UNIVERSIDADE CANDIDO MENDES – UCAM
PROGRAMA DE PÓS-GRADUAÇÃO EM PESQUISA OPERACIONAL E INTELIGÊNCIA COMPUTACIONAL
CURSO DE MESTRADO EM PESQUISA OPERACIONAL E INTELIGÊNCIA COMPUTACIONAL

ANDRÉ BESSA DA SILVA

SOLUÇÃO MOBILE E USO DE TECNOLOGIA API REST PARA PROBLEMA DE ROTEIRIZAÇÃO

CAMPOS DOS GOYTACAZES, RJ.
2020

UNIVERSIDADE CANDIDO MENDES – UCAM
PROGRAMA DE PÓS-GRADUAÇÃO EM PESQUISA OPERACIONAL E INTELIGÊNCIA COMPUTACIONAL
CURSO DE MESTRADO EM PESQUISA OPERACIONAL E INTELIGÊNCIA COMPUTACIONAL

ANDRÉ BESSA DA SILVA

SOLUÇÃO MOBILE E USO DE TECNOLOGIA API REST PARA PROBLEMA DE ROTEIRIZAÇÃO

Dissertação apresentada ao Programa de Mestrado em Pesquisa Operacional e inteligência Computacional da Universidade Candido Mendes – Campos/RJ, para obtenção do grau de MESTRE EM PESQUISA OPERACIONAL E INTELIGÊNCIA COMPUTACIONAL.

Orientador: Prof. Ítalo de Oliveira Matias, D.Sc.

CAMPOS DOS GOYTACAZES, RJ.

Maio de 2020

Catalogação na Fonte

Preparada pela Biblioteca da **UCAM – CAMPOS** **035/2020**

Silva, André Bessa da.

Solução mobile e uso de tecnologia API REST para problemas de roteirização. / André Bessa da Silva. – 2020.
99 f.

Orientador: Ítalo de Oliveira Matias.

Dissertação de Mestrado em Pesquisa Operacional e Inteligência Computacional – Universidade Candido Mendes – Campos. Campos dos Goytacazes, RJ, 2020.
Referências: f. 95-99.

1. Algoritmo genético. 2. Aplicativo. I. Matias, Ítalo de Oliveira, orient. II. Universidade Candido Mendes – Campos. III. Título.

CDU – 004.421

Bibliotecária Responsável: Flávia Mastrogirolamo CRB 7ª-6723

ANDRÉ BESSA DA SILVA

SOLUÇÃO MOBILE E USO DE TECNOLOGIA API REST PARA PROBLEMA DE ROTEIRIZAÇÃO

Dissertação apresentada ao Programa de Pós-Graduação em Pesquisa Operacional e Inteligência Computacional, da Universidade Candido Mendes – Campos/RJ, para obtenção do grau de MESTRE EM PESQUISA OPERACIONAL E INTELIGÊNCIA COMPUTACIONAL.

Avaliado em 29 de Maio de 2020.

BANCA EXAMINADORA

Prof. Ítalo de Oliveira Matias, D.Sc. – Orientador

UCAM – Universidade Cândido Mendes

Prof. Dalessandro Soares Vianna D.Sc.

UFF – Universidade Federal Fluminense

Prof.ª Marcilene Dianin Vianna, D.Sc.

UFF – Universidade Federal Fluminense

CAMPOS DOS GOYTACAZES, RJ

2020

DEDICATÓRIA

Dedico esta experiência de formação à dona Terezinha da Conceição, minha amada avó, exemplo de pessoa e minha incentivadora fiel por toda a minha vida.

Dedico este trabalho ao meu pai Sebastião Paulo e à minha mãe Maria da Penha por sempre me apoiarem em tudo, mesmo nas idéias mais loucas, e, sobretudo são os meus maiores incentivadores quando o assunto é educação.

Dedico também às minhas filhas como um incentivo aos estudos, em suas futuras formações e para suas vidas.

AGRADECIMENTOS

Primeiramente agradeço a Deus pela força necessária concedida para encarar mais este desafio em minha vida.

Agradeço à minha família: meu pai, minha mãe, meus irmãos e minhas filhas que, com seu amor e compreensão, deram-me o combustível necessário para que eu conseguisse concluir esta etapa.

Agradeço também aos meus amigos de trabalho da *Gransystem* pela força e incentivo neste período de estudo.

Agradeço a todos os meus colegas do mestrado que dividiram comigo alegrias, conquistas e conhecimento, em especial, aos meus amigos de república Denilton Macário e Orpheu Ayres - esta conquista também é de vocês que merecem o meu muito obrigado.

Agradeço ao meu orientador, professor Ítalo de Oliveira Matias, que aceitou me orientar neste projeto de dissertação.

Um agradecimento todo especial a UCAM – Campos/RJ e seu corpo docente que realiza um trabalho de excelência neste curso de mestrado, contribuindo com a formação acadêmica e profissional de um grande número de pessoas.

“Parte da jornada é o Fim.” (Antony Edward Stark – personagem de Robert Downey Jr. Em *Vingadores Ultimato*).

RESUMO

O problema do caixeiro viajante se constitui como um dos clássicos problemas de otimização da Pesquisa Operacional, cujo cerne se concentra em encontrar a rota menos custosa, que permita a um caixeiro viajante passar por todas as cidades com o menor tempo possível, ou seja, é um problema de roteirização e que pode ser aplicado a outras realidades, como: serviços de entregas, orquestração de mensagens, dentre outros. Para resolver esse problema de roteirização, no caso do problema do caixeiro viajante, optou-se pela implementação do Algoritmo Genético (AG). Apresentado este cenário, o objetivo deste trabalho de pesquisa é o desenvolvimento de uma solução tecnológica que resolva o problema do caixeiro viajante utilizando algoritmos genéticos. Para alcançar o objetivo proposto serão desenvolvidas duas soluções: um aplicativo mobile que permitirá o usuário marcar rotas em um mapa georeferenciado e a criação de um serviço Web para a execução do AG. Na elaboração deste trabalho será adotada como metodologia a revisão bibliográfica para o embasamento conceitual e teórico, já para criação da solução optou-se por aplicar processos, rotinas e métodos de Engenharia de Software. Como resultado, a pesquisa apresenta um aplicativo mobile compatível com as principais plataformas de mercado e um serviço de processamento em nuvem que pode ser consumido por diversas aplicações que queiram processar rotas com base em coordenadas de um mapa.

Palavras-chave: Aplicativo. Algoritmo Genético. Caixeiro Viajante. Serviço Web.

ABSTRACT

The traveling salesman problem constitutes one of the classic optimization problems of Operational Research, the focus of which is to find the least expensive route, which allows a traveling salesman to go through all cities in the shortest possible time, that is, it is a routing problem that can be applied to other realities, such as: delivery services, message orchestration, among others. To solve this problem of routing, in the case of the traveling salesman problem, we opted for the implementation of the Genetic Algorithm (AG). Presented this scenario, the objective of this research work is the development of a technological solution that solves the traveling salesman problem using genetic algorithms. To achieve the proposed objective, two solutions will be developed: a mobile application that will allow the user to mark routes on a georeferenced map and the creation of a Web service for the execution of the AG. In the elaboration of this work, the bibliographic review for the conceptual and theoretical basis will be adopted as methodology, for the creation of the solution, it was decided to apply Software Engineering processes, routines and methods. As a result, the survey presents a mobile application compatible with the main market platforms and a cloud processing service that can be consumed by several applications that want to process routes based on the coordinates of a map.

Key words: Application. Genetic Algorithm. Traveling Salesman. Web Service.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AG	Algoritmo Genético
API	*Application Programming Interface*
BLE	*Bluetooth Low Energy*
CLR	*Common Language Runtime*
CRUD	*Create Retrive Update Delete*
CTS	*Common type system*
DLL	*Dynamic-link library*
GPS	*Global position system*
HTTP	*Hypertext transfer protocol*
ICP	*Iterative Closed Point*
IDE	*Integrated Development Environment*
MLS	*Mobile Laser Scanning*
MVC	*Model View Controller*
NSMS	*Normalized Sum Of Matching Scores*
NP	*Non Polynomial*
OFDM	*Orthogonal Frequency Division Multiplexing*
PCV	Problema do Caixeiro Viajante
REST	*Representational State Transfer*
SDK	*Software Development Kit*
TLS	*Terrestrial Laser Scanning*
UML	*Unified Modeling Language*
URI	*Uniform Resource Identifier*
VLGA	*Variable Length Genetic Algorithm*
WCF	*Windows Communication Foundation*

SUMÁRIO

1 INTRODUÇÃO

Caixeiro Viajante ou Problema do Caixeiro Viajante – PCV (do inglês *Traveling Salesman Problem* - TSP) constitui-se como um algoritmo cujo objeto é a busca de uma solução viável para estabelecer uma rota de menor percurso que passe por todos os pontos sem repeti-los, sem ser necessariamente a melhor rota. Constitui um problema classificado como NP Completo, em razão da solução ótima demandar um tempo de processamento computacional muito alto. O custo para encontrar a solução ótima aumenta proporcionalmente ao número de pontos a serem percorridos, podendo incorrer em meses, anos ou até décadas.

Reinelt (1994) afirma que o PCV é um dos que mais se destaca quando se trata de problemas de otimização combinatória. Para o autor, o PVC desperta o interesse de muitos pesquisadores - das mais diversas áreas - como: pesquisa operacional, matemática, física, biologia, inteligência artificial, dentre outras. Isso pela simplicidade e pela possibilidade de encontrar questões sobre otimização combinatória, que podem ser resolvidas com o PCV. Sendo assim, além de tudo, ele pode diminuir o tempo que se leva para encontrar uma solução viável.

> O PCV pode ser definido como o problema de encontrar o roteiro de menor distância ou custo que passa por um conjunto de cidades, sendo cada cidade visitada exatamente uma vez. Sua origem é creditada a William Rowan Hamilton, que inventou um jogo cujo objetivo era o de traçar um roteiro através dos vértices de um dodecaedro (vértices equivalem a cidades) que iniciasse e terminasse no mesmo vértice (cidade) sem, contudo, repetir uma visita. (Cunha; Bonasser; Abrahão, 2002, p. 2).

O PCV constitui-se, em um problema de otimização combinatória e é caracterizado pela presença de uma função-objetivo de maximização ou de minimização, que limita o conjunto de soluções.

Heurísticas existem como respostas genéricas que oferecem uma solução viável dentro dos limites de recursos necessários para que uma solução seja disposta, por exemplo, tempo e recursos computacionais. De acordo com Feigenbaum e Feldman (1963), a Heurística não só não garante uma solução ótima como não garante solução nenhuma. Tudo o que se pode afirmar é que ela pode oferecer, na maioria das vezes, uma solução boa o suficiente. Segue o trecho original em que se encontra essa afirmação:

> Uma heurística (regra heurística, método heurístico) é uma regra prática, estratégia, truque, simplificação ou qualquer outro tipo de dispositivo que limite drasticamente à busca por soluções em grandes espaços problemáticos. As heurísticas não garantem soluções ideais; de fato, eles não garantem nenhuma solução; tudo o que se pode dizer de uma heurística é que ela oferece soluções que são boas o suficiente na maioria das vezes. (FEIGENBAUM; FELDMAN, 1963, p. 61).

Neste cenário, a proposta aplica a utilização da heurística dos algoritmos genéticos, que segundo Arroyo (2002, p. 15) seriam:

> ...métodos inteligentes flexíveis, pois possuem uma estrutura com componentes genéricos que são adaptados ao problema que se quer resolver. Estes métodos possuem certa facilidade em incorporar novas situações e exploram o espaço de soluções permitindo a escolha estratégica de soluções piores que as já encontradas, na tentativa de superar a otimalidade local.

Dentre a gama de algoritmos classificados como heurísticas, estão os Algoritmos Genéticos. Os Algoritmos Genéticos (AG) são utilizados como uma técnica amplamente difundida de busca e otimização combinatória, que se inspira na teoria da evolução proposta pelo geneticista Charles Darwin (GOLDBERG, 1989).

O AG busca simular a natureza a partir da evolução de populações viáveis criadas aleatoriamente. No problema de roteirização o indivíduo é uma rota e o cromossomo é a sequência do conjunto de pontos. O processo utiliza abstração de cruzamento, seleção de melhores indivíduos, mutações, de acordo com uma avaliação cria-se um novo conjunto de soluções, que podem ser evoluídas. O ciclo de evolução pode ser interrompido por diversos fatores, sejam eles combinados ou não, como: limitação do tempo de resposta, quantidade de evoluções sem o surgimento de melhores indivíduos, limite de cruzamentos preestabelecidos, dentre outros fatores.

Para a solução do PCV como um problema de roteirização com a utilização de mapas, depara-se com conceitos e técnicas para geoprocessamento, que de acordo com RODRIGUES (1993), é um conjunto de tecnologias de coleta, tratamento, manipulação e apresentação de informações espaciais voltados para um objetivo específico. O geoprocessamento, que é em essência a adoção de tecnologias de informação aplicadas aos dados georeferenciados para localização e

posicionamento em mapas, traz neste processo, suporte para a integração do algoritmo a esta pesquisa.

Com a finalidade de resolver problemas de roteirização, como o do PCV, que integram mapas em sua proposta e levando em consideração avanços tecnológicos como a mobile e webservices este trabalho propõe a articulação dessas soluções com geoprocessamento, e a heurística dos Algoritmos Genéticos, para a criação de um aplicativo mobile.

1.1 OBJETIVOS DA PESQUISA

1.2 OBJETIVOS GERAIS

Como objetivo deste trabalho de pesquisa está uma proposta de desenvolver uma solução que resolva o Problema do Caixeiro Viajante aplicado aos problemas de roteirização, utilizando mapas georeferenciados em um aplicativo mobile.

1.3 OBJETIVOS ESPECÍFICOS

Como forma de atender o objetivo geral foram estabelecidos os seguintes objetivos específicos:

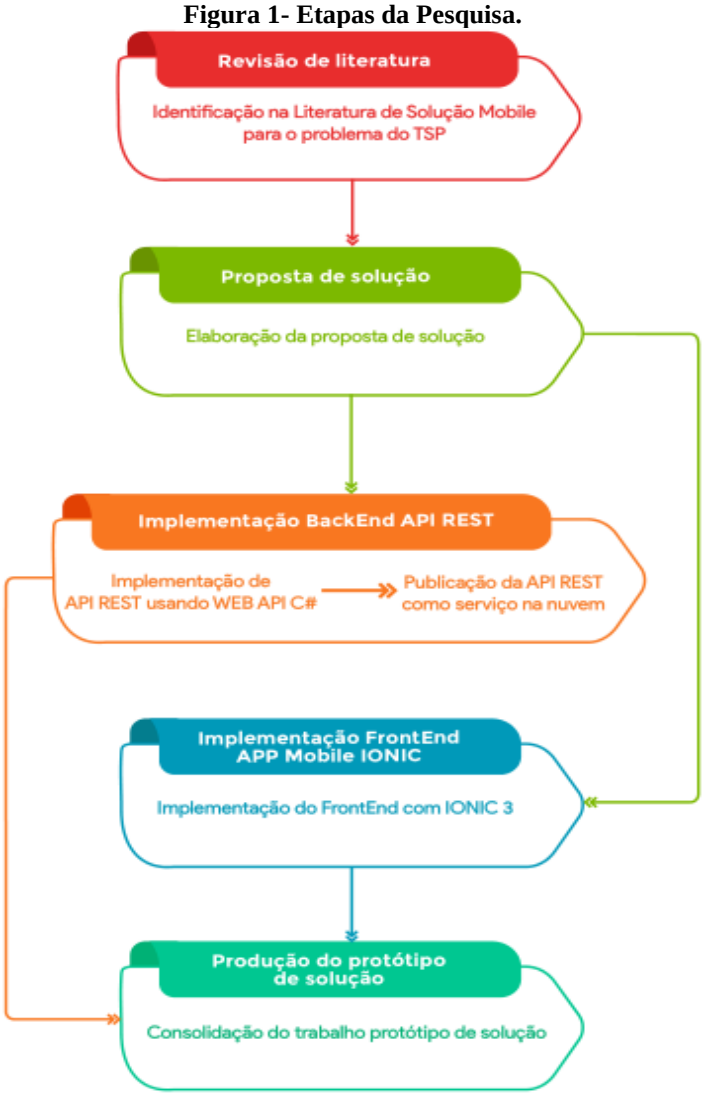
a) Identificar, na literatura, soluções aplicadas com sucesso ao problema do caixeiro viajante e de roteirização.
b) Desenvolver uma solução mobile que utilize mapas para marcação de pontos para traçar uma rota.
c) Criar um serviço web que processe as informações do aplicativo mobile e retorne uma solução viável dentro de conjunto de possibilidades, levando em consideração a disponibilidade dos recursos de hardware e software.

1.4 ETAPAS DA PESQUISA

Como estratégia de pesquisa para a realização deste trabalho foram propostas as seguintes etapas (Figura 1):

a) No âmbito da Revisão de literatura, serão pesquisados artigos indexados nas bases *SCOPUS* e *Scientifc Eletronic Library Online* (Scielo) para busca de soluções de roteirização e problemas do caixeiro viajante;
b) Elaboração de uma proposta de solução computacional para a resolução do Problema do Caixeiro Viajante.
c) Construção e hospedagem de uma REST API em tecnologia Microsoft C# que ofereça como serviço o processamento de informações georeferenciados, utilizando algoritmos genéticos;
d) Desenvolvimento de um aplicativo mobile, utilizando tecnologia móvel híbrida;
e) Consolidação do trabalho com o desenvolvimento de uma solução proposta.

Figura 1- Etapas da Pesquisa.



Fonte: O próprio autor.

1.5 DELIMITAÇÃO DA PESQUISA

Este trabalho se destina a construção de uma solução que envolve um protótipo de aplicativo mobile em tecnologia híbrida e uma REST API hospedada como um serviço na nuvem que implementará a solução de inteligência computacional dos Algoritmos Genéticos, sendo assim a solução a ser criada deve possuir:

a) Desenvolver um aplicativo mobile com uma interface simples e intuitiva;
b) Criação de um serviço web usando o modelo REST;
c) Criação de uma biblioteca (.dll) de software que implementa a heurística dos Algoritmos Genéticos.

1.6 ESTRUTURAÇÃO DO TRABALHO

Este trabalho está dividido em 5 capítulos, estruturados da seguinte forma: Introdução, Revisão de Literatura, Materiais e Métodos, Resultados e Considerações Finais.

I. Capítulo 1: Aborda o PCV como um problema de roteirização de tipo complexo, que pode ser solucionado com a utilização da heurística dos algoritmos genéticos. Partindo para a proposição de uma solução mobile conjugado a um serviço WEB do tipo REST, que irá roteirizar no mapa do dispositivo uma solução viável e, ainda, será realizada um consulta das bases cientificas, buscando por trabalhos que apresentem soluções para PVC.
II. Capítulo 2: Apresenta uma revisão de literatura iniciado pela definição de Algoritmo Genético, focando na solução para o problema do caixeiro viajante, na sequência, abordará a função objetivo usando a fórmula Haversine. Tudo isso tendo como objetivo traçar rotas em mapas utilizando a tecnologia disponibilizada pelo Google. Neste capítulo ainda foi realizada uma bibliometria na busca de trabalhos em consonância com o tema pesquisado e que poderá ser de grande valia para a definição de um ponto de partida para o início da pesquisa.
III. Capítulo 3: Descreve a sequência metodológica, os materiais e as ferramentas utilizadas para o desenvolvimento deste trabalho de pesquisa e do desenvolvimento do aplicativo de roteirização.

IV. Capítulo 4: São apresentadas as telas da solução mobile IONIC e descritas às funcionalidades disponibilizadas.

V. Capítulo 5: São postas as considerações finais e as conclusões do trabalho desenvolvido com foco no projeto, sob uma solução viável para roteirização pela ótica do problema do caixeiro viajante.

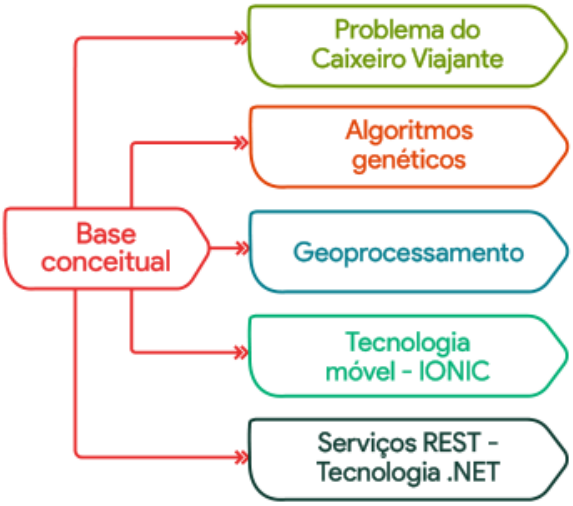
2 REVISÃO DE LITERATURA

Neste capítulo serão descritos os conceitos essenciais que estão relacionados ao entendimento deste trabalho. Além disso, esta revisão de literatura busca realizar uma pesquisa em trabalhos científicos e livros da área de computação e pesquisa operacional em consonância ao tema em questão. O trabalho nesta sessão será subdividido em duas partes: a base conceitual que evidencia os conceitos chaves e as tecnologias usadas além de uma bibliometria (*biblioming*), mostrando o estado da arte.

2.1 BASE CONCEITUAL

Nesta sessão será apresentada a revisão de literatura que abordará conceitos centrais que circundam o tema, com intuito de gerar os pilares teóricos de sustentação da referida pesquisa. Sendo assim, a base conceitual está estruturada conforme a Figura 2:

Figura 2: Base Conceitual.



Fonte: O próprio autor.

2.1.1 PROBLEMA DO CAIXEIRO VIAJANTE

De acordo com Coppin (2012), o problema do caixeiro viajante é um dos clássicos problemas da Inteligência Artificial, sendo considerado um NP-Completo que, resumidamente, significa que as grandes instâncias do problema podem ser algo muito custoso para um programa computacional em que se tem a pretensão de resolvê-lo em um período de tempo razoável.

Na resolução de problemas computacionais, busca-se classifica-los de acordo com o seu grau de complexidade e por quais motivos alguns problemas são difícieis de ser resolvido por computadores, Oliveira (2010). Um problema é considerado de classe P (*Polynomial*) se ele puder ser resolvido em um tempo polinomial e a partir de algoritmos determinísticos, ou seja, à medida que aumenta a complexidade do problema aumenta o tempo de processamento, a solução crescerá conforme uma função polinomial. Os problemas da classe NP (*Nondeterministic Polynomial*) englobam aqueles cuja solução pode ser encontrada em tempo polinomial e com algoritmos não-determinísticos em tempo polinomial.(SISPER, 2007).

De acordo com TOSCANI e VELOSO (2012) problemas classificados como NP-Completo são aqueles cuja complexidade não pode ser resolvida, por um algoritmo polinomial conhecido. Para Arora e Barak (2009) os NP-completos possuem complexidade exponencial e podem ser reduzidos de maneira polinomial a problemas de classe NP. Cobertura de vértices, escalonamento entre multiprocessadores, caminho hamiltoniano, clique em grafos, são alguns problemas NP-completos de acordo com Garey e Johnson (1979).

