

Aplicação do Algoritmo Genético Para Elaboração de Roteiros Turísticos na Serra da Ibiapaba, Ceará

João Victor da Costa¹, Anderson Passos de Aragão¹, Nécio de Lima Veras¹

¹Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)
Campus Tianguá
Av. Tab. Luiz Nogueira Lima, s/n – Santo Antônio – CEP 62324-075
Tianguá – CE — Brasil

victorcosta.ifce@gmail.com; {anderson.aragao, necio.veras}@ifce.edu.br

Abstract. *The region of Ibiapaba mountains, in Ceará, stands out as a region with great tourist potential in the state. Because of this, the present work carried out the application of the Genetic Algorithm (GA) to develop an optimal route, based on the shortest distance, between a set of points of interests in the region. The problem was modeled using graphs and the distances obtained through Google Maps. The results showed that the GA reduced the distance from the proposed initial route by up to 66.58%, performing this function in less than 1.8 seconds. This work was used as a basis for an optimized and personalized tourist itinerary planner application.*

Resumo. *A Serra da Ibiapaba, no Ceará, desponta como uma região com grande potencial turístico no estado. Devido a isso, o presente trabalho realizou a aplicação do Algoritmo Genético (AG) para elaborar uma rota ótima, baseada na menor distância, entre um conjunto de locais turísticos da região. O problema foi modelado através de grafos e as distâncias obtidas através do Google Maps. Os resultados apontaram que o AG reduziu em até 66,58% a distância do roteiro inicial proposto, realizando tal função em um tempo inferior a 1,8s. Este trabalho foi utilizado como base para uma aplicação planejadora de roteiros turísticos otimizados e personalizados.*

1. Introdução

O turismo é uma importante atividade econômica, capaz de gerar renda, emprego e movimentar a economia de uma região [Rabahy 2019]. O artigo de [Tomé 2018] cita algumas condições naturais da região Nordeste do Brasil que fazem dela um polo atrativo para turistas de todo o país e do mundo, como a grande faixa litorânea, o clima tropical, a gastronomia característica e a seu rico artesanato. Além das praias, no estado do Ceará, realçam-se também as serras, como a Serra de Guaramiranga e da Ibiapaba, famosas pelo turismo de aventura, ecoturismo, mirantes e clima ameno, que podem chegar em torno de 16°C [Ceará 2019], [Brasil 2020].

Assim, a Serra da Ibiapaba possui atrativos que oferecem condições para o desenvolvimento de vários segmentos de turismo, entre eles o turismo religioso e histórico, podendo-se destacar também o turismo de aventura e o ecoturismo [Azevedo 2016]. Infelizmente, dos 7 pólos turísticos existentes no Ceará, a Ibiapaba é responsável por apenas

5,6% da demanda turística no estado, sendo a 3ª pior colocação. Estes dados corroboram a hipótese que, embora seja uma região com grande potencial turístico, este mesmo potencial não está sendo de fato aproveitado. Desta forma, uma ferramenta capaz de ajudar o turista a visitar locais de interesse com a menor distância possível pode ser um instrumento auxiliador para o aumento da exploração turística da região.

Portanto, o objetivo deste trabalho consiste em aplicar o Algoritmo Genético para encontrar uma rota ótima, baseada na menor distância, entre um conjunto de locais turísticos da Serra da Ibiapaba. Desta forma, este trabalho realizou a implementação de um AG de forma a verificar sua viabilidade em relação à fornecer uma rota mais curta em comparação com a rota inicial proposta em um espaço de tempo aceitável. Este algoritmo foi selecionado com base na literatura, onde ele é eficientemente aplicado em problemas de otimização de rotas, a exemplo dos trabalhos relacionados abaixo. Os dados utilizados nos testes do algoritmo representam importantes pontos turísticos da região presentes na Serra da Ibiapaba. A modelagem dos locais de interesse (pontos turísticos) foi realizada por meio de grafos. A coleta da longitude, latitude e da distância entre os locais se deu através do *Google Maps*.

Este trabalho está dividido da seguinte forma: a seção 2 apresenta os trabalhos relacionados; na seção 3 ocorre a fundamentação teórica, onde são apresentados breves resumos sobre a região estudada, o modelo de grafos, o Problema do Caixeiro Viajante e sobre o AG; a seção 4 trata sobre a metodologia utilizada no trabalho; a seção 5 demonstra os resultados e discussões e, por fim, as conclusões são descritas na seção 6.

