Figura 29 - Mapa interativo para consulta dos serviços e das rotas otimizadas	46
Figura 30 - Sumário de alguns pontos importantes sobre a otimização	47
Figura 31 - Rotas otimizadas com a ferramenta Workwave	48
Figura 32 - Configuração dos serviços de acordo com a informação partilhada	48
Figura 33 - Configuração das viaturas alterada	49

Figura 34 - Simulação das rotas otimizadas	49
Figura 35 - Rota otimizada da viatura 00AA00	50
Figura 36 - Rota otimizada da viatura 00BB00	50
Figura 37 - Rota otimizada da viatura 00CC00	51
Figura 38 - Possível conflito entre rotas.....	51
Figura 39 - Comparação entre as rotas 00AA00 e 00CC00	52
Figura 40 - Serviços com janelas horárias definidas no período da tarde.....	52
Figura 41 - Rota otimizada da viatura 00EE00	53

Lista de Tabelas

Tabela 1 - Metodologia adotada para o projeto	2
Tabela 2 - Tabela comparativa entre ferramentas de mapas	8
Tabela 3 - Módulos referentes ao Frontend	17
Tabela 4 - Módulos referentes ao Backend	17
Tabela 5 - Listagem de requisitos funcionais	18
Tabela 6 - Ferramentas de suporte ao projeto.....	26
Tabela 7 - Tecnologias escolhidas para o Frontend	28
Tabela 8 - Tecnologias escolhidas para o Backend	29
Tabela 9 - Bases de Dados escolhidas.....	29
Tabela 10 - Tecnologias integradas no projeto.....	30
Tabela 11 - Comparação entre os tempos de trabalho.....	53

Lista de Siglas e Acrónimos

API	Interface de programação de aplicativos
Backend	Aplicação desenvolvidos no lado do servidor de suporte ao frontend
Browser	Motor de busca
Canal Horeca	Refere-se a distribuição porta-a-porta
Cluster	Agrupamento de elementos semelhantes entre si
CRUD	Create, Read, Update and Delete
CSS	Cascading Style Sheets
Dashboard	Interface com informação resumida e útil
ESTS	Escola Superior de Tecnologia de Setúbal
Framework	Conjunto de bibliotecas para fornecer funcionalidades genéricas
Frontend	Desenvolvimento da interface gráfica
IDE	Ambiente de desenvolvimento integrado
IPS	Instituto Politécnico de Setúba
JSON	JavaScript Object Notation
Mockup	Desenho ou rascunho
Open source	Projetos desenvolvidos em código aberto, não propriamente gratuitos
REST	Representational State Transfer
SDK	Pacote (Kit) de desenvolvimento de software
URL	Localizador uniforme de recursos
WEB	World Wide Web

Lista de Símbolos

m³ Metros cúbicos

Kg Quilogramas

Km Quilómetros

Capítulo 1

Introdução

Atualmente assistimos a um aumento dos serviços de entregas de mercadorias nacionais e internacionais até pela envolvimento pandémica que nos encontramos. O consumo de tempo, atrasos devido ao trânsito, paragens obrigatórias e outras circunstâncias fazem com que este tipo de serviço aumente os custos de uma forma significativa. Surge assim uma oportunidade para o desenvolvimento de uma solução de otimização de rotas de modo a minimizar os custos inerentes a estes problemas que tem levado a muitos investigadores estudarem alternativas que possam ajudar as empresas neste tipo de atividade, fazendo o uso de algoritmos inteligentes.

Este projeto insere-se no contexto do transportador nacional com o objetivo de maximizar a utilização de recursos internos, otimizando assim o transporte com o intuito de minimizar custos e o tempo de entrega ao destinatário.

A ideia original deste projeto nasceu após a perceção, nos diferentes contactos com empresas do setor, da necessidade de encontrar uma solução para o problema que os afeta na sua atividade diária. Atualmente estas empresas contam com um vasto número de gestores de frota que estão focados na alocação de recursos e tentar compreender como podem otimizar manualmente todos os serviços que têm em mãos, avaliando todos os recursos e limitações que têm nesse momento.

Se essa tarefa já parece difícil, ao juntar ao problema várias dezenas de viaturas com outras tantas dezenas de motoristas para realizar centenas de cargas diárias, esta tarefa é transformada em algo virtualmente impossível e certamente o resultado não será otimizado (Vasileios Zeimpekis, 2007).

Problema

Atualmente existem diversos algoritmos de otimização de rotas baseados em viaturas ligeiras com algumas variáveis, como número de viaturas, janelas horárias e capacidades. Estes algoritmos são utilizados, em grande parte, no transporte capilar, que é conhecido por “Canal Horeca”, onde a prestação de serviços ocorre dentro de uma pequena área geográfica.

O desafio do transportador em viaturas pesadas é bem mais complexo pois tem de lidar com diferentes variáveis (Route Optimization Algorithm and Big Data, 2022):

- O local da entrega é interdito a viaturas pesadas?
- Qual a melhor rota a utilizar tendo em conta as restrições existentes para este tipo de viaturas?
- Qual o motorista ideal para realizar o trabalho tendo em conta a sua disponibilidade uma

vez que perante a lei europeia o número de horas disponível é limitado em diferentes parâmetros?

- Onde deverá efetuar os descansos obrigatórios de forma a otimizar os tempos de condução?
- Deverá o transporte ser efetuado com um ou dois motoristas? Ou seja, o transporte a pares é mais dispendioso, mas permite conduzir durante mais tempo, minimizando assim o tempo de trabalho.
- Quais as restrições que existe no local de carga ou de descarga para efetuar essas operações?

Estes são apenas alguns dos desafios presentes no projeto onde terá de lidar com N variáveis de difícil avaliação e controlo.

Metodologia

Desde uma fase inicial que se percebeu da dificuldade e dos desafios que este projeto iria trazer, pelo que foi fundamental definir uma metodologia de trabalho para ter sempre uma orientação bem clara, e perceber se o projeto estaria ou não fora do plano traçado. Essa metodologia foi definida e segue os seguintes passos gerais:

Tabela 1 - Metodologia adotada para o projeto

<i>O Problema</i>	Definir e enquadrar bem o problema que é pretendido resolver. Ou seja, clarificar o âmbito do projeto e as suas fronteiras.
<i>Os Objetivos</i>	Uma vez identificado o problema, é necessário definir quais os nossos objetivos. O que pretendemos fazer para tentar resolver o nosso problema?
<i>Estado da Arte</i>	Estando definido os objetivos, é necessário perceber que tipos de trabalhos já foram desenvolvidos na área para tentar resolver estas questões. Existem já padrões identificados para atacar totalmente ou parcialmente alguns dos objetivos?
<i>Estudo de Mercado</i>	É igualmente relevante perceber se já existem soluções no mercado. Que tipo de soluções são, como são apresentadas ao público e que funcionalidades estão disponíveis.
<i>Fundamentos Teóricos</i>	Após estarem identificadas as tecnologias a utilizar, é fundamental entender como as mesmas funcionam. Sem esse conhecimento será difícil perceber o seu comportamento ou tentar identificar e corrigir alguma questão.
<i>Análise e Requisitos Funcionais</i>	Com os objetivos traçados, é necessário decompô-los em requisitos e funcionalidades e assim agrupá-los em módulos bem definidos e delimitados.
<i>Desenho</i>	Com naturalidade a fase de desenho vem de seguida. Até agora já sabemos o que queremos desenvolver, agora temos de perceber como o vamos implementar e concretizar.

<i>Tecnologias</i>	Está na fase de identificar quais as tecnologias necessárias ou que se enquadram neste projeto. É necessário estudar as melhores alternativas e escolher apropriadamente a melhor.
<i>Implementação</i>	Finalmente a fase de implementação inicia, com o desenvolvimento de código e com a integração das ferramentas externas, segundo uma metodologia de desenvolvimento bem definida.
<i>Resultados</i>	É fundamental estudar e analisar os resultados obtidos e proceder às correções necessárias até chegar ao produto final desejado. Será também nesta fase que vamos perceber se o objetivo traçado foi ou não atingido na sua totalidade ou parcialmente.
<i>Conclusões e Trabalhos Futuros</i>	Chegado a esta fase é sempre importante concluir e perceber o que ficou por fazer e identificar o que ainda tem de ser feito para tornar o produto numa solução vendável e que se enquadre bem numa empresa.

Objetivo

Este projeto surge da necessidade de trazer para o mercado de gestão de frotas novos métodos de controlo e otimização no transporte de cargas. Mais concretamente, o objetivo final será desenvolver ou encontrar ferramentas que possam trabalhar todas as questões que serão abordadas de seguida, cruzando os resultados das diferentes restrições e devolver um conjunto de rotas ótimas (ou aproximadamente) para realizar todos os serviços possíveis.

Neste sentido, iremos abordar os objetivos técnicos para a solução a desenvolver.

4.1. Cálculo da rota

O primeiro objetivo deste projeto passa por permitir calcular uma rota de um ponto A ao ponto B, tendo em conta as restrições de uma viatura. Hoje grande parte dos motores de rotas, como o *google maps* ou *waze* apenas permitem o cálculo para viaturas ligeiras e não para viaturas pesadas. Uma viatura pesada obedece a regras diferentes das ligeiras, sendo que o acesso a uma determinada estrada ou zona poderá estar restrita devido às suas características. Além disso, dentro das viaturas pesadas existe um vasto leque de características que podem ter impacto nessa mesma rota, como o peso total, dimensão, número de eixos, tipo de carga transportada, entre outros.

Assim, o primeiro objetivo passará por encontrar ferramentas que permitam calcular a melhor rota entre dois pontos tendo em conta as restrições das viaturas, sejam elas ligeiras ou pesadas.

4.2. Otimização de rotas

Um dos problemas mais famosos neste tipo de solução é o problema do Caixeiro Viajante

em que se pretende encontrar o caminho mais curto para visitar todos os clientes que se encontram em locais distintos. Este problema pode ser exemplificado por um grafo em que os nós representam os locais e os arcos a distância entre os dois nós.

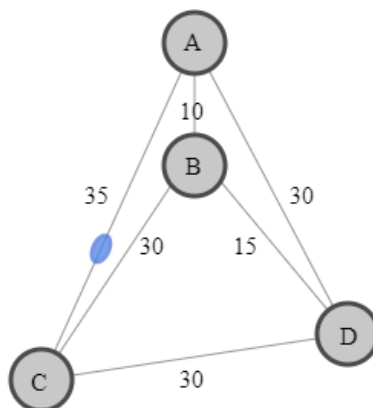


Figura 1 - Grafo exemplificativo do Caixeiro Viajante

Neste exemplo simples apresenta na Figura 1, com apenas 4 locais, conseguimos calcular o caminho mais curto como sendo $35+30+15+10$ que dá 90. Esta questão torna-se cada vez mais difícil de calcular quando o número de locais aumenta. Aqui apenas temos 6 rotas para as 4 posições, mas se aumentarmos para 10 locais o número de rotas aumenta para 362880 e se forem 20 estamos a falar de um valor enorme, ou seja, 2432902008176640000 rotas possíveis (Vehicle Routing, 2022). Obviamente que isto torna-se impossível de calcular manualmente e mesmo os computadores mais sofisticados têm dificuldade em percorrer todas estas opções em tempo útil. Quando falamos de um número bem mais elevado de posições é fisicamente impossível efetuar esses cálculos da forma tradicional, pelo que será fundamental aplicar algoritmos otimizados que nos permitam resolver esse problema de uma forma eficaz e em tempo útil.

Um outro problema a resolver será a otimização tendo em conta a disponibilidade de várias viaturas como está representado na Figura 2. Uma coisa é otimizar uma rota para um conjunto de localizações que será visitado por uma única viatura, diferente será se essas mesmas localizações poderem ser visitadas por múltiplas viaturas. Assim, teremos de encontrar o conjunto ótimo de rotas para várias viaturas (Benmansour, 2020).

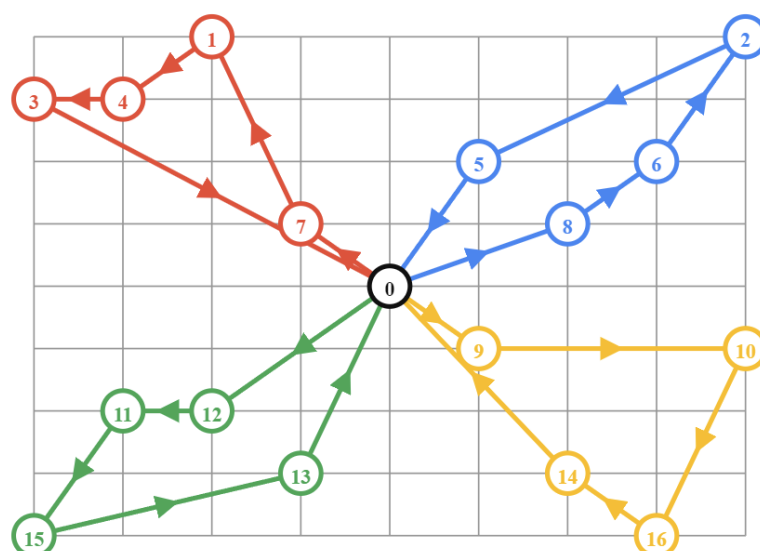


Figura 2 - Grafo exemplificativo com 4 rotas distintas

4.3. Restrições de tempo

Encontrar o caminho mais curto é apenas uma das questões. Para além do mais curto pretende-se combinar a unidade tempo pois o mais curto poderá não ser o mais rápido. A juntar a este problema temos as tais janelas temporais para a entrega ou recolha. Por vezes existe a necessidade de limitar a entrega ou recolha num determinado intervalo de tempo por diferentes motivos, pelo que é fundamental conseguir limitar o sistema a tais restrições (Bräysy, 2005).

4.4. Restrições de capacidade

Para além das restrições faladas anteriormente, a capacidade de uma viatura é claramente relevante aquando da otimização da rota. É lógico pensar que numa frota poderemos ter viaturas com diferentes limitações de capacidade tanto ao nível do peso como do volume, isto em transporte de mercadorias. Quando falamos em transporte de passageiros o tipo de capacidade muda para pessoas ou para tipo de limitação da pessoa se estivermos a falar de ambulâncias (ex: se tem capacidade de levar uma maca ou se tem sistema de oxigénio).

4.5. Restrições de carga ou descarga

Quando falamos de viaturas pesadas é fundamental perceber que tipo de limitações poderão existir no momento da carga ou da descarga. Para além de provavelmente ser necessário ajuda externa para movimentar a carga, é importante saber quanto tempo demoramos a carregar o conteúdo bem como descarregar o mesmo no local de destino. Além disso teremos de perceber quantas viaturas podem carregar em simultâneo no armazém pois nem todos os locais têm a capacidade de iniciar a carga em simultâneo em toda a sua frota.

4.6. O motorista

Após a obtenção das rotas ótimas, será necessário encontrar e alocar um ou dois motoristas que possam efetuar o trajeto em questão. A lei europeia sobre o controlo dos tempos de condução, pausa, descanso e trabalho são bastante complexas, mas bem controladas tanto em paragens realizadas por agentes da autoridade durante o percurso ou mesmo por agentes fiscalizadores que visitam as diferentes empresas e analisam os últimos 2 anos de atividade dos motoristas a fim de encontrar qualquer irregularidade.

Um dos objetivos será então integrar com ferramentas que devolvam a disponibilidade de cada motorista e aí sugerir quais os motoristas disponíveis para realizar cada percurso.

Capítulo 2

Estado da arte

Uma vez conhecido o problema e quais os objetivos retidos para este projeto, nesta seção será abordada soluções que existem disponíveis no mercado e que possam oferecer soluções equivalentes ou parciais, verificar estudos ou trabalhos existentes realizados por terceiros bem como analisar ferramentas, componentes ou tranches de código já existentes que possam ser reutilizados ou integrados neste projeto.

5.1. Análise para cálculos de rotas

Não sendo o desenvolvimento de algoritmos ou de ferramentas para o cálculo de uma rota o objetivo, neste ponto será apresentado um levantamento de sistemas de mapas profissionais para viaturas pesadas e ligeiras que efetuam esse mesmo cálculo através de integrações utilizando API ou SDK. Sendo uma necessidade muito específica, não existe um vasto leque de opções disponíveis no mercado que sejam completas e com uma grande comunidade de forma a ajudar no processo de integração. Outra grande limitação é que todas as ferramentas avaliadas não são gratuitas e nem sempre estão disponíveis para períodos de avaliação. Após esse primeiro levantamento foram eleitas as seguintes 3 ferramentas.

5.1.1. *HERE Technologies*



É uma empresa global com 35 anos de existência e presente em 56 países que está focada no desenvolvimento de serviços para integração com serviços externos de localização, como o cálculo de rotas, georreferenciação inversa e mapas. A documentação avaliada para este projeto foi vasta, mas com foco na **routing-api** (HERE Routing API, 2022) que contém todos os serviços necessários para calcular uma rota entre 2 ou mais pontos bem como um serviço poderoso denominado de **matrix-routing** (Here Matrix Routing API, 2022) que efetua o cálculo da distância e tempo entre as diferentes localizações enviadas.

5.1.2. Tomtom



A Tomtom foi fundada em 1991 e focou-se inicialmente no desenvolvimento de sistemas de navegação para o mercado particular. Mais tarde foi crescendo e entrou no mercado empresarial em particular no desenvolvimento de ferramentas para a gestão de serviços. Só mais recentemente tornou-se numa empresa voltada para o desenvolvimento de serviços para software externos, como mapas, georreferenciação, cálculo de rotas, entre outros. Foi avaliada a documentação com especial foco na API de **routing** (Routing API and Extended Routing API, 2022) e da **matrix-routing** (Routing API and Extended Routing API, 2022).

5.1.3. Sygic



Por fim, a última empresa avaliada é eslovaca e foi fundada em 2004. Focada no desenvolvimento de software para a navegação em Android e iOS foi só mais recentemente que entrou no mercado de serviços web, com o desenvolvimento de um SDK para mapas bem como de serviços para integração via API (Routing API, 2022).

Após percebermos o que são essas empresas e o que podem oferecer, a Tabela 2 ajuda a perceber a diferença entre elas.

Tabela 2 - Tabela comparativa entre ferramentas de mapas

	Tomtom	HERE	Sygic
<i>API para integração</i>	SIM	SIM	SIM
<i>SDK para integração</i>	SIM	SIM	SIM
<i>Comunidade, fóruns e ajuda online</i>	BOA	MÉDIA	FRACA
<i>Preço</i>	ALTO	ALTO	MÉDIO
<i>Flexibilidade para Trial</i>	BAIXA	MÉDIA	MÉDIA
<i>Cálculo para viaturas Ligeiras</i>	SIM	SIM	SIM
<i>Cálculo para viaturas Pesados</i>	SIM	SIM	SIM
<i>Nível de caracterização da viatura pesada</i>	ALTA	ALTA	MÉDIA
<i>Serviço de rotas em matriz</i>	SIM	SIM	NÃO
<i>Leque de serviços disponíveis</i>	EXCELENTE	EXCELENTE	MÉDIO
<i>Facilidade de integração</i>	FÁCIL	FÁCIL	MÉDIA

Após esta análise a *Here Technologies* foi o software eleito para este projeto devido à sua API completa e de fácil implementação, bem como uma documentação extensa e com alguma comunidade online para dar suporte em possíveis complicações na sua integração. Outro ponto que posso nesta escolha foi a sua possibilidade de poder obter uma licença trial para utilização de todas as interfaces necessárias para este projeto.

5.2. Análise para otimização de rotas

Nesta seção será demonstrada o resultado do levantamento efetuado ao mercado sobre o tipo de ferramentas existentes e mais utilizadas na otimização de rotas. O levantamento compreendeu a capacidade de integração das mesmas com ferramentas externas, bem como a possibilidade de as utilizar como ferramentas complementares ao cálculo de rotas.

5.2.1. Widescope

Começando por esta empresa, não pelo seu grau de utilização na comunidade, mas por ser uma empresa portuguesa, formada em 2003, tem como único foco o desenvolvimento de serviços para a otimização de rotas. Estando mais focada no pacote à medida, dispõem de alguns serviços que podem ser utilizados e integrados por entidades externas, mas a sua presença na comunidade é muito reduzida ou praticamente inexistente dificultando qualquer iniciativa de integração. Esta empresa desenvolveu uma solução denominada por Routyn mas cujo a expressão no mercado é muito reduzida (Routyn, 2022).

