



UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Leonardo Nunes Gonçalves

Software para planejamento de Redes IoT: Uma solução baseada em
Algoritmo Genético, Algoritmo de Múltiplas Tentativas e EPSO

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém – Pará – Brasil
2022

Leonardo Nunes Gonçalves

**Software para planejamento de Redes IoT: Uma solução baseada em
Algoritmo Genético, Algoritmo de Múltiplas Tentativas e EPSO**

Dissertação de mestrado apresentada ao
Programa de Pós-Graduação em Engenharia
Elétrica, da Universidade Federal do Pará, como
parte dos requisitos para a obtenção do título de
Mestre em Engenharia Elétrica na área de
Computação Aplicada.

Orientadora: Profa. Dra. Jasmine Priscyla Leite de Araujo

UFPA/ITEC/PPGEE
Campus Universitário do Guamá
Belém – Pará – Brasil
2022

Leonardo Nunes Gonçalves

**Software para planejamento de Redes IoT: Uma solução baseada em
Algoritmo Genético, Algoritmo de Múltiplas Tentativas e EPSO**

Dissertação de mestrado apresentada ao
Programa de Pós-Graduação em Engenharia
Elétrica, da Universidade Federal do Pará, como
parte dos requisitos para a obtenção do título de
Mestre em Engenharia Elétrica na área de
Computação Aplicada.

Conceito: _____

Belém, 20 de julho de 2022.

BANCA EXAMINADORA

Profa. Dra. Jasmine Priscyla Leite de Araújo

Orientadora – PPGEE / UFPA

Prof. Dr. Fabricio Jose Brito Barros

Avaliador interno – PPGEE / UFPA

Prof. Dr. Maria Emília de Lima Tostes

Avaliador interno – PPGEE / UFPA

Prof. Dr. Fabrício de Souza Farias

Avaliador Externo – Campus Universitário do Tocantins / UFPA-CAMETÁ

**Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBDSistema
de Bibliotecas da Universidade Federal do Pará**

Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)

-
- G635s Gonçalves, Leonardo Nunes.
 Software para planejamento de Redes IoT: Uma Solução
 baseada em Algoritmo Genético, Algoritmo de Múltiplas
 Tentativas e EPSO / Leonardo Nunes Gonçalves. — 2022.
 69 f.
- Orientador(a): Prof^ª. Dra. Jasmine Priscyla Leite de Araujo
 Dissertação (Mestrado) - Universidade Federal do Pará,
 Instituto de Tecnologia, Programa de Pós-Graduação em
 Engenharia Elétrica, Belém, 2022.
1. Internet of Things. 2. Planejamento. 3. Redes IoT. 4.
 Algoritmo Genético. I. Título.

“In Memoriam” de Tereza de Melo, Benedito Evilásio e Maria da Luz.

AGRADECIMENTOS

A Deus Su, por todas as proteções, orientações e concessões.

Meus pais Jorge e Iolete e avô Eliésio, pelo suporte e oportunidades que me forneceram, muitas vezes abdicando das suas vontades para promover meu desenvolvimento pessoal, acadêmico e profissional. Meu irmão Leandro que sempre esteve ao meu lado, compartilhando metas e objetivos, assim nossa competitividade nos engrandeceu. A afirmativa é verdadeira FAMÍLIA, É PODER.

Aos meus orientadores, Prof. Dra. Jasmine Priscyla Leite de Araujo e Prof. Dr. Fabrício de Souza Farias, pela compreensão, confiança e incentivo para realização deste trabalho. Aos mestres com carinho, meus mais sinceros agradecimentos.

Por último, mas não menos importante, aos meus amigos e colegas de trabalho, membros do Laboratório de Programação Extrema e do Laboratório de Comunicações e Telecomunicações, pela parceria e trocas de conhecimento e experiências, especialmente a Laciene Melo, Joiner Sá e Hugo Sousa pelas contribuições e nortes compartilhados neste e outros projetos desenvolvidos.

RESUMO

A Internet das Coisas (IoT, do inglês Internet of Things) permite o monitoramento ubíquo de ambientes através de sensores dispostos em determinada área de interesse. Tal coleta de dados gera um conteúdo sem precedentes de informações que são apresentadas a diferentes algoritmos que servem para auxiliar na tomada de decisões associadas a mobilidade urbana, economia, saúde, bem-estar, entre outras. Para garantir o sucesso dessa cadeia de comunicação, definida desde a coleta dos dados até a extração de decisões valorosas, se faz necessário a implantação de uma comunicação ponta a ponta. Para isso a IoT faz uso da tecnologia de comunicação Long Range (LoRa), que por sua vez garante a comunicação sem fio e sem custos entre os sensores instalados nos endnodes dispostos na área de interesse e os pontos de agregação de tráfego de dados instalados na área a ser monitorada, isto é, o gateway. Embora a solução seja prática, existem desafios de minimização dos custos associados a implantação do menor número de gateways na área a ser coberta, assim como, a tarefa de se planejar a rede IoT levando em consideração o posicionamento ótimo dos gateways. Diante deste contexto e para responder aos desafios impostos pelo planejamento de redes IoT, este trabalho tem como objetivo propor um software otimizador de planejamento de redes IoT baseado em Algoritmo Genético, Evolutionary Particle Swarm Optimization (EPSO) e algoritmo de Múltiplas Tentativas, com o intuito de minimizar o número de gateways e determinar a geolocalização para instalação dos mesmos, assim visando garantir a cobertura de todos os endnodes e seus respectivos sensores dispostos no campo.

Palavras-chave: Internet of Things, Planejamento, Redes IoT, Algoritmo Genético, EPSO.

ABSTRACT

The Internet of Things (IoT) allows the ubiquitous monitoring of environments through sensors arranged in a certain area of interest. Such data collection generates unprecedented content of information that is presented to different algorithms that serve to assist in decision-making associated with urban mobility, economy, health, well-being, among others. To ensure the success of this communication chain, defined from the collection of data to the extraction of valuable decisions, it is necessary to implement an end-to-end communication. For this, the IoT makes use of Long Range communication technology (LoRa), which in turn guarantees wireless and cost-free communication between the sensors installed in the endnodes arranged in the area of interest and the data traffic aggregation points installed in the area to be monitored, ie the gateway. Although the solution is practical, there are cost minimization challenges associated with deploying the fewest number of gateways in the area to be covered, as well as the task of planning the IoT network taking into account the optimal positioning of the gateways. Given this context and to respond to the challenges imposed by the planning of IoT networks, this work aims to propose an optimizing software for planning IoT networks based on Genetic Algorithm, Evolutionary Particle Swarm Optimization (EPSO) and Multiple Attempts algorithm, in order to minimize the number of gateways and determine the geolocation for their installation, thus aiming to guarantee the coverage of all endnodes and their respective sensors arranged in the field.

Keywords: Internet of Things, Planning, Network IoT, Genetic Algorithm, EPSO.