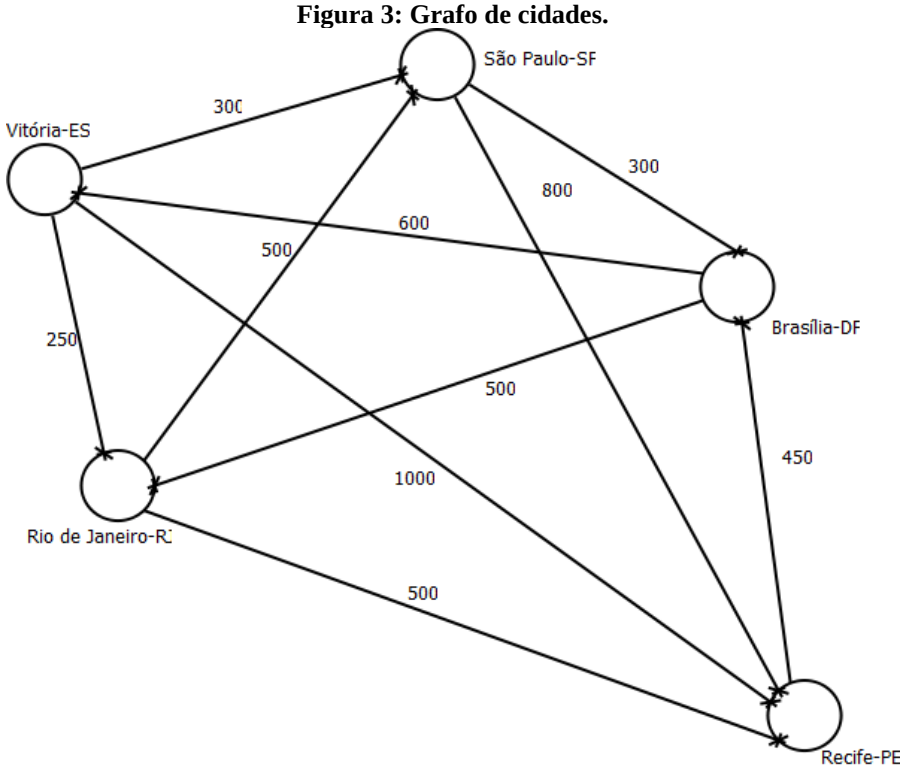
Como exemplos de problemas do tipo NP-completos, na literatura, Garey e Johnson (1979) acrescentam o PCV: um caixeiro viajante deve visitar cada uma das cidades de um conjunto de cidades e retornar para a cidade origem, nesse problema o objetivo é encontrar o menor caminho, de modo que o caixeiro passe por todas as cidades. Dada a explicação, é preciso imaginar a seguinte situação hipotética na qual um caixeiro viajante tem que percorrer cinco capitais brasileiras:

a) Vitória-ES;
b) Rio de Janeiro-RJ;
c) Brasília-DF;
d) Recife-PE;

e) São Paulo-SP.

Morando o caixeiro viajante na capital capixaba, ele tem que visitar as demais capitais antes de retornar para casa. Tomando como uma premissa que ele irá fazer o percurso de avião e que o custo de cada voo seja diretamente proporcional à distância a ser percorrida e que existem voos diretos entre os pares de cidades. Assim, pôde-se montar um grafo conforme o apresentado na Figura 3:

Figura 3: Grafo de cidades.



Fonte: O próprio autor.

O grafo da Figura 3 mostra o relacionamento entre as cidades, sendo que as distâncias descritas não representam as distâncias reais, mas estão expostas apenas com efeito de ilustrar o exemplo. Nessa representação as cidades são os vértices dos grafos e as arestas quantificadas com o custo.

2.1.2 ALGORITMOS GENÉTICOS (AG)

Os Algoritmos Genéticos (AG) é uma heurística que teve suas definições propostas por John Holland nos anos de 1960, que se baseou na Teoria da Evolução natural das espécies (Holland, 1975). O processo do AG inicia-se com a criação de uma população de soluções, que são os cromossomos, que serão

submetidos a operações que irão evoluir a população inicial até uma solução que atenda a alguns critérios de avaliação (Mitchell, 1998). Neste processo cada geração de indivíduos (cromossomos) é submetida a uma função, que busca atestar a aptidão deste como solução do problema. Os indivíduos mais capacitados darão origem a uma nova população de soluções por intermédio de operações como cruzamento e mutação.

Linden (2012) define os algoritmos genéticos como um ramo dos chamados algoritmos evolucionários, classificados como uma técnica de busca baseada na metáfora do processo de evolução biológica, proposto por Charles Darwin, na sua teoria da evolução das espécies. Para Coppin (2012), algoritmos genéticos é uma forma de busca local que usa como método uma abstração baseada em evolução para fazer pequenas alterações, em uma população de cromossomos, na tentativa de identificar uma solução ótima.

> Nos algoritmos genéticos populações de indivíduos são criadas e submetidas aos operadores genéticos: seleção, recombinação e mutação. Estes operadores utilizam uma caracterização da qualidade de cada indivíduo como solução do problema em questão chamada de avaliação e vão gerar um processo de evolução natural destes indivíduos, que eventualmente deverá gerar um indivíduo que caracterizará uma boa solução (talvez até a melhor possível) para o nosso problema. (LINDEN, 2012, p. 43).

Para Goldberg (1989), os algoritmos genéticos são uma técnica de otimização que utiliza paralelismo inspirada na teoria da evolução de Darwin. Conforme Michalewicz (1996), a abordagem de solução com base na utilização destes tem sido aplicada na resolução de problemas de otimização. Como, por exemplo, em abordagens que envolvem a otimização de funções matemáticas, problemas de roteirização, otimização de layoute de circuitos, otimização de negócios e, inclusive, para solução do problema do caixeiro viajante.

Linden (2012), também afirma que, a utilização de uma técnica baseada na evolução natural não objetiva buscar uma solução ótima, mas consiste no processo de colocar diversos indivíduos para competir pelo processo de sobrevivência no qual os mais aptos tendem a sobreviver. Para corroborar com essa afirmação, Tam (1992 apud LOPES, 1995) aborda que a utilização dos algoritmos genéticos para problemas de otimização - embora não encontrem necessariamente a melhor solução - permitem encontrar uma boa solução para o problema que se apresenta.

2.1.2.1 O ALGORITMO

O processo de execução do algoritmo genético padrão pode ser entendido como o fluxograma apresentado na Figura 4.

Figura 4 - Fluxograma do algoritmo genético.

Fonte: Adaptado de Ikeda, 2009.

Independente da forma que irá ser representado se tem:

1. Geração de uma população aleatória de cromossomos, que representa um indivíduo da população, que pode ser entendido como uma possível solução para o problema;
2. Realiza-se uma avaliação de cada indivíduo da população;
3. Realiza-se a seleção dos pais para a geração de uma nova população de indivíduos;
4. Aplicam-se os operadores de cruzamento (*crossover*) e mutação (*mutation*), onde, respectivamente, misturam-se os genes dos selecionados e muta-se um percentual dos novos indivíduos a serem inseridos em uma nova população.
5. Com base em um critério estabelecido, para-se o processo e se encontra uma boa solução para o problema, caso contrário o processo se reinicia avaliando novamente a população corrente.

Nessa representação de alto nível do algoritmo genético, abstraem-se algumas complexidades que são inerentes ao processo como: a) Obtenção da representação do cromossomo, que no desenvolvimento deste estudo é realizada de maneira binária; b) Uma função de aptidão que busca avaliar cada indivíduo para descobrir o grau de adequação de cada indivíduo à solução.

A representação dos indivíduos assume grande importância para o desenvolvimento do algoritmo genético, no qual o objetivo principal é traduzir o problema para o processamento computacional (LINDEN, 2012). O cromossomo, a representação do indivíduo, é uma estrutura de dados que representa uma possível solução para o problema em questão. Dentre as formas mais comuns de se representá-lo está a forma binária, proposta por John Holland, na década de 1960 (Holland, 1975). Na proposta de Holland, uma cadeia de bits vai representar um cromossomo e cada bit irá representar um gene, como ilustra a Figura 5.

Figura 5: Exemplo de representação cromossômica de Holland.

Fonte: Ikeda, 2009.

Com relação às funções que medem aptidão do indivíduo, de acordo com Coppin (2012), para as implementações mais tradicionais de algoritmos genéticos se faz

necessária uma métrica pela qual a aptidão de um indivíduo possa ser determinada objetivamente.

As funções de avaliação ou *fitness* utilizadas em algoritmos genéticos são usadas para determinar e medir o desempenho de um indivíduo como uma solução para o problema, sendo que as mais comuns são referentes aos problemas de otimização e a utilização da própria função-objetivo do problema (RODRIGUES *et al.* 2004). No modelo utilizado para implementação do projeto da dissertação, para o cálculo a distância entre os pontos traçados no mapa, foi utilizado a seguinte função de *fitness*, dada pela fórmula de *Haversine* (Figura 6):

Figura 6: Fórmula Haversine.

$$haver\sin\left(\frac{d}{R}\right) = haver\sin(\Delta\Phi) + \cos(\Phi_1)\cos(\Phi_2)haver\sin(\Delta\lambda)$$

Fonte: SHYLAJA, 2015.

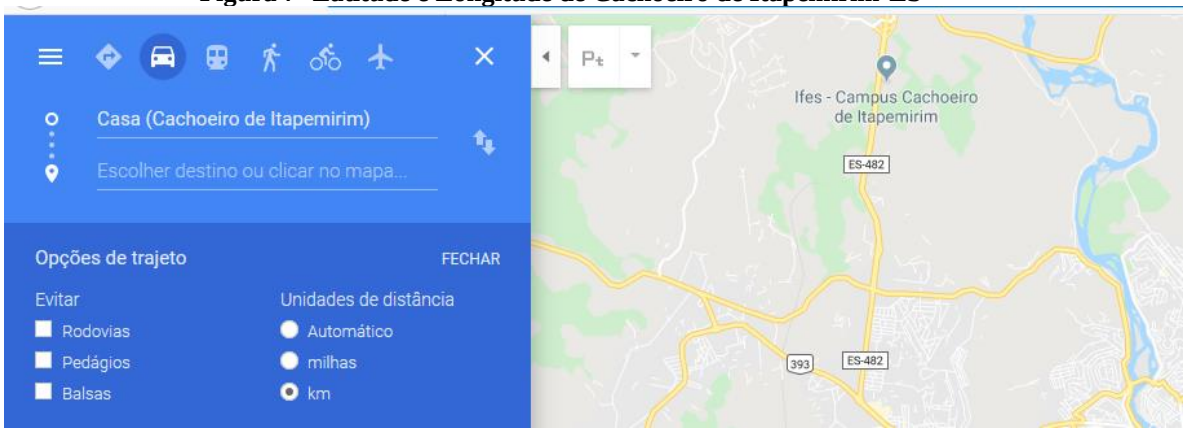
Na fórmula onde se tem:

- ***R*** é o raio da terra que é de 6371 quilômetros.
- ***d*** representa as distâncias entre dois pontos.
- As váriaveis Φ_1 e Φ_2 representam a latitude de dois pontos.
- As váriaveis λ_1 e λ_2 as longitudes respectivas desses dois pontos.
- Delta $\Delta\lambda = \lambda_2 - \lambda_1$ e Delta $\Delta\Phi = \Phi_2 - \Phi_1$

A fórmula de Haversine é uma importante equação usada em fórmulas de navegação, por permitir calcular a distância mínima entre dois pontos em mapas tomando como base informações sobre latitude e longitude (SHYLAJA, 2015).

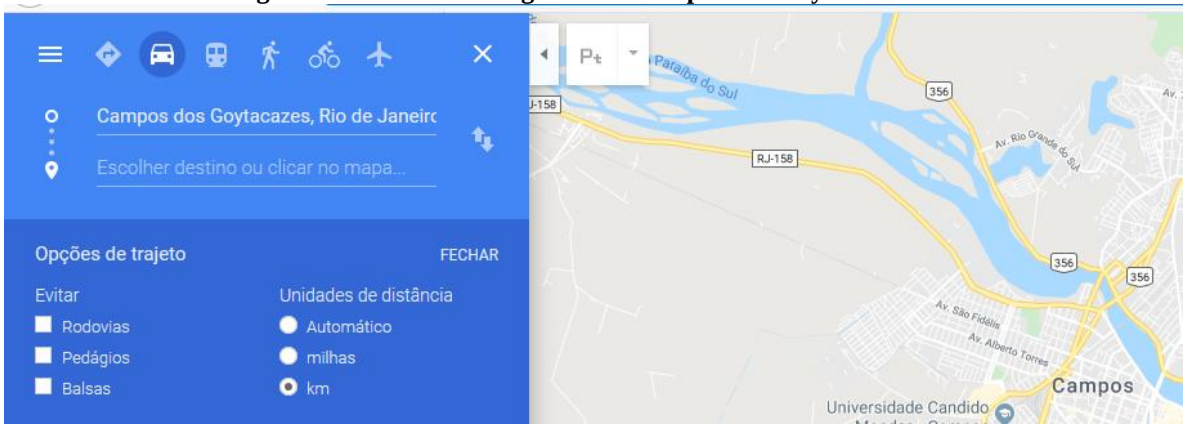
Como exemplo de aplicação da fórmula de Haversine foi obtido, através do Google mapas, as coordenadas entre duas cidades: Cachoeiro de Itapemirim, ES (-20.8761897,-41.0992887) e Campos dos Goytacazes, RJ (-21.7587266,-41.3617384). Figuras 7 e 8, respectivamente:

Figura 7- Latitude e Longitude de Cachoeiro de Itapemirim-ES



Fonte: Google Maps, 2019.

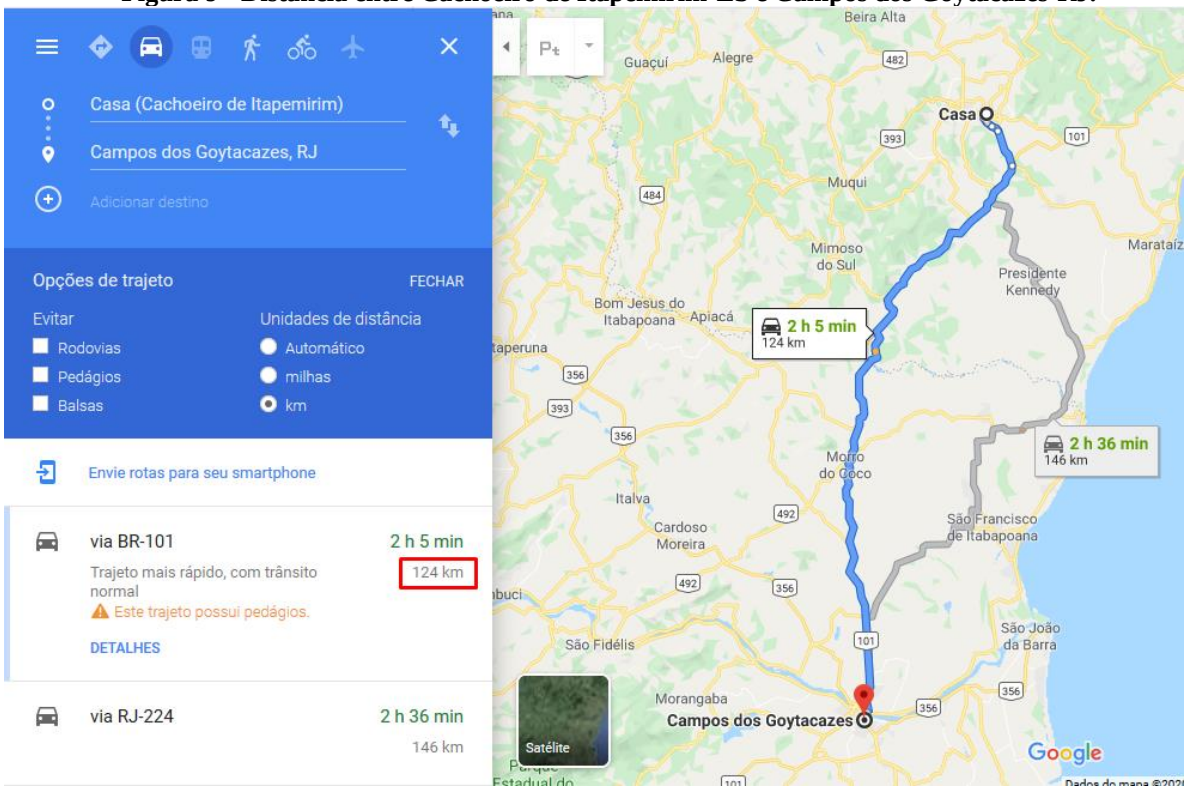
Figura 8 - Latitude e Longitude de Campos dos Goytacazes-RJ.



Fonte: Google Maps, 2019.

Usando o Google Maps, foi encontrada a distância de aproximadamente 124 quilômetros entre as duas cidades, conforme Figura 9.

Figura 9 - Distância entre Cachoeiro de Itapemirim-ES e Campos dos Goytacazes-RJ.



Fonte: Google Maps, 2019.

Para validar a função Haversine, foi implementando um código em linguagem C#, onde se aplicada a fórmula para o cálculo da distância entre as cidades de Cachoeiro de Itapemirim-ES e Campos dos Goytacazes-RJ, tomando como base as mesmas informações sobre latitude e longitude capturada do Google Maps. E, usando a fórmula foi desenvolvido um código de validação, utilizando a linguagem C# (Figura 10 e Figura 11), o resultado é apresentado no terminal do computador (Figura 12):

Figura 10 - Implementação de função Haversine em C#.

```
static double CalculoHaversine(double lat1, double lon1,
                double lat2, double lon2)
{
    //Fator de ajuste de 15% (1.15) e 20% (1.20) para considerar a distância Linear x rodoviária.
    double fatorajuste = 1.2;

    // Distâncias entre longitudes e latitudes
    double distLatitude = (Math.PI / 180) * (lat2 - lat1);
    double distLongitude = (Math.PI / 180) * (lon2 - lon1);

    // conversão para radianos
    lat1 = (Math.PI / 180) * (lat1);
    lat2 = (Math.PI / 180) * (lat2);

    // aplicação da formula Haversine
    double a = Math.Pow(Math.Sin(distLatitude / 2), 2) +
               Math.Pow(Math.Sin(distLongitude / 2), 2) *
               Math.Cos(lat1) * Math.Cos(lat2);
    double rad = 6371;
    double c = 2 * Math.Asin(Math.Sqrt(a));
    return rad * c * fatorajuste;
}
```

Fonte: O próprio autor.

Figura 11 - Implementação de função principal invocando a função *Cálculo Haversine* em C#.

```
static void Main(string[] args)
{
    // Cachoeiro de Itapemirim-ES
    double lat1 = -20.8467011;
    double lon1 = -41.1552396;

    //Campos dos Goytacazes-RJ
    double lat2 = -21.7587266;
    double lon2 = -41.3617384;

    Console.WriteLine(CalculoHaversine(lat1, lon1,lat2, lon2) + " aproximadamente Kilometros.");
}
```

Fonte: O próprio autor.

Figura 12 - Execução do código.

```
124,373227189127 aproximadamente Kilometros.
Pressione qualquer tecla para continuar. . .
```

Fonte: O próprio autor.

2.2 OPERADORES GENÉTICOS

De acordo com Rosa e Luz (2009), o objetivo dos operadores genéticos nos AG é a transformação da população, em cada nova geração, permitindo que tenhamos indivíduos cada vez melhores. Os operadores genéticos são classificados como: função de aptidão, cruzamento, mutação, atualização e finalização. Dentre os operadores Rosa e Luz (2009) destacam os operadores de seleção, cruzamento e mutação que nos ajudam de maneira mais eficiente na escolha da melhor solução.

2.2.1 SELEÇÃO

A seleção é um operador genético que é executado após o cálculo do fitness do indivíduo, de acordo com Izo, Matias e Passos. (2015) este é o momento em que duas ou mais soluções serão selecionadas para a próxima etapa do algoritmo, a quantidade das soluções selecionadas é a mesma da população inicia, sendo os métodos mais utilizados o torneio e a roleta (IZO*,MATIAS e PASSOS,* 2015).

Sobre o torneio Rosa e Luz (2009) argumentam que o processo se dá por meio da seleção de um número de indivíduos da população de forma aleatória. Esses indivíduos disputam entre si (*fitness* avaliado) e os melhores irão compor uma população intermediária, que será inserida na população novamente e entrando num processo até que a população esteja completa novamente.

No processo da execução da seleção pela roleta Izo, Matias e Passos (2015) explica que é criada uma roleta virtual na qual cada indivíduo recebe uma fatia da roleta proporcional a sua avaliação fitness, onde a soma das fatias serão igual a 360 em uma clara alusão aos graus de uma roleta circular, a roleta é girada e os pais selecionados. Esse processo vai ser repetido de acordo com o tamanho da população (IZO*, MATIAS e PASSOS,* 2015). Sobre a técnica da roleta Rosa e Luz (2009, p. 4) dizem:

> Deste modo, os cromossomos que possuírem um alto valor de aptidão ocuparão uma maior fração da roleta, enquanto que os cromossomos com valor de aptidão inferior ocuparão menores frações. Para que seja possível obter o número de pares necessários para a execução dos processos de cruzamento e mutação a roleta é girada quantas vezes forem necessárias.

Para Lurge (2013) é importante ressaltar que em alguns métodos de seleção os indivíduos menos aptos não são descartados inteiramente do processo, pois eles

podem conter algum componente que possa ser de suma importância para a solução do problema, podendo assim em combinação de genes posteriores gerar indivíduos melhores.

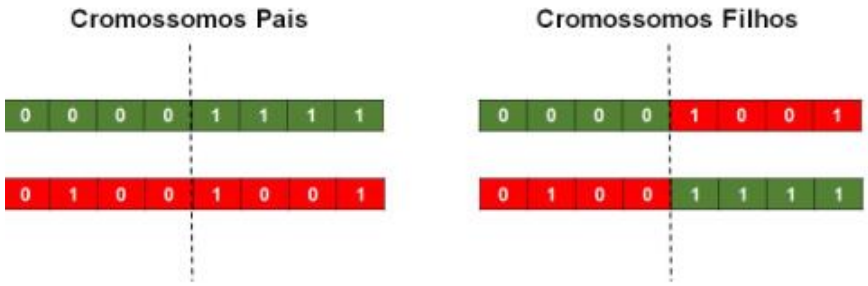
2.2.2 CRUZAMENTO (*Crossover*)

O cruzamento (*crossover*) é a fase onde realizamos a troca de material genético dos selecionados na etapa de seleção, e onde desta combinação de pares de cromossomos teremos novos indivíduos para compor a nova geração da população. O crossover tem como premissa a propagação das melhores características dos indivíduos em uma geração da população.

De acordo com Ikeda (2009) e Linden (2012) o cruzamento é o principal e mais importante método para geração de novos indivíduos no AG, sendo quando há a troca de genes dos pais, que darão origem a outros membros da população, filhos. Essa troca de material genético irá dar origem a indivíduos mais aptos que seus genitores (HORTA, SOLDATI, FREITAS, MATIAS, 2014).

Segundo Rosa e Luz (2009), as formas mais comuns de cruzamento estão as de ponto único e ponto duplo. Na técnica de ponto único é determinado um ponto de corte na representação cromossômica dos pais que dará origem a dois novos indivíduos com a combinação destes pontos (Figura 13).

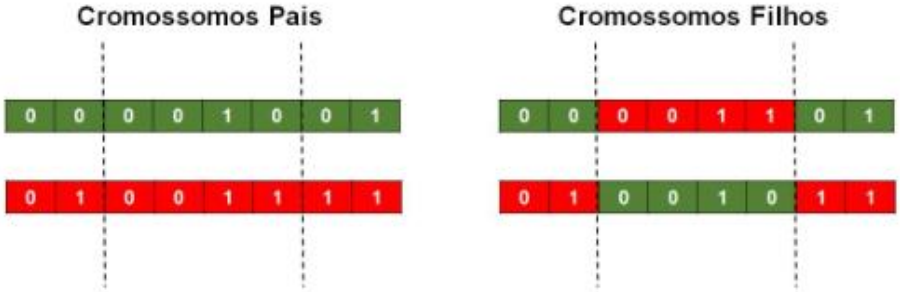
Figura 13 – Cruzamento ponto único.



Fonte: Gomes, Pereira, Marins, Silva (2017).