2. Trabalhos Relacionados

Esta seção apresenta um conjunto de trabalhos para resolução do Problema do Caixeiro Viajante (PCV) aplicados a otimização de rotas, que utilizaram modelagem de grafos, além de utilizar o mesmo algoritmo que foi alvo deste trabalho.

O artigo de [Aydin and Telceken 2015] trata de uma comparação entre 3 algoritmos: o A*, a Otimização por Colônia de Formigas (ACO) e o AG. Esses algoritmos foram utilizados para resolver o PCV para um aplicativo móvel de planejamento de roteiros em uma cidade da Turquia. Cerca de 30 Pontos de Interesse (POI) foram utilizados e os algoritmos foram comparados em relação ao seu tempo de execução, onde testaram vários cenários com diferentes números de POI. Para modelagem do problema, os autores também utilizaram grafos. Em seus resultados os autores demonstram que o ACO obteve um desempenho inferior ao AG em 2 das 3 situações testadas: com 5 e 8 POIs. Acima de 10 pontos, o ACO torna-se mais rápido em sua execução.

[Adhau et al. 2018] propôs um sistema de recomendação para indicar locais populares ao usuário. Ao fim, um roteiro de viagem otimizado é então gerado a partir desses locais previamente selecionados. Os autores modelaram o problema também como um PCV e compararam 2 algoritmos de otimização para resolução do problema, sendo o ACO e AG. Em seus resultados e conclusões, os autores afirmam que ambos os algoritmos são adequados para chegar a uma solução viável para o PCV. Entretanto, o AG obteve um melhor desempenho quando comparado ao ACO, de forma que ele foi mais rápido em tempo de execução e também ofereceu uma distância menor a percorrer no roteiro final.

3. Fundamentação Teórica

3.1. A Serra da Ibiapaba

A Serra da Ibiapaba, situa-se na porção ocidental do Ceará, divisa com o estado do Piauí. É provida de diversas atrações turísticas, onde destacam-se o ecoturismo, o turismo de esporte e aventura além do turismo histórico, arquitetônico e religioso. Possui clima ameno, temperaturas médias que variam de 19°C a 29°C e altitudes de 500m a 1.100m. Como atrativos, destacam-se parques com floresta preservada, grutas, cachoeiras, mirantes, trilhas, além do Parque Nacional de Ubajara onde ainda há um teleférico. Há na região uma parcela histórica e religiosa que conta com um rico patrimônio material e imaterial, como igrejas também históricas, praças, teatro e casarões antigos, eventos culturais e festejos religiosos. [Azevedo 2016],[Silva et al. 2017].

3.2. Grafos

Grafo é um modelo matemático formado por dois conjuntos, onde o primeiro é o conjunto de elementos chamados vértices, pontos ou nodos; e o segundo é o conjunto que contém as relações entre os vértices, também conhecidos como linhas ou arestas. Se dois vértices estiverem relacionados, diz-se que entre eles existe uma aresta. Desta forma, conforme [Simões-Pereira 2013], pode-se modelar estruturas complexas, como estradas, redes de computadores, relações entre pessoas, entre outras. Um grafo é dito completo quando há uma aresta entre cada par de seus vértices. Já um grafo é ponderado quando um valor é associado às suas arestas. A Figura 1 demonstra visualmente um exemplo de grafo completo. Dentro das estruturas de grafos, dado um certo grafo G , um *percurso* ou *passeio* entre os vértices v e w é uma sequência de vértices e arestas, que inicia-se em um dos vértices (neste caso, o vértice v) e encerra-se no outro (vértice w). Existem passeios abertos e fechados. Acerca dos passeios fechados, eles iniciam-se e finalizam-se no mesmo vértice de origem, passando por pelo menos um outro vértice, sendo considerados *circuitos* ou *ciclos*. Um ciclo, quando ele engloba cada vértice contido no conjunto V , é denominado *ciclo hamiltoniano* [Cormen et al. 2012]. Um ciclo hamiltoniano também é exibido na Figura 1, através das arestas em vermelho, tendo o ponto inicial e final o vértice em verde.

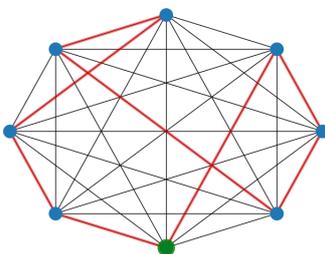


Figura 1. Exemplo de grafo completo contendo um Ciclo Hamiltoniano, resposta aceita para o PCV.