LISTA DE FIGURAS

Figura 1 – Ilustração do Arduino UNO R3.	22
Figura 2 – Ilustração do <i>Shield</i> Dragino LoRa.	23
Figura 3 - Formato de Mensagem CoAP.....	25
Figura 4 – Arquitetura redes IoT.	26
Figura 5 – Cruzamento um ponto de corte.	28
Figura 6 – Cruzamento um ponto de corte.	29
Figura 7 – Exemplo de Diagrama Entidade Relacionamento.....	34
Figura 8 - Modelo lógico do banco de dados do <i>Software</i> de Planejamento de Redes IoT.	37
Figura 9 - Esquema da metodologia de medição.....	40
Figura 10 - Dados medidos.....	41
Figura 11 – Login do <i>software</i> de planejamento de redes IoT.	43
Figura 12 – Dashboard do <i>software</i> de planejamento de redes IoT.	44
Figura 13 – Módulo UFPA Manual.....	45
Figura 14 – Adição manual de <i>gateway</i>	46
Figura 15 – Adição manual dos parâmetros do <i>gateway</i>	47
Figura 16 – Gráficos resultantes. (a) Perda de sinal. (b) RSSI. (c) Dispositivos <i>end nodes</i> discretos.....	48
Figura 17 – Módulo Otimização Geral.....	49
Figura 18 – Processo de adição de entradas do sistema. (a) Carregamento de rota. (b) Definição da área de otimização. (c) Definição das áreas de exclusão.....	50
Figura 19 – Exemplo de arquitetura para otimização.....	51
Figura 20 – Resultados da otimização por múltiplas tentativas. (a) Posicionamento de gateways. (b) Valores de perda calculados. (c) Valores de RSSI calculados.....	52
Figura 21 – Resultados da otimização por algoritmo genético. (a) Posicionamento de gateways. (b) Valores de perda calculados. (c) Valores de RSSI calculados.....	54
Figura 22 – Resultados de convergência do algoritmo genético.	55
Figura 23 - Módulo EPSO.....	57
Figura 24 - Resultado da simulação por EPSO	58
Figura 25 - Exemplo de solução/indivíduo EPSO.....	58

LISTA DE QUADROS

Quadro 1 – Comparação das Tecnologias de Comunicação.	16
Quadro 2 – Principais aspectos de segurança dos protocolos de redes IoT.	24
Quadro 3 – Protocolos utilizados nas camadas do modelo TCP/IP.	24
Quadro 4 – Requisitos funcionais do <i>software</i> de planejamento de redes IoT.....	35
Quadro 5 – Requisitos não funcionais do <i>software</i> de planejamento de redes IoT.....	36
Quadro 6 – Parâmetros utilizados nas campanhas de medições.....	39
Quadro 7 – Coeficientes calculados.	42

LISTA DE SIGLAS

4G	Quarta Geração de telefonia móvel
IoT	Internet of Things
LoRa	Long Range
QoS	Quality of Service
SNR	Signal to Noise Ratio
RSSI	Received Signal Strength Indicator
AG	Algoritmo Genéticos
RAW-Manager	Reliable and Available Wireless
CoAP	Constrained Application Protocol
MQTT	Message Queue Telemetry Transport
PDR	Packet Delivery Ratio
SF	Spreading Factor
MHz	megahertz
dBm	decibel miliwatt
mW	miliwatt
kbps	quilobit por segundo
TCP/IP	Transmission Control Protocol / Internet Protocol
ISO	International Organization for Standardization
IP	Internet Protocol
HTTP	Hypertext Transfer Protocol
UDP	User Datagram Protocol
6LoWPAN	Over Low Power Wireless Personal Area Networks
Wi-Fi	Wireless Fidelity

LPWAN	Low Power Wide Area Network
PSO	Particle Swarm Optimization
EPSO	Evolutionary Particle Swarm Optimization
ER	Entidade relacionamento
DER	Diagrama de entidade relacionamento
FI	Floating Intercept
CI	Close In
Hz	hertz
LCT	Laboratório de Computação e Telecomunicações
UFPA	Universidade Federal do Pará
dBi	Decibéis isotrópicos
MQL	Mínimos Quadrados Lineares

SUMÁRIO

1	Introdução	15
1.1	Justificativa.....	16
1.2	Trabalhos Relacionados	17
1.3	Objetivos	19
1.4	Organização do Trabalho	20
2	Referencial Teórico	21
2.1	Arquitetura de Rede IoT.....	21
2.2	Algoritmos de Planejamento	27
2.2.1	Algoritmos Genéticos	27
2.2.2	EPSO	29
2.2.3	Múltiplas Tentativas	30
2.3	Modelos de Propagação.....	30
2.4	Engenharia de Software.....	32
2.4.1	Requisitos	33
2.4.2	Projeto de Banco de Dados.....	33
3	Software Para Planejamento de Redes IoT.....	35
3.1	Introdução.....	35
3.2	Requisitos	35
3.3	Modelagem do Banco de Dados.....	37
3.4	Medições e Modelagem dos Modelos de Propagação.....	39
3.4.1	Análise de Dados	41
3.5	Resultados	43
3.5.1	Credenciamento e Primeiro acesso.....	43
3.5.2	UFPA Manual.....	44
3.5.3	Otimização Geral.....	49
3.5.4	Resultados Matemáticos	52

3.5.5	Otimização EPSO	56
4	Conclusão	60
	Referências	62

1 INTRODUÇÃO

O advento da *Internet of Things* (IoT) possibilitou a extensão do uso de Internet para um número indefinido de dispositivos/objetos, tendo como princípio fundamental a comunicação sem fio. De acordo com Lopes (2019), os elementos fundamentais de uma rede IoT são os *end devices* / *end nodes*, dispositivos conectados à rede por meio de algum protocolo de comunicação, e os gateways, responsáveis em receber os sinais enviados pelos *end nodes* e redirecioná-los para a Internet.

Essa rede de objetos conectada foi idealizada com intuito de facilitar o dia a dia das pessoas, empresas e cidades, a partir da coleta de dados sempre presentes, mas negligenciados em diferentes ambientes. Objetivando modificar o *status quo* das cidades através do monitoramento ambiental e urbanístico, promovendo planejamento urbano para redução da poluição e emissão de gases carbônico e metano; e a partir da otimização da mobilidade urbana facilitando a movimentação nesses espaços geográficos, através do desenvolvimento de soluções que ajudem a população a se deslocar de forma rápida, segura e eficiente.

Essa coleta de informações gerou a necessidade da criação de bancos de dados para guardar as informações em nuvem, assim como, uma demanda de desenvolvimento de soluções de redes de comunicação sem fio visando garantir a comunicação entre os dispositivos e sensores instalados no campo para aquisição dos dados no ambiente e seu respectivo banco de dados.

Por conta disso, existem diversos projetos sendo realizados com esta temática, como o Projeto de Pesquisa e Desenvolvimento (P&D) do Sistema Inteligente Multimodal da Amazônia (SIMA) que tem como principal objetivo “investir em pesquisa para criar soluções que melhorem a qualidade de vida das pessoas e contribuam para a conservação do meio ambiente” (NORTE ENERGIA, 2022).

Esse projeto foi planejado e arraigado na Cidade Universitária Professor José Silveira Netto da Universidade Federal do Pará (UFPA), em Belém. Inserindo uma infraestrutura de dispositivos interconectados no campus universitário, possibilitando que este seja considerado um *smart campus* (campus inteligente). Diante deste contexto, esta dissertação apresenta uma

parte das soluções desenvolvidas para o projeto SIMA, sendo a parte apresentada a que diz respeito ao *software* voltado para o planejamento de redes IoT.

1.1 Justificativa

A definição da área de cobertura de uma determinada rede sem fio, de forma clássica, é realizada através do estabelecimento do valor da potência efetivamente irradiada pelo transmissor, e o limiar de sensibilidade do receptor (nível mínimo que o equipamento consegue decodificar a mensagem). A partir disso é possível calcular a distância entre o transmissor e o receptor (através da execução de medidas no ambiente ou aplicação de modelos de perda de propagação), consequentemente também é possível se obter o raio de cobertura do sistema sem fio. E assim, definir pontos equidistantes para tentar delimitar a maior área de cobertura da rede (SCHILLER, 2003).