No cruzamento de dois pontos (Figura 14) escolhemos também de maneira aleatória dois pontos de cortes nos cromossomos dos pais, que serão recombinados de maneira intercalada.

Figura 14 – Cruzamento ponto duplo.



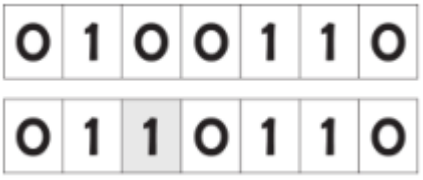
Fonte: Gomes, Pereira, Marins, Silva (2017).

2.2.3 MUTAÇÃO

O processo de mutação de um indivíduo consiste em aplicar uma modificação em um cromossomo selecionado de maneira aleatória. Para Izo, Matias e Passos. (2015) a idéia é que a população atual possua indivíduos com estrutura diferentes o suficiente para que as novas populações não fiquem com as mesmas heranças. Segundo Luger (2013) a mutação é um operador de suma importância ao processo do Algoritmo Genético, esse processo seleciona um indivíduo e troca de forma aleatória alguma de suas características. Uma mutação (Figura 15) pode ocorrer na simples troca de uma informação representada de maneira binária, por exemplo, de 1 para 0 ou vice-versa.

De acordo com Linden (2012), o AG assim como na natureza identifica nos cromossomos dos indivíduos informações que fazem que a população evolua entre as gerações, no tocante a mutação ela possibilita a diversidade por promover internamente ao indivíduo e de forma aleatória a mudança de genes assim como na natureza.

Figura 15 – Mutação troca binária.



Fonte: Ikeda (2009).

2.3 *WEBIBLIOMINING*

Nesta sessão do trabalho foi realizada uma pesquisa sistêmica nas bases de dados acadêmicas, visando identificar artigos e pesquisas acadêmicas que vão de encontro com a linha de pesquisa proposta neste trabalho, para tanto se lançou mão da metodologia *webibliomining*[1] proposta por Costa (2010) a qual busca selecionar produções de cunho acadêmico, permitindo que se alcancem mais informações para a pesquisa em curso ou área de conhecimento, formando o início do referencial teórico utilizado. Ela possui as seguintes etapas:

- Definir amostra de pesquisa;
- Pesquisar a amostra por meio de palavras-chave;
- Identificação dos periódicos com maior número de publicação relacionado ao tema;
- Identificação dos autores com maior número de publicações;
- Cronologia da produção;
- Seleção dos artigos para compor o “ponto inicial” do referencial teórico, que deve conter:
    - Artigos mais relevantes;
    - Identificação dos primeiros autores a escrever sobre o tema;
    - Identificação dos textos mais relevantes.

2.3.1 AMOSTRA DA PESQUISA

A Amostra da pesquisa, de acordo com o modelo de Costa (2010), congrega as bases de dados onde serão realizadas as pesquisas, como por exemplo, *Scopus*, *Web Of Science* e *Scielo*. A base de consulta escolhida para este projeto de pesquisa foi a *Scopus*, que foi acessada pelo portal da Capes[2]. A consulta à uma base de artigos se limitou apenas à base da Scopus por ser considerada o maior repositório de resumos e citações de trabalhos científicos e por estar em constante revisão pela comunidade científica mundial. Embora seja uma limitação que deve ser observada, ela não afeta de maneira significativa os resultados apresentados.

[1] Garimpagem de texto na rede web. Costa (2010).
[2] Coordenação de Aperfeiçoamento de Pessoal de Nível Superior e está vinculada ao Ministério da Educação e Cultura.

2.3.2 PALAVRAS-CHAVE

Com a base de dados de pesquisa definida, no caso deste trabalho a *Scopus*, iniciou-se o processo de escolha de palavras-chave que irão compor a *string*[3] de busca a ser executada na base de dados selecionada. Abaixo as palavras-chave selecionadas para a pesquisa:

- Genetic Algorithms;
- Mobile Application;
- Hibrid Mobile;
- Web Service;
- REST;
- Maps;
- Traveler Salesman Problem;
- Geoprocessing;
- Geolocation.

Tendo as palavras-chave consolidadas, foi definida a string de consulta a ser realizada na base Scopus (Quadro 1), onde foi considerado os campos “Título do Artigo”, “Palavras Chaves” e “Resumo” sintetizada na consulta da base *como TITLE-ABS-KEY*. Na pesquisa como expressão base, foi utilizada, *title-abs-key ("genetic algorithms" or genetic and algorithms) and title-abs-key ("mobile application" or "mobile")* e como filtros adicionais foram *title-abs-key ("traveler salesman problem" or "tsp") or title-abs-key ("rest api" or rest and api or rest) or title-abs-key ("web service" or "webservice" or web and service) or title-abs-key (geolocation or georeferencing or gps or "global positioning system" or "geoprocessing"))* e limitando a busca para considerar o ano de 2018 pelo acréscimo da expressão *pubyear < 2019* .

[3] Uma cadeia de caracter, como por exemplo, uma palavra.

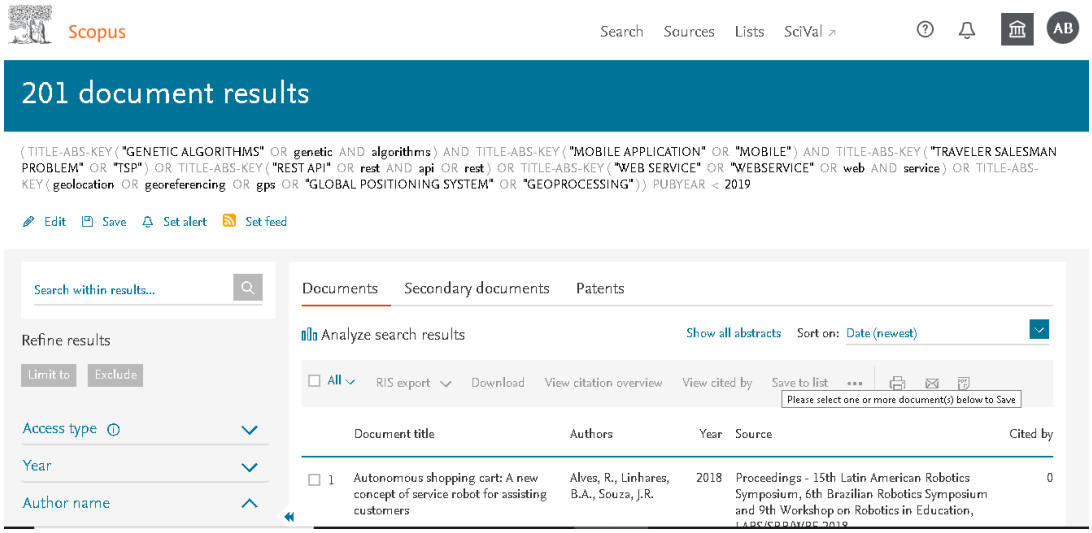
Quadro 1 - String de busca.

(TITLE-ABS-KEY ("GENETIC ALGORITHMS" OR GENETIC AND ALGORITHMS) AND TITLE-ABS-KEY ("MOBILE APPLICATION" OR "MOBILE") AND TITLE-ABS-KEY ("TRAVELER SALESMAN PROBLEM" OR "TSP") OR TITLE-ABS-KEY ("REST API" OR REST AND API OR REST) OR TITLE-ABS-KEY ("WEB SERVICE" OR "WEBSERVICE" OR WEB AND SERVICE) OR TITLE-ABS-KEY (GEOLOCATION OR GEOREFERENCING OR GPS OR "GLOBAL POSITIONING SYSTEM" OR "GEOPROCESSING")) PUBYEAR < 2019.

Fonte: O próprio autor.

Após a execução da string de consulta na base Scopus, foram retornados 201 trabalhos de pesquisa (Figura 16). Nessa consulta foram utilizados os campos Título do Artigo, Resumo e Palavras-Chave expressas pela sintaxe TITLE-ABS-KEY, que é o padrão na base Scopus.

Figura 16 - Resultado da execução da string de busca na base Scopus.



Fonte: O próprio autor.

2.3.3 RESULTADOS DA AMOSTRA

A Tabela 1 mostra o resultado da pesquisa na base Scopus dividindo os resultados por tipo de publicação.

Tabela 1 - Tipos de Publicações.

Tipo de Publicação	Quantidade
Artigo	43
Artigo de conferência	67
Capítulo de Livro	2
Revisão	1
Revisão de Conferência	88
Total	**201**

Fonte: Adaptado do Scopus (2019).

Nesta etapa da pesquisa, registraram-se os percentuais de cada um dos tipos de publicações da amostra, onde os artigos publicados correspondem a 21% de publicações, os artigos de conferências contribuem com 33%, os capítulos de livros e as revisões representam apenas 1% e as revisões de conferências representam o expressivo resultado de 44%.

Na Tabela 2, tem-se a distribuição das publicações por país, ranqueando os 10 primeiros com maior número de publicações relacionadas ao tema da pesquisa. A Tabela 2 apresenta o Brasil (5 publicações) ranqueado na 7º posição de trabalhos relacionados em comparação a China (24 publicações), Índia (19 publicações) e Estados Unidos (11 publicações) que respectivamente ocupam o primeiro, segundo e terceiro lugar.

Tabela 2 - Distribuição por país.

País	Quantidade
China	24
Índia	19
Estados Unidos	11
Taiwan	10
Reino Unido	6
Austrália	5
Brasil	5
Canadá	4
Itália	4
Turquia	4

Fonte: Adaptado do Scopus (2019).

2.3.4 AUTORES POR NÚMERO DE PUBLICAÇÕES

Na análise dos resultados da consulta na base Scopus, identificou-se que dentre os autores que têm trabalhos publicados nesta base, os dez primeiros possuíam pelo menos duas publicações e os demais apenas uma. Na Tabela 3, apresentam-se os 10 primeiros autores com mais publicações.

Tabela 3 - Autores com mais publicações na pesquisa *Scopus*.

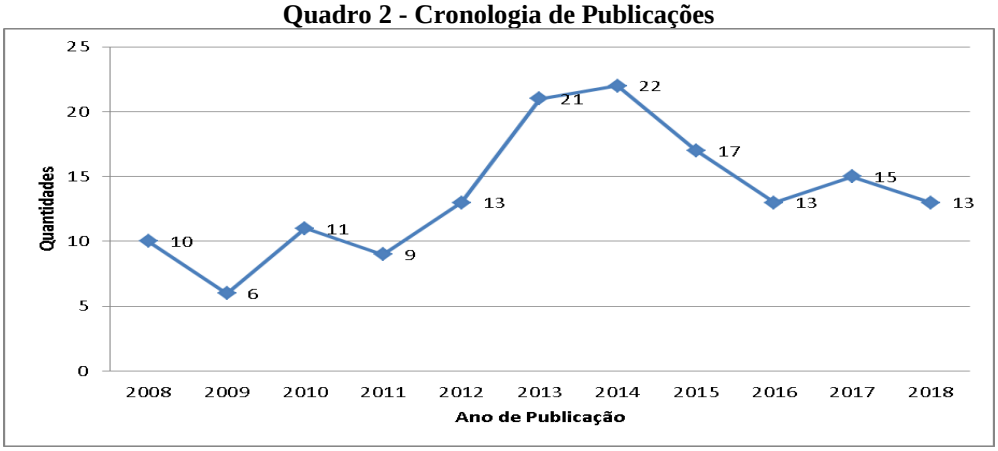
Autores
Altshuler, E.E.
Chakraborty, S.
Chang, C.H.
Chang, S.E.
Chen, C.
Hiramatsu, Y.
Kaushik, S.
Kumaki, K.
Kuwahara, Y.
Liu, C.C.

Fonte: Adaptado do Scopus (2019).

2.3.5 CRONOLOGIA DAS PUBLICAÇÕES

Para análise cronológica das quantidades de publicações relacionados ao tema de pesquisa, foram considerados os últimos dez anos e, no Quadro 2, encontra-se o consolidado de publicações por ano.

Quadro 2 - Cronologia de Publicações



Fonte: Adaptado do Scopus (2019).

O período compreendido entre 2013 e 2015, fica evidenciado como o período de maior número de publicações relacionadas ao tema desta pesquisa levando-se em consideração os filtros realizados na busca.

2.3.6 ESTABELECENDO O PONTO DE PARTIDA

Como proposto por Costa (2010), baseando-se no resultado obtido, foram estabelecido alguns critérios para definir o núcleo de partida do referencial teórico e as regras propostas nesse trabalho são:

a) Na tabela 4, é apresentado uma seleção dos 3 artigos mais antigos da base e de autores diferentes. Costa (2010) propõe este artifício com o intuito da identificação de linhas de pensamentos diferentes.

Tabela 4 - Seleção dos 3 trabalhos mais antigos.

Título documento	Autor (es)	Ano de Publicação	Origem (Periódico)
An improved heuristic crossover operator for TSP	Cong, Z., Jinhua, Z., Wangyi, L.	2008	Proceedings - 4th International Conference on Natural Computation, ICNC 2008 1,4666904, pp. 541-545
Adaptive image delivery based on WAP	Wang, Y.	2008	IET Conference Publications (545 CP), pp. 159-162
Research on constraint-based genetic algorithm approach for mobile supply chain dynamic decisions	Feng, Y., Wang, L.	2008	Proceedings of the 8th International Conference of Chinese Logistics and Transportation Professionals - Logistics: The Emerging Frontiers of Transportation and Development in China pp. 3718-3724

Fonte: Adaptado do Scopus (2019).

b) Costa (2010) sugere a escolha de 15 artigos para compor a seleção de trabalhos mais recentes, contudo nesta bibliometria chegou-se a 13 trabalhos dos mais recentes da base (Tabela 15) e de autores diferentes, aplica-se como critério de seleção a de regra identificação de linhas de pensamentos diferentes com a utilização de autores diversos (COSTA, 2010).

Tabela 5 - Os 13 últimos trabalhos mais recentes. (continua)

Título documento	Autor (es)	Ano de Publicação	Origem (Periódico)
Autonomous shopping cart: A new concept of service robot for assisting customers	Alves, R., Linhares, B.A., Souza, J.R.	2018	Proceedings - 15th Latin American Robotics Symposium, 6th Brazilian Robotics Symposium and 9th Workshop on Robotics in Education, LARS/SBR/WRE 2018

Tabela 6 - Os 13 últimos trabalhos mais recentes. (continuação)

Título documento	Autor (es)	Ano de Publicação	Origem (Periódico)
Node Failure Aware Broadcasting Mechanism in Mobile Adhoc Network Environment	Banumathi, J., Kanthavel, R.	2018	Programming and Computer Software
A Variable Length Genetic Algorithm approach to Optimize Data Collection using Mobile Sink in Wireless Sensor Networks	Anwit, R., Jana, P.K.	2018	2018 5th International Conference on Signal Processing and Integrated Networks, SPIN 2018
Service Selection for Composition in Mobile Edge Computing Systems	Wu, H., Deng, S., Li, W., (...), Yin, J., Zomaya, A.Y.	2018	Proceedings - 2018 IEEE International Conference on Web Services, ICWS 2018 - Part of the 2018 IEEE World Congress on Services
Energy-Aware Service Composition of Configurable IoT Smart Things	Sun, M., Zhou, Z., Duan, Y.	2018	Proceedings - 14th International Conference on Mobile Ad-Hoc and Sensor Networks, MSN 2018
Development and validation of the source code for defining the optimal path of a mobile robot using genetic algorithm based TSP	Guha, S.K., Dutta, S., Saha, S., Samanta, A.	2018	2017 4th International Conference on Opto-Electronics and Applied Optics, Optronix 2017
Clustered Crowd GPS for Privacy Valuing Active Localization	Yucel, F., Bulut, E.	2018	IEEE Access
Registration of TLS and MLS Point Cloud Combining Genetic Algorithm with ICP	Yan, L., Tan, J., Liu, H., Chen, C.	2018	Cehui Xuebao/Acta Geodaetica et Cartographica Sinica
Multi-objective optimization technique for resource allocation and task scheduling in vehicular cloud architecture: A hybrid adaptive nature inspired approach	Midya, S., Roy, A., Majumder, K., Phadikar, S.	2018	Journal of Network and Computer Applications
The TSP Solution for the Supermarket Chains Supply Route Based on 'Ant Colony - Particle Swarm' Algorithm	Lin, C., Wu, Y., Lin, Y.	2018	Chinese Control Conference, CCC

Tabela 7 - Os 13 últimos trabalhos mais recentes. (conclusão)

Título documento	Autor (es)	Ano de Publicação	Origem (Periódico)
16th International Conference on Service-Oriented Computing, ICSOC 2018		2018	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)
Towards cross-site scripting vulnerability detection in mobile web applications	Hydara, I., Sultan, A.B.M., Zulzalil, H., Admodisastro, N.	2018	International Journal of Engineering and Technology(UAE)
Customized Bus Line Design Model Based on Multi-Source Data	Chen, X., Wang, Y., Ma, X.	2018	International Conference on Transportation and Development 2018: Traffic and Freight Operations and Rail and Public Transit - Selected Papers from the International

Fonte: Adaptado do Scopus (2019).

c) Seleções dos 15 trabalhos com maior relevância presentes na base consultada (Tabela 6), é uma métrica também proposta por Costa(2010),a ordenação de relevância pode variar de uma base para outra, no caso da base *Scopus*, o critério adotado foi o número de citações.

Tabela 8 - Os 15 trabalhos de maior relevância na base. (continua)

Título documento	Autor (es)	Ano Publicação	Origem (Periódico)	Citações
Indoor localization without the pain	Chintalapudi, K., Iyer, A.P., Padmanabhan, V.N.	2010	Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM pp. 173-184	621

Tabela 9 - Os 15 trabalhos de maior relevância na base. (continuação)

Título documento	Autor (es)	Ano Publicação	Origem (Periódico)	Citações
Tracking the evolution of smartphone sensing for monitoring human movement	del Rosario, M.B., Redmond, S.J., Lovell, N.H.	2015	Sensors (Switzerland) 15(8), pp. 18901-18933	79
Use of mobile phones as intelligent sensors for sound input analysis and sleep state detection	Krejcar, O., Jirka, J., Janckulik, D.	2011	Sensors 11(6), pp. 6037-6055	64
Cloud-ECG for real time ECG monitoring and analysis	Xia, H., Asif, I., Zhao, X.	2013	Computer Methods and Programs in Biomedicine 110(3), pp. 253-259	61
Indoor localization algorithms for an ambulatory human operated 3D mobile mapping system	Corso, N., Zakhor, A.	2013	Remote Sensing 5(12), pp. 6611-6646	35
Biowep: A workflow enactment portal for bioinformatics applications	Romano, P., Bartocci, E., Bertolini, G., (...), Merelli, E., Milanesi, L.	2007	BMC Bioinformatics 8(SUPPL. 1),S19	34
Design of a recommender system for mobile tourism multimedia selection	Biuk-Aghai, R.P., Fong, S., Si, Y.-W.	2008	IMSAA'08 - 2nd International Conference on Internet Multimedia Services Architecture and Application 4753931	29
Design of a vehicular antenna for GPS/IRIDIUM using a genetic algorithm	Altshuler, E.E.	2000	IEEE Transactions on Antennas and Propagation 48(6), pp. 968-972	29

Tabela 10 - Os 15 trabalhos de maior relevância na base. (continuação)

Título documento	Autor (es)	Ano Publicação	Origem (Periódico)	Citações
Genetic algorithm aided proportional fair resource allocation in multicast OFDM systems	Sharma, N., Madhukumar, A.S.	2015	IEEE Transactions on Broadcasting 61(1),7031928, pp. 16-29	27
Multi-objective optimization technique for resource allocation and task scheduling in vehicular cloud architecture: A hybrid adaptive nature inspired approach	Midya, S., Roy, A., Majumder, K., Phadikar, S.	2018	Journal of Network and Computer Applications 103, pp. 58-84	24
Path planning of a data mule in wireless sensor network using an improved implementation of clustering-based genetic algorithm	Liu, J.-S., Wu, S.-Y., Chiu, K.-M.	2013	Proceedings of the 2013 IEEE Symposium on Computational Intelligence in Control and Automation, CICA 2013 - 2013 IEEE Symposium Series on Computational Intelligence, SSCI 2013 6611660, pp. 30-37	22
Mobile agent evolution computing	Shih, T.K.	2001	Information Sciences 137(1-4), pp. 53-73	22
Efficient route planning for an unmanned air vehicle deployed on a moving carrier	Savuran, H., Karakaya, M.	2016	Soft Computing 20(7), pp. 2905-2920	21
Optimum array configuration to improve null steering time for mobile CRPA systems	Byun, G., Hyun, J.-C., Seo, S.M., Choo, H.	2016	Journal of Electromagnetic Engineering and Science 16(2), pp. 74-79	21

Tabela 11 - Os 15 trabalhos de maior relevância na base. (conclusão)

Título documento	Autor (es)	Ano Publicação	Origem (Periódico)	Citações
An Efficient Resource Allocation Approach Based on a Genetic Algorithm for Composite Services in IoT Environments	Kim, M., Ko, I.-Y.	2015	Proceedings - 2015 IEEE International Conference on Web Services, ICWS 2015 7195613, pp. 543-550	20

Fonte: Adaptado do Scopus (2019).