3.3. O Problema do Caixeiro Viajante

O Problema do Caixeiro Viajante (PCV) está relacionado ao problema do ciclo hamiltoniano. Ele constitui-se como um problema de roteiro de viagem em que um vendedor deve visitar n cidades somente uma vez, retornando à cidade de origem. O objetivo é

encontrar o percurso mais curto entre esse conjunto de cidades. Assim, modelando o problema como um grafo completo com n vértices, pode-se afirmar que o vendedor deseja fazer um percurso visitando cada vértice no grafo exatamente uma vez e terminando no vértice de origem [Russell and Norvig 2013], [Cormen et al. 2012]. O PCV é conhecido por ser da classe NP-difícil, ou seja, não há algoritmo de tempo polinomial conhecido na programação linear para quaisquer problemas NP-difíceis. Devido a essa limitação de recursos, principalmente tempo, a estratégia de resolução do PCV pode ser através de heurísticas. Heurísticas são métodos que, em geral, conduzem a uma boa solução de um problema, entretanto que não garantem produzir a solução com o resultado mais satisfatório dentre todas [Cormen et al. 2012][Russell and Norvig 2013]. Neste trabalho, a heurística chamada de Algoritmo Genético foi aplicada ao PCV a fim de encontrar uma solução viável.

3.4. Algoritmo Genético

O Algoritmo Genético faz parte da família de algoritmos conhecida como bio-inspirados, ou seja, inspirados pela natureza. Esse algoritmo inspira-se em mecanismos propostos na Teoria Evolutiva por meio da Seleção Natural, propostos por Charles Darwin bem como em conceitos da genética [Hirata 2018]. Assim, o AG trata-se de uma meta-heurística que baseia-se em uma analogia com processos naturais de evolução: dada uma certa população, os indivíduos com as melhores características genéticas possuem as maiores chances de sobreviver e de produzirem filhos cada vez mais aptos ao ambiente, enquanto que indivíduos menos aptos tendem a desaparecer [Chaves 2005]. Nos AGs, cada *cromossomo* está associado a uma solução do problema e cada *gene*, um elemento de uma posição do *cromossomo*, está associado a um componente da solução. Desta forma, tem-se um conjunto ou uma população de possíveis soluções para um certo problema. Essas soluções passam por processos de recombinação (como uma *reprodução*) e *mutação*. Esse processo é repetido por várias gerações até que se atinja o critério de parada, que pode ser o número máximo de gerações atingido pelo algoritmo [Hirata 2018], [Chaves 2005].

4. Metodologia

4.1. Pré-processamento

No início do projeto, os pontos turísticos foram selecionados usando indicações da plataforma de viagens [Tripadvisor 2020], do site da Rota Mirantes da Ibiapaba [Ibiapaba 2020], matérias de jornais estaduais como "Jornal O Povo"[Online 2019] e "Diário do Nordeste"[Dourado 2019], além de experiências de moradores locais. Ao total 40 pontos foram selecionados e organizados em um mapa utilizando o *Google My Maps* [Google LLC 2020a]. Finalizada a seleção dos pontos turísticos, foi realizada a etapa de mapeamento que consistiu em realizar o levantamento das coordenadas geográficas (latitude e longitude) dos 40 pontos selecionados. Esse mapeamento se deu através de uma ferramenta que converte endereços postais em coordenadas geográficas, a *Google Maps Geocoding API* [Google LLC 2020b]. Estas coordenadas foram armazenadas no Sistema Gerenciador de Banco de Dados *MySQL*, criado através do ambiente de desenvolvimento XAMPP [Apache Friends 2020] no próprio computador de teste.

O AG foi implementado utilizando a linguagem *Python* [PSF 2020]. Para implementar as estruturas de Grafos, foi utilizado o pacote *NetworkX* [NetworkX 2019] importado à linguagem de programação. Uma matriz de distância também foi criada no banco

de dados, de modo a auxiliar o algoritmo. Essa matriz foi criada com o auxílio da *Google Maps Distance Matrix API*, sendo este um dos diferenciais deste trabalho: as distâncias entre os pontos são as distâncias calculadas pelo *Google Maps*. Por fim, os experimentos foram realizados utilizando um notebook modelo Dell(R) Inspiron(TM) 3442 equipado de um processador Intel(R) Core(TM) i3-4005U CPU @ 1.70GHz, 4GB de memória RAM DDR3 e 256GB de armazenamento SSD.