Diversas tecnologias de comunicação foram desenvolvidas para prover estas redes, o Quadro 1 apresenta algumas tecnologias de comunicação que normalmente podem ser utilizadas na implementação de uma rede IoT.

Quadro 1 – Comparação das Tecnologias de Comunicação.

PROTOCOLO	ALCANCE	FREQUENCIA
WiFi	50m	2.4/5 GHz
3G/4G	35/200km	1900/2100/2500 MHz
LoRaWAN	2/5km	Sub-GHz

Fonte: Adaptado de Lopes (2019).

Através do Quadro 1 percebe-se que a tecnologia de comunicação escolhida impacta diretamente no alcance da rede, sendo desta forma um importante parâmetro de configuração a ser definido ao se implementar redes IoT. Por conta disso, as tecnologias de 3G/4G e LoRaWAN são frequentemente utilizadas. (LEITE, MARTINS e URSINI, 2017)

A tecnologia Long Range (LoRa) é empregada em projetos de IoT que demandam conectividade sem fio de longo alcance e baixo consumo energético, estando no Brasil operando em radiofrequência nas faixas disponíveis no intervalo de 915-928 Mhz. Essa tecnologia é também a base para construção de redes LoRaWAN, que por sua vez, possibilitam o suporte para muitos dispositivos e uma cobertura estendida de 10-15 km, em áreas rurais, e de 2-5 km, em áreas urbanas (LOPES, 2019).

Redes LoRaWAN ajudaram a difundir o conceito de *smart campus* que segundo Bandeira e Araújo Neto (2020) é um conceito embasado nas definições de *smart city* (cidades inteligentes), que generalizando, são cidades e campus que utilizam da tecnologia da informação e comunicação para melhorar sua habitabilidade, capacidade de trabalho e sustentabilidade.

Embora a relevância da investigação de redes IoT e *smart campus*, Jin, Gubbi e Murusic (2013) ressaltam que existem diversos desafios presentes no planejamento de uma rede IoT, por exemplo, os custos, a gestão de dados, e a manutenção e gerenciamento do sistema de controle. Por conta disso, Afzal et al. (2019) discorrem que a utilização dos recursos destinados à implantação deve ser otimizada. Já Silva (2018) destaca que ao solucionar estes desafios, objetiva-se trazer valor efetivo na implementação dos projetos de IoT, isto é, tornando as soluções realísticas e com capacidade de replicabilidade.

Silva (2018) ainda enfatiza que solucionando os desafios, a implantação de projetos de redes IoT conseguirá promover uma análise mais fidedigna às informações geradas pela rede, fazendo com que a análise do conhecimento, possa antever ou até mesmo prever uma solução para os negócios e serviços prestados à sociedade, para isso é necessário garantir a conectividade entre todos ou a maioria dos dispositivos que compõem uma rede IoT. Por conta dessa premissa a execução do planejamento de redes IoT considerando o dimensionamento da quantidade de recursos de infraestrutura é de suma importância para implantação destas redes.

1.2 Trabalhos Relacionados

As *smart cities* ganharam ênfase a partir do momento em que um ambiente inteligente foi definido como um ambiente capaz de adquirir e aplicar conhecimento sobre o espaço e seus habitantes tendo objetivo de melhorar a vivência nesse ambiente, dessa forma Neves, Sarmanho, Júnior e Meiguins (2017) enfatizam que o desenvolvimento de soluções para os problemas de gestão e mobilidades encontrados, podem ser primeiramente solucionados a partir de estudos em ambientes mais restritos, entretanto, não menos complexos, tais como os *smart campus* que implementam gradualmente um sistema interconectado capaz de gerenciar seus recursos como: monitoramento de energia e de sistemas de transporte, localização e outros. (FERREIRA e ARAÚJO, 2022).

Para que essas soluções sejam desenvolvidas, primeiramente é necessário prover a capacidade de comunicação entre os dispositivos que compõe uma rede IoT, a qual acaba

englobando os conceitos de *smart cities* e *smart campus*, na literatura podem ser encontrados diversos trabalhos que permeiam este campo de estudo em busca de soluções que possam tornar o processo de implementação de redes IoT mais rápido e com menores custos operacionais. Nesta seção alguns destes trabalhos serão apresentados com intuito de fornecer embasamento teórico para análise e implementação de um software capaz de simular e prover uma arquitetura de rede eficaz no que diz respeito a comunicação de seus dispositivos.

A demanda por comunicação entre dispositivos IoT é uma crescente contínua, Soares (2019) apresenta um trabalho que tem como o objetivo avaliar cenários de propagação de sinal Long Range (LoRa), por conta disso, efetuou campanhas de medição para que as informações coletadas alimentassem um modelo adaptado de propagação baseado na técnica de Hata, esperando que esse modelo seja utilizado como base para planejamento de redes IoT.

O método apresentado é constituído de cinco etapas, (1) pré-processamento, (2) implementação do modelo de propagação Okumura-Hata, (3) otimização do modelo, onde seus parâmetros constates são variados a partir de um Algoritmo Genético, (4) avaliação estatística, já que, paralelamente as medidas coletadas, possuem um componente estocástico que é induzida pela aplicação da técnica de Bootstrap, que por sua vez, implica que a cada intervalo de medida coletada um processo de reamostragem dos intervalos de confiança deve ser executado. E (5) Unificação das partes determinísticas e estocásticas, assim gerando um modelo expresso graficamente.

Os parâmetros de avaliação para modelagem do novo modelo consideram as necessidades vislumbradas por uma rede 5G com suporte de cobertura LoRa-IoT em questões de Quality of Service (QoS) e propagação, entre os quais destacam-se Signal to Noise Ratio (SNR), Received Signal Strength Indicator (RSSI), taxa de transferência, latência e entre outros. Foram apresentados os resultados para sete cenários de testes em campo, além dos resultados obtidos a partir da otimização do modelo de Okumura-Hata utilizando Algoritmo Genético (AG). Após análise, todos os resultados compuseram a base para implementação do modelo de propagação escolhido para as cidades de Patos de Minas, Uberlândia e Brasília.

Já Venceslau (2021) propõe em sua dissertação um modelo arquitetural de comunicação e o desenvolvimento do sistema Reliable and Available Wireless (RAW-Manager) para gerenciamento dos dispositivos que se conectam aos gateways, oferecendo suporte à configuração e monitoramento da rede. A arquitetura utiliza os protocolos CoAP e Message Queue Telemetry Transport (MQTT), sendo respectivamente responsáveis pelo trecho

de comunicação que não suporta conexão com a Internet e entre o gateway centralizador até as aplicações finais de IoT. A comunicação é provida por meio do modelo *publish-subscriber* do protocolo MQTT, sendo a avaliação do desempenho da arquitetura realizada a partir da qualidade da comunicação aferida com base nos valores de *Packet Delivery Ratio* (PDR) e RSSI entre os nós finais e os gateways da rede. Por fim, o autor espera que o sistema permita a construção de uma rede de sensores sem fio apta a receber outros dispositivos de sensoriamento, que possam monitorar distintos fenômenos de um ambiente de *smart campus*.