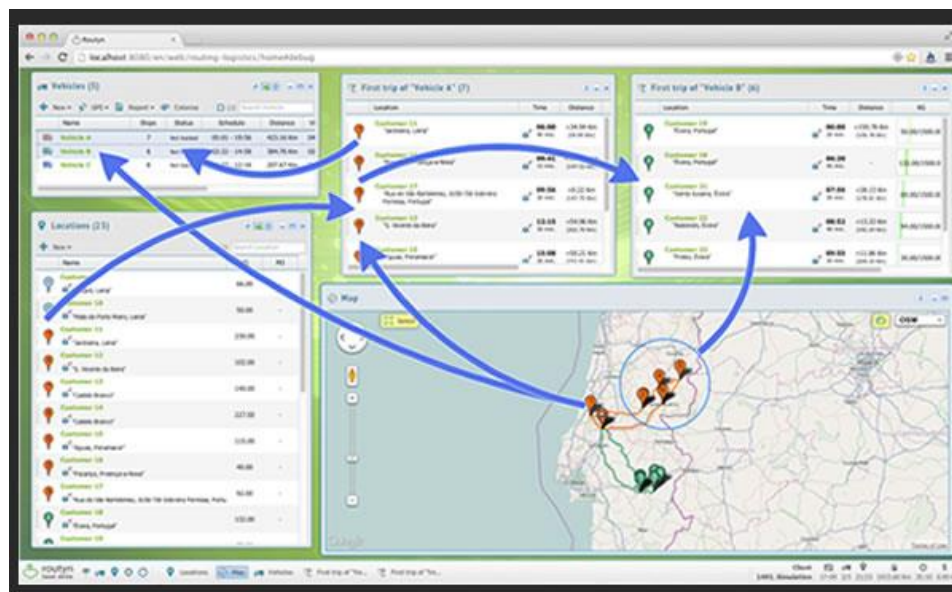


Figura 3 - Cópia de ecrã do software da widescope

5.2.2. Workwave

A **Workwave** é uma empresa americana que entrou no mercado da otimização de rotas após a aquisição da **Viamente**, uma empresa italiana e líder no mercado europeu nessa altura (WorkWave Routing Engine, 2022).

Apesar de ser um software completo que também contempla uma API que permite a

integração com qualquer serviço externo, está limitada a viaturas ligeiras fugindo ao foco deste projeto pois pretendemos a otimização de rotas em viaturas pesadas. Alias esse é o grande problema da generalidade das ferramentas existentes no mercado: não permitem a otimização com viaturas pesadas.

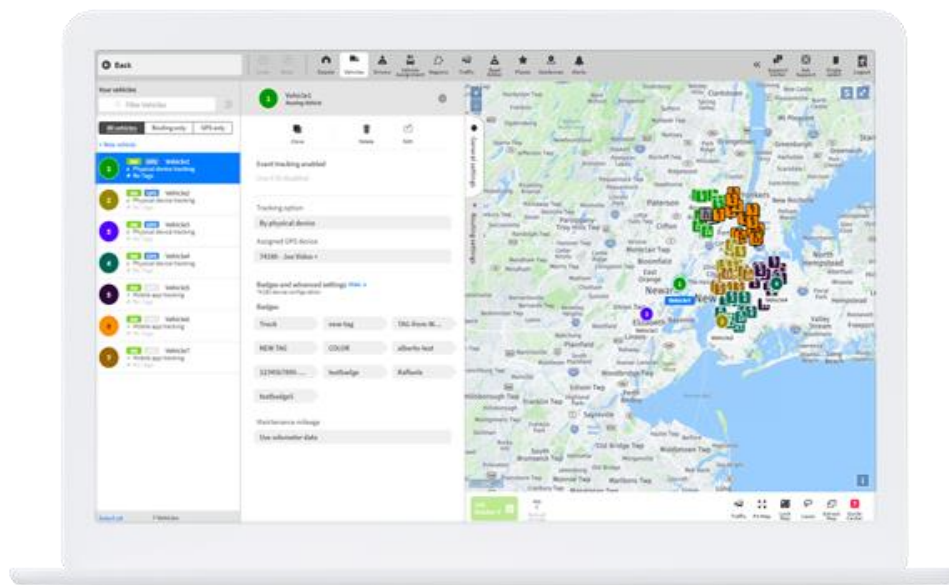


Figura 4 - Cópia de ecrã do software da workwave

5.2.3. *OptimoRoute*

Após uma extensa procura foi possível encontrar uma empresa que apresenta um software de otimização de rotas para viaturas pesadas. Mas depois de ser realizada uma investigação mais detalhada ficou patente que esta ferramenta é uma extensão do software da Workwave já anteriormente abordado, mas com algumas configurações relativamente a viaturas pesadas, mas essas definições são ligeiras e não resolvem o problema dos utilizadores. Apesar deste software ter alguma, mas ainda assim reduzida, presença na comunidade, a informação disponível é pouca o que dificulta perceber a sua qualidade e capacidade de integração (API Reference v1.18, 2022).

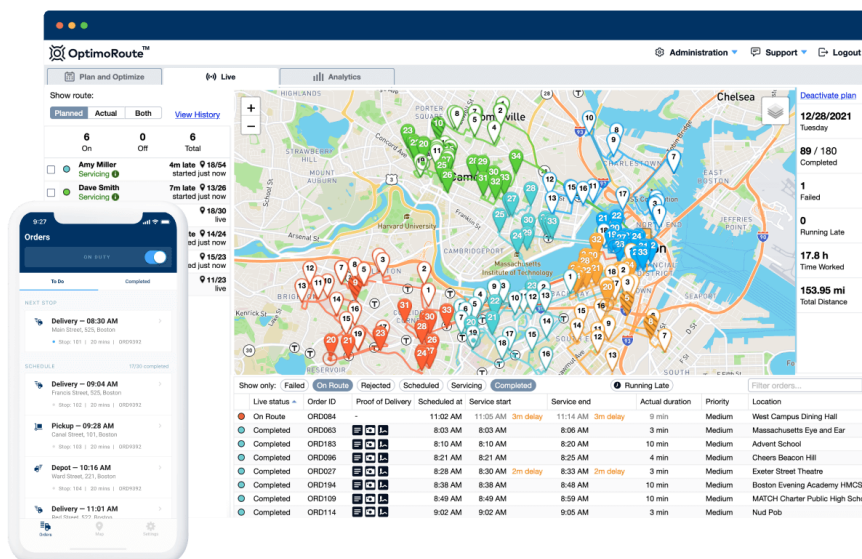


Figura 5 - Cópia de ecrã do software da optimoroute

5.2.4. Google OR-Tools

O OR-Tools é um software *opensource* da Google para otimização de diferentes questões, ajustado para lidar com os problemas mais difíceis do mundo e em particular na definição de rotas de veículos, fluxos, programação linear e na programação de restrições.

Este software pode ser facilmente integrado em qualquer outro, independentemente da linguagem escolhida, sendo assim uma excelente opção para qualquer solução que esteja a ser desenvolvida. Esta ferramenta não tem como objetivo resolver o nosso primeiro problema, que é o cálculo da rota entre dois ou mais pontos, mas sim focar-se na resolução do segundo problema, que é calcular, de uma forma eficiente e otimizada através de algoritmos próprios para o efeito, os caminhos otimizados com as diferentes restrições já aqui abordadas.

Para além de uma documentação completa, a comunidade online parece consistente e forte sendo uma excelente ajuda num processo de implementação, pelo que se torna num excelente candidato para este projeto.

5.3. Análise para gestão de motoristas

O último ponto estudado nesta fase foi a possível integração com ferramentas de gestão dos tempos legais de condução dos motoristas, tempos esses controlados e registados nos tacógrafos. Um tacógrafo é um dispositivo instalado nas viaturas pesadas (e que também serão instalados nas ligeiras já a partir de 2023) e que registam e monitorizam os tempos de uso, distância e velocidade da viatura bem como o motorista ou motoristas que estavam presentes em cada momento (How to deal with tachograph failures, 2022).

Essa informação registada pode ser obtida por serviços externos através de integrações com sistemas de gestão de frota, que disponibilizam essa informação em tempo real nas suas plataformas para ajudar os gestores de frota a avaliarem os respetivos tempos e, assim, otimizarem os tempos dos motoristas e tornar a empresa mais eficiente.

Assim, uma possível integração terá de passar sempre por uma empresa especializada em gestão de frota de viaturas pesadas, com tecnologia de ponta e que permita a integração com sistemas externos. Para esse efeito foi escolhido a empresa TRACKiT devido à sua relação com a Escola Superior de Tecnologia e por ter uma API completa, o que permitirá integrar com alguma facilidade.

Fundamentos Teóricos

Perceber como funcionam os diferentes algoritmos de otimização disponíveis na biblioteca da OR-Tools é fundamental. Sem esse conhecimento não é possível determinar qual o melhor algoritmo a aplicar para a estratégia que temos definido. Além disso, a escolha desse mesmo algoritmo pode influenciar o modo como construímos e definimos todo o sistema.

É também importante perceber que o tempo necessário para resolver uma solução de otimização de rotas cresce exponencialmente à medida que o tamanho do problema também aumenta. Para problemas suficientemente grandes, o OR-Tools (ou qualquer outro software de definição de rotas) pode levar anos para encontrar a solução ideal. Por esse motivo, às vezes o algoritmo retorna soluções que são boas, mas não ótimas, como demonstra a Figura 6, pois também é importante que a resposta seja retornada em tempo útil para o utilizador. Apesar de existirem outras bibliotecas que trabalham melhor o algoritmo ótimo, esta ferramenta enquadra-se melhor na resolução de problemas mais comuns.

A biblioteca vem com diferentes algoritmos, alguns idênticos entre si e outros com paradigmas bastante diferentes, o que importa perceber o mecanismo de cada um. De seguida estão representados os algoritmos tidos em conta nesta implementação, sendo que existem ainda outros que podem ser consultados na documentação oficial (Routing Options, 2022).

Cada algoritmo está orientado a resolver melhor um determinado problema, em que raramente devolvem uma solução ótima, mas sim aproximada pelo que convém escolher corretamente o algoritmo a utilizar para o nosso problema (Yaixin Wu, 2022).

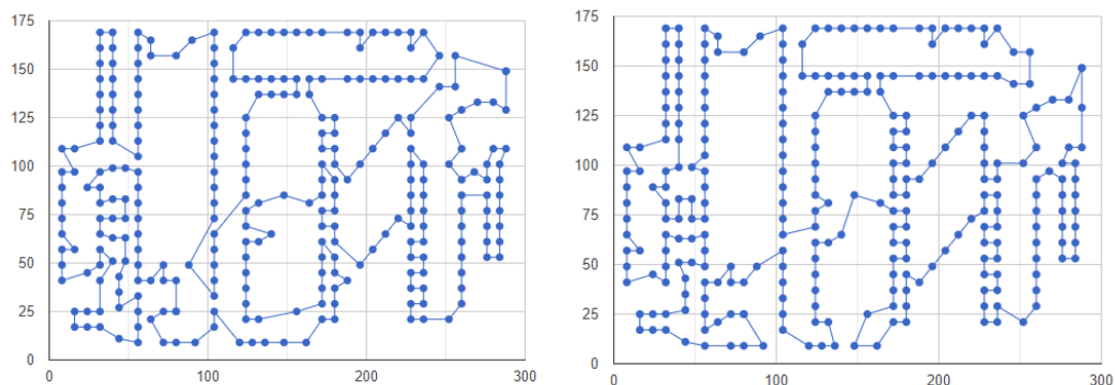


Figura 6 - Solução aproximada à esquerda (boa) vs ótima à direita

PATH_CHEAPEST_ARC

DESCRIÇÃO	A partir de um nó "inicial" da rota, é ligado ao nó que produz o segmento de rota de custo mais baixo e, de seguida, a rota é estendida ao iterar até ao último nó adicionado na rota.
AVALIAÇÃO	Este algoritmo apenas tem aplicação em problemas bastantes simples, aumentado o tempo de execução exponencialmente à medida que vamos acrescentado restrições no sistema. Por esse motivo não será a melhor opção para esta solução.

PATH_MOST_CONSTRAINED_ARC

DESCRIÇÃO	Semelhante ao PATH_CHEAPEST_ARC, mas os arcos são avaliados com um seletor de comparação que favorecerá primeiro o arco mais restrito.
AVALIAÇÃO	Tal como o algoritmo anterior, este tem como base o mesmo mecanismo pelo que falha na sua aplicação para este projeto.

SAVINGS

DESCRIÇÃO	O algoritmo Savings de Clarke & Wright desenvolvido em 1962 foi dos primeiros a ser trabalhado para resolver o problema de entregar diversos clientes a partir de um armazém, apesar de ter sido já desenvolvido para ser executado num computador, foi também definido um conjunto de procedimentos manuais para que fosse utilizado manualmente (Clarke, 1964).
AVALIAÇÃO	O resultado obtido é meramente académico e o mesmo não tem a capacidade de evoluir para ser integrado todo o conjunto de restrições que foi abordado anteriormente.

SWEEP

DESCRIÇÃO	O algoritmo Sweep de Wren & Holliday, tentou resolver o problema de existirem diferentes armazéns de recolha para proceder a diferentes entregas. Até então o paradigma seria sempre a partida de um local para satisfazer as necessidades dos outros (Holliday, 1972).
------------------	---

AVALIAÇÃO	Este algoritmo não está disponível ainda na versão compilada em Python, pelo que não foi possível avaliar o comportamento do mesmo.
------------------	---

CHRISTOFIDES

DESCRIÇÃO	Algoritmo Christofides (é na verdade, uma variante desse algoritmo pois utiliza uma correspondência máxima de um subconjunto em vez de uma correspondência máxima de todo o conjunto, o que não garante o fator de aproximação como acontece no dilema do Caixeiro Viajante). Funciona nos modelos genéricos de cálculo de rotas de viaturas, estendendo uma rota até que nenhum nó possa ser inserido (Christofides, 1976).
AVALIAÇÃO	Apesar de ser um dos primeiros algoritmos a ser trabalhado em graph de ordem 3, não consegue resolver problemas mais complexos.

BEST_INSERTION

DESCRIÇÃO	Constrói iterativamente uma solução inserindo o nó de menor custo na posição também de menor custo. O custo de inserção é baseado na função de custo global do modelo de cálculo de rotas, mas apenas funciona para nós que possam ser opcionais.
AVALIAÇÃO	Será uma boa abordagem para tornar alguns dos serviços opcionais, mas neste caso obriga que todos os nós o sejam, algo que não vai ao encontro das necessidades dos utilizadores.

PARALLEL_CHEAPEST_INSERTION

DESCRIÇÃO	Constrói iterativamente uma solução inserindo o nó com menor custo na posição, também com menor custo. O custo de inserção é baseado na função de custo do arco. Esta solução é a ideal quando trabalhamos com entregas e recolhas, sendo este o caso.
AVALIAÇÃO	Este algoritmo é o que melhor se enquadra no problema aqui retratado pois foi desenvolvido, já em 1981, a pensar em rotas com um conjunto de restrições laterais, tentando assim resolver os problemas comuns na definição de rotas de viaturas. Desde este passo inicial, vários documentos científicos foram escritos baseados neste com o objetivo de evoluir o mesmo, integrando novas restrições no seu cálculo (Savelsbergh, 1989). A base deste algoritmo é separar as responsabilidades de decisão sobre a atribuição de serviços a viaturas da sequência de entregas. Esta novidade foi um avanço para este tipo de processos, mas ao mesmo tempo, trouxe outros desafios pois algumas decisões eram deixadas para o fim. Por exemplo, a restrição das janelas horárias era calculada separadamente, mas tem impacto direto na atribuição a uma viatura, sendo desaconselhado deixar esta decisão para o final. Os algoritmos tradicionais calculavam um resultado em duas fases distintas e consecutivas, em que a primeira fase consistia em associar os serviços a viaturas baseado numa função que resolvia a problemática do custo da viagem. A segunda fase, após a atribuição estar concluída, é o cálculo do percurso o qual é definido através da resolução da problemática do Caixeiro Viajante. Este

processo obrigava a voltar à fase 1 sempre que se encontrava uma restrição não ultrapassável na fase 2 em que esse serviço teria de ser efetuado por outra viatura.

Ao contrário desses algoritmos, este tenta resolver a problemática em paralelo, ou seja, as rotas não são construídas de forma sequencial, mas sim paralelamente. O algoritmo está implementado numa metodologia pensada em inserção, baseada nas seguintes 3 questões como guias:

- a) Como é escolhido o conjunto de m rotas iniciais, cada uma visitando apenas um único cliente?
- b) Qual cliente é o próximo selecionado para ser inserido na solução atual?
- c) Onde será inserido?

A resposta à primeira pergunta, ou seja, o seu estado inicial, é fundamental para o arranque do processo, uma vez que as rotas vão sendo incrementadas segundo este guia. Este estado inicial pode ser calculado utilizando diferentes métodos, como o de cobertura circular, que é dos mais aceites.

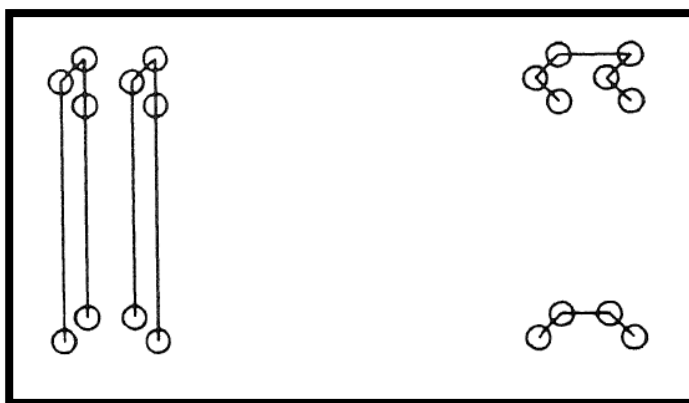


Figura 7 - Diferença entre o método de cobertura em cone e cobertura circular

O método de cobertura circular representado na Figura 7 funciona da seguinte forma. Para cada cliente é determinado o menor círculo com o centro neste de modo que a quantidade total de serviços dentro do círculo esteja próxima da capacidade do veículo. De seguida vamos aumentando os círculos dos pontos associados até incluir todos os pontos. De notar que esta abordagem é independente do local do armazém original.

De realçar que não é parte integrante deste trabalho, a demonstração matemática deste tipo de algoritmos, mas apenas demonstrar superficialmente como os mesmos funcionam e abordar a lógica que está por detrás deste mecanismo.

Dito isto, o próximo passo consiste em determinar os caminhos mais curtos e ao invés de calcular diretamente as distâncias entre cada nó, que vai aumentar exponencialmente o consumo dos recursos das máquinas, é necessário calcular uma árvore de custo mínimo e depois desenvolver uma árvore aumentada, visível na Figura 8, que tem como base a árvore de custo mínimo mais um conjunto de arcos extra para facilitar o acesso entre os nós.

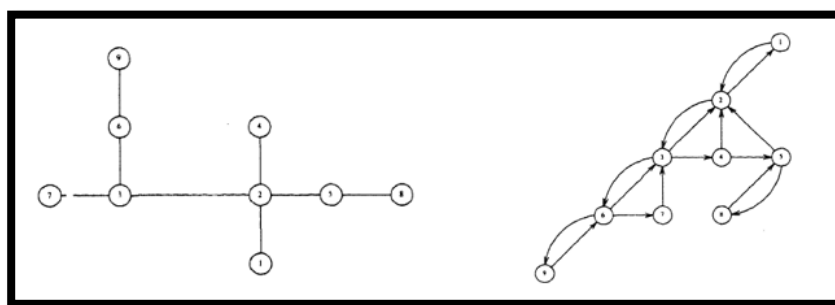


Figura 8 - Árvore de custo mínimo à esquerda e a sua aumentada à direita

Uma vez respondida a primeira pergunta, é necessário determinar qual o próximo serviço selecionado. Nesta fase são determinadas a melhor e segunda melhor rota para cada serviço que ainda não tenha sido atribuído a uma rota, e sempre que um serviço é atribuído será necessário percorrer todos os serviços não atribuídos novamente e calcular uma vez que as melhores rotas e segundas melhores poderão ter alterado. No final desse cálculo, um serviço é selecionado com base na diferença entre o custo associado à melhor e à segunda melhor designação viável.