4.2. Experimentos

Para realização dos experimentos, os pontos turísticos – ou pontos de interesse (POI) – foram ordenados aleatoriamente. Após isso, organizou-se os POI aleatoriamente em 10 grafos independentes, com quantidades múltiplas de 4. Portanto, o menor grafo continha 4 pontos e o maior 40, compreendendo assim todos os POI. Na Figura 2 há o exemplo da metodologia utilizada neste trabalho, onde está ilustrada a situação com um grafo contendo 4 POI, sendo esta metodologia replicada a todos os grafos.

Desta maneira, para criação dos roteiros iniciais bem como calcular as distâncias destes roteiros, cada grafo (Figura 2a) tornou-se uma lista aleatoriamente ordenada de tamanho igual ao seu grafo de origem (Figura 2b). Desta forma a lista gerada equivale a um roteiro proposto, que parte do primeiro item, percorre a lista e encerra-se no ponto de origem, formando assim um ciclo hamiltoniano. Modelando ao AG, cada lista produzida pelo grafo representa um indivíduo da população inicial do problema, sendo então submetida ao algoritmo (Figura 2c). Ao fim, o algoritmo retornava uma lista (Figura 2d) contendo portanto o roteiro otimizado, além da distância final e o tempo com o qual o AG gerou este resultado. Os resultados foram inseridos em uma planilha para que pudessem ser analisados.

O AG foi executado por 50 vezes de modo a validar os resultados de cada experimento. Os parâmetros aplicados à ele foram: tamanho da população = quantidade de vértices do grafo, ou seja, variando de 4 a 40; tamanho do grupo de elite = um terço da quantidade de vértices do grafo atual; taxa de mutação = 0,01; gerações = 100, sendo este o critério de parada.

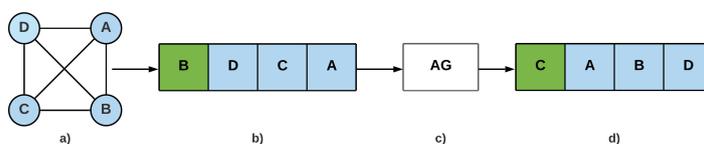


Figura 2. Exemplo da metodologia utilizada no experimento.

5. Resultados e Discussões

Esta seção trata dos resultados do algoritmo empregado juntamente com a discussão acerca de cada um deles.

O primeiro teste verifica o tempo médio de execução do AG na busca pelo roteiro ótimo, nas 50 execuções. A Figura 3 demonstra a média dos tempos de execução do algoritmo com relação aos experimentos, contendo portanto de 4 a 40 POI. Os tempos de execução aqui demonstrados referem-se ao tempo em que o algoritmo leva para encontrar

uma solução, seja essa solução ótima ou não. Em relação à Figura 3, é interessante observar a velocidade com a qual o AG gera roteiros com até 12 POI, ficando abaixo da marca de 0,5s. Entre 16 e 24 POI, ele retorna um roteiro em menos de 1s e somente ultrapassa a marca de 1s a partir de 28 POI. De modo geral, há um destaque para o tempo de execução do AG quando submetido ao teste contendo todos os 40 POI: mesmo aumentando os POI de 28 para 40, o tempo de execução permaneceu dentro da casa de 1s, ficando em na média de 1,77s. Cabe destacar que o baixo tempo de execução é um fator importante, haja vista que a ideia fundamental é oferecer roteiros turísticos otimizados. O turista possui um tempo limitado de permanência na região visitada, portanto, ferramentas auxiliaadoras também necessitam de rapidez em sua execução para oferecer maior conforto ao usuário.

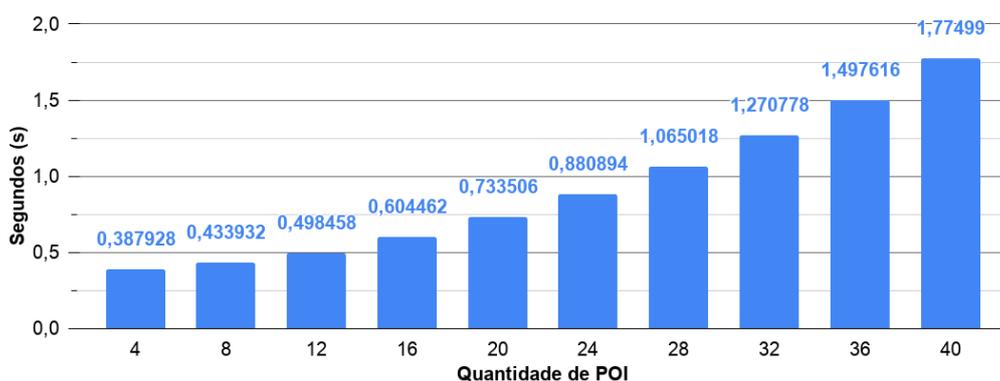


Figura 3. Tempo médio de execução do algoritmo AG, de 4 a 40 POI.