Matni et al. (2020) propõem um modelo de posicionamento de *gateways* LoRaWAN, denominado DPLACE, constituído de três etapas: (1) determinação da quantidade de *gateways* para implantação de uma rede IoT, fundamentado no algoritmo Fuzzy C-Means; (2) estimação da localização do *gateway*, a partir do método de estatísticas de Gap; e (3) cálculo da localização final do *gateway* considerando um conjunto de *gateways* para diferentes cenários de rede. Os autores destacam que os resultados dessa proposição são estimados de forma que os *gateways* que irão compor a rede IoT desconhecem o número de *end nodes* que serão atrelados à mesma, além disso concluíram que as simulações com esta metodologia melhoram em 8% os índices de PDR, se comparado, com um estudo realizado anteriormente pelos autores.

1.3 Objetivos

O objetivo geral desta dissertação é o desenvolvimento de um *software* que auxilie profissionais no planejamento de Redes IoT para *smart cities* / *smart campus*, fundamentado em algoritmos de otimização.

Para alcançar o objetivo geral, foram definidos os seguintes objetivos específicos:

- Realizar medições *in loco* para análise de dados;
- Inferir níveis de limiares para cada modelo de propagação integrado ao *software*;
- Levantar requisitos funcionais e não funcionais do *software*;
- Realizar a modelagem lógica e física do banco de dados do *software*;
- Desenvolver algoritmos de otimização para o planejamento da rede IoT;
- Criação de um painel *web* para realização das simulações de otimização.

1.4 Organização do Trabalho

Esta dissertação é composta por 4 Capítulos, mais as referências bibliográficas e Apêndice. Além deste Capítulo de introdução, o restante deste trabalho está estruturado da seguinte forma:

Capítulo 2 – Referencial Teórico: apresenta toda parte teórica e conceitos sobre as tecnologias e ferramentas utilizados nesta pesquisa;

Capítulo 3 – Software Norte Rotas: apresenta o desenvolvimento, testes e resultados do *software* para o planejamento de redes IoT;

Capítulo 4 – Conclusão: apresenta as considerações finais, trabalhos futuros e as produções acadêmicas publicadas durante esta pesquisa.

2 REFERENCIAL TEÓRICO

Neste Capítulo os conceitos teóricos deparados na literatura sobre os métodos e tecnologias utilizadas neste trabalho são explicitados. Na Seção 2.1, são apresentados os fundamentos da arquitetura de uma rede IoT. A Seção 2.2 versa sobre os conceitos e abordagens dos algoritmos de planejamento utilizados no *software*. A Seção 2.3 aborda os modelos de propagação empregados, enquanto na Seção 2.4 são apresentados os métodos de engenharia de software utilizados para desenvolvimento deste trabalho.

2.1 Arquitetura de Rede IoT

Nas primeiras vezes em que o termo IoT era utilizado, Dodson (2003) fazia referência a implementação de uma rede de comunicação por Rádio Frequência (RFID), que ganhava visibilidade a partir de sua redução de tamanho e custos de produção, e tentava substituir o código de barras global.

Atualmente, Madakam et al (2015) referem-se à IoT como uma rede interconectada de dispositivos inteligentes capazes de compartilhar informações, dados e recursos diante das eventuais modificações no ambiente que os engloba. Para Menezes e Costa (2019), em uma rede IoT de alto nível se identifica a presença de três camadas entrelaçadas: *Hardware*, a camada de infraestrutura da rede, *Middleware*, a camada de software, e *Presentation*, a camada que possibilita a visualização dos dados e interação com o usuário.

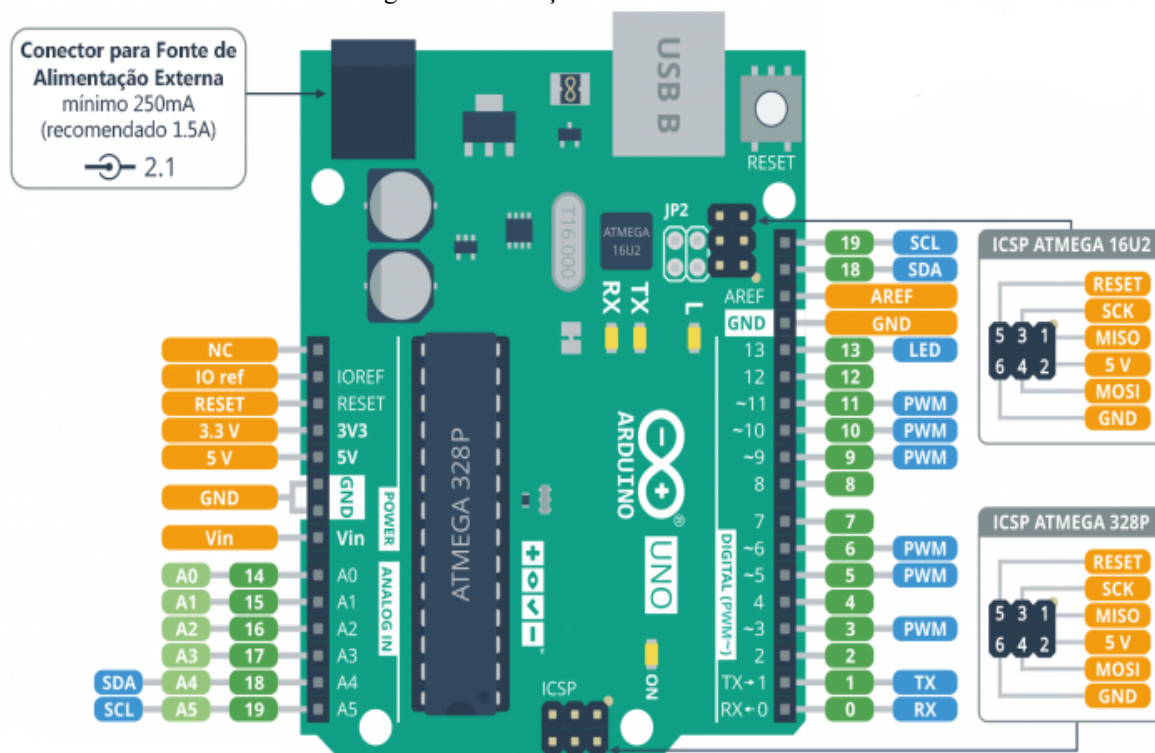
Nesta subseção serão conceitualmente elucidados alguns componentes das camadas que constituem redes de IoT modernas, sendo apresentados de acordo com a ordem de entrelaço acima mencionada.

Para provimento da infraestrutura de uma Rede IoT, existem dois itens essenciais que merecem ênfase e destaque, sendo eles os *gateways* e *end nodes*. Estes equipamentos atuam de forma colaborativa comunicando-se de forma contínua a partir de protocolos específicos, que objetivam reduzir o consumo de energia entre os dispositivos, assim promovendo maior alcance de sinal. Por conta disso, Bor et al. (2016) destacam que a tecnologia LoRa permite tráfego das informações coletadas por diversos dispositivos alocados em uma área geográfica.

Para Proença (2022) *end nodes* são módulos incumbidos da captação das informações aferidas pelos sensores, esses módulos possuem um grau de classificação, em que os dispositivos de classe A são comumente implementados por causa do baixo consumo energético e comunicação bidirecional. Além disso, segundo Bor et al. (2016) quando estes módulos são atribuídos em uma rede LoRaWAN há um ganho de performance na transmissão do sinal, haja vista que estes dispositivos permitem utilização de diferentes fatores de espalhamento, do inglês, *Spreading Factor* (SF).