Finalmente, será necessário atribuir o serviço selecionado a uma rota disponível. Nesta fase são verificados duas dimensões, a viabilidade e a rentabilidade. Primeiramente é determinado se ao inserir o serviço a rota fica viável, ou seja, não incumpra em nenhuma das restrições definidas, como restrições de janelas horárias ou de capacidade. Validada a sua viabilidade, a escolha cai sobre a rota mais rentável, pois o mesmo serviço colocado em rotas diferentes poderá ter rentabilidades dispares, como ter tempos de condução ou distâncias superiores entre si (Bräysy, 2005).

Para uma compreensão mais detalhada sobre o funcionamento do algoritmo, poderá consultar a secção de referências.

Análise e Especificação de Requisitos

Nesta seção será abordado a análise do projeto, nomeadamente a definição dos módulos a implementar e o respetivo levantamento e especificação dos requisitos funcionais.

7.7. Módulos

O desenvolvimento deste projeto foi dividido em 2 grandes grupos, *frontend* e *backend*. Cada grupo é composto por diferentes módulos que estão listados de seguida.

Módulos implementados no *Frontend*

Tabela 3 - Módulos referentes ao Frontend

	PRIORIDADE	DESCRIÇÃO
ARMAZÉNS	BAIXA	Módulo responsável pela gestão de Armazéns, nomeadamente todas as ações CRUD.
VIATURAS	ALTA	Módulo responsável pela gestão de Viaturas, nomeadamente todas as ações CRUD.
SERVIÇOS	MÉDIA	Módulo responsável pela gestão de Serviços, nomeadamente todas as ações CRUD.
DASHBOARD	ALTA	Módulo responsável por aglomerar toda a informação necessária para alimentar o otimizador, bem como fornecer todos os dados necessários para o cliente ser capaz de tomar uma decisão acertada baseada na informação devolvida.

Módulos implementados no *Backend*

Tabela 4 - Módulos referentes ao Backend

	PRIORIDADE	DESCRIÇÃO
REST	ALTA	Módulo responsável por fazer a ligação ao frontend, através da implementação do padrão REST (POST, GET, DELETE e PUT).
OTIMIZADOR	ALTA	Módulo desenvolvido em Python onde consta o algoritmo de otimização de rotas.

De notar que a prioridade aqui definida tem três níveis, BAIXA, MÉDIA e ALTA, sendo a ALTA o módulo que será mais crítico na implementação e desenvolvimento do projeto.

7.8. Requisitos Funcionais

Os requisitos funcionais propostos têm em vista o desenvolvimento do protótipo inicial e não uma solução final que poderá ser desenvolvida ou continuada após esta primeira fase.

Tabela 5 - Listagem de requisitos funcionais

MÓD.	ID	P.	DESCRIÇÃO
FRONTEND	ARMAZÉNS	RF01	M O sistema deverá permitir efetuar todas as ações CRUD sobre um armazém.
		RF02	M O sistema deverá permitir definir o número de cais disponíveis num determinado armazém.
		RF03	S O sistema deverá permitir visualizar no mapa a localização de um armazém.
		RF04	C O sistema deverá permitir editar a localização de um armazém, arrastando-o no mapa.
	VIATURAS	RF06	M O sistema deverá permitir efetuar todas as ações CRUD sobre uma viatura.
		RF07	M O sistema deverá permitir definir configurações genéricas de uma viatura, como tipologia, matrícula, entre outros.
		RF08	M O sistema deverá permitir definir os tempos de trabalho de uma viatura.
		RF09	M O sistema deverá permitir definir os limites de tempo de condução e da distância máxima a percorrer de uma viatura.
		RF10	M O sistema deverá permitir definir as dimensões de uma viatura, como altura, peso, comprimento, entre outras.
		RF11	S O sistema deverá permitir definir ainda outro tipo de restrições sobre uma viatura.
		RF12	M O sistema deverá permitir alterar o estado de uma viatura, entre disponível e indisponível.
	SERVIÇOS	RF13	M O sistema deverá permitir efetuar todas as ações CRUD sobre um serviço.
		RF14	M O sistema deverá permitir definir configurações genéricas de um serviço, como o nome do cliente.
		RF15	M O sistema deverá permitir definir a carga do serviço a entregar em quilogramas e em volume.
		RF16	M O sistema deverá permitir definir o local de recolha e de entrega da carga do serviço.
		RF17	M O sistema deverá permitir definir a janela horária da recolha e da entrega do serviço.
		RF18	M O sistema deverá permitir definir a duração esperada para recolha e entrega do serviço.
		RF19	S O sistema deverá permitir visualizar no mapa a localização do serviço.
		RF20	C O sistema deverá permitir editar a localização do local de descarga, arrastando-o no mapa.

BACKEND		RF21	M	O sistema deverá permitir carregar os serviços para um dia específico de atividade.
		RF22	S	O sistema deverá permitir transformar uma morada em coordenadas (geocode) ao adicionar um serviço.
	DASHBOARD	RF23	M	O sistema deverá permitir consultar toda a informação relevante para o otimizador.
		RF24	M	O sistema deverá permitir otimizar um conjunto de serviços para um determinado dia.
		RF25	C	O sistema deverá permitir alterar o estado da disponibilidade de uma viatura para facilitar a interação com o otimizador.
		RF26	C	O sistema deverá permitir alterar o estado de um serviço para facilitar a interação com o otimizador.
		RF27	M	O sistema deverá permitir consultar a distribuição dos serviços pelas viaturas.
		RF28	M	O sistema deverá permitir visualizar os somatórios das diferentes métricas, como duração da viagem, distância percorrida ou carga transportada.
		RF29	S	O sistema deverá permitir visualizar no mapa os serviços de cada rota.
		RF30	S	O sistema deverá permitir visualizar no mapa o trajeto definido de cada rota.
		RF31	C	O sistema deverá permitir consultar um conjunto de métricas gerais sobre o resultado obtido pelo otimizador.
	REST	RF32	M	O sistema deverá disponibilizar um conjunto de interfaces que permitam gerir um armazém.
		RF33	M	O sistema deverá disponibilizar um conjunto de interfaces que permitam gerir uma viatura.
		RF34	M	O sistema deverá disponibilizar um conjunto de interfaces que permitam gerir um serviço.
		RF35	M	O sistema deverá disponibilizar uma interface que permita otimizar um conjunto de serviços.
		RF36	M	O sistema deverá ainda disponibilizar interfaces secundárias que sejam necessárias para auxiliar o sistema.
	OTIMIZADOR	RF37	M	O sistema deverá calcular uma rota de um ponto A ao ponto B, com diversos pontos intermédios, tendo em conta as características e restrições da viatura.
		RF38	M	O sistema deverá encontrar o caminho mais curto entre os diferentes locais a visitar tendo em conta o conjunto de viaturas disponíveis.
		RF39	M	O sistema deverá ter em conta as restrições das janelas horárias das recolhas e das entregas aquando da otimização.
		RF40	M	O sistema deverá ter em conta a duração do serviço das recolhas e das entregas aquando da otimização.
		RF41	M	O sistema deverá ter em conta as restrições dos pesos e volumes das recolhas e das entregas aquando da otimização.
		RF42	W	O sistema deverá ter em conta o número de cais disponíveis em cada armazém para ter em conta as cargas em simultâneo aquando da otimização.

		RF43	C	O sistema deverá ter em conta a duração de uma atividade, que poderá ser diária ou não.
--	--	------	----------	---

No caso dos requisitos, o sistema de priorização adotado foi o **MoSCoW**, **M** para **MUST HAVE**, **S** para **SHOULD HAVE**, **C** para **COULD HAVE** e **W** para **WILL NOT HAVE**. O Requisitos do tipo M são não-negociáveis, ou seja, são completamente necessários para ter um produto final e funcional. Os do tipo S são também importantes para o projeto, mas não vitais. Já os C são iniciativas se for possível implementar, mas caso não o sejam, o impacto será bem reduzido. Por fim, os do tipo W são funcionalidades que, à partida, não farão parte desta implementação inicial.

7.7. Requisitos Não Funcionais

Uma vez que se trata de um desenvolvido de um protótipo, os requisitos não funcionais, tanto de Ambiente como de Qualidade, não foram especificados nesta fase. Será importante, numa segunda fase de desenvolvimento e implementação final da solução, proceder ao levantamento exaustivo destes requisitos, com especial foco nos requisitos de qualidade, de forma a garantir a qualidade esperada pelo utilizador nas suas diferentes características.

Desenho

Após uma primeira análise, e a especificação de requisitos, iniciou-se a fase de desenho ainda de alto nível, de forma a especificar o que será implementado.

8.1. Arquitetura Geral

A arquitetura do sistema demonstrado na Figura 9 está dividida em 2 grandes áreas, *frontend* e *backend*. A aplicação *frontend*, totalmente desenvolvida em JavaScript e com apoio de algumas *frameworks*, irá ser executada num browser no dispositivo do utilizador. Essa aplicação web irá comunicar através de pedidos AJAX com a API REST que está a ser executada no lado do servidor, num ambiente nodejs. Nesse mesmo servidor será também executado o otimizador desenvolvido em Python.

Todo esse sistema está suportado por uma base de dados não relacional, orientada a documentos.

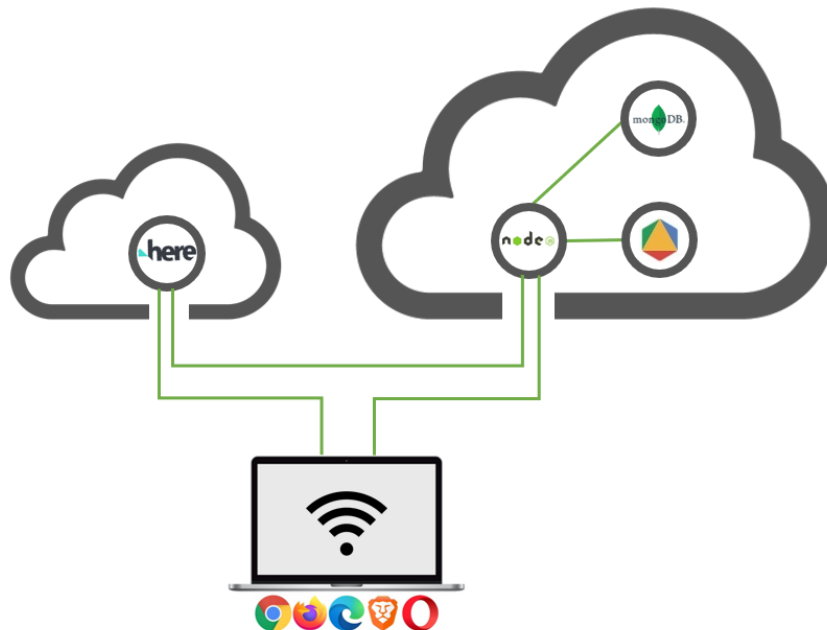


Figura 9 - Diagrama da arquitetura geral do sistema

8.2. Arquitetura Física

A arquitetura do sistema foi pensada e implementada em componentes fisicamente isolados e independentes para que a sua evolução e manutenção seja fácil. A Figura 10 demonstra como esses componentes estão organizados e que interfaces estão disponíveis para que estes comuniquem entre si.

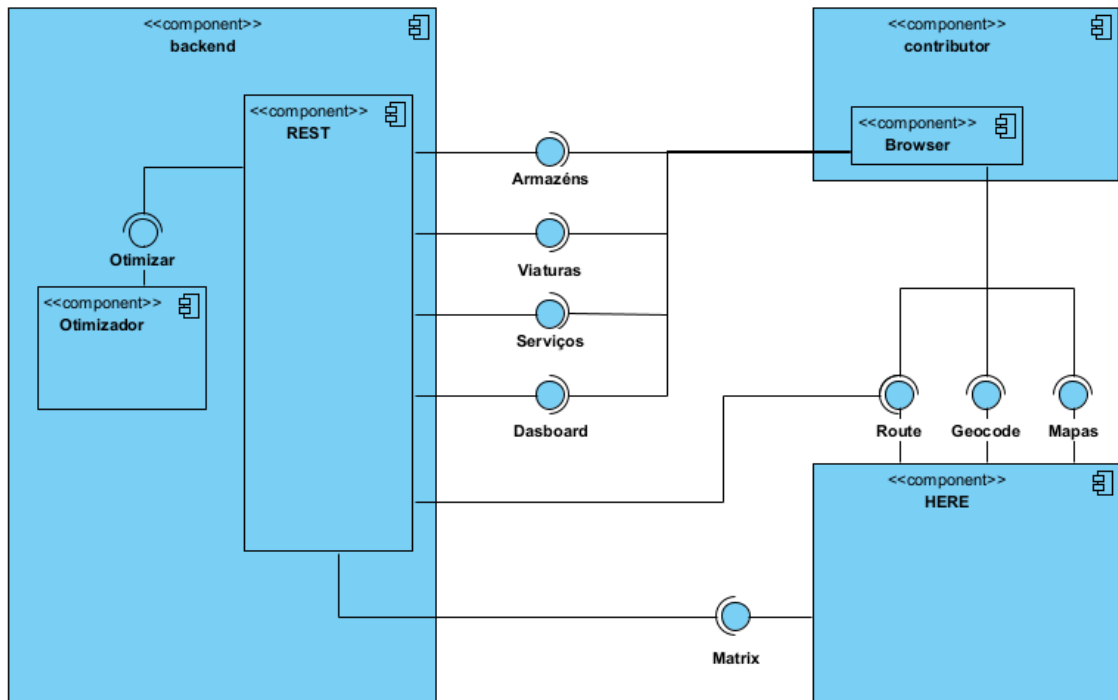


Figura 10 - Diagrama de componentes

8.3. Processos de negócio

O diagrama representado na Figura 11 tenta demonstrar como é que o processo de otimização de rotas funciona, quais os intervenientes e os locais onde decorrem os mesmos.

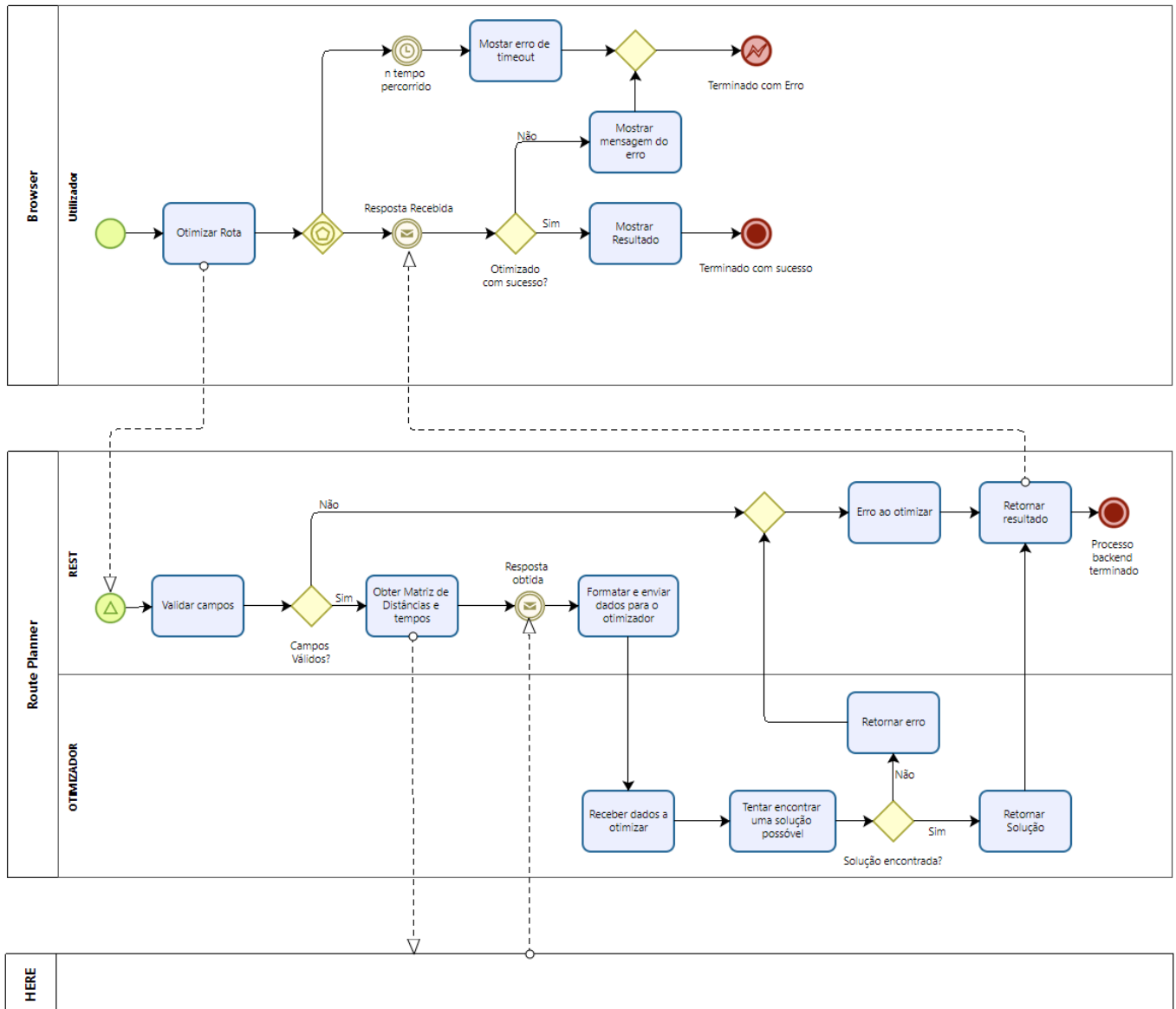


Figura 11 - Processo de negócio da otimização de rotas

8.4. Interface com o Utilizador

A interface foi planeada antes da sua implementação através de um *mockup* de baixa fidelidade presente na Figura 12 de forma a perceber quais os elementos necessários para compor o painel principal denominado de *dashboard*.

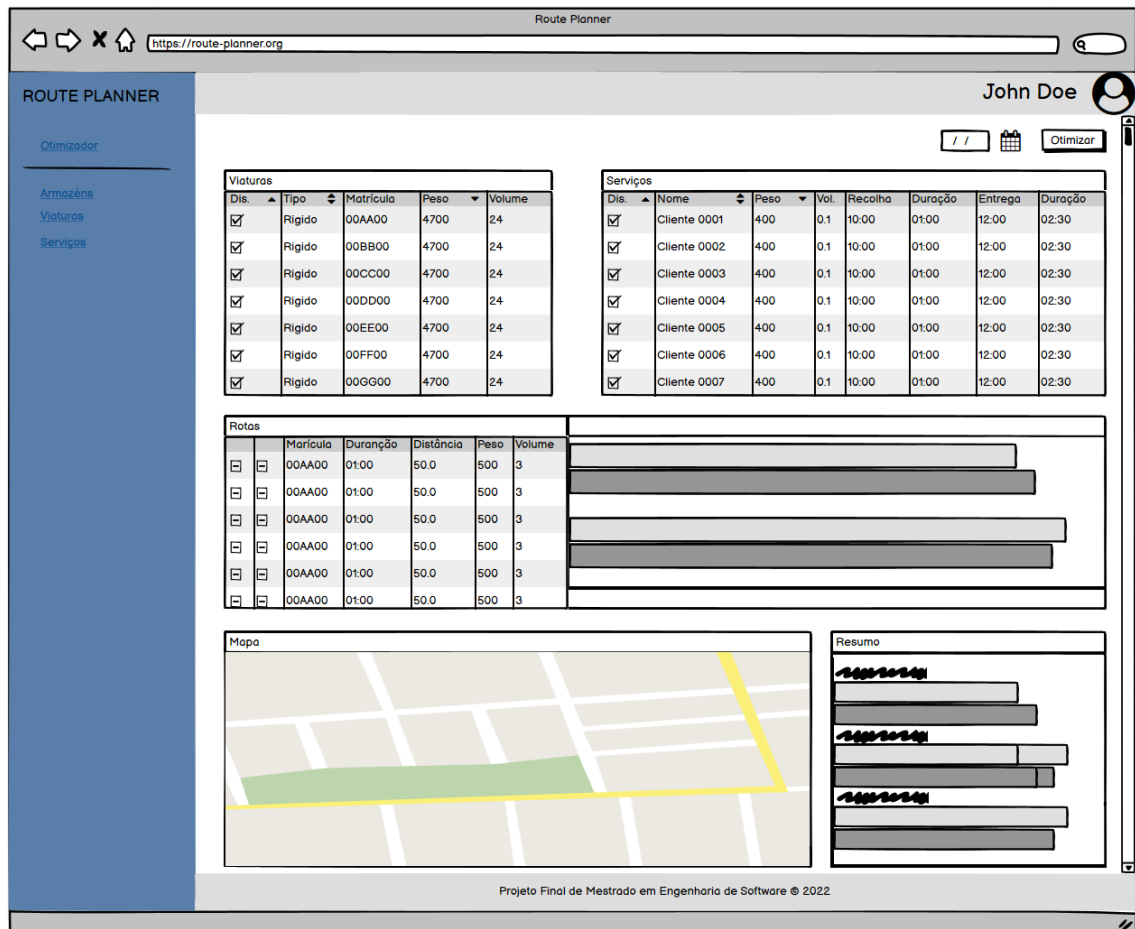


Figura 12 - Mockup de baixa definição do dashboard

Relatório Final

Otimização de rotas em viaturas pesadas

8.5. Planeamento

Na fase inicial foi definido o planeamento para todo o projeto, sendo que o mesmo teve data de previsão de conclusão e entrega no dia 23 de outubro de 2022. O planeamento foi dividido em 3 fases distintas, a fase de proposta que não está presente no diagrama da Figura 13, a fase de projeto que decorre ao longo de todo o período onde consta a elaboração dos relatórios, apresentação, definição de objetivos bem como decisões tecnológicas e, por fim, a fase de elaboração do protótipo com implementação em 4 grupos distintos, mas complementares e evolutivos da solução.