Na tabela 7 está o compilado que compõe o ponto de partida dos artigos iniciais, ele é o resultado da união de resultados dos itens anteriores (a, b e c). Abaixo a tabela final de resultados:

Tabela 12 - Compilado de artigos iniciais. (continua)

Título documento	Autor (es)	Ano Publicação	Origem (Periódico)
An improved heuristic crossover operator for TSP	Cong, Z., Jinhua, Z., Wangyi, L.	2008	Proceedings - 4th International Conference on Natural Computation, ICNC 2008 1,4666904, pp. 541-545
Adaptive image delivery based on WAP	Wang, Y.	2008	IET Conference Publications (545 CP), pp. 159-162
Research on constraint-based genetic algorithm approach for mobile supply chain dynamic decisions	Feng, Y., Wang, L.	2008	Proceedings of the 8th International Conference of Chinese Logistics and Transportation Professionals - Logistics: The Emerging Frontiers of Transportation and Development in China pp. 3718-3724

Tabela 13 - Compilado de artigos iniciais. (continuação)

Título documento	Autor (es)	Ano Publicação	Origem (Periódico)
Autonomous shopping cart: A new concept of service robot for assisting customers	Alves, R., Linhares, B.A., Souza, J.R.	2018	Proceedings - 15th Latin American Robotics Symposium, 6th Brazilian Robotics Symposium and 9th Workshop on Robotics in Education, LARS/SBR/WRE 2018
Node Failure Aware Broadcasting Mechanism in Mobile Adhoc Network Environment	Banumathi, J., Kanthavel, R.	2018	Programming and Computer Software
A Variable Length Genetic Algorithm approach to Optimize Data Collection using Mobile Sink in Wireless Sensor Networks	Anwit, R., Jana, P.K.	2018	2018 5th International Conference on Signal Processing and Integrated Networks, SPIN 2018
Service Selection for Composition in Mobile Edge Computing Systems	Wu, H., Deng, S., Li, W., (...), Yin, J., Zomaya, A.Y.	2018	Proceedings - 2018 IEEE International Conference on Web Services, ICWS 2018 - Part of the 2018 IEEE World Congress on Services
Energy-Aware Service Composition of Configurable IoT Smart Things	Sun, M., Zhou, Z., Duan, Y.	2018	Proceedings - 14th International Conference on Mobile Ad-Hoc and Sensor Networks, MSN 2018
Development and validation of the source code for defining the optimal path of a mobile robot using genetic algorithm based TSP	Guha, S.K., Dutta, S., Saha, S., Samanta, A.	2018	2017 4th International Conference on Opto-Electronics and Applied Optics, Optronix 2017
Clustered Crowd GPS for Privacy Valuing Active Localization	Yucel, F., Bulut, E.	2018	IEEE Access
Registration of TLS and MLS Point Cloud Combining Genetic Algorithm with ICP	Yan, L., Tan, J., Liu, H., Chen, C.	2018	Cehui Xuebao/Acta Geodaetica et Cartographica Sinica

Tabela 14 - Compilado de artigos iniciais. (continuação)

Título documento	Autor (es)	Ano Publicação	Origem (Periódico)
Multi-objective optimization technique for resource allocation and task scheduling in vehicular cloud architecture: A hybrid adaptive nature inspired approach	Midya, S., Roy, A., Majumder, K., Phadikar, S.	2018	Journal of Network and Computer Applications
The TSP Solution for the Supermarket Chains Supply Route Based on 'Ant Colony - Particle Swarm' Algorithm	Lin, C., Wu, Y., Lin, Y.	2018	Chinese Control Conference, CCC
16th International Conference on Service-Oriented Computing, ICSOC 2018		2018	Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)
Towards cross-site scripting vulnerability detection in mobile web applications	Hydara, I., Sultan, A.B.M., Zulzalil, H., Admodisastro, N.	2018	International Journal of Engineering and Technology(UAE)
Customized Bus Line Design Model Based on Multi-Source Data	Chen, X., Wang, Y., Ma, X.	2018	International Conference on Transportation and Development 2018: Traffic and Freight Operations and Rail and Public Transit - Selected Papers from the International
Indoor localization without the pain	Chintalapudi, K., Iyer, A.P., Padmanabhan, V.N.	2010	Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM pp. 173-184
Tracking the evolution of smartphone sensing for monitoring human movement	del Rosario, M.B., Redmond, S.J., Lovell, N.H.	2015	Sensors (Switzerland) 15(8), pp. 18901-18933
Use of mobile phones as intelligent sensors for sound input analysis and sleep state detection	Krejcar, O., Jirka, J., Janckulik, D.	2011	Sensors 11(6), pp. 6037-6055
Cloud-ECG for real time ECG monitoring and analysis	Xia, H., Asif, I., Zhao, X.	2013	Computer Methods and Programs in Biomedicine 110(3), pp. 253-259

Tabela 15 - Compilado de artigos iniciais. (continuação)

Título documento	Autor (es)	Ano Publicação	Origem (Periódico)
Indoor localization algorithms for an ambulatory human operated 3D mobile mapping system	Corso, N., Zakhor, A.	2013	Remote Sensing 5(12), pp. 6611-6646
Biowep: A workflow enactment portal for bioinformatics applications	Romano, P., Bartocci, E., Bertolini, G., (...), Merelli, E., Milanesi, L.	2007	BMC Bioinformatics 8(SUPPL. 1),S19
Design of a recommender system for mobile tourism multimedia selection	Biuk-Aghai, R.P., Fong, S., Si, Y.-W.	2008	IMSAA'08 - 2nd International Conference on Internet Multimedia Services Architecture and Application 4753931
Design of a vehicular antenna for GPS/IRIDIUM using a genetic algorithm	Altshuler, E.E.	2000	IEEE Transactions on Antennas and Propagation 48(6), pp. 968-972
Genetic algorithm aided proportional fair resource allocation in multicast OFDM systems	Sharma, N., Madhukumar, A.S.	2015	IEEE Transactions on Broadcasting 61(1),7031928, pp. 16-29
Multi-objective optimization technique for resource allocation and task scheduling in vehicular cloud architecture: A hybrid adaptive nature inspired approach	Midya, S., Roy, A., Majumder, K., Phadikar, S.	2018	Journal of Network and Computer Applications 103, pp. 58-84
Path planning of a data mule in wireless sensor network using an improved implementation of clustering-based genetic algorithm	Liu, J.-S., Wu, S.-Y., Chiu, K.-M.	2013	Proceedings of the 2013 IEEE Symposium on Computational Intelligence in Control and Automation, CICA 2013 - 2013 IEEE Symposium Series on Computational Intelligence, SSCI 2013 6611660, pp. 30-37
Mobile agent evolution computing	Shih, T.K.	2001	Information Sciences 137(1-4), pp. 53-73

Tabela 16 - Compilado de artigos iniciais. (conclusão)

Título documento	Autor (es)	Ano Publicação	Origem (Periódico)
Efficient route planning for an unmanned air vehicle deployed on a moving carrier	Savuran, H., Karakaya, M.	2016	Soft Computing 20(7), pp. 2905-2920
Optimum array configuration to improve null steering time for mobile CRPA systems	Byun, G., Hyun, J.-C., Seo, S.M., Choo, H.	2016	Journal of Electromagnetic Engineering and Science 16(2), pp. 74-79
An Efficient Resource Allocation Approach Based on a Genetic Algorithm for Composite Services in IoT Environments	Kim, M., Ko, I.-Y.	2015	Proceedings - 2015 IEEE International Conference on Web Services, ICWS 2015 7195613, pp. 543-550

Fonte: Adaptado do Scopus (2019).

2.3.7 DISCUSSÃO DA BIBLIOMETRIA

Nesta sessão da bibliometria é realizada uma breve análise de cinco artigos que constam dentro da lista compilada, após a aplicação das etapas proposta por Costa (2010).

Wu, Li, Deng, Zomaya (2018) no artigo *Service Selection for Composition in Mobile Edge Computing Systems,* apresenta uma proposta de nova heurística denominada GAMEC, que busca combinar Algoritmos Genéticos e a metaheurísticas do Recozimento Simulado para a seleção de serviços de computação de borda. Na abordagem proposta pelos autores eles transformam um problema de seleção em um problema genético a ser resolvido com o AG e introduzem a variável temperatura proveniente do algoritmo de Recozimento Simulado.

Yan *et al.* (2018) em *Registration of TLS and MLS Point Cloud Combining Genetic Algorithm with ICP*, propõem a utilização dos Algoritmos Genéticos associados a métodos de registros automáticos de pontos e ICP (*Iterative Closed Point*) aplicada a nuvem de pontos TLS e MLS que fazem uso de coordenadas geodésicas. O objetivo é usar o AG para obter uma solução inicial e usar a eficiência do ICP quando o AG tender a um ótimo local. No processo é realizada a captura da posição aproximada dos pontos que são incorporadas ao GPS, sendo aplicado um modelo de melhoria de precisão denominado NSMS, de acordo com os resultados obtidos a eficácia aumenta em 50% da combinação dos AG com o ICP.

Jana e Anwit (2018) no artigo *A Variable Length Genetic Algorithm approach to Optimize Data Collection using Mobile Sink in Wireless Sensor Networks*, os autores propõem um novo algoritmo de otimização da coleta de informações por um coletor móvel em uma rede de sensores sem fio, onde o coletor percorre todos os nós de sensores desta rede, tendo que levar em consideração alguns pontos de parada. O Objetivo do novo algoritmo VLGA (*Variable Length Genetic Algorithm*) é encontrar o número ideal dos pontos de parada usados no percurso do coletor móvel, a fim de mostrar a eficácia do novo algoritmo é realizada uma série de simulações comparando o algoritmo VLGA com o Algoritmo genético aplicado ao problema do Caixeiro Viajante (TSP). De acordo com os autores o desempenho do VLGA sobre o TSP é melhor quando analisado sob a ótica do comprimento do caminho e o tempo para a coleta de informações.

Kim e Ko (2015) no artigo *An Efficient Resource Allocation Approach Based on a Genetic Algorithm for Composite Services in IoT Environments* propõem a utilização de Algoritmos Genéticos para alocar recursos de maneira mais eficiente em um ambiente de IoT (*Internet of Things*), o objetivo dos autores é a minimização da transmissão de dados entre diferentes gateways para execução de serviços compostos, para este fim eles utilizam a heurística dos Algoritmos Genéticos. Das razões apontadas para a utilização do AG, o artigo aponta que uma solução possível é viável quando está pode ser determinada antes da criação da árvore de conexões que será utilizada, mesmo que aleatoriamente, a segunda razão é que o problema em questão possui outros objetivos como a viabilidade de alocação de recursos, procurar evitar a violação de restrições entre gateways além da própria minimização de transmissão de dados.

Sharma e Madhukumar (2015), em *Genetic algorithm aided proportional fair resource allocation in multicast OFDM systems* após uma pesquisa sobre problemas de sub-canais e alocação de energia em um ambiente do tipo OFDM (*Orthogonal Frequency Division Multiplexing*), técnica de transmissão de dados que faz uso de bandas divididas, os autores propõem a maximização da capacidade do sistema utilizando uma nova implementação do Algoritmo Genético, levando em consideração restrições como largura de banda e taxa de transmissão de dados.

Yucel e Bulut (2018), no artigo *Clustered Crowd GPS for Privacy Valuing Active Localization*, os autores realizam um estudo que propõe uma solução e métricas para a busca de objetos perdidos de maneira ativa por meio da utilização

da tecnologia GPS e Bluetooth. De acordo com Yucel e Bulut (2018), com o surgimento da tecnologia *Bluetooth Low Energy* (BLE) e com o advento das redes do tipo *Beacon*, que são para dispositivos de geolocalização focados em ambientes fechados, possibilitam a criação de soluções para a localização de objetos ou mesmo pessoas perdidas como crianças ou idosos.

Krejcar *et al.* (2011), no artigo *Use of Mobile Phones as Intelligent Sensors for Sound Input Analysis and Sleep State Detection*, foi desenvolvido uma solução mobile para windows que realiza o monitoramento do sono de pessoas por períodos definidos de acordo com o alarme do dispositivo celular. O aplicativo desenvolvido foi o *wakeNsmile* que além de monitorar também registra os períodos de sono e fazendo uso do microfone embutido para sugerir a melhor hora para o pessoa acordar e assim conseguir um sono de qualidade.

2.3.8 RESULTADOS DA BIBLIOMETRIA

O objetivo desta etapa da pesquisa foi estabelecer um referencial bibliográfico inicial, relacionando um conjunto de artigos, valendo-se para tal de uma pesquisa do tipo Bibliométrica, como a proposta por Costa (2010), que teve como resultados:

- Três trabalhos mais antigos da base, atendendo o período da pesquisa, publicados nos últimos 10 anos;
- Treze trabalhos mais recentes presentes na base;
- Quinze últimos trabalhos mais relevantes presentes no resultado da pesquisa, usando como critério o maior número de citações.

Nessa pesquisa bibliométrica foi possível identificar, ainda, os autores com maior número de trabalhos, a distribuição cronológica dos trabalhos no período dos 10 últimos anos, além de identificar os países com o maior número de publicações sobre os temas, sendo que o Brasil ficou como o 7º país em número de publicações.

3 MATERIAIS E MÉTODOS

Nesta seção do trabalho será descrita a metodologia, os materiais e os métodos utilizados para o desenvolvimento das soluções, juntamente com os

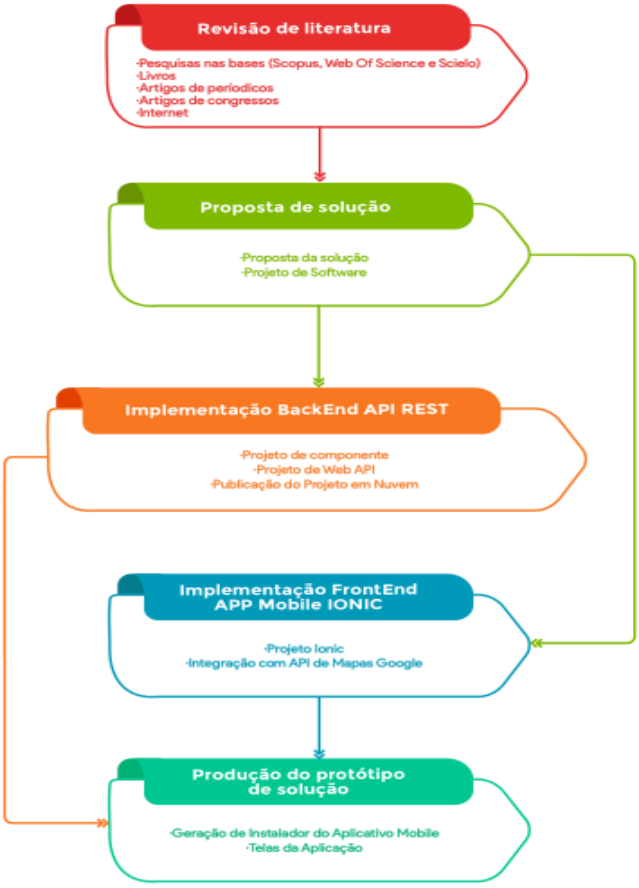
respectivos processos que permitiram o desenvolvimento deste trabalho no seu viés teórico e prático.

Para o desenvolvimento da pesquisa, iniciou-se por uma consulta sistêmica das bases de conhecimento como: *Scopus* e a *Scielo.* Além de estudos e buscas em livros, artigos de periódicos, artigos de congressos, sites de internet em conjunto com monografias e dissertações no intuito de dar maior subsídio informacional para a definição da base teórica e fundamentação deste trabalho.

Recorreu-se, também, às pesquisas bibliográficas de cunho técnico para suprir necessidades de conhecimento que foram de suma importância para a implementação da solução proposta, corroborando com o argumento de Gil (2002) quando afirma que a pesquisa bibliográfica permite ao pesquisador a ampliação do saber em determinadas áreas do conhecimento.

A estruturação e o planejamento desta pesquisa tiveram como intuito atender aos objetivos geral e específicos, apresentados na sessão 1 deste trabalho, enquanto etapas que estão detalhas na Figura 17:

Figura 17 - Etapas detalhadas da pesquisa.



Fonte: O próprio autor.

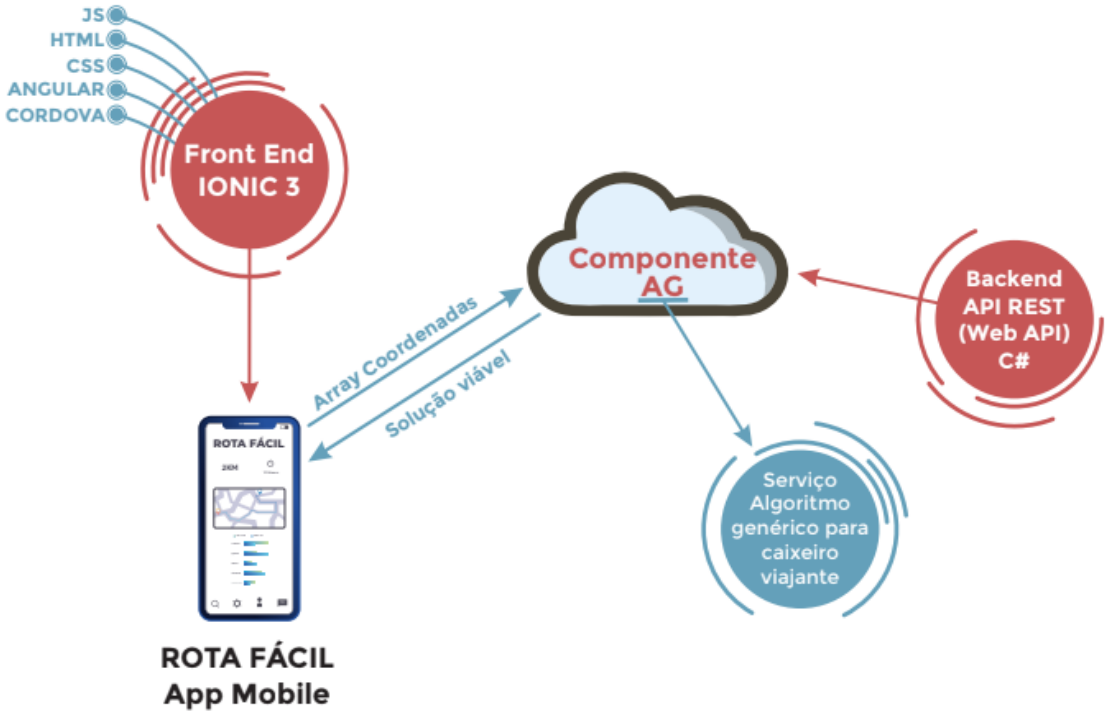
Nesta sessão do trabalho de pesquisa será delineada a proposta da solução e algumas das ferramentas de softwares, métodos e metodologias utilizadas para elaboração das soluções propostas.

3.1 PROPOSTA DE SOLUÇÃO

O projeto visa criar um aplicativo Mobile denominado *ROTA FÁCIL App Mobile*, que será desenvolvido utilizando um pool de tecnologias: *IONIC Framework, Web API C#* dentre outras, já detalhadas neste trabalho (Sessão 2 - Referencial Teórico).

Para o desenvolvimento da solução mobile, que faz uso de um serviço online de processamento de dados que utiliza a solução de algoritmos genéticos com enfoque no problema do caixeiro viajante, exposto na sessão 2.1.1, foi projetado a solução (Figura 18).

Figura 18 - Proposta da solução ROTA FÁCIL App Mobile.



Fonte: O próprio autor.

3.2 FERRAMENTAS

Nesta sessão serão apresentadas as ferramentas tecnológicas utilizadas na implementação das soluções mobile e do serviço Web.

3.2.1 IONIC

O desenvolvimento de aplicações mobile tem sido destaque no mundo tecnológico, deixando de ser uma tendência promissora e passando a ser uma realidade predominante. Contudo, desenvolver um aplicativo que atenda à diversidade de plataformas existente (Android, Windows, IOS) pode se tornar bastante oneroso, haja vista as peculiaridades e o tempo de programação dispensada a cada uma delas.

De acordo com Nunes (2013), existe uma grande necessidade por parte das empresas de tecnologia em criar aplicativos ou *web pages* que funcionem adequadamente nas diversas plataformas de mercado, como: Android, IOS, Windows Phone e Blackberry.

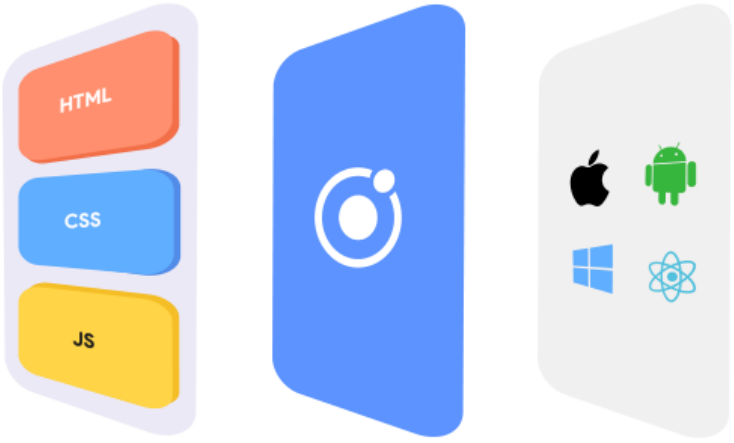
O objetivo deste trabalho de pesquisa é desenvolver uma solução que englobe a criação de um aplicativo mobile, para tal foram realizadas pesquisas bibliográficas em livros e artigos técnicos científicos na busca por soluções para implementações de aplicativos que funcionem nas principais plataformas mobile de mercado. Assim, o Framework escolhido para o desenvolvimento deste trabalho de pesquisa foi o IONIC na sua versão 3. Blini (2016) elenca alguns dos motivos que tornam pertinente a escolha do IONIC Framework como ferramenta para a criação de aplicações híbrida:

- Baixa curva de aprendizado;
- Utilização do framework Angular;
- Possibilidade de visualizar o aplicativo diretamente no browser durante o desenvolvimento, sem a necessidade de emuladores adicionais;
- Documentação vasta e abrangente;
- Criação de aplicações leves;

- Comunidade de desenvolvedores ativa e em contínua expansão;
- Trabalha na filosofia *Write once, run anywhere* (escreva uma vez, rode em qualquer lugar). Assim, o código desenvolvido uma única vez será compatível com a plataforma Android, IOS, Windows Phone e a maioria dos navegadores disponíveis de mercado;
- Possibilidade de suporte para equipes e empresas que contribuem no desenvolvimento e manutenção do framework;
- Tecnologia gratuita.

O IONIC é uma ferramenta que apresenta uma série de recursos que facilitam e aumentam a produtividade no desenvolvimento de aplicações mobile e o Framework é uma mescla de frameworks como o PhoneGap e o AngularJS. O IONIC é um Open Source SDK que trabalha sob o conceito de *native-feelings mobile apps*, preconizando o desenvolvimento de aplicativos móveis utilizando tecnologias web expostos na Figura 19, como: HTML, CSS, Javascript. (GOIS, 2017), AngularJS (VIEBRANTZ; CAMPOS, 2015).

Figura 19 - Tecnologias IONIC.



Fonte: Adaptado de IONICframework.com.

O IONIC 3 utiliza conceitos do Angular 4, no qual a linguagem principal é uma extensão do Javascript que é o TypeScript, cujo objetivo é o desenvolvimento em larga escala usando Javascript. De acordo com Bierman, Abadi, Torgersen (2014), a linguagem TypeScript possui algumas vantagens em relação ao JavaScript, como por exemplo, conceitos comi módulos, classes e interfaces, presentes em

linguagens de programação orientada a objetos de mercado, na Figura 20 há um trecho Typescript utilizado na solução mobile.

Figura 20 - Trecho TypeScript do projeto.

```
//Função que carrega o mapa para exibição;
loadMap(lat,lng)
{

  let latLng = new google.maps.LatLng(lat,lng);
  let mapOption= {
    center: latLng,
    zoom:14,
    mapTypeId:google.maps.MapTypeId.ROADMAP,
    disableDefaultUI:true
  }

  let element = document.getElementById('map');
  this.map = new google.maps.Map(element,mapOption);
  this.ag.map = this.map;
  this.directionsDisplay.setMap(this.map);

  // Adicionando evendo no mapa
  this.map.addListener('click', (event) => {
     var lat = event.latLng.lat().toFixed(6);
     var lng = event.latLng.lng().toFixed(6);
     this.AddArray(lat,lng);
     this.createMarker(lat,lng);
    });

  }
```

Fonte: O próprio autor.

No tocante à interface das aplicações, o IONIC utiliza HTML5 e CSS, o framework já disponibiliza diversos componentes customizáveis em HTML que permitem, ao desenvolvedor, como é apresentado na Figura 21, maximizar o trabalho para o projeto de design e são utilizadas folhas de estilos CSS3 possibilitando uma customização ainda maior do layout de telas e permitindo a formatação de fontes, bordas, cores de fundo, dentre outros recursos. (LOTAR, 2010).

Figura 21 - Trecho HTML e tags IONIC do projeto.

```
<ion-header>
  <ion-navbar color="primary">
    <ion-title>
      Rota Fácil - MPOIC
    </ion-title>
    <ion-buttons end>
    </ion-buttons>
  </ion-navbar>
</ion-header>
<ion-content>
    <div id="map"></div>
    <ion-buttons right >
      <button ion-button (click)="executaAG()"><ion-icon name="add"></ion-icon>AG-TSP</button>
    </ion-buttons>
</ion-content>
```

Fonte: O próprio autor.

De acordo com Gois (2017), no IONIC o foco fica no desenvolvimento da interface do aplicativo, front view que permite o acesso aos componentes na view da aplicação, com o uso de recursos do Angular. E para o acesso aos recursos de hardware do dispositivo, utiliza-se o framework Cordova, por meio da instalação de plugins específicos.

3.2.2 REST

O Rest, *REpresentational State Transfer* é uma metodologia para criação de Webservices proposto por Roy Thomas Fielding (FIELDING,2000), em sua tese de doutorado. O protocolo REST é guiado pelas boas práticas do HTTP, uma vez que Fielding é um dos coautores desse protocolo (SAUDATE, 2014). Dentre as boas práticas citadas por Saudate (2014), têm-se:

1. Uso adequado dos métodos HTTP;
2. Uso adequado das URL's;
3. Padronização de códigos para representação de sucessos ou falhas;
4. Interligação entre vários recursos diferentes.