Ana (2016) ressalta que o desenvolvimento de um sistema deve começar pela avaliação da arquitetura, para que os componentes desta permitam o alcance do objetivo final, por isso é frequente a adoção de plataformas de desenvolvimento que oferecem recursos de hardware que possibilitem o desenvolvimento de sistemas com rapidez e baixos custos. A Figura 1 ilustra um dos dispositivos eletrônicos mais popular do mundo, o Arduino na versão UNO R3.

Figura 1 – Ilustração do Arduino UNO R3.

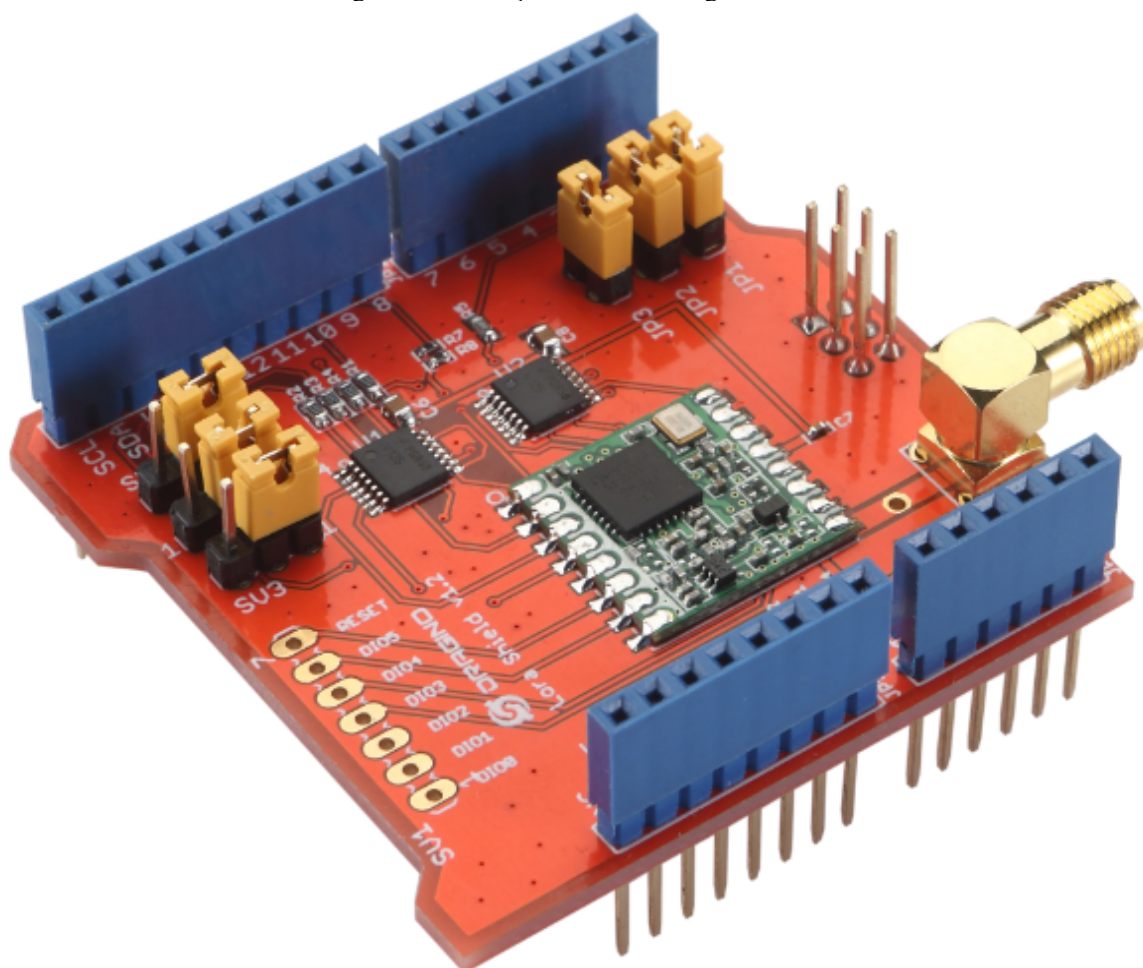


Fonte: Curso Arduino (2022)

Devido sua compatibilidade com inúmeros sensores, o Arduino é utilizado em diversos projetos e aplicações. Este foi criado em 2005 por um grupo de pesquisadores com objetivo de ser um facilitador no ensino de eletrônica, e para isso disponibiliza um microcontrolador ATMEL ATMEGA328 de 8 bits operando em 20 MHz, 28 pinos dos quais 23 podem ser

utilizados como sinais de entrada ou saída (EMBARCADOS, 2013). Para construir um dispositivo capaz de realizar comunicação bidirecional, pode-se adicionar ao Arduino um *shield*, conforme o ilustrado na Figura 2, o que possibilita a adição de sensores tais como: temperatura, luz, umidade, presença e outros, além de permitir a integração com a tecnologia LoRa Wireless.

Figura 2 – Ilustração do *Shield* Dragino LoRa.



Fonte: Dragino Shield (2022)

Atualmente, o *shield* Dragino Lora presente no mercado opera na frequência de 915 MHz, com constante de saída de rádio frequência +20 dBm – 100 mW, potência +14 dBm, *bit rate* programável até 300 kbps, possui alta sensibilidade de -148 dBm, e aceita modulações FKS, GFSK, MSK, GMSK, LoRaTM e OOK.

A utilização destes módulos é garantida por meio de diferentes protocolos de comunicação, que podem ser considerados como a primeira etapa da implementação da camada de *middleware*. Para Estevam e Guedes (2019) protocolo é um tipo de linguagem que permite

que dois ou mais módulos troquem informações entre si, através da definição de regras para que um módulo receptor “entenda” as informações enviadas por um módulo transmissor.

Em projetos de redes IoT, o protocolo TCP/IP é um dos mais utilizados. Para Santo e Ordoñez (2018), uma das razões para essa escolha se deve ao fato de o TCP/IP oferecer segurança a partir do suporte à integridade, autenticidade e privacidade dos dados. O Quadro 2 apresenta o detalhamento dos requisitos de segurança.

Quadro 2 – Principais aspectos de segurança dos protocolos de redes IoT.

Tipo	Descrição
Integridade	Dados provenientes não podem ser reescritos, assegurando que não haja manipulação da informação transmitida.
Autenticidade	A identificação única de um módulo nunca poderá ser modificada.
Privacidade	Não permitir que o histórico de informações transmitidas seja acessado, sem que haja uma verificação da autenticidade do requisitante.

Fonte: elaborado pelo autor.

Além das características de segurança, de acordo com Muniz (2019) o protocolo TCP/IP tornou-se mais popular devido a necessidade de padronização na comunicação entre *hardwares* diferentes. Além disso quando comparado ao modelo de comunicação estabelecido pela *International Organization for Standardization* (ISO), este protocolo possui outra vantagem relevante que é a sua garantia de entrega de dados de uma ponta a outra da aplicação.

Para Tanenbaum (2003), sua composição em quatro camadas, isto é Aplicação, Transporte, Internet e Rede, e seu modo de operação único na camada de Rede (sem conexão), possibilita maior eficácia para protocolos simples de solicitação/resposta, o que também justifica sua popularidade.

Rotta et al. (2017) defendem que a comunicação dos módulos IoT foi fundamentada sobre o modelo TCP/IP, já que, as aplicações IoT eram desenvolvidas inicialmente para Internet. O Quadro 3 apresenta a correlação entre os protocolos utilizados nas camadas do modelo TCP/IP e do modelo IP estabelecido para IoT.

Quadro 3 – Protocolos utilizados nas camadas do modelo TCP/IP.