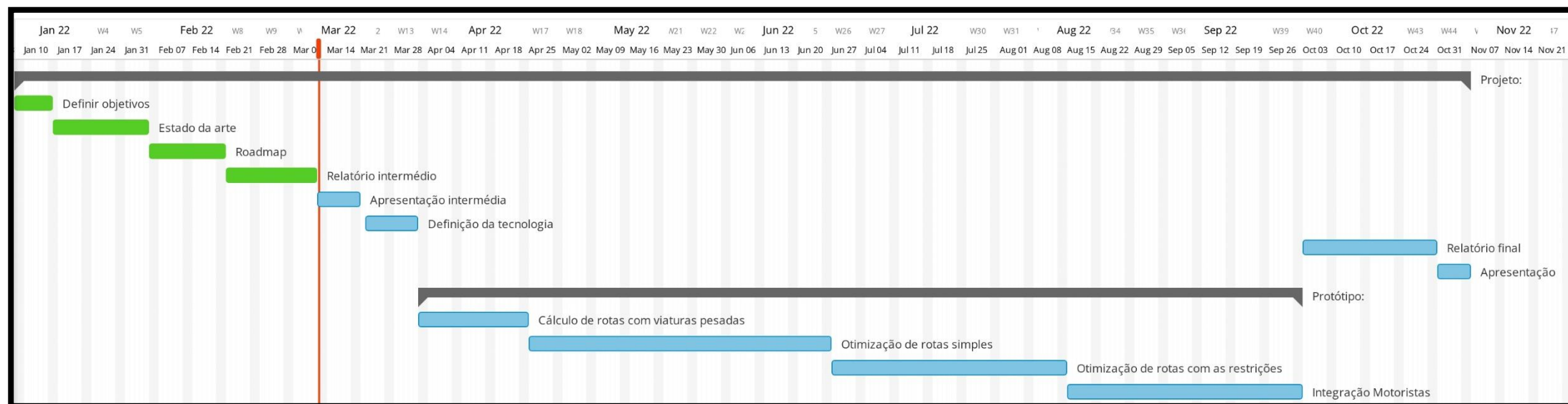


Figura 13 - Gráfico gantt com o planeamento do projeto

Implementação

Neste seção será abordada a fase de implementação, passando pelas tecnologias selecionadas após um levantamento inicial do estado da arte, perceber qual a metodologia de desenvolvimento utilizada nesta fase e olhar para o resultado final do protótipo, perceber as suas funcionalidades e resultados obtidos.

7.1. Metodologia e Ferramentas

Devido à natureza imprevisível deste projeto, nomeadamente no estudo e implementação de soluções que requerem um elevado nível de estudo e com uma curva de aprendizagem longa e desconhecida, foi adotado o Kanban como metodologia ágil. Esta metodologia é flexível quanto à definição das tarefas, principalmente na definição de datas de entrega e tempos de execução. Ainda como suporte ao desenvolvimento e implementação do projeto, foram utilizadas as ferramentas presentes na Tabela 6.

Tabela 6 - Ferramentas de suporte ao projeto

 Jira	Jira	<i>Gestor de projetos (KANBAN)</i>
 Confluence	Confluence	<i>Suporte documental</i>
 Gitlab	Gitlab	<i>Versionamento de código</i>
 Visual Paradigm	Visual Paradigm	<i>Ferramenta para criação de esquemas e diagramas</i>
 Basamiq	Basamiq	<i>Desenvolvimento de mockups</i>
 IntelliJ	IntelliJ	<i>IDE para desenvolvimento do código</i>

Num desenvolvimento de melhoria contínua, o quadro Kanban representado na Figura 14 é uma ajuda fundamental para ajudar uma equipa de desenvolvimento a perceber e gerir toda a quantidade de trabalho bem como a determinar o ponto de situação a qualquer instante, podendo ajustar a sua estratégia (What is a kanban board, 2022).

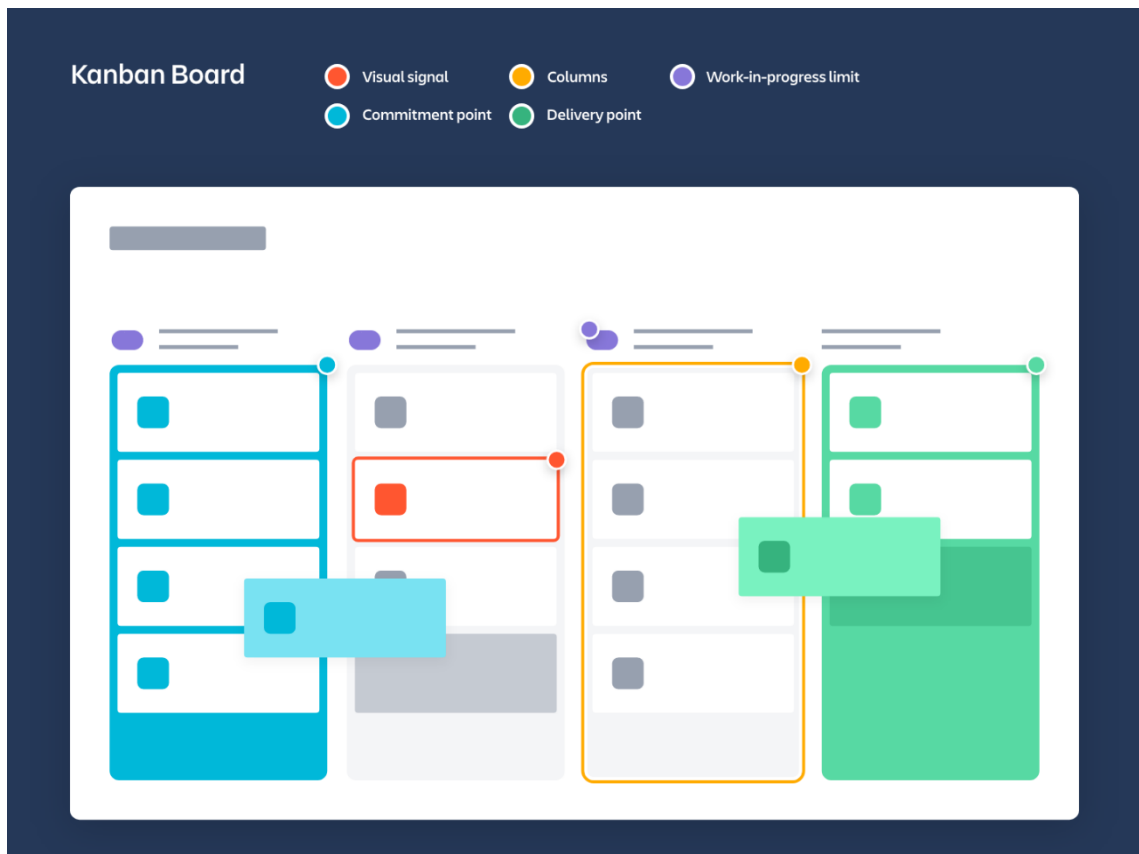


Figura 14 – Exemplo de um quadro Kanban

7.2. Tecnologias

Durante o planejamento e elaboração do projeto foram utilizadas diferentes ferramentas, ferramentas essas que foram escolhidas mediante uma análise do estado da arte bem como alguns testes locais ou públicos tanto ao nível de performance, preço ou funcionalidade.

7.2.1. Tecnologias no Frontend

O *frontend*, sendo um protótipo, foi pensado para auxiliar o utilizador na sua gestão diária de serviços, possibilitando ao mesmo configurar os diferentes elementos, como armazéns, viaturas e serviços, mas com especial foco na preparação da otimização e análise do otimizador, percebendo graficamente os resultados obtidos. Nesse sentido foram estudadas diferentes tecnologias, sendo que a Tabela 7 lista as tecnologias escolhidas.

Tabela 7 - Tecnologias escolhidas para o Frontend

	Bootstrap	<i>Framework para gestão DOM e Gráfica</i>
	jQuery	<i>Framework para manipulação DOM</i>
	D3js	<i>Ferramenta para construção de gráficos em JS</i>
	JavaScript	<i>Linguagem de programação</i>
	CSS	<i>Definição de Estilos no documento web</i>
	HTML	<i>Construção DOM</i>

Tratando-se de uma ferramenta web, é fundamente a escolha de uma *framework* para a gestão da complexidade que existe aquando da utilização de diferentes browsers, e por esse motivo a escolha recaiu sobre o *Bootstrap* e *jQuery*. A par destes, foram analisados o *React* e o *SenchaJS* mas, apesar destes sempre bem mais completos que os escolhidos, são também muito mais pesados e complexos pelo que, para o projeto aqui definido, não seriam necessários todas as funcionalidades fornecidas por estes.

Já a biblioteca D3js foi utilizada para a elaboração de alguns gráficos. Esta escolha caiu no facto de ser necessário elaborar alguns gráficos que não são facilmente encontrados noutra ferramentas, como o ChartJS. A D3js disponibiliza um conjunto de bibliotecas e métodos que nos permite construir com gráfico de acordo com as nossas necessidades. A Figura 15 demonstra um exemplo da sua aplicação no projeto, na demonstração das rotas otimizadas.

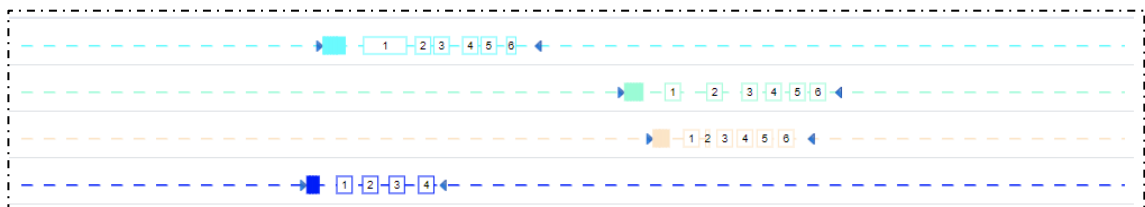


Figura 15 - Exemplo da aplicação da D3js

7.2.2. Tecnologias no Backend

O *backend* foi definido a pensar em 3 grandes responsabilidades: disponibilizar a página web; responder e fornecer os dados necessários para a sua visualização; bem como alojar o algoritmo e código do otimizador, de uma forma eficiente e eficaz. Nesse sentido foram estudadas

diferentes alternativas, sendo que a Tabela 8 lista as tecnologias selecionadas.

Tabela 8 - Tecnologias escolhidas para o Backend

	nodejs	<i>Ambiente de servidor</i>
	JavaScript	<i>Linguagem de programação</i>
	Python	<i>Linguagem de programação</i>

O nodejs é um ambiente de servidor baseado no google V8 que permite, entre outros, executar código JavaScript, fornecer páginas web, alojar serviços REST bem como executar scripts externos escritos em linguagens diferentes, como Python ou C++. Uma vez que a escolha da linguagem no *frontend* ser JavaScript bem como a base de dados ser MongoDB, em que guarda os registos nessa mesma estrutura (JSON), foi também escolhido o nodejs de forma inteligente para efetuar esta ligação de uma forma mais transparente, sem grande necessidade no tratamento dos dados e registos. Por fim, este ambiente de servidor dispõe de um conjunto de bibliotecas que permite uma fácil integração com diferentes ferramentas externas utilizadas neste projeto.

Já o otimizador foi desenvolvido em Python, utilizando a biblioteca OR-Tools da Google, sendo que o motivo e análise da sua escolha pode ser lido mais à frente na seção “Otimização”.

7.2.3. Base de Dados

Mesmo o resultado final sendo um protótipo, é necessário efetuar a persistência de alguns dados para alimentar o otimizador, facilitando assim o utilizador na sua atividade diária. Era pretendido uma base de dados que fosse fácil de alimentar e manter, e que permitisse crescer e acompanhar facilmente este projeto nas suas necessidades futuras, ficando as bases de dados tipicamente relacionais excluídas à partida desta seleção.

Tabela 9 - Bases de Dados escolhidas

	MongoDB	<i>Base de dados orientada a documentos</i>
---	---------	---

A escolha final caiu sobre o MongoDB pois é uma base de dados orientada a documentos, guardados e geridos no formato JSON (na verdade BSON, uma vez que estão em formato

binário). Além disso é uma base de dados fácil de manter e de escalar, acompanhado um possível crescimento do projeto. Esta escolha final foi efetuada após a comparação entre o MongoDB e Cassandra, da Apache, sendo dois sistemas bem maduros e com provadas dadas, a escolha recaiu para o MongoDB devido a alguns estudos de benchmarking que davam alguma vantagem sobre esta.

7.2.4. Integração

Sendo uma aplicação com o propósito final de otimizar rotas, é fundamental a visualização gráfica do resultado do otimizador no respetivo mapa. Além disso foram estudadas diferentes bibliotecas com algoritmos preparados para otimizar rotas, sendo que a listagem seguinte demonstra as tecnologias eleitas para essas integrações.

Tabela 10 - Tecnologias integradas no projeto

	HERE Technologies	Motor de rotas e mapas
	OR-Tools	Biblioteca de otimizador de rotas

Começando pela HERE Technologies, que tem um conjunto de ferramentas especializadas em mapas, localização e rotas, ela foi integrada e utilizada em diferentes plenitudes, desde a renderização de mapas para a visualização gráfica das ruas e cidades, a marcação nesses mapas das localizações de algumas entidades, como serviços, locais de recolha ou entregas, bem como o desenho gráfico das rotas calculadas para cada viatura.

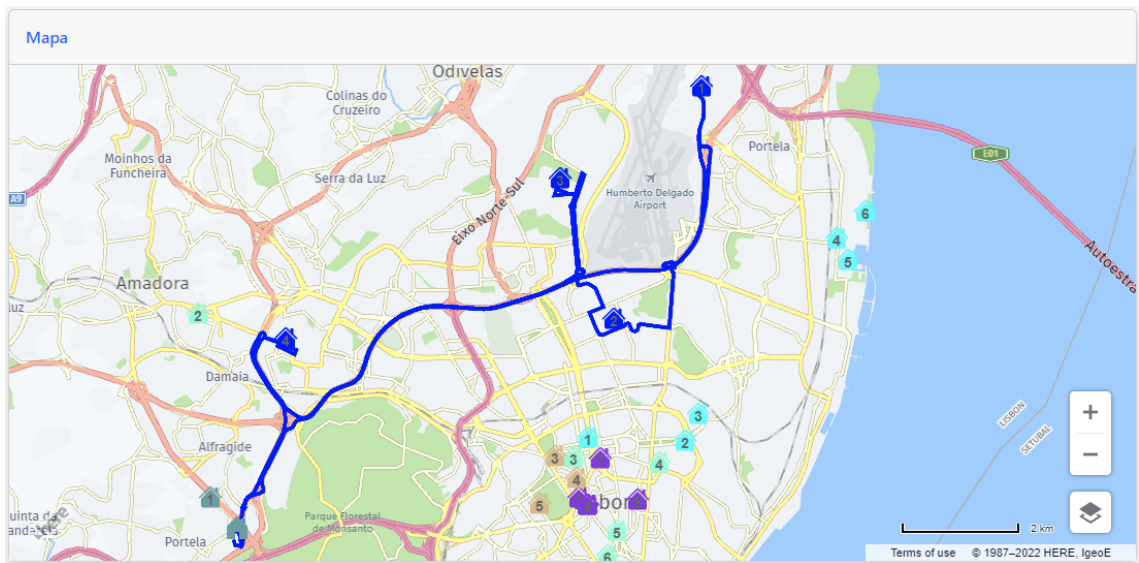


Figura 16 - Exemplo da utilização da HERE Technologies

Para além das ferramentas disponíveis na componente gráfica, foram também integrados alguns serviços no *backend*, nomeadamente:

- **geocode**: para ajudar a transformar uma morada numa posição geográfica, algo fundamental aquando da criação dos serviços pois, muitas vezes, é necessário proceder a essa transformação na introdução dos dados por parte do cliente. A Figura 17 mostra um exemplo simplificado de um resultado para a pesquisa da morada “Av. Bento Gonçalves 22B, 2910-431 Setúbal”. Para mais informações sobre este serviço consulte a documentação oficial (HERE Geocoding & Search API 7, 2022).

```
1 {
2   "items": [
3     {
4       "title": "Avenida Bento Gonçalves 22b, 2910-431 Setúbal, Portugal",
5       "address": {
6         "label": "Avenida Bento Gonçalves 22b, 2910-431 Setúbal, Portugal",
7         "countryCode": "PRT",
8         "countryName": "Portugal",
9         "county": "Setúbal",
10        "city": "Setúbal",
11        "district": "Setúbal",
12        "street": "Avenida Bento Gonçalves",
13        "postalCode": "2910-431",
14        "houseNumber": "22b"
15      },
16      "position": {
17        "lat": 38.52751,
18        "lng": -8.8831
19      }
20    }
21  ]
22 }
```

Figura 17 – Exemplo simplificado da integração "geocode"

- **calculateroute**: para obter a rota entre dois ou mais pontos, sendo estes pontos os serviços resultantes do otimizador. Dando as restrições da viatura fornecidos ao otimizador, como, entre outros, a tipologia, peso e altura, este serviço retorna a rota mais viável para visitar os locais pela ordem definida. O exemplo seguinte mostra o resultado simplificado para uma rota definida por 10 pontos, sendo que o campo “sections” contém uma lista de cada trajeto entre esses pontos. Neste caso, para simplificar a visualização apenas é apresentado uma secção. Para mais informações sobre este serviço pode consultar a documentação oficial (HERE Routing API 7, 2022).

```

1 {
2   "routes": [
3     {
4       "id": "7f681a3b-5e75-4d57-9c98-e8af8d037500",
5       "sections": [
6         {
7           "id": "d964dclb-d582-4bcf-a4d8-e57da857b1d6",
8           "type": "vehicle",
9           "departure": {
10            "time": "2022-10-05T10:50:54+01:00",
11            "place": {
12              "type": "place",
13              "location": {
14                "lat": 38.72226,
15                "lng": -9.2145
16              },
17              "originalLocation": {
18                "lat": 38.7223871,
19                "lng": -9.2139406
20              },
21              "waypoint": 3
22            }
23          },
24          "arrival": {
25            "time": "2022-10-05T11:06:50+01:00",
26            "place": {
27              "type": "place",
28              "location": {
29                "lat": 38.7889551,
30                "lng": -9.1242271
31              },
32              "originalLocation": {
33                "lat": 38.7889502,
34                "lng": -9.1242342
35              },
36              "waypoint": 4
37            }
38          },
39          "summary": {
40            "duration": 956,
41            "length": 13233,
42            "baseDuration": 919
43          },
44          "polyline": "BGott7pCnityRjSwwCzK4DzFkDrE8GTsE8BoQA8GnBw.....",
45          "transport": {
46            "mode": "truck"
47          }
48        }
49      ]
50    }
51  ]
52 }

```

Figura 18 - Exemplo simplificado da integração "calculateroute"

- **calculatematrix**: este serviço disponibilizado pela API foi fundamental no desenvolvimento deste projeto. Dado um conjunto de localizações, esta interface vai devolver uma matriz de distâncias e de tempos entre cada localização, ou seja, conseguimos perceber qual a distância e tempo necessário para percorrer de um local para qualquer outro destino. Esta matriz foi depois utilizada para alimentar o otimizador nas suas diferentes variantes. A Figura 19 tem um exemplo de uma matriz de distância entre 10 localizações distintas em que podemos ver, como exemplo, que a distância entre o ponto 2 e o ponto 4 é de 0.29, já a distância entre o ponto 4 e o ponto 7 é de 0.2146.