De acordo com Saudate (2014), o recurso é considerado o "marco zero" e tudo pode ser definido em termos de recursos, sendo esse o conjunto de dados a serem trafegados pelo protocolo e representados pelas URI's (*Universal Resource Identifier*) - o identificador do recurso.

Para Gonçalves e Silva (2012), o REST é formado por um conjunto de recursos no qual cada um possui um identificador único, podendo possuir várias

representações que podem se conectar através de *hyperlinks* e há métodos que permitem manipulação de suas representações. Como proposto por Fielding (2000), os princípios que norteiam a metodologia REST podem ser balizados em:

- Arquitetura Cliente-Servidor: Uma das principais implementações arquiteturais em sistemas WEB, onde um servidor congrega um conjunto de serviços que recebem e processam requisições de clientes;
- *Stateless* (sem estado): Na comunicação cliente servidor, não deverá ser guardado nenhum tipo de estado no servidor, sendo assim, estados de sessão devem ser guardadas no cliente;
- Uso de Cache: Como medida de otimizar o desempenho da aplicação, o uso de cache permite que uma resposta possa ser setada e utilizada em futuras requisições;
- Interface: O REST padroniza sua interface para a comunicação entre seus componentes, que são quatro: (i) Identificação de recursos; (ii) Manipulação dos recursos, por meio de sua representação; (iii) Mensagens auto-descritivo; (iv) Hipermídia.

Sobre o HTTP como protocolo padrão de utilização do REST, Moro, Dorneles, Rebonatto (2011, p. 41) afirmam que:

> Atualmente, o HTTP é o protocolo chave, sendo o mais indicado para trabalhar com Web services REST. O mesmo provê uma interface uniforme com quatro métodos básicos para as quatro operações mais comuns. GET para recuperar uma representação de um recurso, PUT para criar um novo recurso ou modificar um existente, DELETE para deletar um recurso e POST comumente utilizado para criação de um novo recurso.

- Multicamada: Com o intuito de possibilitar a escalabilidade de soluções baseadas em REST, a tecnologia traz a possibilidade da divisão em camadas, separando-as em funcionalidades.
- *Code on Demand*: Concentra-se na possibilidade do cliente requisitar e baixar um código, diretamente no lado cliente.

A idéia central no funcionamento do REST é a utilização do URI para a identificação do recurso requerido de maneira única. Uma vez que, com base no método invocado (POST, PUT, GET, DELETE) e nos dados passados na requisição,

a URI terá um comportamento diferenciado, tomando como base um exemplo hipotético de um Web Service para manutenção de produtos de uma loja virtual de eletrônicos com as funções básicas de operações de CRUD (*create, retrieve, update* e *delete*). Para realizar a invocação do serviço REST, identificado por uma URI, é necessário somente o identificador do serviço requerido para o caso de uma consulta ou *delete*, pois o método HTTP utilizado já indica a operação e o recurso a serem utilizados. Para os casos de um cadastro com os métodos *put* ou *post,* no corpo da requisição, também são enviados os dados.

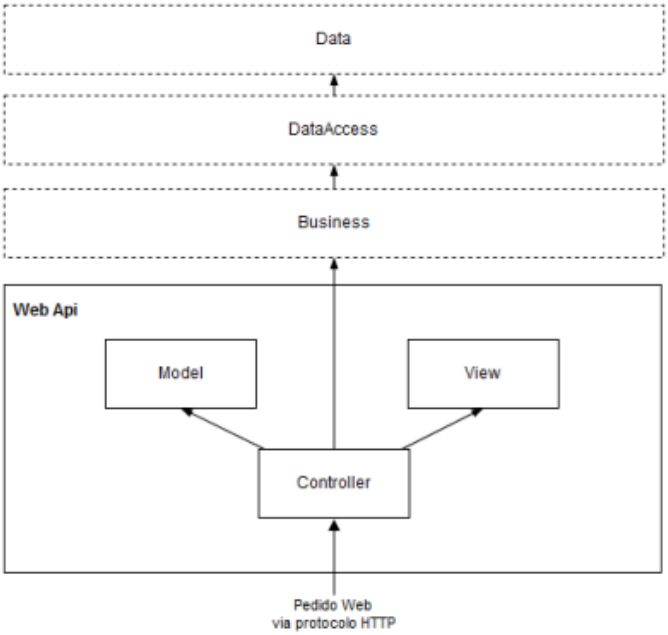
3.2.3 WEB API

No desenvolvimento deste trabalho de pesquisa a tecnologia REST foi utilizada para o desenvolvimento do *Backend* da aplicação mobile, no formato de um serviço WEB. A tecnologia Web API é um recurso disponibilizado pela Microsoft dentro do seu pacote de soluções para o desenvolvimento WEB, o ASP. NET, e por meio dessa API é possível a criação de serviços do tipo Restful (MICROSOFT, 2019).

A Arquitetura da Web API é estruturada no padrão MVC (*Model-View-Controller*), nesse modelo a aplicação é dividida em 3 grupos (MARTINS, 2016), na Figura 22 está um diagrama dessa divisão:

- O *Model* é a representação da lógica de negócios da aplicação e a representação de dados específicos, ele também pode manter dados da aplicação;
- Na *View* é onde se encontram os componentes gráficos de interação com os usuários (páginas, formulários etc.);
- Já o *Controller* é a interface de comunicação do *Model* com a *View*, ficando encarregado de coletar os dados da View e encaminhá-los para o modelo e vice-versa.

Figura 22 - Arquitetura MVC de uma Web API.



Fonte: MARTINS (2016).

3.2.4 LINGUAGEM DE PROGRAMAÇÃO C#

De acordo com Lotar (2010), C# é uma linguagem de programação simples, porém poderosa e ideal para o desenvolvimento de aplicações dos mais variados tipos. Sua linguagem implementa conceitos de orientação aos objetos, o que permite a criação de classes que herdam atributos (características) e métodos (comportamentos) de outras classes, tanto escrita em C# ou mesmo Visual Basic .NET. Isso possibilita, aos desenvolvedores, a modularização e o reaproveitamento dos códigos produzidos. A linguagem deriva do C e C++, por esse motivo, sua sintaxe é considerada relativamente simples, como exposto abaixo na Figura 23, o que leva à uma curva de aprendizado reduzida (LOTAR, 2010).

Figura 23 - Sintaxe da linguagem C#.

```
using System;

class Hello
{
    static void Main()
    {
        Console.WriteLine("Hello, World");
    }
}
```

Fonte: MICROSOFT (2019).

Para a criação de sistemas robustos - que se utilizam do C# - há recursos como: *Coleta de lixo* - que limpa a memória de objetos já não mais referenciados no programa; Tratamento de exceções - a linguagem oferece uma forma estruturada para a detecção e recuperação de erros; Fortemente tipada - essa característica é interessante por impossibilitar a utilização de variáveis não inicializadas ou indexação de vetores e matrizes além de seus limites (MICROSOFT, 2019).

3.2.5. NET FRAMEWORK

Em 15 de janeiro de 2002, a Microsoft introduz no mercado a plataforma .NET que é uma ferramenta extremamente poderosa e que possibilitou a comunidade de desenvolvedores a criação de aplicativos dos mais variados fins. Nesse ambiente é possível a criação rápida de aplicativos de maneira profissional e abstraindo o programador de uma grande quantidade de detalhes técnicos (ARAÚJO, 2006). A .NET é formada por dois componentes de suma importância: Um ambiente de execução (*Common Language Runtime* - CLR) que funciona como a máquina virtual, que irá executar os códigos .NET; Uma biblioteca de classes (*Class Library*) com diversas implementações (LOTAR, 2010).

De acordo com a Microsoft (2019), os principais serviços oferecidos pela .NET Framework são:

- Gerenciamento de memória: Em linguagens de programações mais tradicionais como *Delphi*, C e C++, a liberação de memória era uma

responsabilidade do desenvolvedor, em .NET essa é uma responsabilidade da CLR;

- Especificação de *Common Type System* (CTS): Novamente, em linguagens de programação tradicionais, a definição de tipos de váriaveis era realizada pelo compilado, o que dificultava a interoperabilidade entre diferentes linguagens. Em .NET existe seu próprio sistema de tipos, que permite que qualquer linguagem que atenda às especificações da CTS possa se comunicar;
- Biblioteca de classes (*Class Library*): Disponibilizada pela plataforma, ela agrega um conjunto de classes, interfaces e tipos que permite acesso ao sistema, sendo a base no qual são construídas as aplicações em .NET (LOTAR, 2010);
- A .NET: Possui bibliotecas específicas para diversos tipos de projetos de tecnologias, como o ASP .NET - para criação de aplicações WEB, *ADO, ADO.NET* e *Entity Framework* e para acesso aos dados - e o *Windows Communication Foundation* (WCF) - para aplicações orientadas aos serviços - e o *Windows Presentation Foundation* (WPF) - para aplicações no Windows;
- Interoperabilidade entre linguagens da plataforma: Os compiladores da linguagem .NET geram um CIL (*Common Intermediate Language*) que é compilado em tempo de execução para a CLR (*Common Language Runtime*);
- Compatibilidade de versões: De acordo com a Microsoft (2020), tirando raras exceções, as aplicações desenvolvidas em diferentes versões mais antigas do framework mantêm compatibilidade;
- Possibilidade da execução: Paralela de várias versões da CLR no mesmo computador;
- Quando da utilização do .NET Standard: Os desenvolvedores podem escrever códigos e bibliotecas para uma gama de plataformas .NET.

3.2.6 GEOPROCESSAMENTO

De acordo com Silva (2009), o geoprocessamento como conceito cresceu e vem evoluindo, sobretudo com a adoção cada vez maior de métodos e tecnologias, tornando-se de grande importância na classificação de ambientes com base em suas características. Possibilitando melhor gestão e planejamento de ambientes que, embora sejam campos técnico-científicos, são enormemente beneficiados pelo uso do geoprocessamento. Nesse cenário, em complemento a Silva (2009), obtém-se: Os sistemas de informações geográficas que são ferramentas cada vez mais utilizadas no processo de pesquisa e planejamento econômico, territorial e ambiental (CROSTA, 1997).

Para um melhor entendimento das tecnologias abarcadas sob o termo geoprocessamento há um conjunto de geotecnologias, como: Sensoriamento Remoto, Cartografia, Sistemas de Posicionamento Global (GPS), dentre uma série de outras tecnologias (SILVA, 2009). Corroborando com essa classificação, Silva (2003) define o geoprocessamento como o processamento de dados georeferenciados que envolve as técnicas já citadas e com acréscimo dos sistemas de informações geográficas (SIG).

Neste projeto de pesquisa, no que tange ao geoprocessamento e à geolocalização, foi utilizado o *Google Maps* como um dos diversos serviços disponibilizados pela empresa Google.com. A empresa disponibiliza acesso aos mapas de estradas, satélite, relevo e outros recursos para que desenvolvedores do mundo inteiro passem a implementar soluções em geoprocessamento, tanto para dispositivos móveis quanto para soluções WEB ou desktops. Para Costa e Matos (2013, p. 33) o “*Google Maps* é possível com a utilização de um mapa genérico por adicionar características locativas, permitindo a utilização de recursos como fotos, sons ou vídeos”. Para acesso à ferramenta online, utiliza-se o endereço de internet: *http:maps.google.com.br*. que mostra, ao ser acessado, a imagem conforme a da Figura 24.

Figura 24 - Página inicial do serviço do Google Maps.

Fonte: Google Maps, 2019.

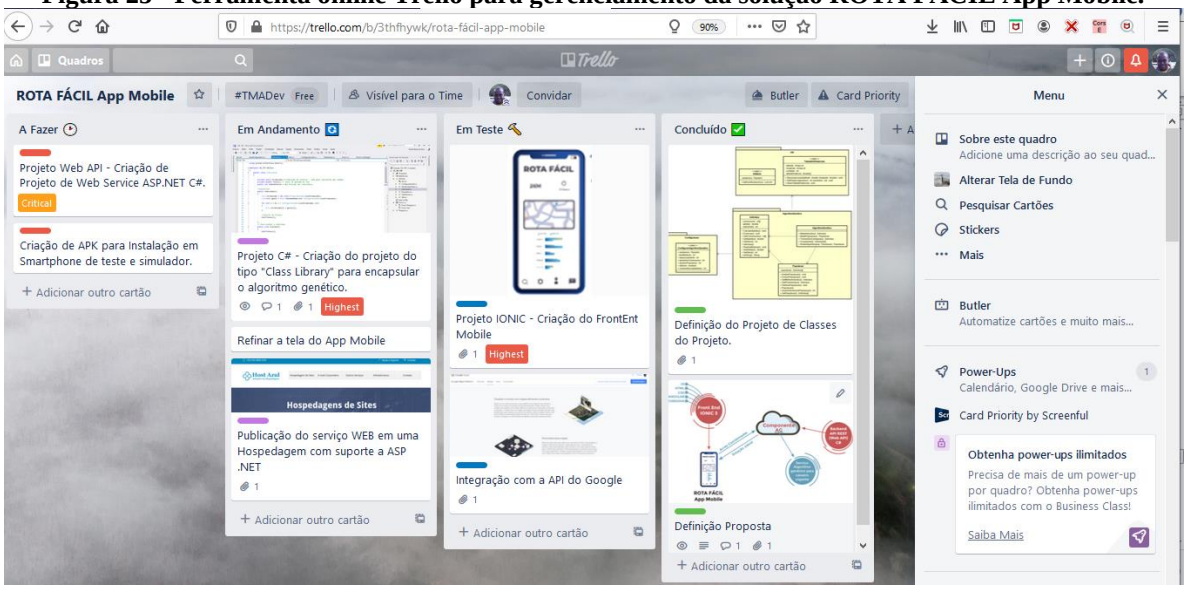
Scussel (2013) descreve o *Google Maps* como uma tecnologia Geoespacial disponibilizada a todos, não somente aos profissionais de tecnologia, ele cita ainda que em um passado recente essa tecnologia estava disponível exclusivamente para os departamentos de informações geográficas e, na atualidade, encontra-se disponível ao alcance de grande parte da população, por meio da internet, sobretudo com a popularização de equipamentos como smartphones e computadores pessoais.

3.3 PROJETO DE SOFTWARE

Como explica Larman (2007), o projeto tem como em sua ênfase a criação de uma solução conceitual, seja ela de software ou hardware, a fim de satisfazer os requisitos sem foco na sua implementação como, por exemplo, os esquemas de bancos de dados, os diagramas como a visão geral da solução apresentada na Figura 15. Na confecção deste trabalho, optou-se para pela utilização de um projeto baseado na utilização de conceitos oriundos da engenharia de software [4] como a gestão de projetos ágeis com a utilização da ferramenta online *Trello,* Figura 25, para gerenciamento de projetos, na orientação aos objetos e para a codificação da solução.

[4] Ciência que estuda metodologias e padrões para o desenvolvimento de software. (LOBO, 2008)

Figura 25 - Ferramenta online Trello para gerenciamento da solução ROTA FÁCIL App Mobile.



Fonte: O próprio autor.

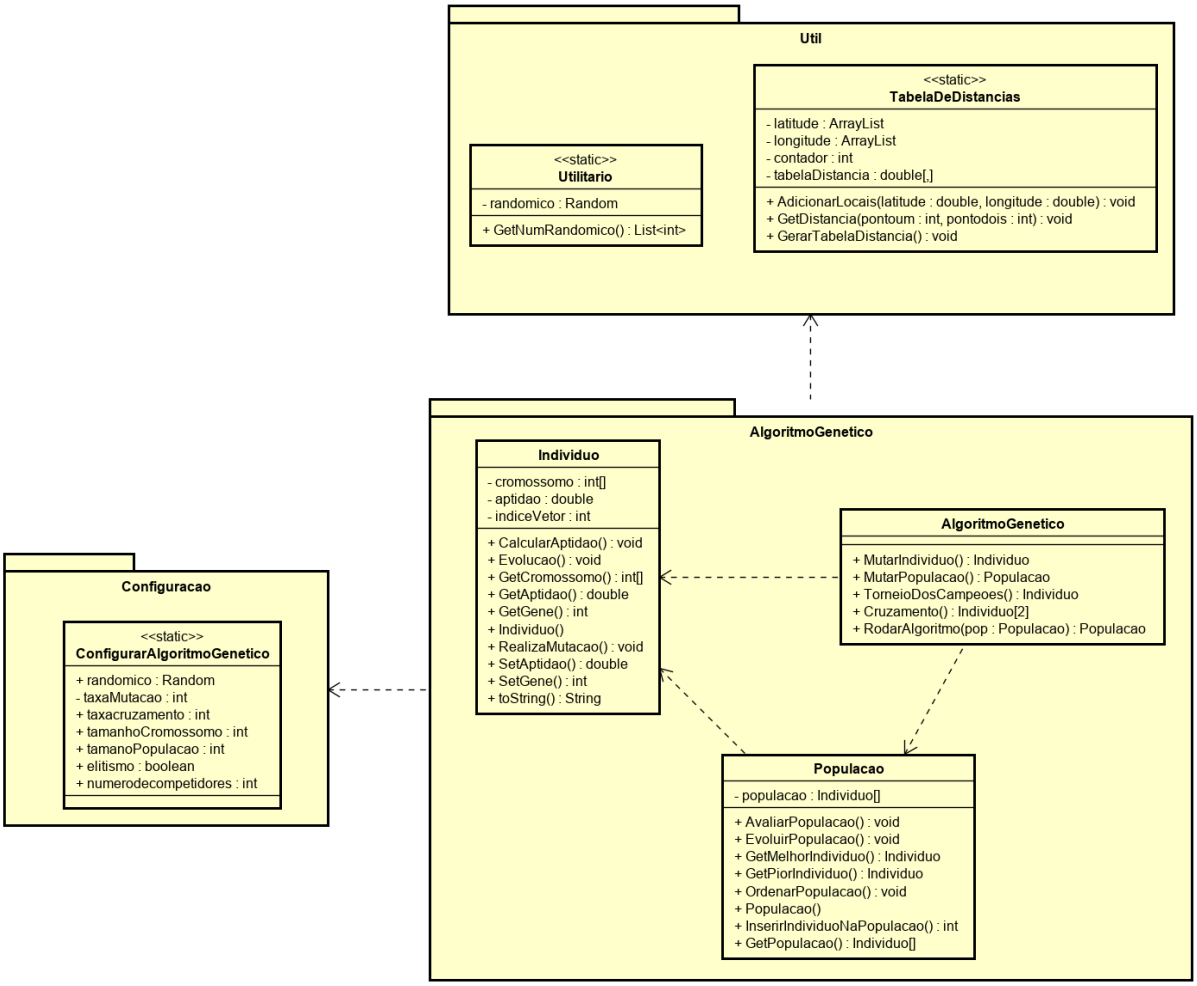
O Trello é uma ferramenta online para gerenciamento de projetos que utiliza o paradigma Kanban[5] e foi desenvolvida pela empresa Toyota. O Trello é extremamente versátil por permitir a criação de quadros de acordo com a necessidade de cada projeto (TECMUNDO, 2015).

Usando como metodologia de desenvolvimento dos projetos englobadas na solução do Aplicativo Mobile, optou-se pela orientação aos objetos, que é um padrão de desenvolvimento que se baseia em classes que modelam representações do mundo real (LOBO, 2008). De acordo com Engholm (2013), a orientação aos objetos é um paradigma no qual o foco não está nos procedimentos, mas sim nas informações que os objetos irão manipular e/ou armazenar. Como apresentado na Figura 26, foi criado um diagrama de classes usando a linguagem UML[6], a qual fornece uma série de diagramas de modelagem que podem ser utilizados durante o ciclo de desenvolvimento de um software, é útil para documentação e especificação do sistema e/ou solução.

[5] O método Kanban nasceu no Japão com o objetivo de gerenciar o abastecimento e fluxo dos materiais em estoque para as linhas de produção, sem desperdícios ou atrasos, de uma forma bastante visual e fácil de perceber: usando cartões coloridos. Fonte: https://www.heflo.com/pt-br/agil/metodo-kanban/.

[6] UML Linguagem de Modelagem Unificada (*Unified Modeling Language*) é uma linguagem visual ou notação utilizada para ajudar na especificação e na documentação de sistemas de informação orientados aos objetos (GÓES, 2014).

Figura 26 - Diagrama de Classe UML do projeto ComponenteAG que abstrai a solução do algoritmo genético da solução ROTA FÁCIL App Mobile.



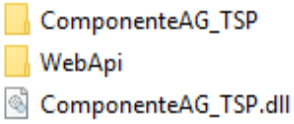
Fonte: O próprio autor.

As classes do Projeto do algoritmo genético foram distribuídas em pacotes para melhor organização lógica, abaixo a descrição das classes:

1. Indivíduo - Classe que representa a abstração de um indivíduo na população;
2. População - Classe que representa um conjunto de indivíduos;
3. Algoritmo Genético - Classe que implementa o algoritmo genético, que realiza os cruzamentos, mutações, evolui e ordena a população;
4. Configura Algoritmo Genético - Classe estática que contém as configurações padrão para execução do algoritmo;
5. Tabela De Distância - Classe estática que irá tratar as rotas como uma tabela de pontos com latitude e longitude;
6. Utilitário - Classe estática, importante para projeto, para tratar a aleatoriedade do processo ao ficar encarregada e gerar números randômicos (aleatórios).

Todas essas classes que compõem a solução do algoritmo genético serão compiladas em um componente de software, mais precisamente em uma DLL, como apresentada na Figura 27, para assim possibilitar o reaproveitamento do código do algoritmo genético na solução de rotas em outros projetos e nas soluções baseadas na plataforma. NET.

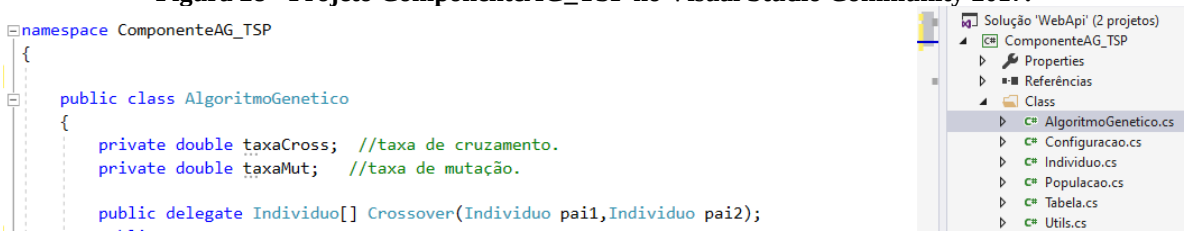
Figura 27 - ComponenteAG_TSP.dll.



Fonte: O próprio autor.

Para a codificação do algoritmo genético, que é o cerne do projeto, foi criado um novo projeto do tipo biblioteca de classes ferramenta Visual Studio *Community* 2017, usando a linguagem de programação orientada aos objetos C#, na Figura 28 é apresentado um recorte da tela de codificação do projeto.

Figura 28 - Projeto ComponenteAG_TSP no Visual Studio Community 2017.



Fonte: o próprio autor.