Camadas	Internet TCP/IP	IP IoT
Aplicação	http / https / ftp / ssh	CoAP / MQTT
Transporte	TCP / UDP	UDP
Internet	IPv4 / IPv6	6LowPAN

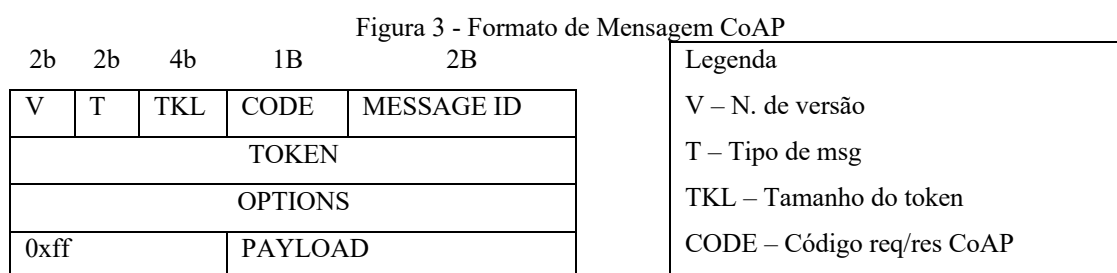
Rede	IEEE 803.2 Ethernet / 802.11 Wi-Fi	IEEE 802.11 / 802.15
------	------------------------------------	----------------------

Fonte: Rotta et al. (2017).

Cada camada da arquitetura de rede é responsável pela execução de uma função bem definida objetivando que o processo de comunicação entre os módulos obtenha sucesso nos mais diversos ambientes. Neste contexto, Tanenbaum (2003) elucida que a camada de aplicação é responsável por englobar todos os protocolos de mais alto nível; já a camada de transporte deve permitir que os módulos conectados, sejam eles, de origem ou destino mantenham uma conversação direta; na camada de Internet se deve possibilitar que os módulos de origem conectados enviem pacotes de qualquer rede, garantindo que estes pacotes trafeguem independentemente até o módulo de destino correspondente; e a camada de rede deve possibilitar a conexão à rede a partir de qualquer protocolo que possibilite o envio de pacotes IP.

Dessa forma, segundo Embarcados (2022) “a natureza adaptável e confiável inerente do IP o torna aceitável para transmissão entre módulos IoT”, e por conta disso alguns dos principais protocolos de IoT: CoAP, MQTT, 6LoWPAN, Wi-Fi, e LoRaWAN, devem ser considerados como alternativa de comunicação.

Ugrenovic e Gardasevic (2015) introduzem que o protocolo CoAP utiliza uma arquitetura REST, onde as funcionalidades do *Hypertext Transfer Protocol* (HTTP) foram modificadas para execução sobre o protocolo *User Datagram Protocol* (UDP) simples, sem conexão. Arvind e Narayanan (2019) apresentam a estrutura das mensagens que são trafegadas entre os módulos finais, conforme apresentado no Quadro 3.



Fonte: adaptado de DA PORCIÚNCULA, 2018

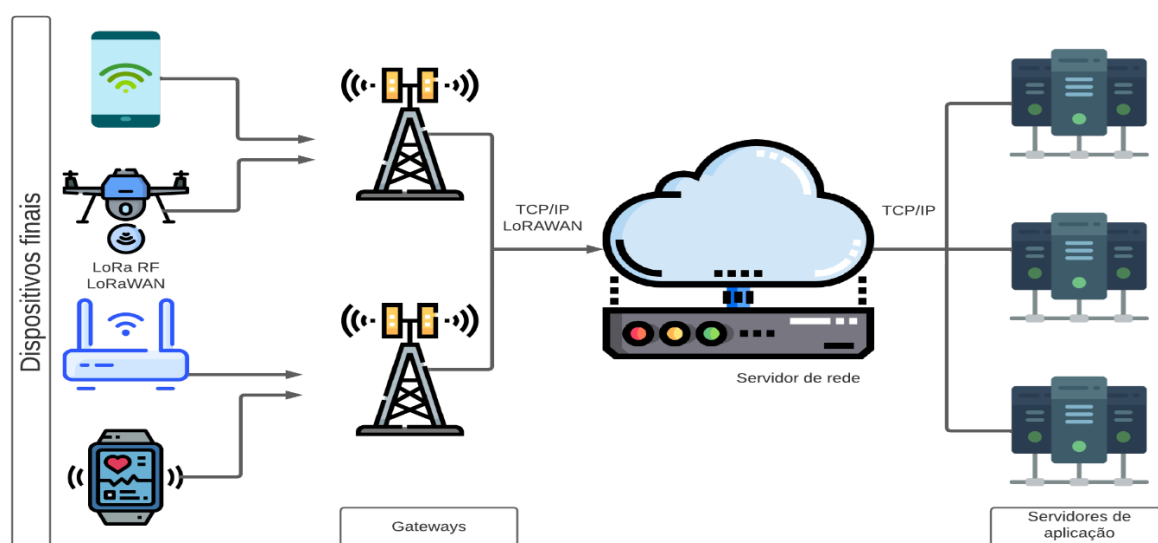
A estrutura de mensagens do protocolo CoAP é codificada em formato binário, onde cada mensagem possui um cabeçalho de comprimento fixo, um campo de opções e um campo de carga útil que possui comprimento variável e, de acordo com Rotta et al. (2017) “apresenta menor consumo computacional e energético”

Para Muniz et al. (2019), o MQTT é um protocolo simples e leve para tráfego de mensagens que utiliza o protocolo TCP para realizar atividades de “publicação/assinatura” (*publish/subscribe*). Esse protocolo foi projetado para módulos restritos, conectados a uma rede de baixa largura de banda e alta latência. O autor ainda conceitua que o protocolo *Over Low Power Wireless Personal Area Networks* (6LoWPAN) faz uma proposição de “encapsular o protocolo IPv6 para redes sem fio de curto alcance, utilizando o padrão IEEE 802.15.4”, dessa forma a principal tarefa deste protocolo é realizar a compressão do cabeçalho e mensagem enviada, além de permitir a fragmentação e reagrupamento dos pacotes transmitidos pela rede.

Provavelmente o protocolo de comunicação mais conhecido seja o WiFi, o qual é definido pela Cisco (2002) como uma tecnologia de rede que através de um roteador sem fio realiza a conexão de computadores, dispositivos móveis e outros equipamentos com a Internet, possibilitando que estes equipamentos troquem informações entre si. Embarcados (2022) elucida que este protocolo utiliza uma topologia de rede em estrela, onde o roteador é um ponto de acesso que pode ser utilizado como *gateway* para Internet. O padrão comumente utilizado é o 802.11-b/g/n, que por sua vez opera na frequência de 2.4 GHZ e atua em uma área em torno de 50 metros.

Segundo Tutui (2019), LoRaWAN é um padrão de arquitetura de rede que objetiva padronizar a implementação das *Low Power Wide Area Network* (LPWAN), a partir da construção de uma camada física baseada em modulação de Rádio Frequência por meio da tecnologia LoRa. Essa arquitetura é dividida em quatro partes: *end nodes*, *gateways*, *network server* e *application server*, conforme apresentado na Figura 4.

Figura 4 – Arquitetura redes IoT.



Fonte: Adaptada de LORA ALLIANCE TECHNICAL MARKETING WORKGROUP (2015).