Quanto à Figura 4, ela demonstra um comparativo entre a distância inicial e a distância média dos roteiros retornados pelo algoritmo. Nela, é possível observar que em todos os testes o AG foi eficiente para otimizar a distância inicial. A menor redução do roteiro inicial ocorreu no teste com 4 POI, onde o AG diminuiu a distância de 80 km para 63,7 km, cerca de 20,36% a menos. Já a maior redução ocorreu no teste contendo 20 POI, cujo roteiro foi otimizado em 66,58%, de 962,7 km para 321,7 km. A redução média dos roteiros ficou por volta de 53,38%. Essa redução média implica que o algoritmo foi eficiente em seu objetivo de otimizar o roteiro inicial, haja vista que a distância foi reduzida em mais da metade. Essa redução oferece benefícios ao turista, como uma maior comodidade, economia de tempo e dinheiro, além da possibilidade de conhecer mais atrativos em um menor período de tempo.

6. Conclusões

Este trabalho apresentou um algoritmo genético aplicado na solução do problema do caixeiro viajante com o intuito de verificar se este algoritmo é capaz de elaborar uma rota ótima entre um conjunto de locais turísticos da região em um tempo hábil. Para modelagem do problema, foram utilizados pontos turísticos ou de interesse turístico localizados na Serra da Ibiapaba, microrregião do Ceará, considerando as distâncias reais entre eles.

Frente ao exposto, por meio dos resultados encontrados, pode-se constatar que o algoritmo obteve resultados satisfatórios em relação à ambos os objetivos deste trabalho. O AG foi capaz de reduzir em até 66,58% o roteiro inicial proposto, realizando tal função em um tempo hábil, abaixo de 1,8s.

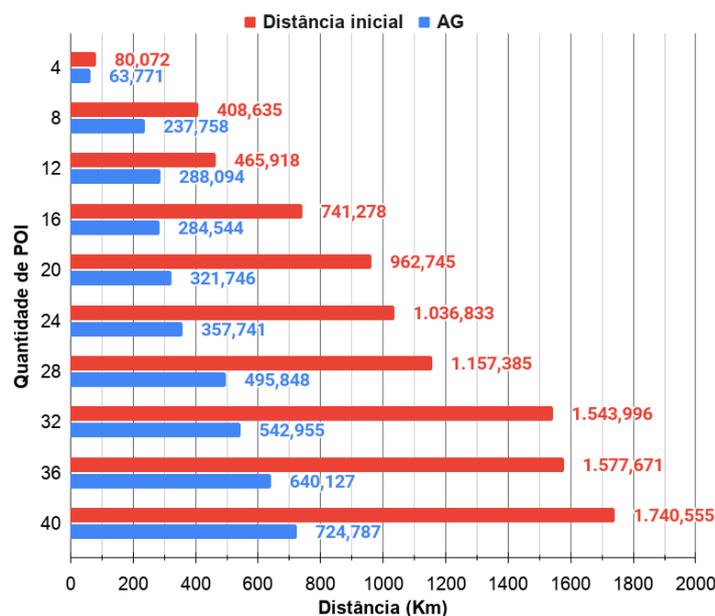


Figura 4. Comparação entre a distância inicial e a distância média gerada pelo AG.

Desta forma, com os resultados obtidos, comprova-se a possibilidade de se utilizar o AG para calcular rotas entre pontos turísticos na região da Ibiapaba. Espera-se futuramente aumentar a quantidade de pontos turísticos mapeados na região, e portanto, com o auxílio deste algoritmo e de outras ferramentas, uma aplicação de recomendação de roteiros turísticos possa ser desenvolvida, com o intuito de levar maior comodidade e auxílio ao passeio dos turistas que visitam a região – otimizando seu tempo e dinheiro–, podendo elevar sua satisfação bem como sua estadia na Ibiapaba.