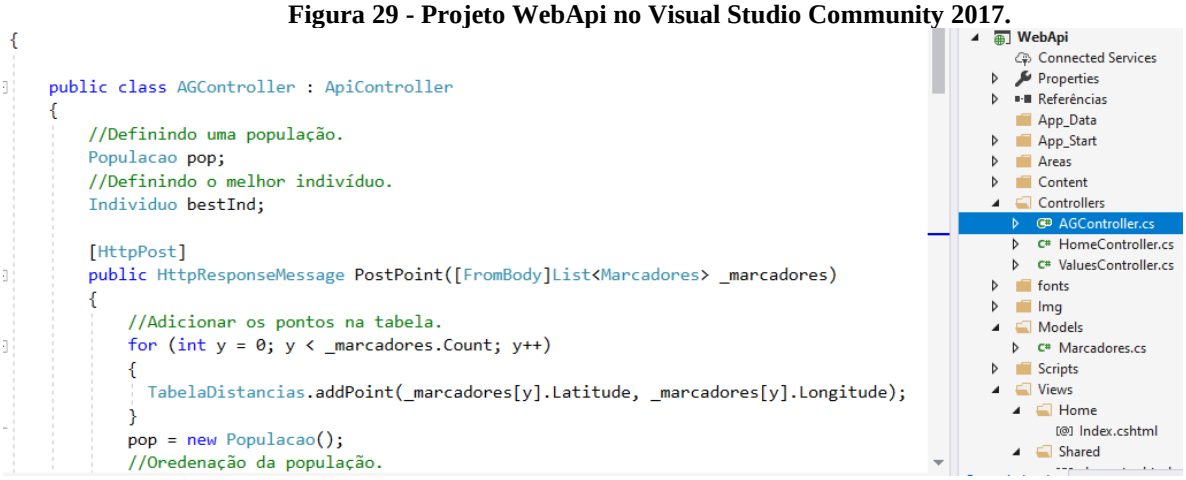
3.3.1 IMPLEMENTAÇÃO BACKEND API REST

O Backend do projeto é composto pela solução do ComponenteAG explicitado na sessão 3.3, e pela solução Web, um webservice do tipo REST, que trabalha com o protocolo Web HTTP e fica hospedado em um servidor em nuvem. Sobre o protocolo de requisição WEB HTTP se pode criar expor e consumir dados fornecidos como serviços na rede. Por ser de fácil uso e por ser suportado por uma grande parte das bibliotecas de programação, esse protocolo permite que se trabalhar com um grande número de plataformas como a web, mobile e desktop.

Para utilizar essas funcionalidades e todos os recursos disponíveis, o.NET Framework desenvolveu as WEB API que nada mais são do que framework usados para a construção de serviços baseados no HTTP. Sendo assim, consegue-se colocar um serviço criado em C#, escutando requisições HTTP e retornando uma informação útil aos usuários.

Como parte da solução, foi criado um projeto denominado WEBAPI desenvolvido com ASP. NET e com a linguagem C#, conforme a Figura 29, que possui como parte integrante o componenteAG_TSP.DLL, apresentado na sessão 3.2, assim tem-se o algoritmo genético rodando como um serviço WEB.

Figura 29 - Projeto WebApi no Visual Studio Community 2017.



Fonte: O próprio autor.

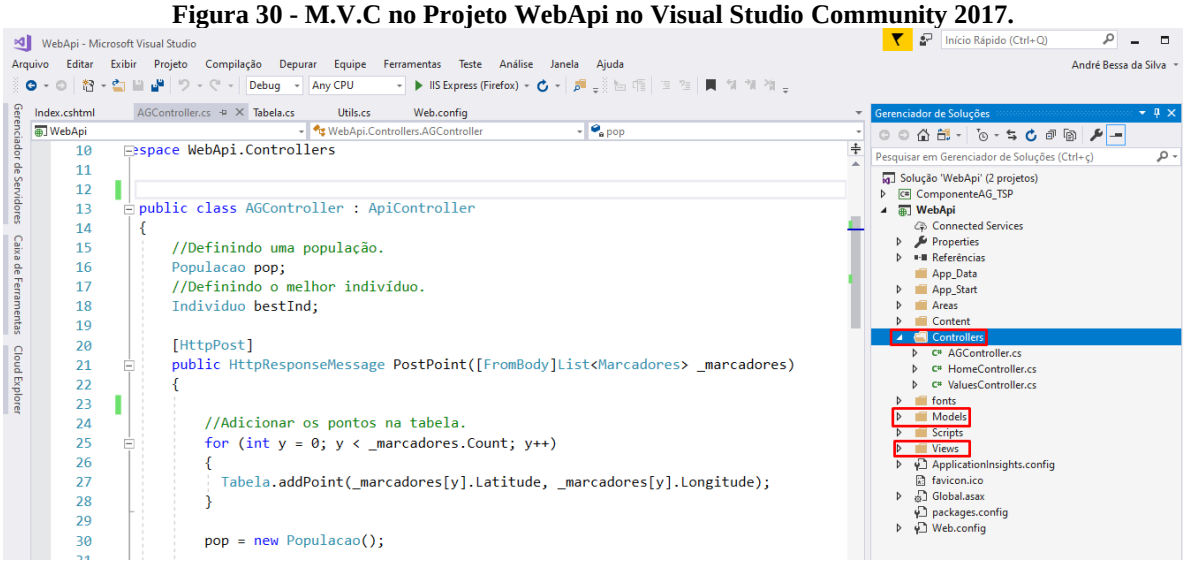
Com a solução da WEBAPI desenvolvida, foi necessária a hospedagem da mesma em um sevidor WEB disponibilizando o serviço para ser acessível tanto para ambientes de produção quanto para os ambientes de testes. Como requisito para a escolha do servidor, dentre fatores, foram analisados disponibilidade e o preço de hospedagem que estão no suporte ao .NET Framework e ao ASP .NET versão 4.8, isso para a correta execução da WEB API. O Serviço web se encontra disponibilizado no link: *http://www.qcode.com.br/.*

Um projeto do tipo WEBAPI é criando tomando como sua base estrutural o paradigma M.V.C (Model, View e Controller), o que pode ser analisado na Figura 30. De acordo com Araújo (2017), o M.V.C é um padrão arquitetural no qual suas siglas ajudam em sua definição:

- *Model* - É o componente deste padrão arquitetural que trata das regras de negócios;

- *View* - É o componente da arquitetura que busca as informação do modelo e as apresenta aos usuários;
- *Controller* - É o responsável pela interação entre o model e a view.

Figura 30 - M.V.C no Projeto WebApi no Visual Studio Community 2017.



Fonte: O próprio autor.

3.3.2 IMPLEMENTAÇÃO FRONTEND APP MOBILE IONIC

O projeto visa criar um aplicativo Mobile denominado ROTA FÁCIL – MPOIC, que será desenvolvido utilizando a tecnologia para desenvolvimento de soluções mobile IONIC.

O IONIC *Framework* é uma solução para desenvolvimento de soluções móveis híbridas, de maneira ágil e fácil. O framework nasceu em 2013 pelas mãos dos desenvolvedores Max Lynch, Ben Sperry e Adam Bradley da Drifty Co. O IONIC é, na verdade, uma pilha de frameworks na qual há dois componentes centrais que são:

- AngularJS - *Framework* javascript que se encarrega da criação da WEB App que compõe a solução;
- Apache Cordova - *Framework* que se encarrega de integrar a solução com os recursos nativos dos dispositivos como câmera, som, dentre outros.

A instalação do *Framework* IONIC segue os passos descritos abaixo:

1. Instalação do SDK do Java para o desenvolvimento, que pode ser encontrado no site *www.Oracle.com*.;

2. Instalação do Node.JS para a criação do projeto local IONIC, também se faz necessária a instalação do Node JS disponível em *Nodejs.org*. O NodeJS que é um *framework* Javascript que roda no lado servidor de uma solução web.

Uma vez instalado o Node.JS é possível utilizar o gerenciador de pacotes NPM para instalação via linha de comando, o Cordova e o IONIC, de modo global, conforme apresentado na Figura 31.

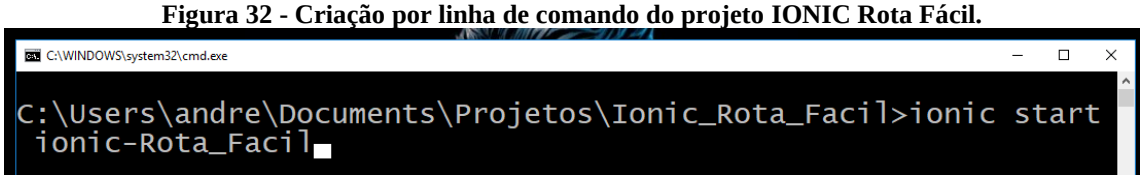
Figura 31 - Comando para instalação do framework IONIC e Cordova de modelo global.

```
npm install -g ionic cordova
```

Fonte: IONICframework.com (2018)

Para o desenvolvimento do protótipo da proposta, foi desenvolvido um projeto do tipo IONIC *Blank* que é uma solução vazia no modelo Web view. Criação do projeto Rota Fácil pela linha de comando em um ambiente Windows 10 (Figura 32).

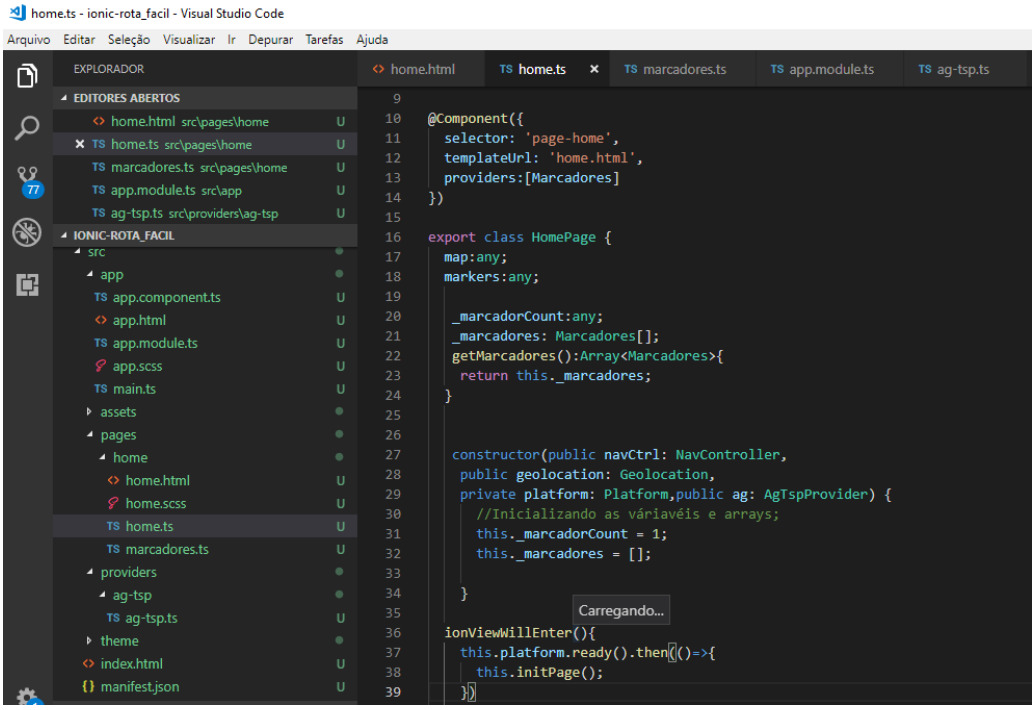
Figura 32 - Criação por linha de comando do projeto IONIC Rota Fácil.



Fonte: O próprio autor.

Como Ferramenta de desenvolvimento e depuração foi utilizada *o Visual Studio Code*, apresentada na Figura 33, que é uma ferramenta Multiplataforma e gratuita da Microsoft.

Figura 33 - Projeto Rota Fácil no Visual Studio Code.



Fonte: O próprio autor.

Nesta parte do projeto é utilizado o Javascript na sua variante *Typescript* que permite, também, trabalhar sob o paradigma da orientação aos objetos. Sendo assim, como parte da solução IONIC foi estruturada uma classe denominada Marcadores, apresentados na Figura 33, conterá a longitude e latitude a serem capturadas pelo dispositivo do usuário e serão passadas via requisição POST para a solução Web API.

Figura 34 - Classe Marcadores.

```
export class Marcadores{

    public latitude: number;
    public longitude: number;
    public constructor(){}

}
```

Fonte: O próprio autor.

Para este projeto também foi desenvolvida uma classe do tipo provider, (Figura 35), que faz o ponto entre a solução IONIC e algum serviço a ser consumido pela aplicação, como um *WebService*.

Figura 34 - Classe provider AgTspProvider da solução IONIC Rota Fácil.

```
import { Injectable } from '@angular/core';
import {HttpClient,HttpHeaders} from '@angular/common/http';

@Injectable()
export class AgTspProvider {

  private API_URL = 'http://localhost:49475/api/AG/';

  constructor(public http: HttpClient){
  }

public conectaComAG(array:any)
{

      let data = array;
      let headers= new HttpHeaders();

      headers.append('Content-Type','application/x-www-form-urlencoded');
      this.http.post(this.API_URL,JSON.stringify(data),{headers:headers})
      .catch(error =>{

        console.log(error.status);
        console.log(error.error);
        alert(error.status);
        console.log(error.headers);

      })


 }
}
```

Fonte: O próprio autor.

3.3.3 INTEGRAÇÃO COM A API DE MAPAS DO GOOGLE

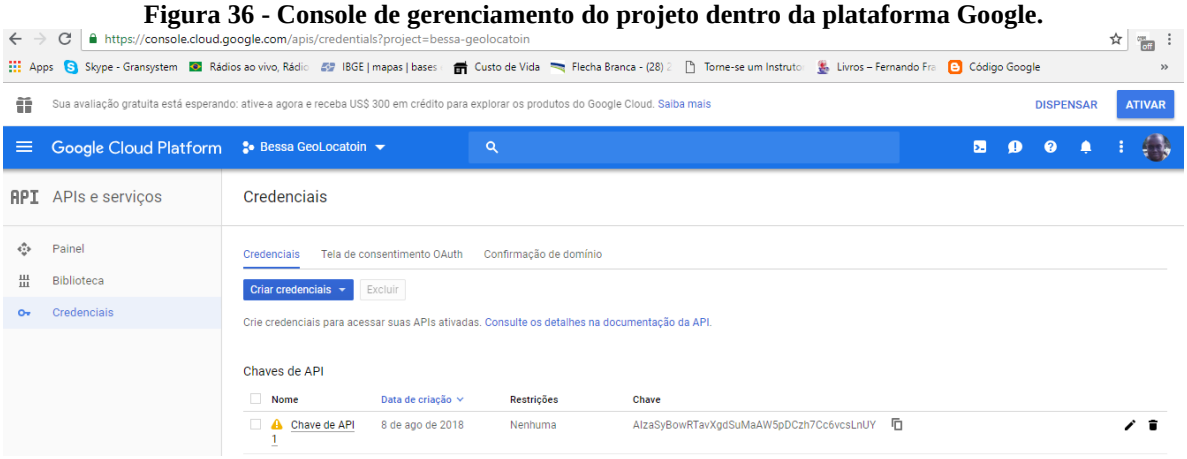
O projeto utiliza a API de mapas disponibilizado pela *Google.com* no que tange ao geoprocessamento, contudo, para utilizarmos as apis do *Google*, fez-se necessário o cadastro da solução dentro da plataforma *Google Cloud,* Figura 36*,* que congrega diversas soluções, inclusive a de mapas a serem utilizados para recursos de geoprocessamento.

Figura 35 - Página Inicial do Google Cloud.

Fonte: O próprio autor.

Após o cadastro na plataforma, o *Google* disponibiliza uma console, Figura 37, para gerenciamento de projetos, onde cada projeto gera uma *API*[7] *KEY* que é um código que será utilizado dentro da aplicação para acessar o serviço de mapas da empresa.

Figura 36 - Console de gerenciamento do projeto dentro da plataforma Google.

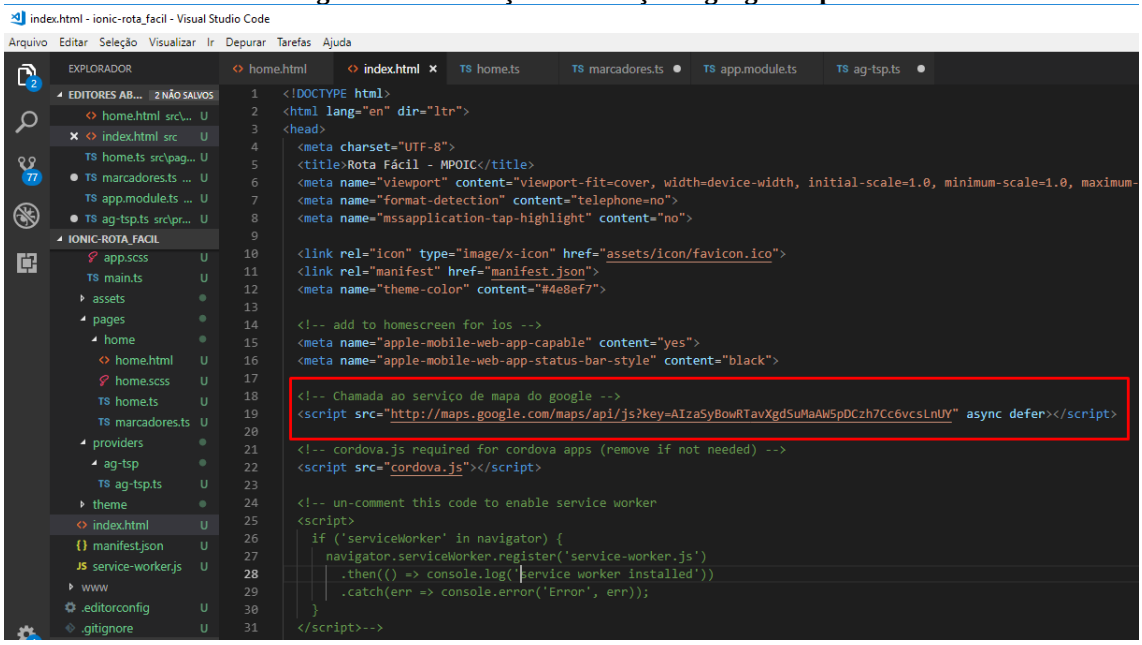


Fonte: O próprio autor.

No projeto IONIC Rota Fácil, a invocação da *API* se dá por meio de uma chamada à uma URL via javascript (Figura 38):

[7] A API é um conjunto de definições e protocolos usado no desenvolvimento e na integração de software de aplicações. API é um acrônimo em inglês que significa interface de programação de aplicações. (RED HAT, 2020).

Figura 37 - Invocação do serviço do google maps.



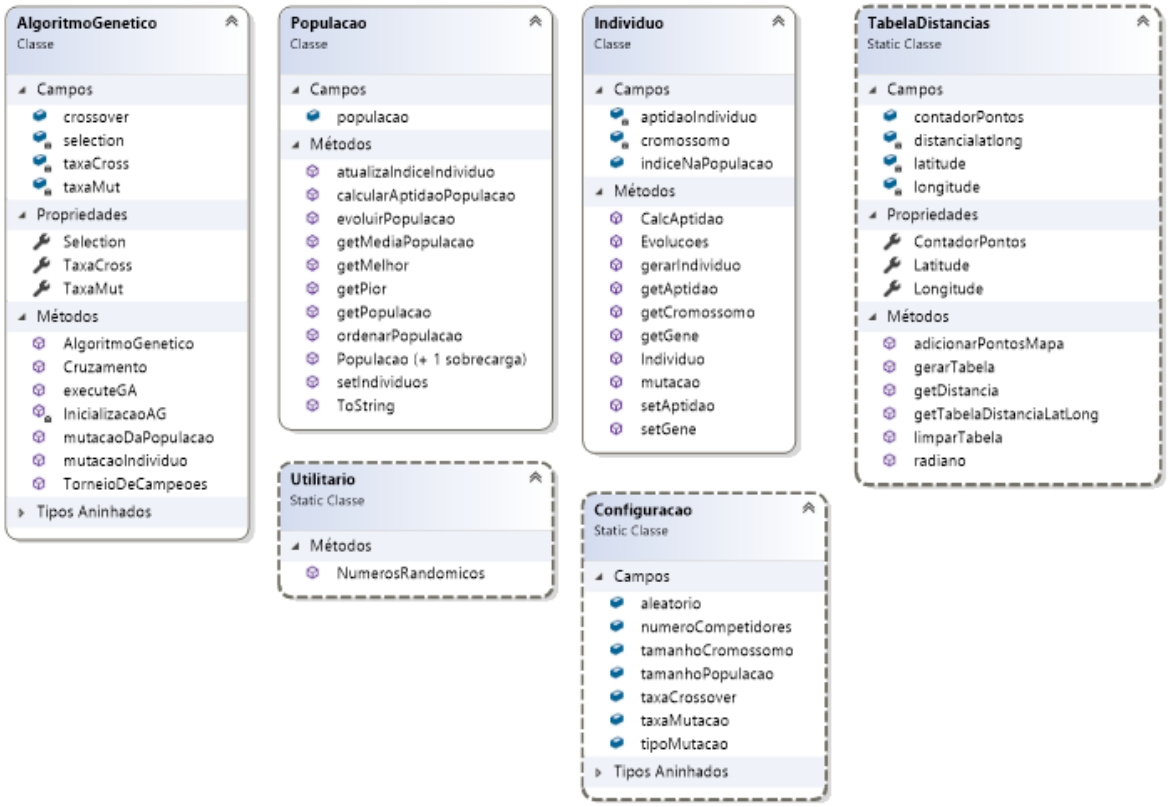
Fonte: O próprio autor.

3.4 PRODUÇÃO DO PROTÓTIPO DA SOLUÇÃO

Para a produção do protótipo funcional da solução do aplicativo Rota Fácil, primeiramente foram desenvolvidos dois projetos que correram em paralelo ao móbile, desenvolvido com IONIC 3 e outro da API que implementam o algoritmo genético, para processar os pontos traçados no mapa. O projeto da API foi dividido em dois subprojetos: ComponenteAG e WebApi.

O projeto Componente AG consiste em um projeto que cria uma *dynamic-link library* ou simplesmente dll com toda a lógica do algoritmo genético, com a intenção de ser um componente reaproveitável em outros projetos. Neste projeto, usando os conceitos de programação orientada aos objetos, são definidas algumas classes (Figura 39), que é a imagem gerada pela ferramenta Visual Studio *Community* 2017.

Figura 38 - Projeto de classes da solução ComponenteAG.



Fonte: O próprio autor.

Algoritmo Genético é uma classe responsável por implementar a solução baseada no problema do caixeiro viajante e, na Figura 40, segue um recorte da classe AlgoritimoGenetico no Visual Studio 2017.

Figura 39 - Recorte inicial da classe Algoritmo Genético.

```
public AlgoritmoGenetico()
{
  InicializacaoAG();
}

private void InicializacaoAG()
{
    this.crossover = Cruzamento;
    this.Selection = TorneioDeCampeoes;
    this.TaxaCross = Configuracao.taxaCrossover;
    this.TaxaMut = Configuracao.taxaMutacao;
}

public Populacao executeGA(Populacao pop)
{
    Populacao novaPopulacao = new Populacao();
    List<Individuo> populacaoTemporaria = new List<Individuo>();
    for (int i = 0; i < Configuracao.tamanhoPopulacao; i++)
    {
      populacaoTemporaria.Add(pop.getPopulacao()[i]);
    }
    for (int i = 0; i < (Configuracao.tamanhoPopulacao/2); i++)
    {
        Individuo pai = Selection(pop);
        Individuo mae = Selection(pop);

        double cruzamento = Configuracao.aleatorio.NextDouble();
        if (cruzamento <= TaxaCross){
            Individuo[] filho =crossover(pai, mae);
            if (Configuracao.tipoMutacao == Configuracao.Mutacao.Novo)
            {
                filho[0] = mutacaoIndividuo(filho[0]);
                filho[1] = mutacaoIndividuo(filho[1]);
            }
```

Fonte: O próprio autor.

A classe ***População*** representará o conjunto de indivíduos, que são as possíveis soluções para o problema do caixeiro viajante, como apresentado na Figura 41.