De forma resumida, Da Silva et al. (2021) ressaltam que nessa arquitetura os *end nodes* enviam informações coletadas, por meio de sensores em que estão conectados, para todos os *gateways* ao alcance. Estes *gateways* conectam-se com a Internet a partir do protocolo IP e transmitem as informações recebidas para os *network server* que têm a responsabilidade de rotear as informações para a *application server* (camada de *presentation*). Além disso, os autores enfatizam que os *network server*, podem ser configurados para comunicação de *uplink*, com tráfego de informações na direção dos sensores para o aplicativo ou *downlink*, com tráfego na direção do aplicativo para os sensores.

2.2 Algoritmos de Planejamento

Middlewares IoT, de acordo com Pinto (2022), são definidos como um conjunto de pacotes ou *software* que operam na interface entre a infraestrutura física e a aplicação final. Normalmente executando alguma atividade em “segundo plano” que fornece informações processadas por meio de algum algoritmo.

Geralmente estes algoritmos são fundamentados em meta heurísticas desenvolvidas com o objetivo de constituírem técnicas baseadas em operações suficientemente gerais para serem aplicáveis à maior parte dos problemas combinatórios, em geral fundamentam-se em conceitos relativamente genéricos.

2.2.1 Algoritmos Genéticos

A partir dos estudos de Darwin, que explica a mudança das espécies considerando o princípio de seleção natural, e Mendel que introduzia como surgia variabilidade genética dos indivíduos, foi possível modelar algoritmos que simulem o processo de evolução natural.

Algoritmos genéticos são técnicas de busca/otimização baseados em mecanismos de seleção natural e genética, que objetivam percorrer o espaço de soluções e encontrar uma solução próxima da solução ideal de um determinado problema. Estes algoritmos são definidos a partir de um conjunto de inicial de indivíduos (população), ou seja, um conjunto de potenciais soluções para o problema; que quando submetido a um *loop*, genericamente, transforma estes indivíduos a partir dos operadores de: seleção, combinação e mutação; até que uma condição de parada seja satisfeita. (GASPAR-CUNHA & TAKAHASHI & ANTUNES, 2012)

O operador de seleção é responsável por selecionar os indivíduos mais aptos de uma população, inferindo numa população intermediária, a partir de então são aplicados os

operadores de reprodução para gerar novos indivíduos. A seguir, alguns métodos de seleção são apresentados.

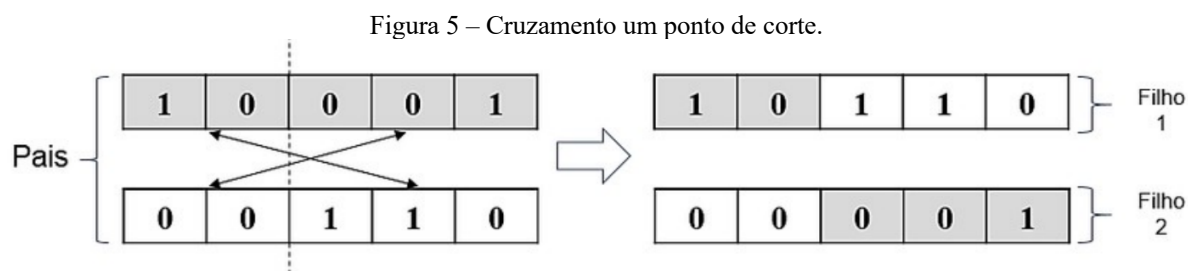
- Seleção por classificação: todos os indivíduos da população são ordenados de acordo com seus valores de aptidão (*fitness*), dessa forma sua probabilidade de escolha é atribuída conforme a posição que ocupam.

- Seleção por roleta: uma seleção probabilística é realizada de modo que cada indivíduo possua chances de sorteio proporcionais a seus respectivos valores de aptidão (*fitness*), ou seja, quanto maior o desempenho de um indivíduo, maior é a chance de ser selecionado para a próxima geração.

- Seleção por torneio: promove-se um torneio entre um grupo de n indivíduos, sendo $n > 2$, dessa forma o indivíduo que possuir o maior valor de aptidão é selecionado para compor a nova geração enquanto os demais são descartados. Este processo termina quando se realiza uma quantidade de torneios igual ao tamanho da população.

O operador de combinação é responsável pela recombinação de características dos pais durante a reprodução, este operador extrai de diferentes indivíduos seus melhores genes e combina-os para formar indivíduos com melhores características. Este operador também apresenta variações, onde as mais utilizadas são:

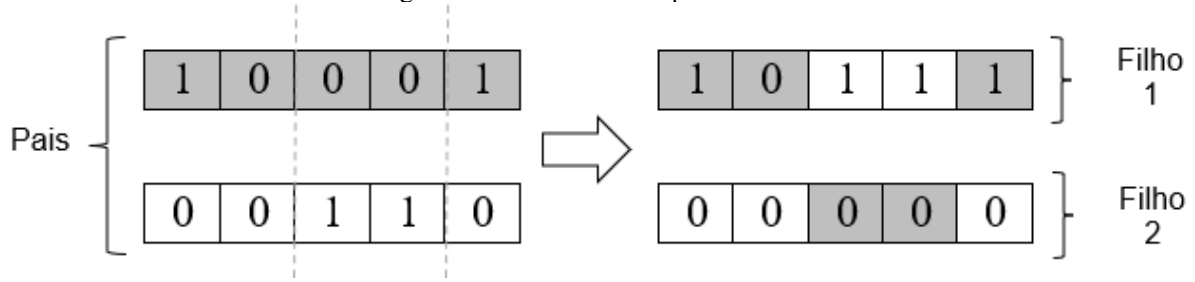
- Um ponto: um ponto de cruzamento é escolhido e a partir deste ponto as informações genéticas dos pais são trocadas, conforme apresentado na Figura 5. As informações anteriores a este ponto em um dos pais são ligadas às informações posteriores a este ponto no outro pai.



Fonte: elaborada pelo autor.

- Multipontos: é uma generalização da ideia de troca de material genético através de pontos, onde muitos pontos de cruzamento podem ser utilizados, conforme apresentado na Figura 6.

Figura 6 – Cruzamento um ponto de corte.



Fonte: elaborada pelo autor.

- Uniforme: não utiliza pontos de cruzamento, mas determina através de um parâmetro global, qual a probabilidade de cada variável ser trocada entre os pais.

O operador de mutação é necessário para a introdução e manutenção da diversidade genética da população, onde altera-se arbitrariamente um ou mais genes de um indivíduo, garantindo que as características que eventualmente desapareçam tenham a possibilidade de reaparecer em gerações futuras. Este operador apresenta diversas variantes, como por exemplo mutação por codificação real, permutações e máquinas de estado finito.

2.2.2 EPSO

Kennedy e Eberhart (1995) apresentaram um algoritmo de otimização fundamentado no comportamento social de animais, o *Particle Swarm Optimization* (PSO) tendo por objetivo encontrar a solução ótima, em um espaço de busca, a partir da troca de informações entre os indivíduos de uma população estabelecendo a trajetória que cada um destes indivíduos deve tomar no espaço. De acordo com Pacheco (2016) este algoritmo é matematicamente elucidado a partir das Equações 1 e 2, respectivamente.

$$v_{k+1} = wv_k + c_1r_1(p_{best_k} - x_k) + c_2r_2(g_{best} - x_k) \quad (1)$$

$$x_{k+1} = x_k + v_k \quad (2)$$

onde numa população com K partículas, v_{k+1} e x_{k+1} respectivamente, são a velocidade e a posição de uma partícula. w o coeficiente de inércia, p_{best_k} a melhor posição conhecida da partícula k , g_{best} a melhor posição conhecida de todas as partículas, c_1 e c_2 constantes de aceleração referentes ao melhor individual e global, por fim r_1 e r_2 números aleatórios extraídos do intervalo $[0,1]$.