Referências

- Adhau, S., Shinde, S., Monghal, T., and Hole, V. (2018). Globetrotter: An optimal travel sequence generator. *2nd International conference on Electronics, Communication and Aerospace Technology (ICECA 2018)*. <https://ieeexplore.ieee.org/document/8474589>.
- Apache Friends (2020). Xampp apache + mariadb + php + perl. https://www.apachefriends.org/pt_br/index.html. Acessado em 23 de abril de 2020.
- Aydin, A. and Telceken, S. (2015). Comparison of three search algorithms for mobile trip planner for eskisehir city. *2015 International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*. <https://ieeexplore.ieee.org/document/7276791>.
- Azevedo, J. N. A. (2016). Turismo de aventura no planalto da ibiapaba: A oferta de serviços nas cidades de tianguá/ce e ubajara/ce. Mestre em gestão de negócios turísticos, Curso de Mestrado Profissional em Gestão de Negócios Turísticos. Disponível em <https://tinyurl.com/2959z23z>.
- Brasil, I. C. M. d. C. d. B. I. (2020). Clima. <https://tinyurl.com/59n9bbc5>. Acessado em 18 de março de 2020.

- Ceará, F. C. d. M. e. R. H. F. (2019). Municípios do ceará registram temperaturas mínimas em torno dos 16°C. <http://www.funceme.br/?p=5338>. Acessado em 20 de março de 2020.
- Chaves, A. A. (2005). Heurísticas híbridas com busca através de agrupamentos para o problema do caixeiro viajante com coleta de prêmios. Dissertação de mestrado, Pós-graduação em Computação Aplicada. Disponível em <https://tinyurl.com/4e4ktfc6>.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2012). *Algoritmos: Teoria e Prática*. Elsevier, Rio de Janeiro – RJ – Brasil, 3 edition.
- Dourado, G. (2019). Ecoturismo na serra da ibiapaba reserva experiências de conexão consigo e com a natureza. <http://abre.ai/bvjI>. Acessado em 30 de agosto de 2020.
- Google LLC (2020a). Google my maps. <https://www.google.com/maps/about/mymaps/>. Acessado em 30 de agosto de 2020.
- Google LLC (2020b). Web services - geocoding api. <https://tinyurl.com/9y7va3nd>. Acessado em 3 de abril de 2020.
- Hirata, R. T. (2018). Algoritmos de otimização para problemas de roteamento. Iniciação Científica (Unicamp). <https://tinyurl.com/vkyufksp>.
- Ibiapaba, R. M. d. (2020). Rota mirantes da ibiapaba. <https://www.rotamirantesdaibiapaba.com.br/>. Acessado em 18 de março de 2020.
- NetworkX (2019). Networkx - network analysis in python. <https://networkx.github.io/>. Acessado em 22 de agosto de 2020.
- Online, J. O. P. (2019). Da serra ao mar - chapada de ibiapaba - especiais o povo. <https://tinyurl.com/8fsrtyw6>. Acessado em 30 de agosto de 2020.
- PSF, P. S. F. (2020). Python. <https://www.python.org/>. Acessado em 20 de agosto de 2020.
- Rabahy, W. A. (2019). Análise e perspectivas do turismo no brasil. *Revista Brasileira de Pesquisa em Turismo (RBTUR)*, 14:1–13. <https://tinyurl.com/34fnu2nr>.
- Russell, S. and Norvig, P. (2013). *Inteligência Artificial*. Elsevier, Rio de Janeiro – RJ – Brasil, 3 edition.
- Silva, M. M. N. d., de Freitas Silva, M. N., and de Sousa Silva, N. G. (2017). Interiorização, regionalização e internacionalização do turismo no nordeste. *XV Simpósio Nacional de Geografia Urbana (SIMPURB)*. <https://tinyurl.com/3x8zfkzj>.
- Simões-Pereira, J. M. d. S. (2013). *Grafos e Redes: Teoria e Algoritmos Básicos*. Editora Interciência, Rio de Janeiro – RJ – Brasil, 1 edition.
- Tomé, L. M. (2018). Panorama do turismo no brasil e oportunidades para a região nordeste. *Caderno Setorial ETENE - Banco do Nordeste*, Ano 3, Nº 59. <https://tinyurl.com/td9st3nt>.
- Tripadvisor (2020). O que fazer: Serra da ibiapaba, ce. <https://tinyurl.com/5yka2npk>. Acessado em 18 de março de 2020.