Figura 40 - Recorte inicial da classe Populacao.

```
public class Populacao
{
    public Individuo[] populacao ;

    public Populacao(Individuo[] populacao)
    {
        this.populacao = populacao;
    }

    public Populacao()
    {
        this.populacao = new Individuo[Configuracao.tamanhoPopulacao];

        for (int i = 0; i < Configuracao.tamanhoPopulacao; i++)
        {
            this.populacao[i] = new Individuo();
            this.populacao[i].indiceNaPopulacao = i;
        }

       this.calcularAptidaoPopulacao();
    }
```

Fonte: O próprio autor.

Indivíduo é uma classe que irá representar uma solução do conjunto de soluções viáveis, conforme exposto na Figura 42:

Figura 41 - Recorte inicial da classe Indivíduo.

```
public class Individuo
{
    public Individuo() => gerarIndividuo();

    public void gerarIndividuo()
    {
        cromossomo = new int[Configuracao.tamanhoCromossomo];
        List<int> genes = Utilitario.NumerosRandomicos(0, Configuracao.tamanhoCromossomo);
        for (int i = 0; i < Configuracao.tamanhoCromossomo; i++)
        {
            this.cromossomo[i] = genes[i];
        }
        CalcAptidao();
    }

    public void Evolucoes() => CalcAptidao();
```

Fonte: O próprio autor.

Tabela Distância, apresentada na Figura 43, é uma classe do tipo estática que representará uma tabela de distância entre a memória e as coordenadas passadas. No Quadro 4 segue a implementação da função geraTabela que implementa a função objetivo, com base na fórmula Harversina apresentada na Figura 10 deste trabalho.

Figura 42 - Recorte parcial da classe Tabela Distância.

```
public static class TabelaDistancias
{
    private static ArrayList latitude = new ArrayList();
    private static ArrayList longitude = new ArrayList();
    private static double[,] distancialatlong;
    public static int contadorPontos = 0;
    public static void adicionarPontosMapa(double xlatitude, double ylongitude)
    {
        latitude.Add(xlatitude);
        longitude.Add(ylongitude);
        contadorPontos++;
        gerarTabela();
    }
    public static void gerarTabela()
    {
        distancialatlong = new double[contadorPontos, contadorPontos];
        for (int i = 0; i < contadorPontos; i++)// Para X
        {
            for (int j = 0; j < contadorPontos; j++)// Para Y
            {
                distancialatlong[i, j] = 6371 * (

                Math.Acos(
                            Math.Cos(Radians(90 - double.Parse(latitude[i].ToString())))  *
                            Math.Cos(Radians(90 - double.Parse(longitude[i].ToString()))) +
                            Math.Sin(Radians(90 - double.Parse(latitude[i].ToString())))  *
                            Math.Sin(Radians(90 - double.Parse(longitude[i].ToString()))) *
                            Math.Cos(Radians(double.Parse(latitude[i].ToString()))- double.Parse(longitude[i].ToString()))
                            )
                    ) * 1.15;
            }
        }
        Configuracao.tamanhoCromossomo = contadorPontos;
    }
```

Fonte: O próprio autor.

Configuração é uma classe do tipo estática que conterá as configurações gerais para a execução do algoritmo genético. Na Figura 44 está o recorte da classe de configuração.

Figura 43 - Recorte parcial da classe Configuração.

```
public static class Configuracao
{
    public static int tamanhoCromossomo = 0;
    public static int tamanhoPopulacao = 600;
    public static Random aleatorio = new Random();
    public static double taxaCrossover = 0.75;
    public static double taxaMutacao = 0.03;
    public static Mutacao tipoMutacao = Mutacao.Populacao;
    public enum Mutacao
    {
        Novo,
        Populacao,
        Genes,
        Aleatoria,
        Especifica
    }
    public static int numeroCompetidores = 3;
}
```

Fonte: O próprio autor.

Útil é uma classe do tipo estática que conterá métodos ou outras definições que sejam de alguma utilidade para a solução, mas que conceitualmente não se encaixam em nenhum conceito relacionando ao algoritmo genético, pois é um tipo de classe muito utilizada para uma melhor organização de projetos de softwares, classe Útil.

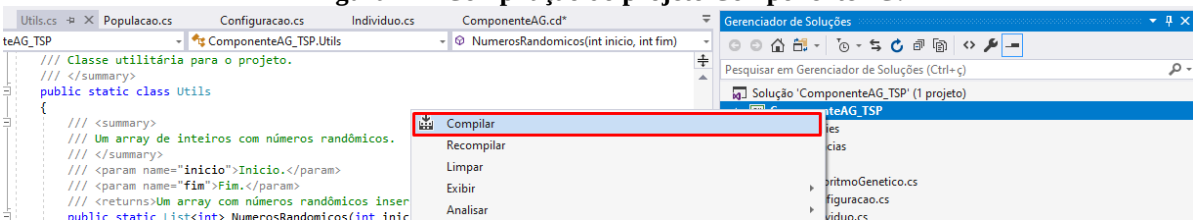
Quadro 3 - Método gerar Tabela da classe Tabela Distâncias.

```
{
        tableDist = new double[contadorPontos, contadorPontos];

        for (int i = 0; i < contadorPontos; i++)// Para X
        {
          for (int j = 0; j < contadorPontos; j++)// Para Y
           {
              tableDist[i, j] = 6371 * (
               Math.Acos(
                                          Math.Cos(Radians(90 -
        double.Parse(X[i].ToString()))) * Math.Cos(Radians(90 - double.Parse(Y[i].ToString()))) +
                                          Math.Sin(Radians(90 -
        double.Parse(X[i].ToString()))) * Math.Sin(Radians(90 - double.Parse(Y[i].ToString()))) *
                                          Math.Cos(Radians(double.Parse(X[i].ToString()))-
        double.Parse(Y[i].ToString())) * 1.15;
             }
        }

 Configuracao.tamanhoCromossomo = contadorPontos;
}
```

Fonte: O próprio autor.

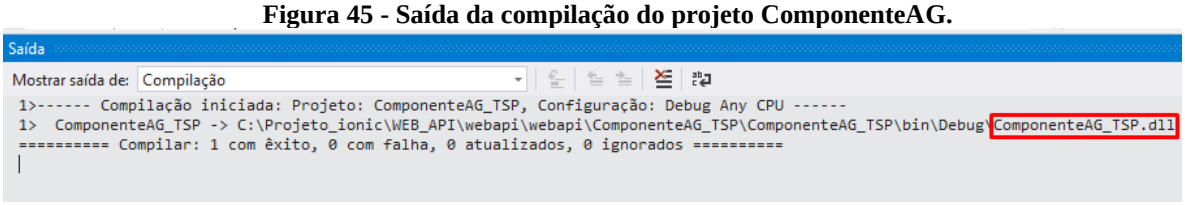
Definidas as classes e suas respectivas codificações, o passo seguinte foi gerar - por meio do *Visual Studio Community* 2017 - a .dll que será utilizada pelo projeto da API web. Para isso foi necessário, somente, compilar todo o projeto, como apresentado na Figura 45 e na Figura 46, e a saída da compilação.

Figura 44 - Compilação do projeto ComponenteAG.



Fonte: O próprio autor.

Figura 45 - Saída da compilação do projeto ComponenteAG.

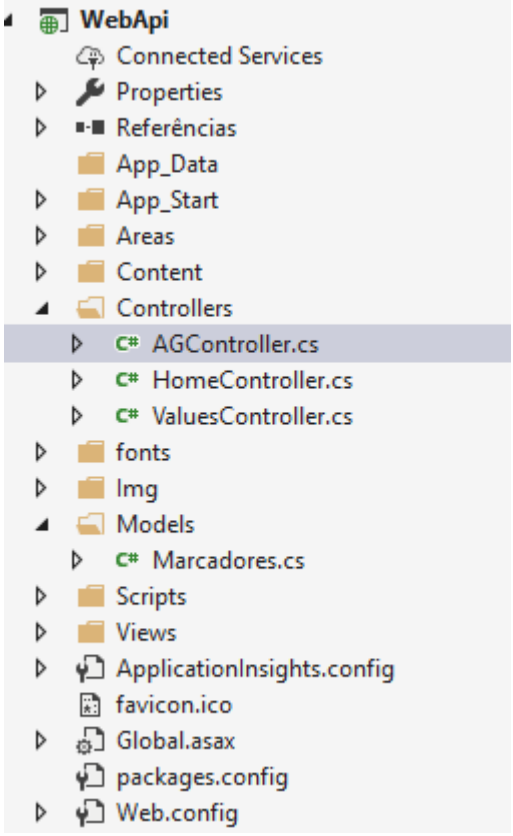


Fonte: O próprio autor.

O outro projeto desenvolvido foi o do tipo WEB API que implementa um serviço web que consumirá ComponenteAG como um serviço de processamento que utiliza o algoritmo genético no backend[8]. No desenvolvimento da Web Api, por dentro do Visual Studio Community 2017, a ferramenta já estrutura todo o projeto no padrão MVC, o que facilita a organização do projeto como um todo, na Figura 47 é apresentado a estrutura definida no projeto.

[8] Backend é a parte da aplicação que roda de “trás” da aplicação, é no backend que se implementam as regras de negócios da aplicação.

Figura 46 - Estrutura MVC do projeto WebApi no Visual Studio Community 2017.



Fonte: O próprio autor.

No projeto WebApi foram adicionadas duas classes, uma classe de modelo chamada Marcadores, Figura 48, que abstrai um marcador do mapa com as propriedade de Longitude e Latitude e outra classe do tipo Controller chamada AGController, quadro 5, encarregada de receber as requisições via método post que vão conter as coordenadas (marcadores) e executar as chamadas das funções de algoritmos genéticos da dll componenteAG.

Figura 47 - Classe de modelo Marcadores do projeto WebApi.

```
namespace WebApi.Models
{
    public class Marcadores
    {
      public Double Latitude { get; set; }
      public Double Longitude { get; set ; }
    }
}
```

Fonte: O próprio autor.

Quadro 4 - Classe controller AGController do projeto WebApi.

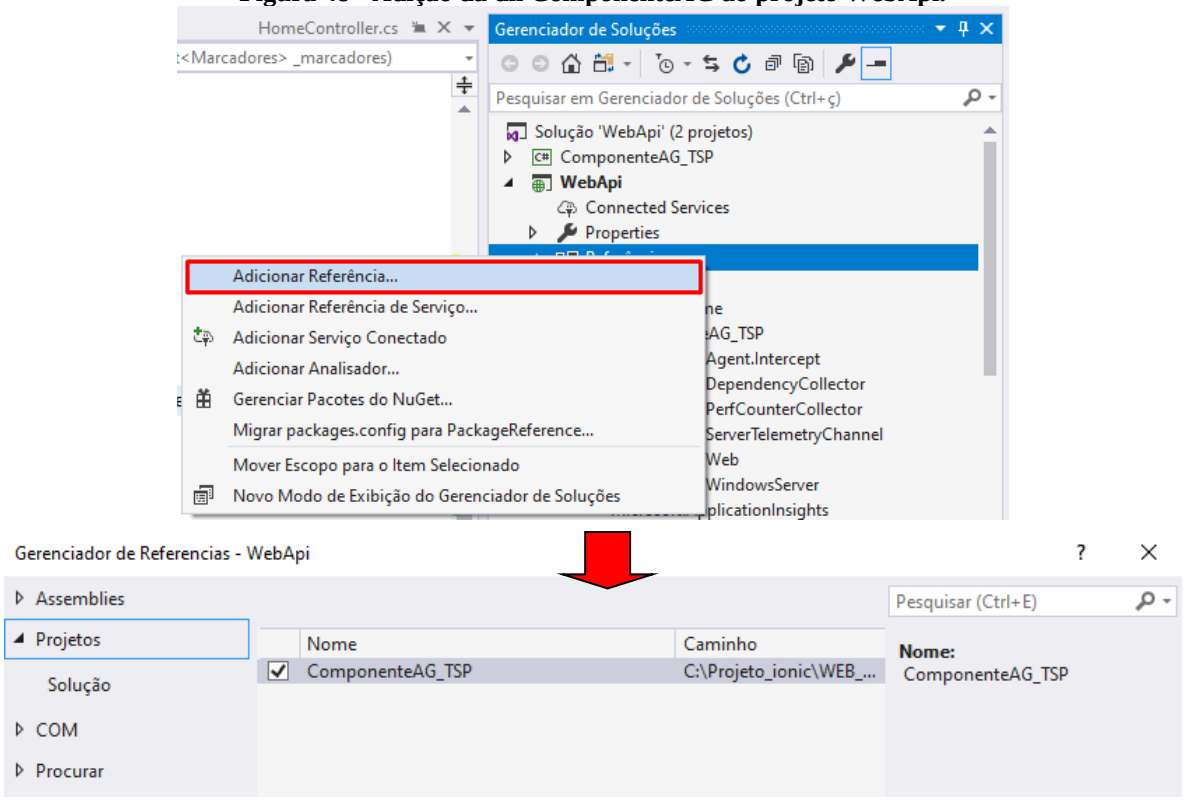
```
public class AGController : ApiController
    {
        //Definindo uma população.
        Populacao pop;
        //Definindo o melhor indivíduo.
        Individuo bestInd;

        [HttpPost]
        public HttpResponseMessage PostPoint([FromBody]List<Marcadores> _marcadores)
        {
            //Adicionar os pontos na tabela.
            for (int y = 0; y < _marcadores.Count; y++)
            {
              TabelaDistancias.addPoint(_marcadores[y].Latitude, _marcadores[y].Longitude);
            }
            pop = new Populacao();
            //Oredenação da população.
            pop.ordenarPopulacao();

            //Criar meu objeto de algoritmo genético.
            AlgoritmoGenetico AG = new AlgoritmoGenetico();

            //Evoluir a população em 500 vezes.
            for (int i = 0; i < 500; i++)
            {
                pop = AG.executeGA(pop);
            }
            //Pega o melhor indivíduo da população;
            bestInd = pop.getMelhor();

            //Limpar recursos
            TabelaDistancias.clear();
            pop = null;

            //Montando o response com a melhor rota, até o momento.
            var response = Request.CreateResponse(HttpStatusCode.Created,bestInd.getCromossomo());
            string uri = Url.Link("DefaultApi", new { });
            response.Headers.Location = new Uri(uri);
            return response;
        }

    }
```

Fonte: O próprio autor.

Na classe **AGController** é definido um método do tipo *HttpResponseMensage*, chamado *PostPoint,* que será invocado numa URL e receberá como parâmetro uma lista de marcadores de latitudes e longitudes. De posse dessa lista, a WebApi utiliza o componente e executa o algoritmo genético em um total de 500 vezes e retorna à melhor solução encontrada pela invocação do método *getMelhor*() do objeto bestInd.

Para adicionar o ComponenteAG ao projeto WebApi é necessário ir ao gerenciador de soluções do projeto e no item referências para adicionar a dll correspondente, vide Figura 49.

Figura 48 - Adição da dll ComponenteAG ao projeto WebApi.



Fonte: O próprio autor.

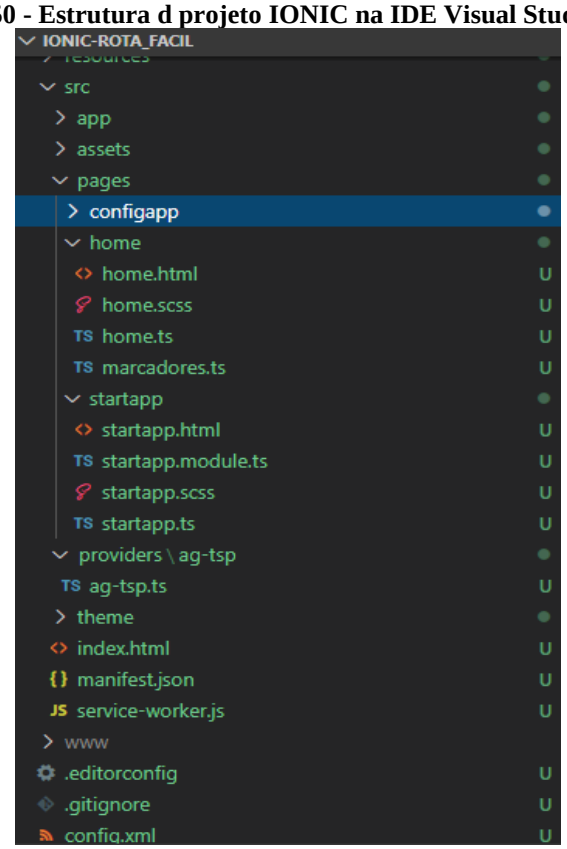
Após a criação do projeto e a execução em máquinas de teste, foi adquirido o domínio *www.qcode.com.br* para hospedagem da solução WebApi, (Figura 50). Para a hospedagem o critério foi o suporte ao ASP.NET versão 4, para execução do projeto online, com isso o serviço escolhido foi o *host Azul* (www.hostazul.com.br).

Figura 49 - WebApi hospedado no domínio qcode.com.br.

Fonte: O próprio autor.

Para o desenvolvimento do aplicativo mobile, optou-se pelo IONIC versão 3 usando como IDE o Visual Studio Code. Isso porque um projeto IONIC propicia desenvolver aplicações de maneira híbrida para as principais plataformas de mercado IOS, Android e Windows Phone. Na Figura 51 é apresentada a estrutura do projeto IONIC na IDE.

Figura 50 - Estrutura d projeto IONIC na IDE Visual Studio Code.



Fonte: O próprio autor.

Na construção do aplicativo foram desenvolvidas duas páginas. Uma delas é de introdução do aplicativo *startapp.html* e *home.html* que irá renderizar o mapa para utilização pelo aplicativo, cada qual com seus respectivos arquivos .ts, e .scss. É no arquivo .ts que é codificado todo o typescript da aplicação. Foi desenvolvido, ainda, o arquivo *ag-tsp.ts* com as funções responsáveis pela conexão com a Api Web, para o envio das coordenadas no formato correto e pela recepção da resposta e marcação no mapa.

A *startapp.html* é a página inicial da aplicação, responsável por redirecionar o usuário para a tela do mapa. Na Figura 52 é mostrada a página inicial da aplicação

rodando no emulador embutido, no Quadro 5 é apresentada a codificação do Startapp.ts com o código typescript de navegação.

Figura 51 - Página inicial da aplicação Startapp.html.

Fonte: O próprio autor.

Quadro 5 - Codificação do arquivo startapp.ts.

```
import { Component } from '@angular/core';
import { IonicPage, NavController, NavParams, Platform } from 'ionic-angular';
import { HomePage } from '../home/home';
import { ConfigappPage } from '../configapp/configapp';

@IonicPage()

@Component({
  selector: 'page-startapp',
  templateUrl: 'startapp.html',
})
export class StartappPage {

  plataforma :any;

  constructor(public navCtrl: NavController, public navParams: NavParams,platform: Platform) {
    this.plataforma = platform;
  }
  ionViewDidLoad() {console.log('ionViewDidLoad StartappPage');}
  intoHome(){this.navCtrl.push(HomePage);}
  intoConfig(){this.navCtrl.push(ConfigappPage);}
  intoExit(){this.plataforma.exitApp();}
}
```

Fonte: O próprio autor.

O arquivo *home.html* pode ser conferido sendo renderizado na tela da aplicação, (Figura 53). Nessa página é renderizado o mapa que será usado para o traçado da rota. Nos Quadros 6 e 7 é apresentado um trecho da codificação do *home.ts* e do *home.html* onde se apresenta o método a ser invocado pelo botão processar.

Figura 52 - Página do mapa da aplicação home.html.

Fonte: O próprio autor.

Quadro 6 - Trecho codificação do arquivo home.ts.

```
executaAG()
  {
    this.ag.conectaComAG(this._marcadores);
  }
```

Fonte: O próprio autor.

Quadro 7 - Codificação do arquivo home.html.

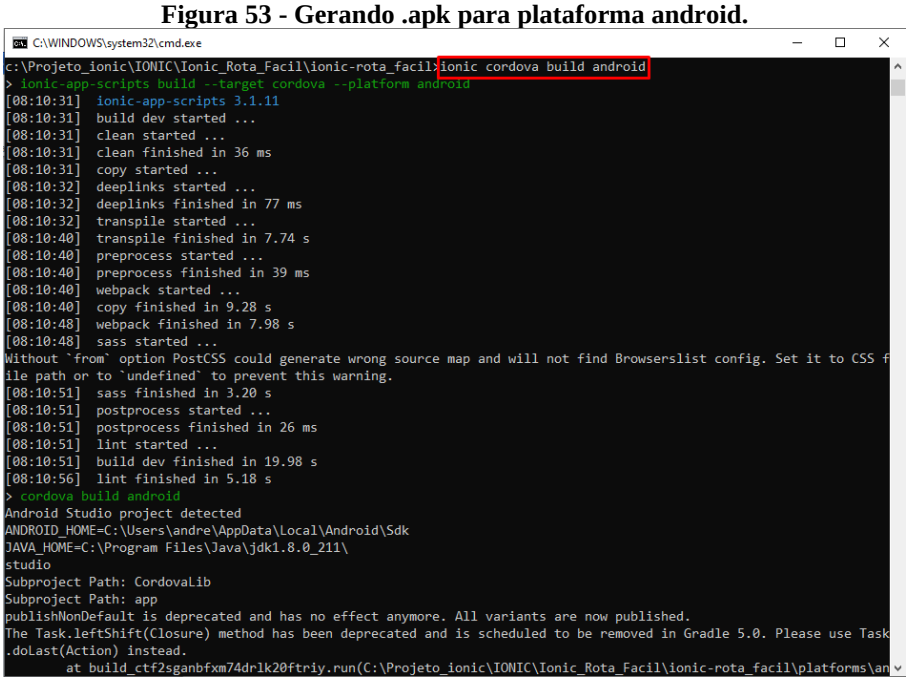
```
<ion-header>
  <ion-navbar color="primary">
    <ion-title>
      Rota Fácil - MPOIC
    </ion-title>
    <ion-buttons end>
      <!-- Botões de Ação para adicionar os marcadores.-->
    </ion-buttons>
  </ion-navbar>
</ion-header>
<ion-content>
   <div id="map"></div>
   <!-- Botão que vai executar o AG -->
   <ion-buttons  >
      <button ion-button full (click)="executaAG()"><ion-icon name="briefcase"></ion-
icon>Processar</button>
   </ion-buttons>
</ion-content>
```

Fonte: O próprio autor.

No arquivo *ag-tsp.ts* do projeto IONIC *3* foi codificado o endereço da Api Web para a passagem dos pontos do mapa para processamento. Nesse arquivo se encontra a classe que contém o método que se conecta ao serviço e renderiza o retorno no mapa.

Para criação do aplicativo, para execução no aparelho celular, foi utilizado o utilitário de linha de comando ***IONIC cordova build android*** que permite gerar um arquivo .apk a ser instalado no aparelho android. Nas figuras 54 e 55 está a execução do comando.

Figura 53 - Gerando .apk para plataforma android.



Fonte: O próprio autor.

Figura 54 - Localização do arquivo gerado.

```
BUILD SUCCESSFUL in 34s
46 actionable tasks: 3 executed, 43 up-to-date
Built the following apk(s):
       c:\Projeto_ionic\IONIC\Ionic_Rota_Facil\ionic-rota_facil\platforms\android\app\build\outputs\apk\debug\app-d
ebug.apk

c:\Projeto_ionic\IONIC\Ionic_Rota_Facil\ionic-rota_facil>
```

Fonte: O próprio autor.

4 RESULTADOS

Neste capítulo do trabalho são apresentados os resultados obtidos pelo projeto. Ao abrir a aplicação no celular, o usuário irá se deparar com a imagem da Figura 56, em execução no emulador para as principais plataformas mobile do mercado.

Figura 55 - Tela Inicial da aplicação para as plataformas IOS, Android e Windows Phone.

Fonte: O próprio autor.