1	0	0.1129	0.2283	0.29	0.4016	0.5018	0.4774	0.09234	0.1143	0.1382
2	0.1129	0	0.1833	0.1932	0.3372	0.4053	0.376	0.149	0.05639	0.1916
3	0.2283	0.1833	0	0.1604	0.1764	0.3375	0.3541	0.2982	0.1737	0.3467
4	0.29	0.1932	0.1604	0	0.2093	0.223	0.2146	0.3294	0.1925	0.3707
5	0.4016	0.3372	0.1764	0.2093	0	0.2668	0.3272	0.4666	0.3325	0.5156
6	0.5018	0.4053	0.3375	0.223	0.2668	0	0.1022	0.5326	0.3994	0.5657
7	0.4774	0.376	0.3541	0.2146	0.3272	0.1022	0	0.4942	0.3724	0.5204
8	0.09234	0.149	0.2982	0.3294	0.4666	0.5326	0.4942	0	0.1516	0.07304
9	0.1143	0.05639	0.1737	0.1925	0.3325	0.3994	0.3724	0.1516	0	0.1981
10	0.1382	0.1916	0.3467	0.3707	0.5156	0.5657	0.5204	0.07304	0.1981	0
	1	2	3	4	5	6	7	8	9	10

Figura 19 - Exemplo de uma matriz de distância entre 10 localizações

De notar que o gráfico anterior é simétrico, ou seja, a distância entre dois pontos A e B é igual na sua forma inversa, B->A, mas na realidade o trajeto entre A->B e B->A pode ser diferente devido a restrições de passagem, como sentidos proibidos pelo que é necessário calcular estes valores para cada sentido. A Figura 20 mostra o resultado simplificado obtido utilizando este serviço, em que o *startIndex* é o ponto de partida (linhas) e o *destinationIndex* é o ponto de destino (colunas). Para mais informações sobre este serviço pode consultar a documentação oficial (HERE Routing API 7, 2022).

```

1  [
2  {
3      "startIndex": 0,
4      "destinationIndex": 0,
5      "summary": {
6          "distance": 10496,
7          "travelTime": 1887,
8          "costFactor": 2213
9      }
10 },
11 {
12     "startIndex": 0,
13     "destinationIndex": 1,
14     "summary": {
15         "distance": 10496,
16         "travelTime": 1887,
17         "costFactor": 2213
18     }
19 },
20 {
21     "startIndex": 0,
22     "destinationIndex": 2,
23     "summary": {
24         "distance": 10496,
25         "travelTime": 1887,
26         "costFactor": 2213
27     }
28 },
29 {
30     "startIndex": 0,
31     "destinationIndex": 3,
32     "summary": {
33         "distance": 10496,
34         "travelTime": 1887,
35         "costFactor": 2213
36     }
37 },
38 {
39     "startIndex": 0,
40     "destinationIndex": 4,
41     "summary": {
42         "distance": 10496,
43         "travelTime": 1887,
44         "costFactor": 2213
45     }
46 }
47 ]

```

Figura 20 - Exemplo simplificado da integração "calculatematrix"

OTIMIZAÇÃO

Por fim, foi integrado a OR-Tools da google (Routing Options, 2022), nomeadamente o pacote desenvolvido e especializado em resolver questões relacionadas com a otimização de rotas, denominada por *Vehicle Routing*. A sua documentação oficial é bastante completa, mas também complexa, sendo *open source*, disponibiliza todos os algoritmos e documentação para suportar os mesmos, tornando complexa a sua implementação, mas por outro lado, é bastante flexível e adaptável a diferentes cenários de implementação (J.J. Schneider, 2006).

Toda a implementação foi desenvolvida em Python, estando este módulo integrado diretamente no projeto nodejs, no ambiente de servidor. Este algoritmo está integrado na plataforma e é executado após o utilizador definir todas as regras necessárias para executar o mesmo. Neste seção vamos abordar como o algoritmo foi implementado e quais as características ou restrições aceites na sua execução, deixando a análise do algoritmo em si para seção de [fundamentos teóricos](#), já abordado anteriormente.

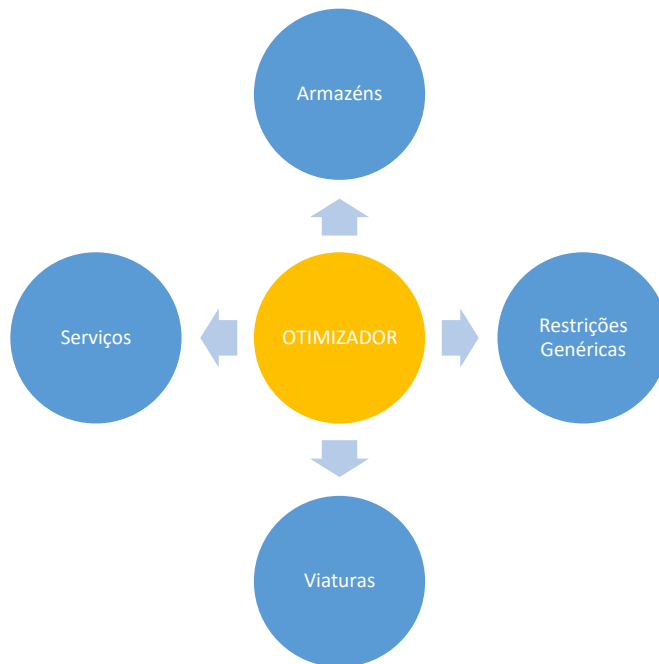


Figura 21 - Configurações do otimizador

O algoritmo foi desenvolvido a pensar nas restrições definidas nos objetivos da solução, cujo a Figura 21 tenta demonstrar, e que passamos a detalhar de seguida.

Armazéns

Os armazéns são locais de recolha ou de entrega de material, bem como o local de partida e de destino final de cada viatura. No caso de viaturas pesadas, é necessário ter em atenção o número total de cais de embarque disponíveis para carregar e/ou descarregar. Isso permite perceber quantas viaturas podem estar encostadas ao cais ao mesmo tempo, garantindo assim que os tempos de carga e descarga são respeitados.

A configuração do armazém é definida da seguinte forma:

```
{
  "_id" : "Identificação única do armazém",
  "alias" : "O nome do armazém",
  "docks" : "O número de docas disponíveis",
  "lat" : "A latitude em graus decimais",
  "lng" : "A longitude em graus decimais"
}
```

Viaturas

A viatura é o elemento crítico do processo de otimização, sendo também o mais complexo e o que pode variar mais entre cada elemento. Cada viatura tem um conjunto de características que pode variar entre cada uma, como a sua dimensão, capacidade máxima ou a capacidade de

transportar diferentes tipos de produtos, como líquidos, cimento, etc. Além disso uma viatura pesada obedece a um conjunto de regras diferentes quando comparada com uma ligeira. Por exemplo, o tempo de condução dos motoristas nas viaturas pesadas é regulado pelas autoridades nacionais e europeias, e as ligeiras não têm quaisquer restrições nesse sentido.

A configuração da viatura é definida da seguinte forma:

```
{
  "_id" : "Identificação única da viatura",
  "available" : "Estado da viatura (disponível/indisponível)",
  "license_place" : "A matrícula",
  "id_model" : "Identificação do modelo da viatura",
  "max_weight" : "Peso máximo transportado",
  "max_volume" : "Volume máximo transportado",
  "max_driving_time" : "Tempo máximo em condução",
  "max_working_time" : "Tempo máximo em trabalho",
  "max_distance" : "Distância máxima permitida",
  "speed_factor" : "Fator velocidade (normal ou rápido)",
  "vehicle" : {
    "shipped_hazardous_goods" : "Tipo de mercadoria perigosa transportada",
    "height" : "Altura da viatura",
    "width" : "Largura da viatura",
    "length" : "Comprimento da viatura",
    "tunnel_category" : "A categoria de túnel máximo permitido",
    "axle_count" : "Nº total de eixos",
    "type" : "Tipo de viatura (ligeira, camião, trator, ...)",
    "trailer_count" : "Nº total de atrelados"
  },
  "workday" : {
    "from" : "Hora mínima para iniciar o trabalho",
    "to" : "Hora máxima para iniciar o trabalho "
  }
}
```

Serviços

O grande objetivo desta solução é otimizar e concretizar o maior número de serviços possíveis, minimizando a distância e a duração da viagem, minimizando assim o custo do transporte e da atividade. Um serviço é algo que tem de ser recolhido num determinado local, como um armazém, ou entregue num destinatário, como um cliente final. Tudo isto num determinado dia de atividade.

A configuração de um serviço é definida da seguinte forma:

```
{
  "_id" : "Identificação única do serviço",
  "active" : "Estado do serviço (ativo/inativo)",
  "name" : "Nome do serviço ou cliente",
  "date" : "Data para realizar o serviço",
  "pickup" : {
    "lat" : "A latitude em graus decimais do local de recolha",
    "lng" : "A longitude em graus decimais do local de recolha ",
    "duration" : "A duração estimada da recolha",
    "time_window" : "Lista de janelas horárias para a recolha"
  },
  "delivery" : {
    "lat" : "A latitude em graus decimais do local de descarga",
    "lng" : "A longitude em graus decimais do local de descarga",
    "duration" : "A duração estimada da descarga",
    "time_window" : "Lista de janelas horárias para a entrega"
  },
  "load" : {
    "weight" : "Peso da carga",
    "volume" : "Voluma da carga"
  }
}
```

Restrições genéricas

Para além das restrições abordadas anteriormente, que estão diretamente ligadas a um determinado elemento, existem ainda outras restrições globais ou transversais:

- **Chegada a um armazém**

É possível obrigar que o tempo de trabalho inclua os tempos de deslocação desde o último serviço até ao armazém, onde finalizará o seu dia de atividade.

- **Duração máxima da atividade**

Uma das limitações mais relevantes no sistema é o tempo. O algoritmo irá tentar encaixar o maior número de serviços dentro de um período definido de atividade. Por exemplo, para atividade capilar normalmente são 24h.

Execução do algoritmo

Uma vez recolhidos todos os requisitos e condições necessárias para executar o módulo de otimização, este pode ser executado baseado em diferentes algoritmos, devolvendo assim diferentes possíveis soluções. Independentemente disso, o formato da resposta é sempre igual para que possa ser facilmente manipulado e integrado. Na Figura 22 é possível visualizar um exemplo simplificado de um resultado do otimizador e no [ANEXO I](#) é possível consultar o *output* obtido desse módulo. Neste exemplo é apenas apresentado uma rota, e foram apenas mantidos alguns passos para facilitar a visualização.

```

1  {
2      "solution_strategy": "PARALLEL_CHEAPEST_INSERTION",
3      "exec_time": 958,
4      "total_time": 103313,
5      "total_distance": 304421,
6      "total_weight": 8339548,
7      "total_volume": 25554,
8      "total_service_time": 59040,
9      "routes": [
10         {
11             "vehicle_id": "62ff57eb9b0d0f8ca62b4f24",
12             "service_time": 10620,
13             "time": 17119,
14             "distance": 43400,
15             "weight": 1326073,
16             "volume": 3003,
17             "steps": [
18                 {
19                     "index": 0,
20                     "from": 23616,
21                     "to": 23616,
22                     "distance": 0,
23                     "service_time": null,
24                     "weight": null,
25                     "volume": null,
26                     "type": "departure"
27                 },
28                 {
29                     "index": 1,
30                     "from": 23616,
31                     "to": 23736,
32                     "distance": 0,
33                     "service_time": 120,
34                     "weight": 769320,
35                     "volume": 1523,
36                     "type": "pickup",
37                     "service_id": "63089f2f12e45db6e16d0ccc"
38                 },
39                 {
40                     "index": 2,
41                     "from": 26794,
42                     "to": 30154,
43                     "distance": 9497,
44                     "service_time": 3360,
45                     "weight": 270576,
46                     "volume": 846,
47                     "type": "delivery",
48                     "service_id": "63089f2f12e45db6e16d0ccc"
49                 },
50                 {
51                     "index": 3,
52                     "from": 40735,
53                     "to": 40735,
54                     "distance": 43400,
55                     "service_time": null,
56                     "weight": null,
57                     "volume": null,
58                     "type": "arrival"
59                 }
60             ]
61         }
62     ]
63 }

```

Figura 22 - Exemplo simplificado do resultado do otimizador

Poderá consultar as diferenças entre os diversos algoritmos de estratégias de solução na seção de Fundamentos Teóricos.

Capítulo 3

Resultados

Neste seção iremos abordar o resultado final com o intuito de perceber o que foi implementado em cada um dos módulos propostos, passando em detalhe por cada um dos objetivos genéricos definidos anteriormente.

- **Cálculo da rota:** Um dos objetivos mais importantes e que foi implementado com sucesso e na sua plenitude, foi a capacidade de calcular uma rota de um ponto A ao ponto B, passando por outros pontos intermédios, para diferentes tipologias de viaturas, ligeiras ou pesadas, tendo em conta as suas características como o peso, altura, largura, número de eixos, entre muitas outras. Esta solução foi implementada recorrendo à integração com a *HERE Technologies*, a qual dispõe de uma API capaz de satisfazer esta necessidade, garantindo assim que a rota gerada apenas apresenta ruas ou vias permitidas para a viatura em questão.
- **Restrições de tempo:** Esta restrição foi analisada e implementada em diferentes dimensões. É possível definir uma restrição do tempo global da atividade, garantido que todo o trabalho é efetuado, por exemplo, durante um dia útil de 24 horas. Outra dimensão foi garantir que o tempo total de viagem ou de trabalho de uma viatura é cumprido, garantindo assim que possíveis restrições contratuais ou legais são garantidas durante a otimização. Podemos ainda definir a velocidade média de uma viatura, se é normal ou mais rápida, influenciando os tempos de viagem entre os diferentes pontos. E por fim, é possível garantir e definir as janelas horárias do serviço, tanto para a recolha como para a entrega, garantindo mais uma vez que a viatura chega ao local da recolha dentro do horário estipulado, e que entrega o produto de acordo com o horário contratualizado.
- **Restrições de capacidade:** A capacidade foi abordada e pensada para 2 dimensões, o peso e o volume, com o objetivo de garantir que uma viatura efetua apenas a quantidade de serviços de acordo com a sua capacidade máxima, respeitando o seu limite máximo de peso e volume. Para isso, é possível definir essas mesmas restrições na configuração da viatura e, no momento da criação do serviço, é necessário preencher o total de peso e volume a carregar.

- **Restrições na carga:** Uma restrição bastante comum nos armazéns é a sua capacidade de servir viaturas em simultâneo. Sendo que é difícil perceber ou ter acesso a este tipo de informação nos destinatários, ou seja, nos clientes finais, esta restrição foi apenas pensada para os armazéns que pertençam ao nosso cliente utilizador. Este pode definir a quantidade de cais disponíveis em cada armazém para limitar assim a quantidade de viaturas que podem estar a carregar ao mesmo tempo, garantindo assim que a rota e os seus respetivos tempos sejam criados corretamente.
- **Motorista:** Uma vez otimizado, é necessário perceber qual ou quais os motoristas necessários para efetuar este serviço. Para isso é calculado o tempo total de viagem e de trabalho de cada rota otimizada para depois permitir a pesquisa dos motoristas com essa disponibilidade. Esta seção está pensada para ser integrada posteriormente com um serviço disponibilizado pelo cliente de forma a ter acesso e em tempo real, aos tempos de condução e de trabalho legais de cada elemento.
- **Otimização de rotas:** Todos os objetivos definidos anteriormente são encaminhados para este, que vai englobar todas as restrições já abordadas e depois fornecer a melhor solução viável de acordo com a estratégia abordada. Este objetivo foi alcançado através da integração com diferentes tecnologias, como a *HERE Technologies* e a sua API que devolve a distância e tempos de condução em matriz entre todos os locais de carga e descarga, bem como a utilização de bibliotecas da *OR-Tools* que contêm diferentes algoritmos de otimização já pensados e desenvolvidos para este tipo de problemática.
- **Protótipo:** Por fim, mas não menos importante, está o desenvolvimento do protótipo da aplicação web para permitir ao utilizador interagir com o sistema, podendo gerir todos os inputs necessários, gerar uma otimização e consultar os resultados obtidos. O feedback obtido pelos utilizadores ou por quem acompanhou um pouco o projeto foi positivo, pelo que acreditamos estar no bom caminho para uma implementação definitiva desta solução.

Nos pontos seguintes iremos abordar, módulo a módulo, a implementação efetuada e perceber graficamente o que foi implementado e as suas principais funcionalidades.

10.1.Módulo Armazéns

Um armazém serve para definir locais de partida e chegada de viaturas, bem como permitir definir locais de carga das viaturas. O painel foi pensado para o utilizador conseguir gerir os seus armazéns por completo, permitindo criar, consultar, editar ou remover (CRUD) qualquer registo de uma forma fácil e intuitiva. Além disso, o utilizador pode consultar graficamente onde estão localizados os seus armazéns, bem como ajustar a sua posição no mapa, caso seja necessário.

Na Figura 23 conseguimos visualizar uma listagem de 8 armazéns diferentes, sendo que o mapa está centrado na zona metropolitana de Lisboa, onde estão visíveis 3 armazéns.

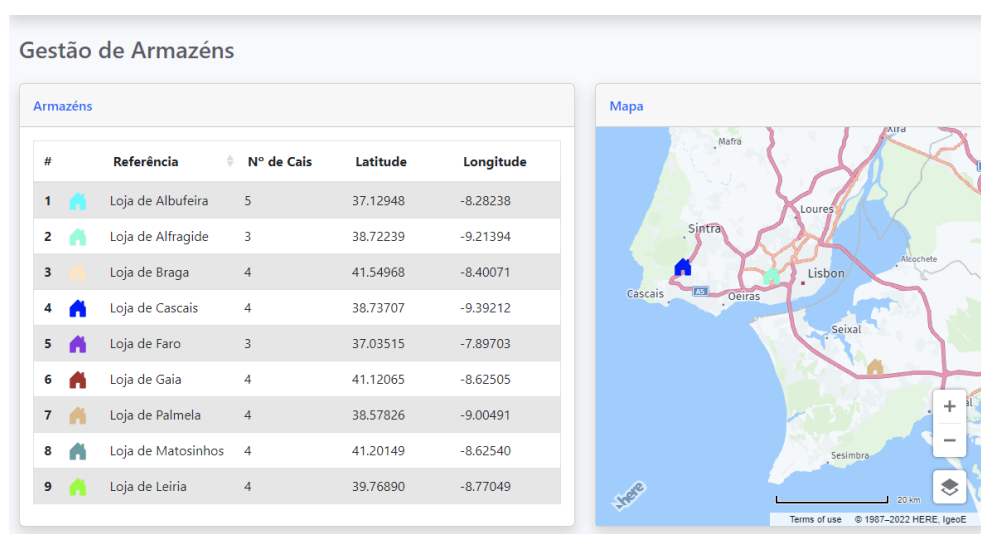


Figura 23 - Cópia de ecrã do painel de gestão de Armazéns

10.2.Módulo Viaturas

Uma viatura é definida por um conjunto largo de características cujo configuração pode influenciar o resultado obtido pelo otimizador. Por esse motivo foi pensado um painel onde seja possível visualizar todas essas características e os seus valores definidos e poder comparar entre as diferentes viaturas, tendo assim uma visão global da configuração do sistema. Aqui é possível realizar qualquer operação CRUD sobre uma viatura em particular, sendo que foram desenvolvidos mecanismos reutilizáveis de edição de cada tipo de dados, como estados, texto, números, listas, datas ou campos do tipo hora, que estão presentes em qualquer formulário.