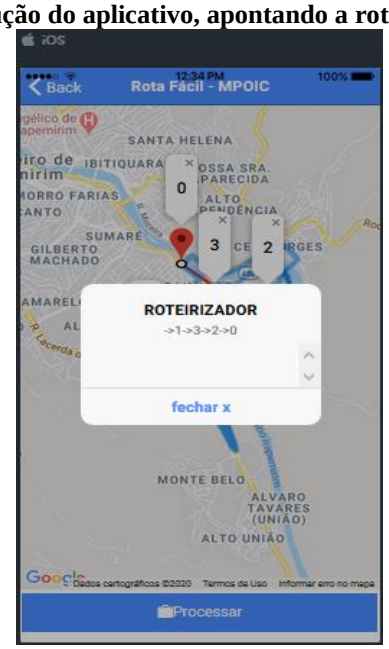
O usuário do aplicativo pode marcar no mapa, usando o touch da tela, os pontos que deseja processar e traçar de uma determinada rota, usando a heurística dos algoritmos genéticos que está disponível como um serviço online. Na Figura 57 é apresentada a funcionalidade de marcação dos pontos.

Figura 56 - Marcação dos pontos na tela do aplicativo.

Fonte: O próprio autor.

Traçado os pontos a serem percorridos no mapa renderizado, na tela do aplicativo, clica-se em processar e neste momento é criada uma estrutura de dados contendo as coordenadas dos pontos marcados e o retorno é processado pelo aplicativo que exibe a ordem a ser percorrida (Figura 58).

Figura 57 - Execução do aplicativo, apontando a rota a ser percorrida.



Fonte: O próprio autor.

Neste ponto da execução já é traçada a rota no mapa, como retas que ligam os pontos no mapa, além de renderizar a rota usando a API do *google maps* para traçar o percurso, como apresentado na Figura 59.

Figura 58 - Rota traçada na tela do aplicativo.

Fonte: O próprio autor.

Por se tratar de tecnologias web para o desenvolvimento de uma solução mobile, a depuração do aplicativo também pode ser realizada através do navegador na máquina de desenvolvimento (Figura 60), a tecnologia IONIC possibilita esse tipo de execução.

Figura 59 - Rota traçada no aplicativo em execução no navegador.

Fonte: O próprio autor.

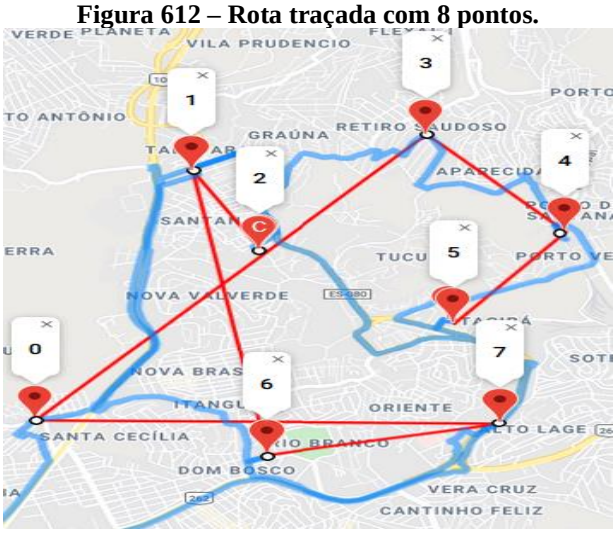
Na Figura 61, foram estabelecidos no aplicativo alguns pontos a serem percorridos, realizada de forma consecutiva o processamento das coordenadas o que ocasionou duas possíveis boas soluções de rotas.

Figura 60 - Rota traçada no google maps execução consecutivas.

Fonte: O próprio autor.

Na Figura 62 são adicionados 8 pontos, seguindo a premissa de aumento da complexidade pela adição de novos pontos a serem percorridos, simulando o problema do caixeiro viajante.

Figura 612 – Rota traçada com 8 pontos.



Fonte: O próprio autor.

Na Figura 63 são adicionados pontos numa visualização mais ampla do mapa onde são inseridas as cidades de Sorrido-MT (7), Vitória da Conquista-BA (6), Brasília-DF (1), Vitória-ES (5), Belo Horizonte (4), Ribeirão Preto-MG (3), Uberlândia-MG (2) e Campo Grande-MTS (0).

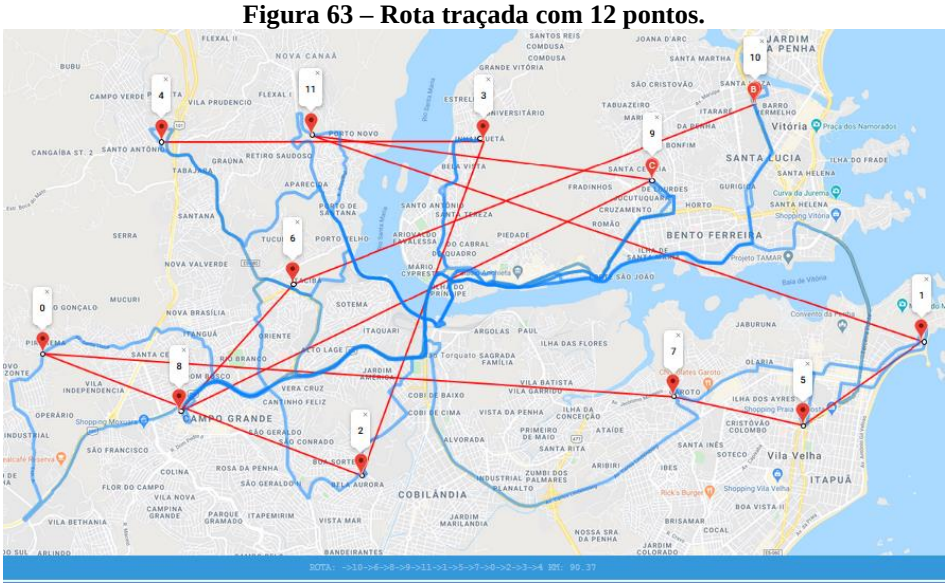
Figura 623 – Rota traçada com 8 cidades.

Fonte: O próprio autor.

Na Figura 64 são adicionados mais de 12 pontos a serem percorridos na região metropolitana da grande Vitória-ES. Os pontos presentes no mapa foram

escolhidos de forma aleatória como forma de aumentar a complexidade do percurso e avaliar rota sugerida pelo aplicativo.

Figura 63 – Rota traçada com 12 pontos.



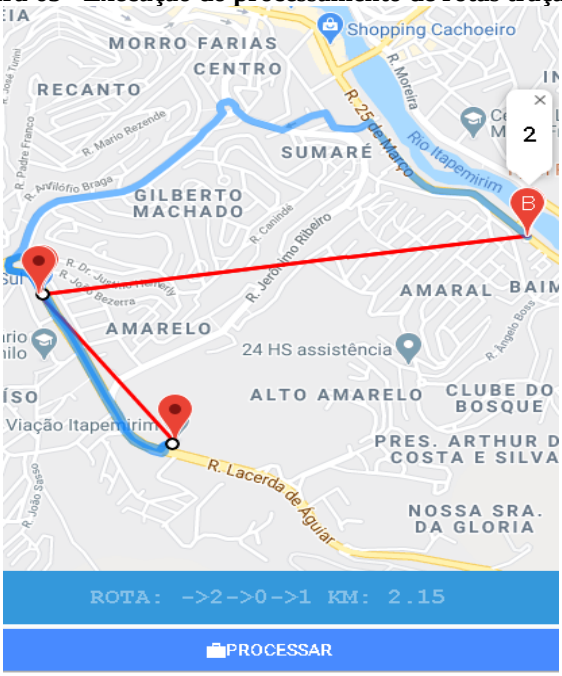
Fonte: O próprio autor.

Para validação do protótipo não foram encontradas aplicações semelhantes à disposição no que se refere aos aplicativos que possibilitem marcar de maneira fácil, na tela do mapa, uma série de pontos. Contudo, existem alguns serviços online como o *google maps* ou *waze carpool* que permitem que sejam traçadas rotas. Para este trabalho, foram comparadas a rota traçada junto ao *google maps* e a sugerida pela aplicação, medindo a distância total em quilômetros. A seguir, na Figura 65, está a imagem do google maps e, na Figura 66, o total de quilômetros em uma rota alternativa sugerida pela aplicação do Rota Fácil.

Figura 64 - Rota traçada no google maps.

Fonte: O próprio autor.

Figura 65 - Execução do processamento de rotas traçadas.



Fonte: O próprio autor.

5 CONSIDERAÇÕES FINAIS

5.1 CONCLUSÕES

O desenvolvimento deste trabalho de pesquisa se iniciou com um estudo bibliográfico na busca de trabalhos científicos que abordariam os seguintes temas: Algoritmos Genéticos, Problema do Caixeiro Viajante, desenvolvimento mobile e criação de API's online. Com isso, buscou-se a identificação de trabalhos e autores relevantes aos temas estudados, evidenciando ainda que há vasta área de estudos que pode ser explorada com relação a esses temas.

Um dos objetivos do trabalho foi mostrar a viabilidade de implementação de soluções tecnológicas modernas e com custo relativamente acessível para problemas de otimizações reais, motivo pelo qual se optou pela heurística dos algoritmos genéticos e o desenvolvimento de um protótipo funcional atestou a viabilidade técnica de sua produção.

Portanto, ao fim deste trabalho de pesquisa, obteve-se uma versão beta de um aplicativo de móbile, que consome um serviço hospedado em nuvem, rodando a heurística do algoritmo genético, focado na implementação do problema do caixeiro viajante.

Com base no planejamento traçado no início deste trabalho e dos resultados obtidos na produção da solução, acredita-se que os objetivos de pesquisa foram alcançados.

5.2 TRABALHOS FUTUROS

Como propostas para trabalhos futuros foram identificados alguns pontos a serem melhorados, que são eles:

- Melhorias do aplicativo mobile:
  - Criação de tela de login, com possibilidade de integração com as API's Facebook e Google;
  - Salvar as dez últimas melhores rotas em um banco de dados SQLite no próprio aplicativo;
  - Redefinição do design das telas do aplicativo por um profissional da área de UX (User Experience).

- Melhorias no serviço web:
  - Adição de outras heurísticas de rotas a API;
  - Criação de um pool de API, ofertando outros serviços de otimização para aplicativos;
  - Busca de um serviço de hospedagem mais profissional.

REFERÊNCIAS

ARAÚJO, A. V. **Treinamento Avançado em .NET**. São Paulo: Digerati Books, 2006.

ARAÚJO, E. C. **ASP.NET MVC 5 crie aplicações web na plataforma Microsoft**. São Paulo: Casa do Código, 2017.

ARORA, S.; BARAK, B. **Computational complexity:** a modern approach. New York: Cambridge University Press, 2009.

ARROYO, J. E. C. **Heurísticas e metaheurísticas para otimização combinatória multiobjectivo**. 2002. Tese (Doutorado em Engenharia Elétrica) - Universidade Estadual de Campinas, Campinas, 2002. Disponível em: http://repositorio.unicamp.br/bitstream/REPOSIP/260313/1/Arroyo_JoseEliasClaudio_D.pdf. Acesso em: 13 nov. 2018.

BIERMAN, G.; ABADI, M.; TORGERSEN, M. Understanding typescript. *In:* OBJECT-ORIENTED PROGRAMMING, 28., 2014, Uppsala. **Proceedings** [...]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. p. 257–281. Disponível em: http://dx.doi.org/10.1007/978-3-662-44202-9_11. Acesso em: 16 dez. 2019.
.

BLINI, FELIPE. **Desenvolvimento de Aplicativos Móveis com Javascript:** IONIC, React Native e NativeScript. Qual escolher? Medium, 2016. Disponível em: https://medium.com/@felipeblini/desenvolvimento-de-aplicativos-m%C3%B3veis-com-javascript-IONIC-react-native-e-nativescript-c303b17fba0d. Acesso em: 21 nov. 2019.

COSTA, H. G. Modelo para *webibliomining*: proposta e caso de Aplicação. **Revista da FAE**, Curitiba, v. 13, n. 1, p. 115-126, jun. 2010.

COSTA, D. A. O.; MATOS, M. A. E. Projeto de trabalho para ensinagem em Geografia “meu Brasil e suas capitais” no ensino fundamental. **Revista de Ensino de Geografia**, Uberlândia, v. 4, n. 6, p. 31-50, jan./jun. 2013. Disponível em: http:\\ww.revistaensinogeografia.ig.ufu.br. Acesso em: 25 dez. 2019.

COPPIN, B. **Inteligência Artificial.** Tradução e revisão técnica Jorge Eduardo Pires Valério. Rio de Janeiro: LTC, 2012.

CROSTA, A. P. **Sensoriamento Remoto**. Anuário Fator gis 97: o guia de referência do Geoprocessamento. Curitiba: Sagres, 1997.

CUNHA, C. B.; BONASSER, U. O.; ABRAHÃO, F. T. M. **Experimentos computacionais com heurísticas de melhorias para o problema do caixeiro viajante**. São Paulo: ANPET, 2002. Disponível em: https://www.researchgate.net/profile/Claudio_Cunha4/publication/228434832_Experimentos_computacionais_com_heuristicas_de_melhorias_para_o_problema_do_caixeiro_viajante/links/54803ccb0cf2ccc7f8bb2c18.pdf. Acesso em: 13 nov. 2018.

ENGHOLM Júnior, H. **Análise e design:** Orientados a Objetos. São Paulo: Novatec, 2013.

FIELDING, R. T. **Architectural Styles and the Design of Network-based Software Architectures**. 2000. Tese (Doutorado em Informação e Ciência da Computação) – Universidade da Califórnia, Califórnia, 2000.

FEIGENBAUM, E. A., FELDMAN, J. **Computers and thought**. New York: McGraw-Hill Inc., 1963.

GAREY, M. R.; JOHNSON, D. S. **Computers and intractability:** a guide to theory NP-completeness. New York: WH Freeman & Co, 1979.

GIL, A. C. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2002. 176 p.

GÓES, W. G. **Aprenda UML por meio de estudos de casos**. São Paulo: Novatec, 2014.

GOOGLE MAPS. Disponível em: https://www.google.com.br/maps/. Acesso em: 21 ago. 2019.

GOMES, F. M.; PEREIRA, F. M.; MARINS, F. A. S; SILVA, M. B. Estudo comparativo entre os Métodos Gradiente reduzido Generalizado e Algoritmo Genético em otimização com Múltiplas respostas. **Revista Produção Online,** Florianópolis, SC, v.17, n. 2, p. 592-619, 2017.

GONÇALVES, P. R.; SILVA, R. F. Web Services: Uma Análise Comparativa. **Revista das Faculdades Integradas Claretianas**, Rio Claro, SP, v.1 n. 5, p. 7-18, 2012.

GOIS, A. **IONIC Framework:** construa aplicativos para todas as plataformas. São Paulo: Casa do Código, 2017.

GOLDBERG, D. **Genetic Algorithms in Search, Optmization and Machine Learning**. New York: Addison Wesley Publishing Co. Inc, 1989.

HOLLAND, J. H. **Adaptation in natural and artificial systems:** An introductory analysiswith applications to biology, control, and artificial intelligence. Oxford, England: University Michigan Press, 1975.

HORTA, D. A.; SOLDATI, F.; FREITAS JUNIOR, N.; MATIAS, I. O. Utilização de algoritmos genéticos na movimentação autônoma de agentes em jogos eletrônicos. *In*: SIMPEP – SIMPÓSIO DE ENGENHARIA DE PRODUÇÃO, 21., 2014, Bauru, SP. **Anais** [...]. Bauru, SP: SIMPEP, 2014. p. 2-14.

IKEDA, P. A. **Introdução aos Algoritmos Genéticos**. 2009. Disponível em: https://www.ime.usp.br/~gold/cursos/2009/mac5758/PatriciaGenetico.pdf. Acesso em: 11 set. 2018.

IZO, F.; MATIAS, I. O.; PASSOS, U. R. C. Otimização no Operador de Crossover (Cruzamento) para Resolução do Problema das N-Rainhas. *In*: EINEPRO – ENCONTRO MINEIRO DE ENGENHARIA DE PRODUÇÃO, 11., 2015, São João da Barra, RJ. **Anais** [...]. São João da Barra, RJ: EINEPRO, 2015, v. 1. p. 1-12.

JANA, P. K., ANWIT, R. A Variable Length Genetic Algorithm approach to Optimize Data Collection using Mobile Sink in Wireless Sensor Networks**.** *In*: INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING AND INTEGRATED NETWORKS (SPIN), 5., 2018. Noida. **Annals** […]. Noida: SPIN, 2018. p. 73-77.

KREJCAR, O.; JIRKA, J.; JANCKULIK, D. Use of Mobile Phones as Intelligent Sensors for Sound Input Analys is and Sleep State Detection*.* **Sensors,** Basel, v. 11, n. 6, p. 6037-6055, 2011.

KIM, MinHyeop; KO, In-Young. An Efficient Resource Allocation Approach Based on a Genetic Algorithm for Composite Services in IoT Environments. *In:* IEEE INTERNATIONAL CONFERENCE ON WEB SERVICES, 11., 2015, New York. **Annals** […]. New York: IEEE International Conference on Web Services, 2015. p. 543-550.

LARMAN, G. **Utilizando UML e Padrões:** Uma introdução á análise e ao projeto orientado a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007.

LINDEN, R. **Algoritmos Genéticos**. 3. ed. Rio de Janeiro: Ciência Moderna, 2012.

LOPES, L. S. A. **Uma heurística baseada em algoritmos genéticos aplicada ao problema de cobertura de conjuntos**. 1995. Dissertação (Mestrado em Computação Aplicada) – INPE, São Jose dos Campos, São Paulo, 1995.

LOBO, E. J. T. **Curso de Engenharia de Software.** São Paulo: Digerati Books, 2008.

LOTAR, A. **Como Programar com ASP. NET e C#**. São Paulo: Novatec, 2010.

LUGER, George, F. **Inteligência Artificial**. 6. ed. São Paulo: Person Education do Brasil, 2013.

MARTINS, F. A. R. **Controlo e monitorização de informação clínica partilhada**. 2016. Dissertação (Dissertação em Engenharia Informática) – Instituto Universitário de Lisboa, Lisboa, 2016.

MORO, T. D.; DORNELES, C.; REBONATTO, M. T. Web Services WS-* versus Web Services REST. **Revista Eletrônica de Iniciação Cientifica em Computação**, Porto Alegre, v 11, n. 1, 2011. Disponível em: https://seer.ufrgs.br/reic/article/view/22140/12928. Acesso em: 03 ago. 2019.

.

MICROSOFT: **ASP. NET Web APIs**. 2019. Disponível em: https://dotnet.microsoft.com/apps/aspnet/apis. Acesso em: 20 dez. 2019.

MICROSOFT: **Introdução ao .NET Framework**. 2019. Disponível em: https://docs.microsoft.com/pt-br/dotnet/framework/get-started/. Acesso em: 01 jan. 2019.

MICROSOFT: **Um tour pelo C# idioma**. 2020. Disponível em: https://docs.microsoft.com/pt-br/dotnet/csharp/tour-of-csharp/index/. Acesso em: 01 jan. 2020.

MITCHELL, M. **An Introduction to Genetic Algorithms.** MIT Press, Cambridge, Massachussetts: MIT Press, 1998.

NUNES, F. **Desenvolvendo aplicativos móveis Multiplataforma**. 2013. Disponível em: http://imasters.com.br/desenvolvimento/desenvolvendo-aplicativos-moveis-multiplataforma/. Acesso em: 14 abr 2019.

OLIVEIRA, I. C. **Complexidade computacional e o problema P vs NP**. 2010. Dissertação (Mestrado em Ciência da Computação) - Instituto de Computação, Universidade Estadual de Campinas, Campinas, SP, 2010.

RED HAT: **O que significa API e como ela funciona**. 2020. Disponível em: https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces. Acesso em: 03 Jan. 2020.

REINELT, G. **The Traveling Salesman:** Computational Solutions for TSP Applications. Berlin: Springer Verlag, 1994.

RODRIGUES, F. L. *et al.* Metaheurística algoritmo genético para solução de problemas de planejamento florestal com restrições de integridade. **Revista Árvore,** Viçosa, MG, v. 28, n. 2, p. 233-245, 2004.

RODRIGUES, M. Geoprocessamento: um retrato atual. **Fator Gis**: a Revista do Geoprocessamento, Curitiba, v. 1, n. 2, p. 20-23, 1993.

ROSA, T. de O., LUZ, H. S. Conceitos Básicos de Algoritmos Genéticos: Teoria e Prática. *In:* Encontro de Estudantes de Informática do Tocantins,11., 2009, Palmas. **Anais** [...]. Palmas: Centro Universitário Luterano de Palmas, 2009. p. 27-37.

SAUDATE, A. **REST:** Construa API's inteligentes de maneira simples. São Paulo: Casa do Código, 2014.

SCOPUS. **Content coverage guide.** Amsterdam, Netherlands: Elsevier BV, 2019.

SCUSSEL, A. **Por dentro do Google Maps**. Jul 2013. Disponível em: https://mundogeo.com/2013/07/01/artigo-por-dentro-do-google-maps/. Acesso em: 01 jan. 2020.

SHARMA, N.; MADHUKUMAR, A. S. Genetic Algorithm Aided Proportional Fair Resource Allocation in Multicast OFDM Systems. **IEEE Transactions on Broadcasting,** Ottawa, v. 61, n. 1, p. 16-29, 2015.

SHYLAJA, B. S. From navigation to star hopping: forgotten formulae. **Resonance 20**, [s.l.], p. 352-359. 2015. Disponível em: https://link.springer.com/article/10.1007%2Fs12045-015-0190-7. Acesso em: 20 jun. 2019.

SILVA, J. S. V**. Análise multivariada em zoneamento para planejamento ambiental.** Estudo de caso: Bacia Hidrográfica do Rio Taquari MS/MT. 2003. 307 f. Tese (Doutorado em Engenharia Agrícola) - Universidade Estadual de Campinas, Campinas, SP, 2003.

SILVA, J. X. O que é Geoprocessamento? **Revista do Crea RJ**, Rio de Janeiro, v. 79, p. 42-44, out./nov. 2009.

SISPER, M. **Introdução a Teoria da Computação**. 2. ed. São Paulo: Thomson Learning, 2007.

TECMUNDO. **Trello:** como esta ferramenta pode ajudar você a organizar a sua vida. Fev. 2015. Disponível: https://www.tecmundo.com.br/organizacao/75128-trello-ferramenta-ajudar-voce-organizar-vida.htm. Acesso em: 17 nov. 2019.

TOSCANI, L. V.; VELOSO, P. A. S. **Complexidade de Algoritmos**. 3. ed. Porto Alegre: Bookman, 2012.

VIEBRANTZ, A. F. P. M.; CAMPOS, G. F. S. Construindo Aplicativos Híbridos com Ionic Framework. *In*: ESCOLA REGIONAL DE INFORMÁTICA DE MATO GROSSO, 6., 2015, Cuiabá, MT. **Anais** […]. Mato Grosso: Escola Regional de Informática de Mato Grosso, 2015. p. 60-72.

WU, H.; LI, W.; DENG, S.; ZOMAYA, A. Service Selection for Composition in Mobile Edge Computing Systems. *In:* IEEE INTERNATIONAL CONFERENCE ON WEB SERVICES, 2., 2018, San Francisco, **Annals** […]. San Francisco: IEEE, 2018, p. 355-358.

YAN, L. TAM, J. LIU, H. CHEN, C. Registration of TLS and MLS Point Cloud Combining Genetic Algorithm with ICP. **Cartographica Sinica**, Cehui Xuebao, v. 47, n. 10, p. 528-536, abr. 2018.

YUCEL, F.; BULUT, E. Clustered Crowd GPS for Privacy Valuing Active Localization. **IEEE Access**, New York, v. 6, n. 10, p. 23213-23221, abr. 2018.