Aqui o utilizador pode alterar o estado da viatura, tornando-a disponível ou indisponível para realizar trabalho, bem como configurar as suas características genéricas como a sua tipologia, matrícula, marca ou modelo. Pode também alterar a configuração dos tempos de trabalho e a sua duração máxima bem como os tempos máximos de condução ou distâncias máximas a percorrer. Pode ainda alterar as dimensões da viatura, como altura, largura, comprimento,

número de eixos, peso e volumes máximos, a sua categoria nos túneis bem como o tipo de materiais perigosos transportados (se for o caso).

Gestão de Viaturas

Search

Geral						Trabalho			Cond.	Dist.	Dimensões						Restrições			
#	Dis.	Tipo	Ref.	Marca	Modelo	Fact. Vel.	Início (h:m)	Final (h:m)	Máx. (h:m)	Máx. (h:m)	Máx. (km)	Alt. (m)	Lar. (m)	Comp. (m)	Eixos (nº)	Peso (Kg)	Vol. (m³)	Atr. (nº)	Cat. Túnel	Mat. Perigosos
1	☒	Rígido	00AA00	daif	LF 210	Normal	06:30	19:00	09:00	08:00	800.0	2.86	2.54	7.11	2	4700	24	0	-	-
2	☒	Rígido	00BB00	desot	LF 210	Normal	06:30	19:00	09:00	08:00	800.0	2.86	2.54	7.11	2	4700	24	0	B	Inflamável
3	☒	Rígido	00CC00	deutz fahr	LF 45.180	Normal	06:30	19:00	09:00	08:00	800.0	2.86	2.55	7.11	2	1500	40	0	-	-
4	☒	Rígido	00DD00	ducati	FUSO CANTER 9C18	Normal	06:30	19:00	09:00	08:00	800.0	2.15	1.88	5.05	2	3785	7	0	-	-
5	☒	Rígido	00EE00	fiat	LF 45.180	Normal	06:30	19:00	09:00	08:00	800.0	2.86	2.54	7.11	2	1500	40	0	C	Inflamável
6	☒	Rígido	00FF00	ford	FUSO CANTER 9C18	Normal	06:30	19:00	09:00	08:00	800.0	2.15	1.88	5.05	2	3785	7	0	-	-
7	☒	Rígido	00GG00	fuso	LF 45.180	Normal	06:30	19:00	09:00	08:00	800.0	2.86	2.54	7.11	2	1500	40	0	-	-
8	☒	Rígido	00HH00	honda	LF 45.180	Normal	06:30	19:00	09:00	08:00	800.0	2.86	2.54	7.11	2	1500	40	0	-	-

Figura 24 - Cópia de ecrã do painel de gestão de Viaturas

10.3.Módulo Serviços

Na sua base, um serviço é composto por uma data, uma carga, uma recolha e uma entrega, sendo que a recolha e entrega ocorrem em locais distintos. Este painel foi pensado para o utilizador conseguir gerir facilmente os serviços definidos para um determinado dia, e perceber a sua localização. Pode ainda alterar o estado do serviço, garantindo assim que o mesmo não é realizado ou tido em conta aquando da otimização deste dia de atividade.

O módulo é composto por dois painéis, sendo o primeiro uma tabela com todos os serviços criados para o dia selecionado, podendo realizar quaisquer ações CRUD sobre os mesmos, como definir o nome, as dimensões da carga (peso e volume), as janelas horárias para a carga e descarga bem como a duração estimada que deverão levar para carregar a viatura e descarregar no destinatário. O segundo painel é um mapa onde estão representados todos os serviços com os seus locais de recolha e entrega, podendo assim ter uma visão da distribuição do trabalho bem como poder ajustar manualmente os respetivos locais.

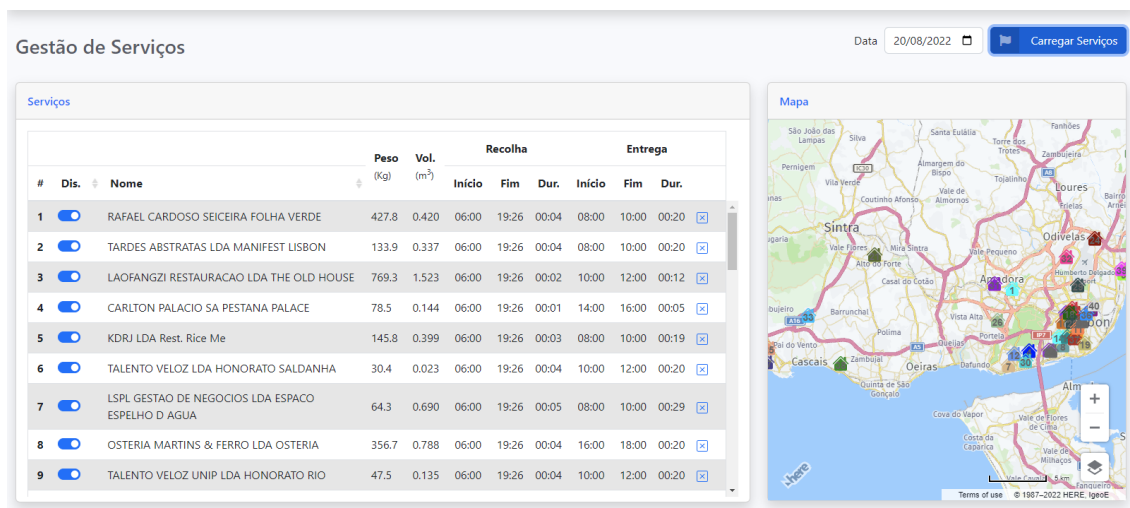


Figura 25 - Cópia de ecrã do painel de gestão de Serviços

10.4. Módulo Otimizador

A componente gráfica do otimizador foi pensada como um *dashboard*, ou seja, um painel resumo de todos elementos que fazem parte do input do algoritmo, bem como analisar e compreender o resultado obtido por esse motor. Com este único painel, é possível consultar as principais configurações das viaturas e dos serviços que foram carregados para esse dia de serviço em particular. E após executar o cálculo das rotas, ou seja, otimizar, é possível perceber a distribuição dos serviços pelas viaturas, tendo assim a visão da atividade que terá de ser realizada.

Decompondo o painel nas suas seções, e começando pela primeira parte, reparamos na Figura 27 que no topo estão duas listagens, a das viaturas que compõem o sistema bem como a listagem dos serviços carregados para o dia em questão. Na listagem de viaturas é possível consultar algumas características mais relevantes, bem como alterar o seu estado de disponibilidade, sendo que apenas viaturas disponíveis serão tidas em conta no algoritmo. O mesmo acontece na listagem de serviços, onde constam as características mais relevantes e é também possível alterar o estado do serviço para ser removido deste dia de atividade.

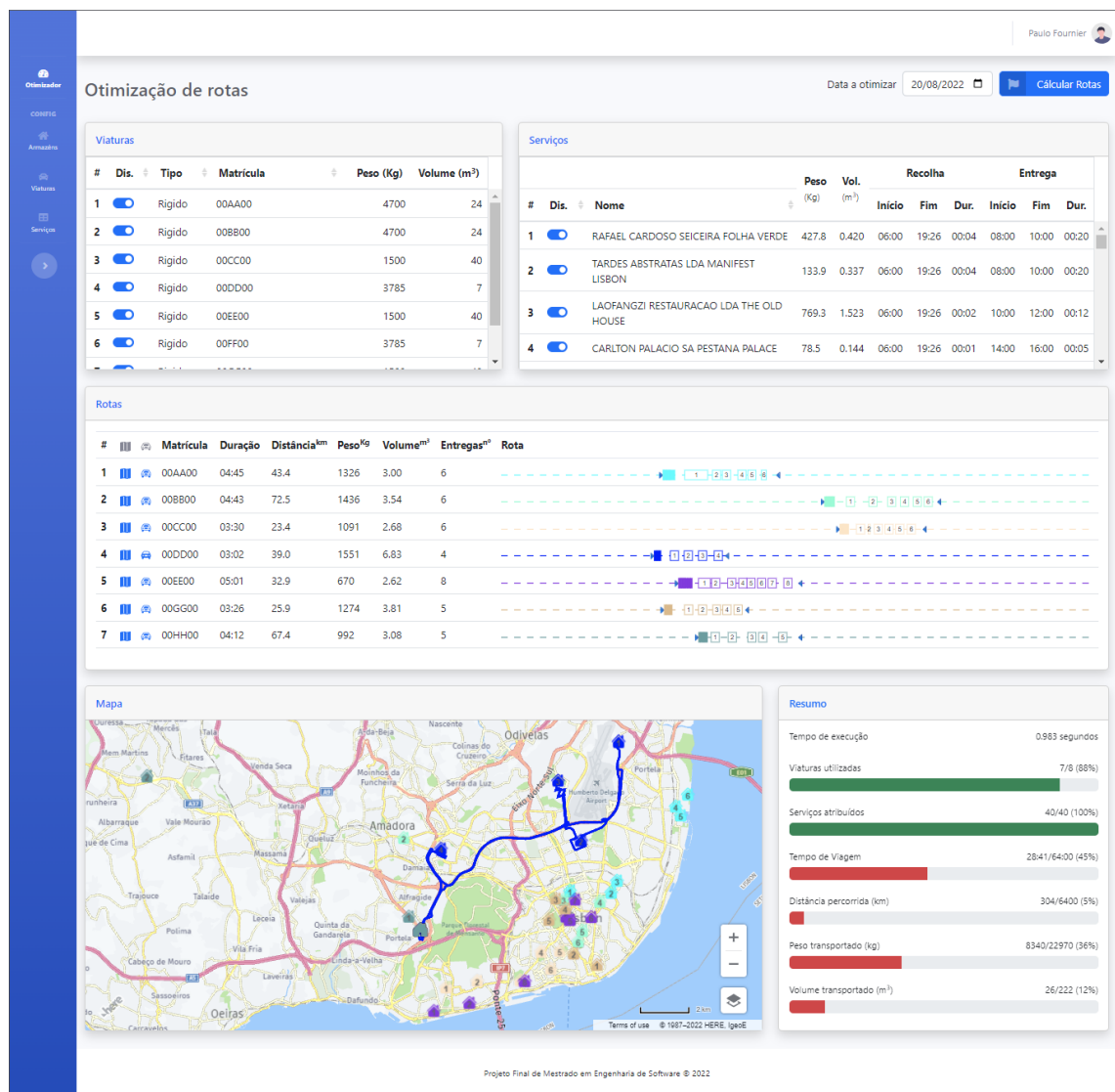


Figura 26 - Cópia de ecrã da aplicação, com foco no Otimizador

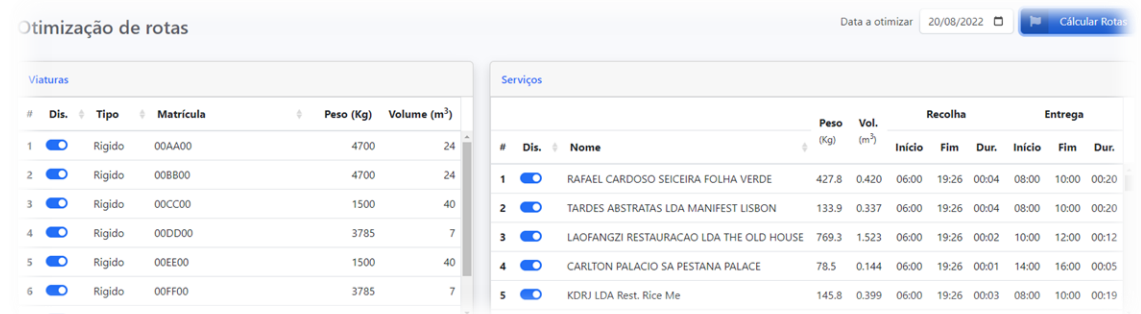


Figura 27 - Painéis com listagem de viaturas e serviços

Após executar o otimizador, carregado no botão “Calcular Rotas”, os painéis “Rotas”, “Mapa” e “Resumo” são carregados com o resultado obtido pelo otimizador. Começando pelo primeiro, o painel “Rotas”, este irá conter todas as rotas otimizadas para cada viatura, mas apenas aquelas que tiverem serviços atribuídos. Neste exemplo temos 8 viaturas disponíveis, mas foram

necessárias apenas 7 para realizar todos os serviços.

Cada linha nesta tabela, representa o resultado obtido para uma viatura. Por exemplo, na primeira linha podemos confirmar que a viatura 00AA00 terá uma atividade de 4:45 horas, e irá percorrer uma distância de 43.400km, carregando 1.326Kg com 3 metros cúbicos em volume num total de 6 serviços (cada um composto por uma recolha e uma entrega).

De seguida temos um gráfico por viatura, sendo que cada uma tem uma cor diferente para ser mais fácil a sua diferenciação, tanto aqui como no mapa. Esse gráfico demonstra a distribuição dos serviços ao longo do dia de atividade, onde temos as setas azuis a demarcar o início e final da atividade e na sua área intermédia estão definidas as recolhas e entregas dos serviços. As recolhas são opacas e estão normalmente no início da atividade, mas podem aparecer no meio no caso de a viatura ter de voltar ao armazém para recolher mais produtos, e as entregas estão numeradas de forma sequencial. Como podemos perceber na Figura 28, ao colocar o cursor do rato por cima de uma destas caixas de recolha ou entrega, temos informação detalhada do serviço a realizar, bem como os tempos e distância do serviço anterior e seguinte. Além disso permite interagir com o mapa uma vez que se clicar numa dessas caixas, o mapa é centrado para percebermos geograficamente onde será realizada essa recolha ou entrega.

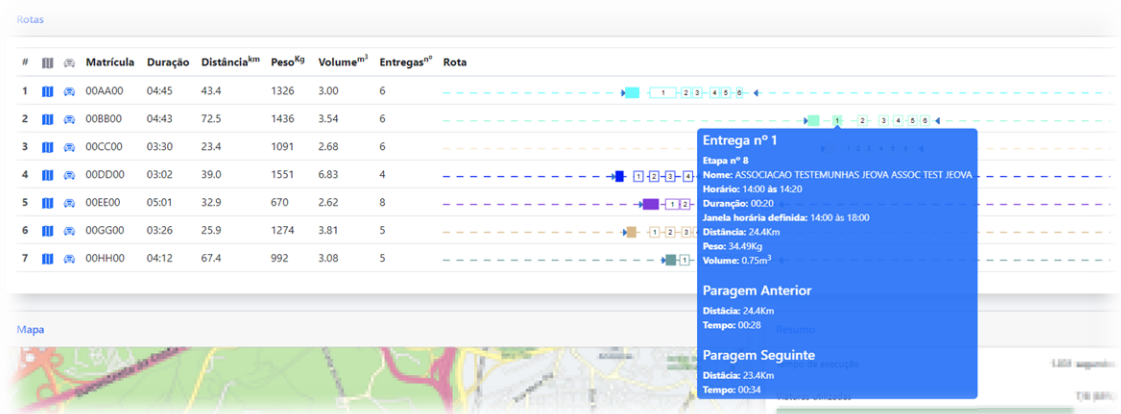


Figura 28 - Painel de consulta das rotas otimizadas

Para além dessa interação com o mapa, é possível ainda esconder ou mostrar os locais de recolha e entrega dessa rota ou de todas as rotas para mais facilmente identificar os serviços de cada viatura. Além disso, é possível consultar a rota definida para essa viatura, ou todas, mostrando no mapa esse mesmo trajeto com a mesma cor do gráfico dessa viatura. O trajeto mostrado tem em conta todas as restrições e configurações definidas para essa viatura e o trajeto é apenas calculado neste momento para diminuir os tempos de carregamento e de cálculo, aumentando assim a experiência de utilização do utilizador.

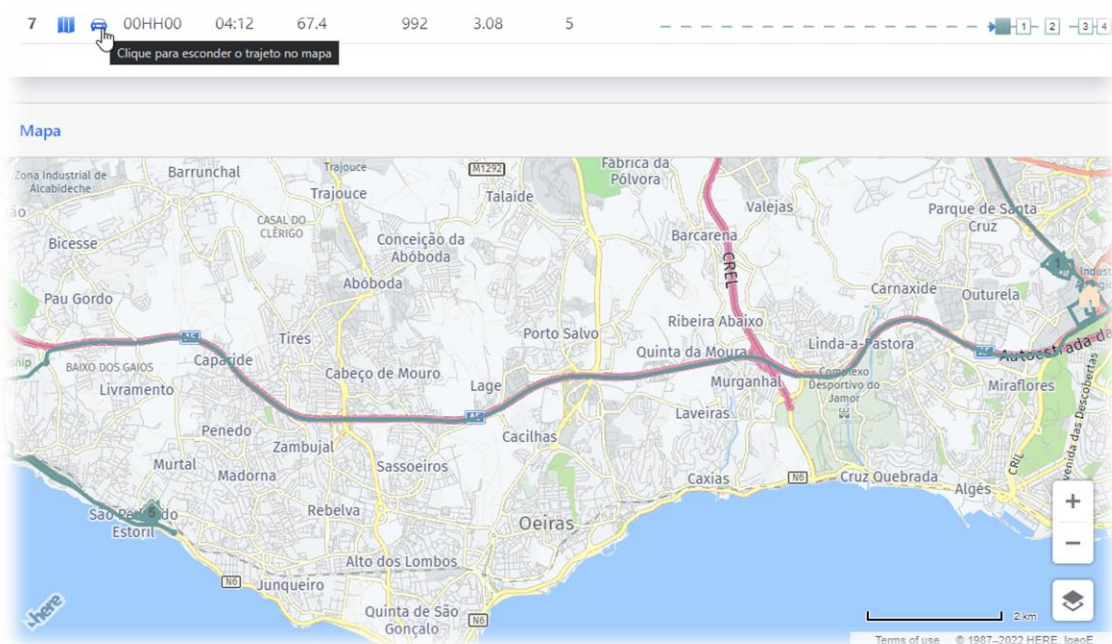


Figura 29 - Mapa interativo para consulta dos serviços e das rotas otimizadas

Por fim, temos um pequeno painel de resumo do resultado do otimizador demonstrado na Figura 30, onde estão um conjunto de parâmetros a analisar, nomeadamente:

- **Tempo de execução:** podemos observar neste exemplo que o tempo de execução foi de apenas 1.031 segundos. Bastante rápido para um conjunto de 40 serviços e 8 viaturas. Este valor é algo a ter em conta, pois quanto maior o tempo de espera, menor a vontade do utilizador em utilizar esta ferramenta.
- **Viaturas utilizadas:** um dos maiores custos deste tipo de empresas é o da viatura, pois sem a mesma o transporte não pode ser realizado. Quanto maior o número de viaturas melhor, pois significa que o parque automóvel do cliente está dimensionado corretamente para a quantidade de serviços a realizar. Por outro lado, se ficarem várias viaturas de fora significa que o cliente tem vários recursos por utilizar e que estão a ser pagos.
- **Serviços atribuídos:** este parâmetro demonstra quantos serviços foram colocados ou, por outro lado, quantos serviços ficaram por alocar. Um serviço pode ficar não alocado por diferentes motivos, como não existir capacidade física para ser transportado numa viatura, ou a sua janela horária não ser exequível ou ficar muito distante, não dando tempo de uma viatura sair e voltar a tempo ao seu armazém, garantindo o seu período de trabalho, de condução ou de distância máximos.
- **Tempo de viagem e Distância percorrida:** demonstram os valores totais de todas as

viaturas versus a capacidade máxima da empresa. Por exemplo, o tempo total em viagem será de 28:41 horas, para uma capacidade máxima da empresa de 64 horas, distribuídas por cada viatura. Mais uma vez, quanto maior o tempo de condução utilizado, mas eficiente será a empresa.

- **Peso transportado e Volume transportado:** tal como no ponto anterior, estes campos comparam os pesos e volumes totais transportados versus a capacidade da empresa, ou seja, o somatório de todas as capacidades máximas de cada viatura, sendo que, quanto maior o valor, mas eficiente também será a empresa.



Figura 30 - Sumário de alguns pontos importantes sobre a otimização

10.5. Análise dos resultados

Uma vez terminada a implementação é fundamental agora avaliar o trabalho desenvolvido e validar a fiabilidade dos resultados obtidos. As primeiras validações ocorreram manualmente com a ajuda de mapas e de ferramentas para calcular as distâncias e tempos de percurso e perceber se os resultados obtidos eram humanamente exequíveis, ou seja, se de facto um motorista conseguia efetuar os percursos que estavam ali definidos. De salientar ainda que este projeto não contempla a previsão do trânsito, algo que pode influenciar imenso os tempos previstos do percurso.

Após essas primeiras validações, que decorreram com sucesso, seria interessante comparar um dia de atividade real otimizada por outra ferramenta já existente no mercado e compará-la com o resultado obtido. A empresa TRACKiT Consulting (Soluções de Mobilidade com controlo via GPS, 2022) teve a amabilidade de fornecer um dia real de atividade de 4 viaturas de um cliente não identificado, dia esse composto por 40 serviços.

Na Figura 31 podemos observar uma cópia de ecrã da plataforma Workwave com o resultado obtido, sendo importante realçar novamente que esta plataforma não tem em conta a tipologia da viatura, definindo-as como viaturas ligeiras no cálculo das rotas. Ao analisar os dados partilhados, percebemos que os serviços foram distribuídos por 4 viaturas no total de 448.30km percorridos. Podemos ainda verificar que as rotas têm períodos de pausa, algo que não foi contemplado nesta primeira fase de construção do algoritmo, pelo que é de esperar algumas diferenças quando comparados os 2 sistemas.

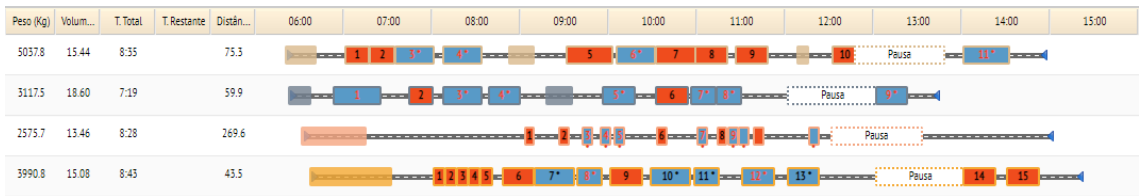


Figura 31 - Rotas otimizadas com a ferramenta Workwave

Após a receção dos dados enviados, os mesmos foram importados para o sistema tendo em conta as configurações dos mesmos, como o peso e volume a carregar e entregar, a posição geográfica de todos os locais e a definição das janelas horárias. Foi também tido em conta os tempos de carga e descarga definidos para cada serviço. Na Figura 32 é possível consultar a configuração de alguns desses serviços bem como a sua distribuição geográfica.

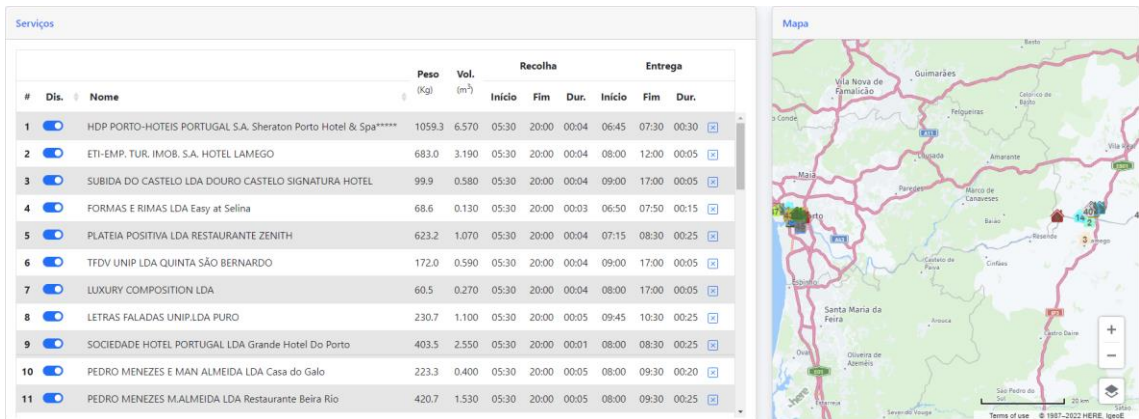


Figura 32 - Configuração dos serviços de acordo com a informação partilhada

As configurações das viaturas também foram alteradas para irem ao encontro das viaturas do cliente em análise, ficando com as características definidas na figura que se segue.

Geral						Trabalho			Cond.	Dist.	Dimensões						
#	Dis.	Tipo	Ref.	Marca	Modelo	Fact. Vel.	Início (h:m)	Final (h:m)	Máx. (h:m)	Máx. (h:m)	Máx. (km)	Alt. (m)	Lar. (m)	Comp. (m)	Elcos (m²)	Peso (kg)	Vol. (m³)
1		Rígido	00AA00	DAF	LF 210	Normal	05:30	20:00	09:00	08:00	800.0	2.86	2.54	7.11	2	4700	24
2		Rígido	00BB00	DAF	LF 210	Normal	05:30	20:00	09:00	08:00	800.0	2.86	2.54	7.11	2	4700	24
3		Rígido	00CC00	DAF	LF 45.180	Normal	05:30	20:00	09:00	08:00	800.0	2.86	2.55	7.11	2	5000	40
4		Rígido	00DD00	DAF	FUSO CANTER 9C18	Normal	05:30	20:00	09:00	08:00	800.0	2.15	1.88	5.05	2	5000	50
5		Rígido	00EE00	DAF	LF 45.180	Normal	05:30	20:00	09:00	08:00	800.0	2.86	2.54	7.11	2	2500	40

Figura 33 - Configuração das viaturas alterada

Uma vez executado o algoritmo, percebemos que o resultado é idêntico na sua distribuição, tendo apenas uma viatura com menos serviços pois, como não existe tempo de pausa, as restantes viaturas tiveram mais serviços atribuídos. Percebemos também que o total de quilómetros percorridos passou a 377.80km contra os 448.30km da outra plataforma, dando uma poupança de 70,50km que significa uma redução em 15%, algo bastante significativo.

#		Matrícula	Duração	Distância ^{km}	Peso ^{kg}	Volume ^{m³}	Entregas ^{nº}	Rota
1		00AA00	05:45	35.7	3795	17.60	13	
2		00BB00	06:04	34.3	3142	11.63	10	
3		00CC00	08:28	274.3	4312	21.00	14	
4		00EE00	02:30	33.5	1633	5.68	3	

Figura 34 - Simulação das rotas otimizadas

Olhando com mais detalhe, rota a rota, encontramos um padrão semelhante entre as 2 plataformas e, mais importante, conseguimos perceber que as entregas estão agrupados por pequenos clusters que estão assinalados nas figuras seguintes com uma elipse laranja.

A rota da viatura 00AA00 percorre apenas 35.7km numa duração de 05:45 horas servindo 13 clientes, todos eles na zona de gaia, exceto um que é servido no caminho entre o armazém e esse aglomerado.

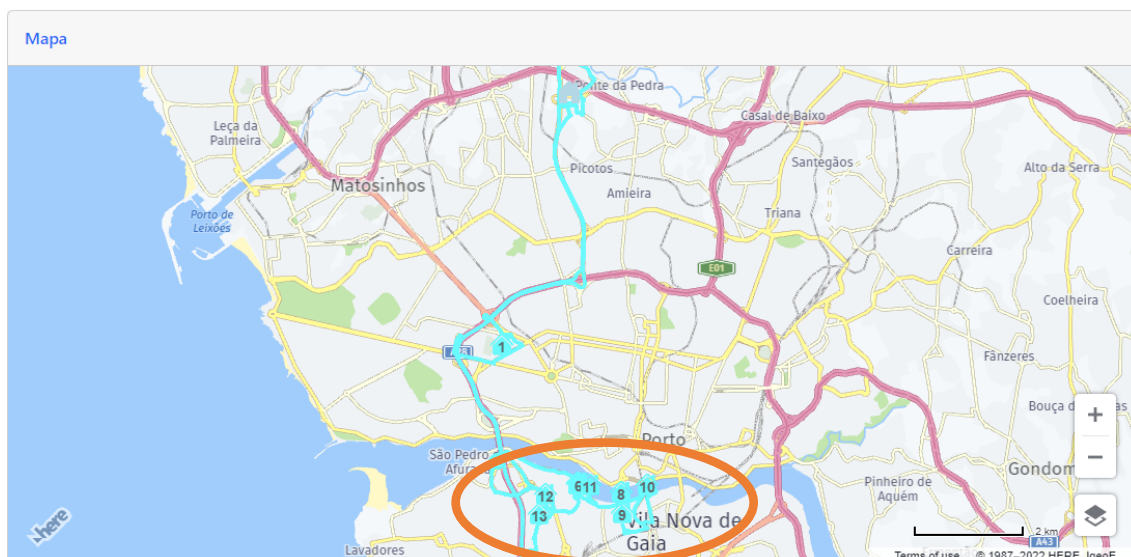


Figura 35 - Rota otimizada da viatura 00AA00

A rota da viatura 00BB00 está concentrada na zona sul da cidade do Porto onde se vê claramente mais um aglomerado de clientes. A viatura percorre 34.3km em 06:04 horas de trabalho.

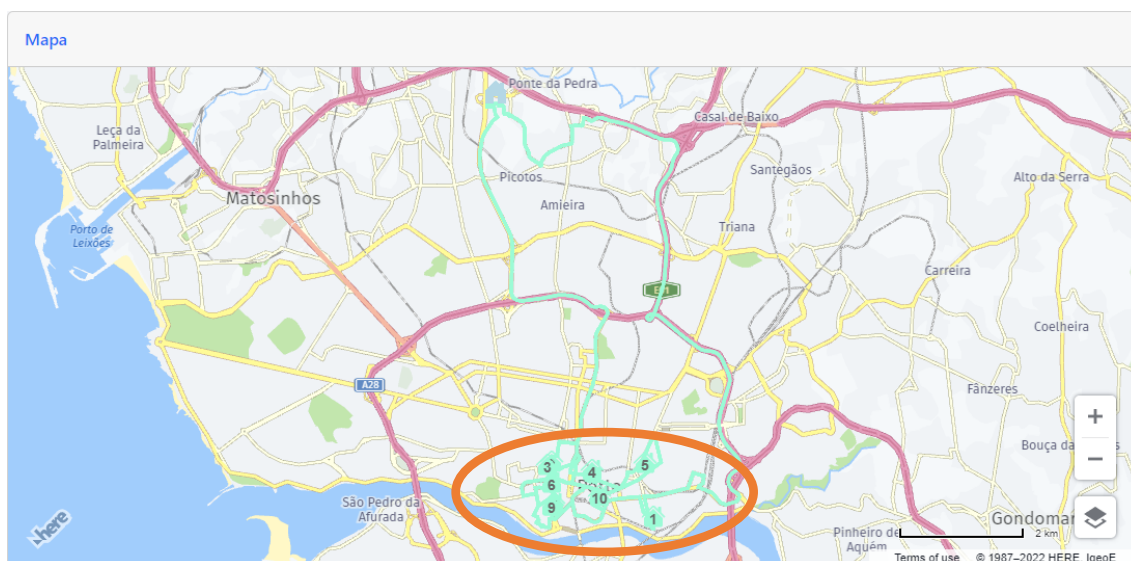


Figura 36 - Rota otimizada da viatura 00BB00

Analisando agora a rota da viatura 00CC00 o nosso olhar é dirigido automaticamente para o cluster presente na zona do Peso da Régua que está delimitada à direita do mapa, mas se observamos com mais atenção apercebemo-nos de um pequeno aglomerar na zona à esquerda, mais concretamente na zona de Gaia.

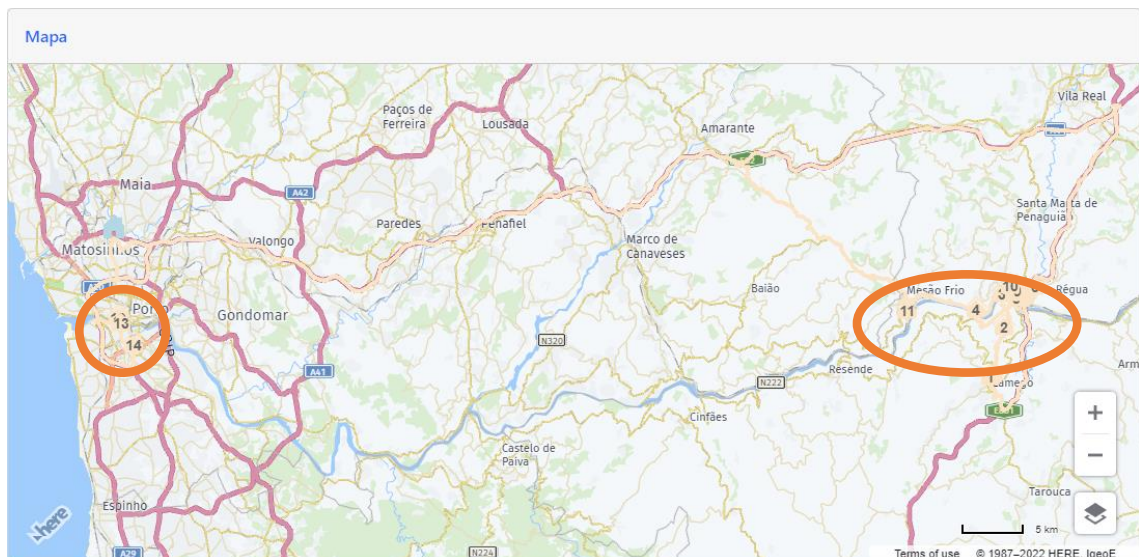


Figura 37 - Rota otimizada da viatura 00CC00

A rota em questão termina com a visita a 3 locais, assinalados e marcados como 12, 13 e 14 na Figura 38. Esses locais estão claramente na zona da primeira rota aqui analisada, a da viatura 00AA00. A questão que importa agora fazer é: qual o motivo para viatura 00AA00 não ter procedido à entrega desses serviços e os mesmos tiveram de ser entregues por uma segunda viatura, desviando-se e a percorrendo esses quilómetros extra?

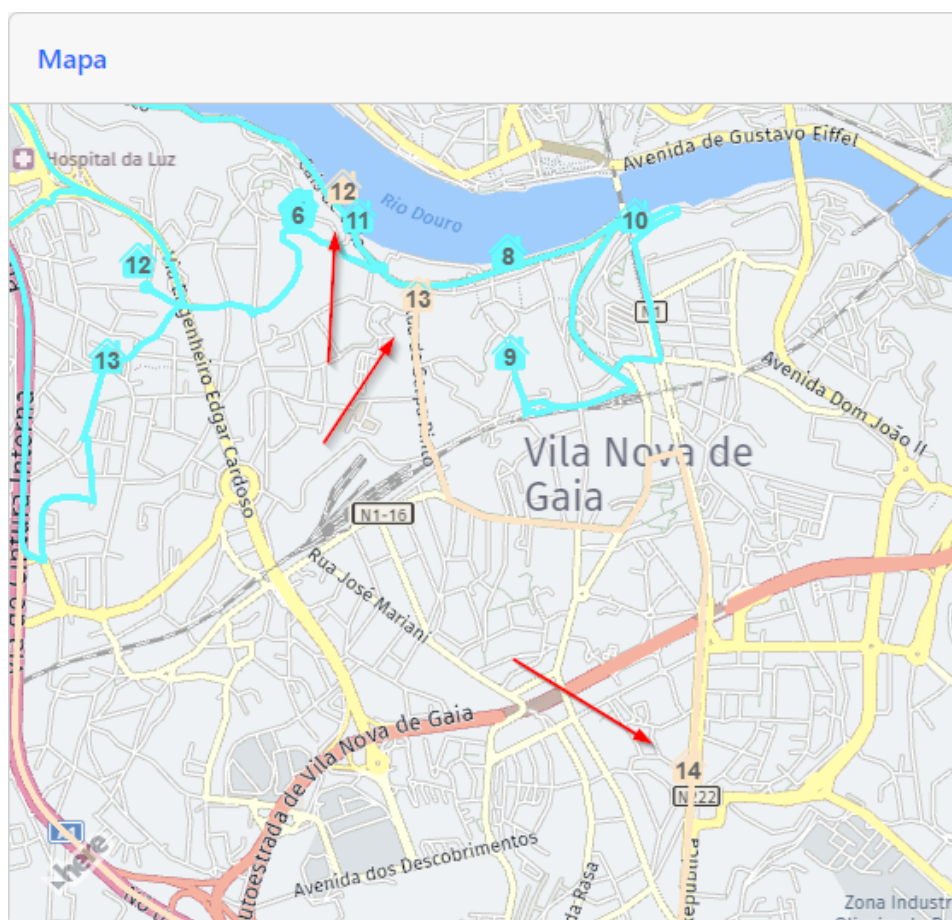


Figura 38 - Possível conflito entre rotas

Na análise que se segue conseguimos perceber que as restrições estão a funcionar e de acordo com o esperado. Os 3 serviços assinalados têm uma janela horária que os obriga a serem visitados no período da tarde, mas se vemos os gráficos fica ainda a dúvida pois aparentemente a viatura 00AA00 ainda tinha disponibilidade de efetuar essas entregas. Algo visível graficamente na figura que se segue.

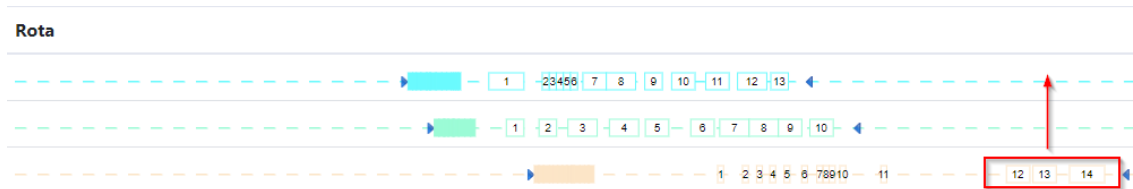


Figura 39 - Comparação entre as rotas 00AA00 e 00CC00

Se o problema não está na definição das janelas horárias, que outro tipo de restrição poderia ser? Olhando para a carga transportada vimos que a viatura 00AA00 está a carregar um total de 3.795Kg para um limite de 4.700Kg. Ou seja, a viatura apenas tem mais 905Kg de carga disponível, sendo que a conjugação dos 3 serviços somam 1.846,2Kg. Uma vez que a viatura não tinha capacidade de levar consigo toda a carga necessária para servir todos os clientes na mesma zona, restava 2 opções. Ou a viatura voltava ao armazém para se reabastecer com a restante carga, ou os serviços seriam entregues por outra viatura disponível. Neste caso, a segunda opção foi escolhida pelo otimizador por ser a mais eficiente. A viatura 00CC00 procedeu à entrega de todos os serviços na zona do Peso da Régua no período da manhã e depois serviu esses 3 clientes na zona de Gaia no percurso de volta ao armazém.

Serviços									
#	Dis.	Nome	Peso (Kg)	Vol. (m ³)	Recolha			Entrega	
					Início	Fim	Dur.	Início	Fim
1	☒	SABERSAL PROM TUR E IMOB SA HOTEL HILTON HILTON PORTO GAIA	485.1	1.053	05:30	20:00	00:04	14:30	17:00
2	☒	BLACK TULIP HOTELS SA Black Tulip Hotel	1006.9	5.230	05:30	20:00	00:03	14:00	17:00
3	☒	DOURO AZUL SOC MAR TUR S A QUEEN ISABEL	354.2	1.460	05:30	20:00	00:05	13:30	14:30

Figura 40 - Serviços com janelas horárias definidas no período da tarde

Ficaram ainda 3 serviços por realizar, todos com janelas horárias definidas entre as 9:00 e as 12:00. Algo impossível de ser integrado nas 3 rotas já analisadas, pelo que foi necessário ativar uma quarta viatura, a 00EE00, para realizar estes serviços.

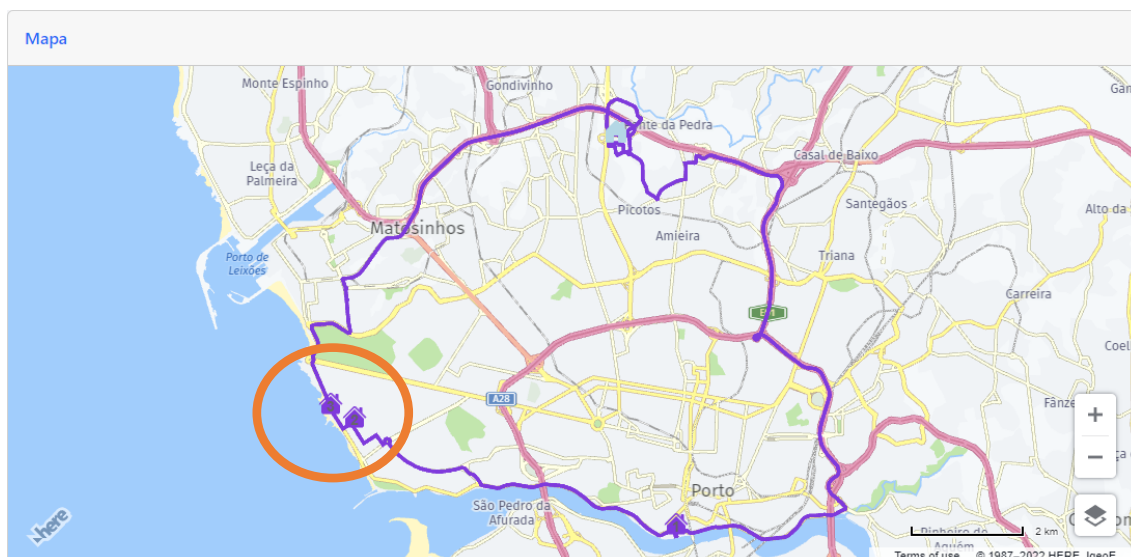


Figura 41 - Rota otimizada da viatura 00EE00

Este pequeno teste comprovativo foi um sucesso uma vez que o resultado foi muito positivo e o mesmo foi também validado pelo parceiro, dando indicações que o projeto está numa boa fase de maturidade e pronta para partir para as melhorias previstas no trabalho futuro.

Outro aspeto ainda a ter em conta, para além do resultado da otimização das rotas, será o tempo poupado pelo utilizador ou pelo responsável da empresa em organizar o trabalho para o dia seguinte. Segundo o parceiro, a pessoa responsável por este tipo de tarefa demora em média 2h por dia a organizar o seu trabalho, independentemente da dimensão da empresa. Isso acontece, pois, em empresas de maior dimensão acabam por dividir a frota ou o serviço por polos ou zonas, dando a responsabilidade a cada polo de gerir o trabalho para o dia seguinte com a gestão das viaturas que lhes foram assignadas. Algo que também poderá ser otimizado no futuro com a introdução de uma ferramenta deste género numa empresa.

Acreditamos que uma pessoa treinada nesta ferramenta consegue introduzir os serviços, configurar e manter o sistema em poucos minutos, poupando aqui horas de trabalho semanal e mensal. Se um utilizar gastar apenas 15 minutos por dia em otimizar as rotas e serviços para o dia seguinte, em comparação com os 120 minutos já partilhados, estamos a falar de 44 horas poupadas por mês. Imenso tempo que pode e irá ser dedicado a outras tarefas que não esta, aumentando assim a produtividade da empresa e também do trabalhador.

Tabela 11 - Comparação entre os tempos de trabalho

<i>Tempo manual</i>	2:00
<i>Tempo automatizado</i>	0:15
<i>Poupança</i>	1:45
<i>Poupança mensal</i>	44:00

10.5.1. *Dificuldades*

Desde o início do projeto, ainda durante a fase de elaboração da proposta, que as dificuldades foram surgindo e novos desafios foram somando à medida que o projeto foi ganhando forma. Finalmente aprovado o projeto nesta unidade curricular, as primeiras dificuldades surgiram ainda na fase de levantamento do estado da arte. Grande parte dos produtos disponíveis são privados e carecem de licença para que possam ser acedidos e assim estudados. Muitas das avaliações foram baseados na documentação dos proprietários ou de comentários disponíveis no mundo web, o que dificultou o arranque.

Dificuldades surgiram no estudo de algoritmos de otimização de rotas. Apesar de existirem diversos documentos e estudos científicos sobre a temática, muitos se não a totalidade, são estudo teóricos sem grande aplicabilidade ou forma fácil de aplicar no projeto, sendo a OR-Tools uma das melhores soluções que foi encontrada neste sentido. Mas, devido à sua especificidade e o nicho de negócio a que se destina, a comunidade online é bastante reduzida e foi difícil encontrar exemplos práticos da sua aplicação, obrigando ao estudo aprofundado do código aberto da solução e muita tentativa e erro para obter os resultados desejados. Isto aleado ao facto de a aplicação deste algoritmo ter sido implementada em Python, uma linguagem ao qual detínhamos pouco conhecimento, trouxe ainda dificuldades acrescidas nesta temática.

Passando agora para o mundo dos mapas, e a HERE Technologies, a eleita para esta implementação, trouxe também as suas dificuldades. Apesar de possuir uma documentação bastante completa e interessante, a sua comunidade online é ainda reduzida pois não dispõem de licenças gratuitas, de teste ou limitadas para a comunidade. Essa foi mesmo a primeira dificuldade, obter uma licença para poder testar e aceder à API completa, nomeadamente o uso da rotas e matrizes. Algo fundamental para poder arrancar e apresentar esta solução.

Uma vez garantido o acesso às diferentes ferramentas, trabalhar e integrar todas elas é, por si só, um grande desafio, pois obriga ao estudo aprofundado de toda a documentação necessária para integrar e aceder aos serviços, bem como identificar cada uma das suas opções e perceber os resultados obtidos.

Ainda durante a integração da HERE Technologies deparamo-nos com problemas que não estavam inicialmente contemplados como o facto da matriz apenas retornar uma matriz por localização, bem como o facto dessa mesma matriz estar limitada a um número máximo 100 de destinos, ou seja, para 50 serviços seria o limite pois cada serviço é composto por 2 locais, carga e descarga. Por fim, cada pedido apenas permite um tipo de configuração da tipologia de viatura, o que obrigava a que todas as viaturas fossem do mesmo tipo, por exemplo, todas pesadas com as mesmas características. Para contornar todos esses casos, foi necessário decompor o pedido em múltiplos e depois conjugar as respostas numa só matriz.

Por fim, e focando no desenvolvimento e implementação do algoritmo foram surgindo vários desafios, como garantir que cada restrição é cumprida e ter em conta as diferentes configurações de cada viatura ou local de carga e descarga. Foi também um desafio garantir que o mesmo serviço se mantivesse dentro da mesma rota, ou seja, a recolha tem de ocorrer na mesma rota onde será entregue, e tem de ocorrer dentro de uma janela horária.

Finalmente, a escolha do melhor algoritmo para calcular a estratégia da solução não foi tarefa fácil pois a biblioteca dispõe de uma grande variedade de soluções, mas a documentação sobre as mesmas é reduzida ou limitada, obrigando mais uma vez ao estudo aprofundado de cada uma delas.

Trabalhos Futuros

Sendo este projeto um protótipo, existe ainda caminho a percorrer para tornar esta ferramenta completa para que possa ser utilizada por empresas na sua atividade diária real. No que toca ao algoritmo de otimização existem ainda outras restrições que podem ser melhoradas ou implementadas, como garantir que apenas um determinado grupo de viaturas possa realizar um serviço, como por exemplo apenas viaturas com capacidade de transportar líquidos podem distribuir gasolina, ou se for para entregar produtos congelados terá de ser uma viatura com caixa frigorífica.

Existe também a necessidade de adicionar outro tipo de restrições de capacidade, não trabalhando apenas com peso e volume. Muitas empresas trabalham com unidades próprias como número de paletes ou número de caixas, e empresas de transporte de passageiros como autocarros ou ambulâncias trabalham com número de pessoas.

Foi ainda identificado a necessidade de definir períodos de descanso ou pausa. Estes tempos são fundamentais para o motorista poder descansar ou alimentar-se e, no que toca a viaturas pesadas, pausas e descansos são obrigatórias e regulamentadas pelas autoridades.

Hoje o algoritmo só devolve um resultado se a estratégia aplicada tiver pelo menos uma solução possível, mas seria também interessante devolver o melhor resultado possível encontrado, mesmo que não satisfaça todas as restrições. Este aspeto é especialmente complexo pois existem inúmeras restrições aplicadas a diferentes elementos, e gerir os limites que podem ser quebrados, sem comprometer a performance, não será tarefa fácil.

Por fim, seria interessante desenvolver um conjunto de relatórios de forma a conseguir comparar o que foi otimizado com o que de facto ocorreu, mas isso só é possível com integração com sistemas de geolocalização instalados nas viaturas.

Conclusão

Ao longo deste relatório foi demonstrado que os principais objetivos propostos na implementação do projeto, mesmos os mais ambiciosos, foram cumpridos com sucesso, desenvolvendo um protótipo funcional capaz de otimizar um conjunto de rotas para viagens pesadas tendo em conta um conjunto alargado de restrições.

É de salientar também a importância na definição da metodologia de trabalho que funcionou como bússola ao longo de todo o projeto. Este fio condutor ajudou a manter o foco e a orientação mesmo quando surgiram todas as dificuldades já mencionadas anteriormente.

Mesmo trabalhando nesta área por algo tempo, este trabalho foi especialmente desafiante uma vez que obrigou a um levantamento inicial exaustivo e ao teste das diferentes ferramentas que foram levantadas e que, por diferentes motivos, tinham especial interesse. Além disso foi adquirido um largo conhecimento nestas tecnologias, bem como na área de negócio aqui presente no decorrer das diferentes reuniões que foram ocorrendo durante a implementação do projeto.

Bibliografia

API Reference v1.18. (22 de 03 de 2022). Obtido de OptimoRoute:

<https://optimoroute.com/api/#introduction>

Benmansour, S. (05 de 2020). Optimizing Routes Using the Vehicle Routing Problem.

Bräysy, O. (02 de 2005). TRANSPORTATION SCIENCE. Vehicle Routing Problem with Time Windows, Part II: Metaheuristics.

Christofides, N. (1976). Worst-case analysis of a new heuristic for the travelling salesman problem. Report 388, Graduate School of Industrial Administration, CMU.

Clarke, G. &. (1964). Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. Operations Research, Vol. 12, pp. 568-581.

HERE Geocoding & Search API 7. (05 de 04 de 2022). Obtido de HERE Developer:

https://developer.here.com/documentation/geocoding-search-api/dev_guide/topics/endpoint-geocode-brief.html

Here Matrix Roating API. (05 de 04 de 2022). Obtido de Developer Guide:

<https://developer.here.com/documentation/matrix-routing-api>

HERE Roating API. (06 de 04 de 2022). Obtido de Developer Guide:

<https://developer.here.com/documentation/routing-api>

HERE Routing API 7. (06 de 04 de 2022). Obtido de HERE Developer:

https://developer.here.com/documentation/routing/dev_guide/topics/resource-calculate-route.html

HERE Routing API 7. (06 de 04 de 2022). Obtido de HERE Developer:

https://developer.here.com/documentation/routing/dev_guide/topics/resource-calculate-matrix.html

Holliday, A. W. (1972). Computer Scheduling of Vehicles from One or More Depots to a Number of Delivery Points. Operational Research Quarterly (1970-1977), Vol. 23, No. 3, pp. 333-344.

How to deal with tachograph failures. (10 de 03 de 2022). Obtido de trans.info:

<https://trans.info/en/how-to-deal-with-tachograph-failures-88938>

J.J. Schneider, S. K. (04 de 2006). Stochastic Optimization.

Route Optimization Algorithm and Big Data. (02 de 04 de 2022). Obtido de GSM Tasks:

<https://gsmtasks.com/route-optimization-algorithm-and-big-data/>

Routing API. (10 de 04 de 2022). Obtido de Sygic Developers:

<https://www.sygic.com/developers/professional-navigation-sdk/windows/api-examples/routing-api>

Routing API and Extended Routing API. (08 de 04 de 2022). Obtido de Tomtom Developers: <https://developer.tomtom.com/routing-api>

Routing API and Extended Routing API. (08 de 04 de 2022). Obtido de TomTom Developers: <https://developer.tomtom.com/routing-api/documentation/matrix-routing-v2/matrix-routing-v2-service>

Routing Options. (05 de 04 de 2022). Obtido de OR-Tools | Google Developers: https://developers.google.com/optimization/routing/routing_options

Routyn. (01 de 05 de 2022). Obtido de Routing Software Solutions: <https://routyn.com/solutions/>

Savelsbergh, M. (1989). A parallel insertion heuristic for vehicle routing with side constraints.

Soluções de Mobilidade com controlo via GPS. (01 de 05 de 2022). Obtido de TRACKiT Consulting: <https://www.trackit.pt/solucoes-de-mobilidade-gps/>

Vasileios Zeimpekis, C. D. (03 de 2007). Dynamic Fleet Management.

Vehicle Routing. (01 de 05 de 2022). Obtido de OR-Tools: <https://developers.google.com/optimization/routing>

What is a kanban board. (06 de 04 de 2022). Obtido de Atlassian: <https://www.atlassian.com/agile/kanban/boards>

WorkWave Routing Engine. (01 de 05 de 2022). Obtido de Route Optimization API: <https://www.workwave.com/routing-engine/>

Yaoxin Wu, W. S. (09 de 2022). Learning Improvement Heuristics for Solving Routing Problems. IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.

ANEXO I

```

MINGW64/c/PROJETS/Faculdade/route-planner
AzurEd-PauloFournier@DESKTOP-P4PBMG5 MINGW64 /c/PROJETS/Faculdade/route-planner (main)
$ tail -100f log/route-optimizer.log
[2022-10-06 15:52:45,791][root][INFO] read data from stdin
[2022-10-06 15:52:46,816][root][INFO] solution found in 1024ms
[2022-10-06 15:52:46,817][root][INFO] output solution
[2022-10-06 15:52:46,817][root][DEBUG] [0] LIMITS time:23616/86400, distance:0/9999999, weight:0/4700000, volume:0/24000
[2022-10-06 15:52:46,817][root][DEBUG] [03] LIMITS time:23616/86400, distance:0/9999999, weight:769320/4700000, volume:1523/24000
[2022-10-06 15:52:46,817][root][DEBUG] [09] LIMITS time:23736/86400, distance:0/9999999, weight:81670/4700000, volume:1658/24000
[2022-10-06 15:52:46,817][root][DEBUG] [039] LIMITS time:23976/86400, distance:0/9999999, weight:995841/4700000, volume:2011/24000
[2022-10-06 15:52:46,817][root][DEBUG] [040] LIMITS time:24216/86400, distance:0/9999999, weight:1029681/4700000, volume:2090/24000
[2022-10-06 15:52:46,817][root][DEBUG] [010] LIMITS time:24456/86400, distance:0/9999999, weight:1055497/4700000, volume:2157/24000
[2022-10-06 15:52:46,817][root][DEBUG] [0113] LIMITS time:24696/86400, distance:0/9999999, weight:1132672/4700000, volume:3003/24000
[2022-10-06 15:52:46,817][root][DEBUG] [0153] LIMITS time:26794/86400, distance:9497/9999999, weight:1326373/4700000, volume:3003/24000
[2022-10-06 15:52:46,817][root][DEBUG] [0160] LIMITS time:30852/86400, distance:11552/9999999, weight:1326073/4700000, volume:3003/24000
[2022-10-06 15:52:46,817][root][DEBUG] [0180] LIMITS time:32296/86400, distance:12211/9999999, weight:1326073/4700000, volume:3003/24000
[2022-10-06 15:52:46,817][root][DEBUG] [0179] LIMITS time:34550/86400, distance:18734/9999999, weight:1326073/4700000, volume:3003/24000
[2022-10-06 15:52:46,817][root][DEBUG] [0149] LIMITS time:36000/86400, distance:19662/9999999, weight:1326073/4700000, volume:3003/24000
[2022-10-06 15:52:46,817][root][DEBUG] [0143] LIMITS time:37996/86400, distance:22275/9999999, weight:1326073/4700000, volume:3003/24000
[2022-10-06 15:52:46,817][root][DEBUG] [0188] LIMITS time:40735/86400, distance:43400/9999999, weight:1326073/4700000, volume:3003/24000
[2022-10-06 15:52:46,817][root][DEBUG] [1181] LIMITS time:47227/86400, distance:0/9999999, weight:0/4700000, volume:0/24000
[2022-10-06 15:52:46,817][root][DEBUG] [1111] LIMITS time:47227/86400, distance:0/9999999, weight:404332/4700000, volume:547/24000
[2022-10-06 15:52:46,817][root][DEBUG] [1122] LIMITS time:47467/86400, distance:0/9999999, weight:613890/4700000, volume:761/24000
[2022-10-06 15:52:46,817][root][DEBUG] [131] LIMITS time:47707/86400, distance:0/9999999, weight:1043478/4700000, volume:1961/24000
[2022-10-06 15:52:46,817][root][DEBUG] [118] LIMITS time:47947/86400, distance:0/9999999, weight:1205888/4700000, volume:2293/24000
[2022-10-06 15:52:46,817][root][DEBUG] [133] LIMITS time:48187/86400, distance:0/9999999, weight:1240378/4700000, volume:3034/24000
[2022-10-06 15:52:46,818][root][DEBUG] [136] LIMITS time:48427/86400, distance:0/9999999, weight:1435666/4700000, volume:3545/24000
[2022-10-06 15:52:46,818][root][DEBUG] [173] LIMITS time:50400/86400, distance:24361/9999999, weight:1435666/4700000, volume:3545/24000
[2022-10-06 15:52:46,818][root][DEBUG] [157] LIMITS time:53574/86400, distance:47751/9999999, weight:1435666/4700000, volume:3545/24000
[2022-10-06 15:52:46,818][root][DEBUG] [158] LIMITS time:55422/86400, distance:58292/9999999, weight:1435666/4700000, volume:3545/24000
[2022-10-06 15:52:46,818][root][DEBUG] [176] LIMITS time:58292/86400, distance:60333/9999999, weight:1435666/4700000, volume:3545/24000
[2022-10-06 15:52:46,818][root][DEBUG] [151] LIMITS time:60155/86400, distance:62249/9999999, weight:1435666/4700000, volume:3545/24000
[2022-10-06 15:52:46,818][root][DEBUG] [1162] LIMITS time:61753/86400, distance:63351/9999999, weight:1435666/4700000, volume:3545/24000
[2022-10-06 15:52:46,818][root][DEBUG] [189] LIMITS time:64251/86400, distance:72470/9999999, weight:1435666/4700000, volume:3545/24000
[2022-10-06 15:52:46,818][root][DEBUG] [282] LIMITS time:49462/86400, distance:0/9999999, weight:0/1500000, volume:0/40000
[2022-10-06 15:52:46,818][root][DEBUG] [214] LIMITS time:49462/86400, distance:0/9999999, weight:112640/1500000, volume:191/40000
[2022-10-06 15:52:46,818][root][DEBUG] [223] LIMITS time:49907/86400, distance:0/9999999, weight:209809/1500000, volume:426/40000
[2022-10-06 15:52:46,818][root][DEBUG] [214] LIMITS time:49942/86400, distance:0/9999999, weight:288303/1500000, volume:570/40000
[2022-10-06 15:52:46,818][root][DEBUG] [28] LIMITS time:50002/86400, distance:0/9999999, weight:644996/1500000, volume:1358/40000
[2022-10-06 15:52:46,818][root][DEBUG] [215] LIMITS time:50242/86400, distance:0/9999999, weight:851859/1500000, volume:2248/40000
[2022-10-06 15:52:46,818][root][DEBUG] [212] LIMITS time:50482/86400, distance:0/9999999, weight:1090843/1500000, volume:2676/40000
[2022-10-06 15:52:46,818][root][DEBUG] [252] LIMITS time:51864/86400, distance:6192/9999999, weight:1090843/1500000, volume:2676/40000
[2022-10-06 15:52:46,818][root][DEBUG] [244] LIMITS time:53571/86400, distance:7826/9999999, weight:1090843/1500000, volume:2676/40000
[2022-10-06 15:52:46,818][root][DEBUG] [263] LIMITS time:54464/86400, distance:9952/9999999, weight:1090843/1500000, volume:2676/40000
[2022-10-06 15:52:46,818][root][DEBUG] [260] LIMITS time:56700/86400, distance:11654/9999999, weight:1090843/1500000, volume:2676/40000
[2022-10-06 15:52:46,818][root][DEBUG] [155] LIMITS time:56700/86400, distance:12578/9999999, weight:1090843/1500000, volume:2676/40000
[2022-10-06 15:52:46,818][root][DEBUG] [248] LIMITS time:59273/86400, distance:13681/9999999, weight:1090843/1500000, volume:2676/40000
[2022-10-06 15:52:46,818][root][DEBUG] [190] LIMITS time:62121/86400, distance:23388/999999
```

1 - Output do otimizador na sua execução