Bernardes (2013) elucida que nesse algoritmo, cada solução candidata é uma partícula que repetidamente se movimenta no espaço de busca à procura de regiões que possuam valores adequados de maximização ou minimização da função de avaliação, dessa forma a posição de cada partícula é modificada utilizando informações compartilhadas entre o enxame (todas as soluções candidatas), implicando na mudança de cada partícula para uma posição em que ela e o enxame conseguiram o melhor valor resultante da função avaliação.

A partir da proposição realizada por Kennedy e Eberhart (1995), Zhang e Ishikawa (2008) explicam que o *Evolutionary Particle Swarm Optimization* (EPSO), um algoritmo variante do PSO, é constituído de duas partes: um *loop* externo implementando métodos de algoritmos evolutivos e um *loop* interno utilizando os princípios do PSO; o objetivo principal da abordagem é encontrar valores apropriados para os parâmetros do PSO resolvendo de forma eficiente qualquer problema de otimização.

2.2.3 Múltiplas Tentativas

Uma proposta extremamente básica e clássica utilizada para solução de diferentes problemas computacionais é a implementação de um algoritmo que verifica todas as alternativas viáveis de resposta resultando a melhor solução, ou solução mais próxima possível da correta. Rodrigues (2017) ressalta que esta proposição sempre encontrará uma solução caso exista, porém, essa abordagem exige um custo computacional proporcional ao número de possibilidades para solução.

Essa abordagem é comumente utilizada para solução de diversos problemas pois é um método genérico e não limitado a nenhum domínio específico, entretanto há uma certa desvantagem por causa de seu tempo de execução, tornando-o ineficiente para problemas que necessitam de respostas em “tempo real”.

2.3 Modelos de Propagação

Esta Subseção trata de alguns conceitos fundamentais para criação de projetos de bancos de dados de *softwares*. Um projeto de banco de dados bem planejado é de fundamental importância para o sucesso no processo de sua implantação em uma empresa.

Neste trabalho, a camada de *middleware* é encarregada de executar cálculos matemáticos que objetivam minimizar o número de *gateways* garantindo cobertura máxima de

uma determinada região. Para isso um estudo aprofundado de alguns modelos de propagação que possuem boa adaptação em cenários amazônicos é apresentado nesta seção.

De acordo com Cosenza (2017) o modelo de propagação *Floating Intercept* (FI) não possui parâmetros baseados em fundamentos físicos, ele utiliza fatores de ajuste de curva, calculados pelo método dos mínimos quadrados. Esse modelo é representado pela Equação 3:

$$PL_{FI}[dB] = \alpha + 10\beta \log_{10}(d) + \chi_{\sigma_{FI}} \quad (3)$$

onde α é o coeficiente de interceptação flutuante em dB, β é a inclinação da reta, d é a distância entre as antenas em metros e $\chi_{\sigma_{FI}}$ é a variável de desvanecimento lento, que descreve as flutuações de sinal em grande escala sobre a perda de percurso médio, com média zero e desvio padrão σ .

Em contrapartida ao modelo FI, o modelo *Close In* (CI) é corroborado pelos princípios fundamentais de propagação vinculado as fórmulas de Friis e Bullington, ancorado fisicamente com a perda no espaço livre, garantindo uma relação fixa e contínua entre a potência transmitida sobre a distância. Sua definição é de acordo com a Equação 4.

$$PL_{CI}[dB] = FSDL(f, d_0) + 10n \log_{10}\left(\frac{d}{d_0}\right) + \chi_{\sigma_{CI}} \quad (4)$$

onde d é a distância entre as antenas em metros, n é o coeficiente de atenuação, $\chi_{\sigma_{CI}}$ é a variável de desvanecimento lento, e o valor de $FSDL(f, d_0)$ é definido conforme Equação 5.

$$FSDL(f, d_0)[dB] = 10 \log_{10}\left(\frac{4\pi d_0 f}{c}\right)^2 \quad (5)$$

onde a frequência f é dada em Hz e c é a velocidade da luz em m/s. A distância d_0 igual a 1 metro, pois é uma referência adotada em diversos modelos capaz de fornecer uma abordagem padronizada de modelagem.

Há um modelo de propagação que foi proposto especificamente para ambientes amazônicos, este modelo foi apresentado por Castro (2010) e possui como objetivo principal a capacidade de predição de perdas de propagação obedecendo a Equação 6.

$$L_{MP} = K_1 \log(d) + K_2 \log(f) + K_0 \quad (6)$$

onde, K_1 e K_2 representam os parâmetros a serem obtidos pelo método dos mínimos quadrados lineares, d a distância em metros, f a frequência em MHz e K_0 o fator de correção que correlaciona as alturas das antenas transmissoras, receptoras e as alturas de possíveis obstruções, esse fator é definido pelo polinômio de primeiro grau apresentado na Equação 7.

$$K_0 = a - bX \quad (7)$$

onde a e b serão ajustados pelo método de mínimos quadrados lineares, X a variável do polinômio que é definido conforme a Equação 8.

$$X = \frac{(H_T + H_R)\lambda}{0.1H_{OB}} \quad (8)$$

onde, H_T representa a altura da estação base, H_R representa a altura do rádio do cliente, λ representa o comprimento de onda e H_{OB} representa a altura média das obstruções, ressaltando que todas estas variáveis são calculadas em metros.

2.4 Engenharia de Software

De acordo com Sommerville (2007) a engenharia de *software* é essencial para o funcionamento das sociedades modernas, haja vista que, os veículos de comunicação, entretenimento, produção agrícola e industrial, além do mercado financeiro são controlados por algum sistema computacional. O principal objetivo da engenharia de *software* é sustentar o desenvolvimento de sistemas computacionais a partir de técnicas que amparam a especificação, projeto e evolução destes sistemas.

Atualmente há a necessidade de se manter a evolução contínua de sistemas computacionais, pois ocorre atualizações regulares nas linguagens e *frameworks* de desenvolvimento, por conta disso a engenharia de *software* estabelece alguns modelos de desenvolvimento, no qual o modelo iterativo e incremental é demasiadamente utilizado.

O modelo iterativo e incremental é fundamentado em entregas contínuas das funcionalidades do sistema em um período previamente acordado entre a equipe de desenvolvimento e os requisitantes da aplicação, dessa forma a cada nova entrega a equipe de desenvolvimento recebe um *feedback* dos usuários do sistema em desenvolvimento, isso faz com que o software final atenda todas as necessidades dos seus requisitantes.

2.4.1 Requisitos

Outro importante processo no desenvolvimento de um sistema computacional, é o levantamento de requisitos, que segundo Mendonça (2014) é a inicialização do processo de desenvolvimento de um sistema computacional, onde os analistas realizam as primeiras reuniões com os requisitantes/usuários da aplicação a fim de conhecer as funcionalidades do sistema que será desenvolvido.

Nessas reuniões são identificados e levantados os requisitos funcionais e não funcionais do sistema a ser desenvolvido que de acordo com Sommerville (2007) descrevem o que o sistema deve fazer, bem como os serviços que deve oferecer e as restrições de seu funcionamento. Os requisitos funcionais descrevem os serviços que o sistema deve possibilitar, como reagir a entradas específicas e como se comportar em determinadas situações. Os requisitos não funcionais descrevem as restrições aos serviços e funções do sistema.

2.4.2 Projeto de Banco de Dados

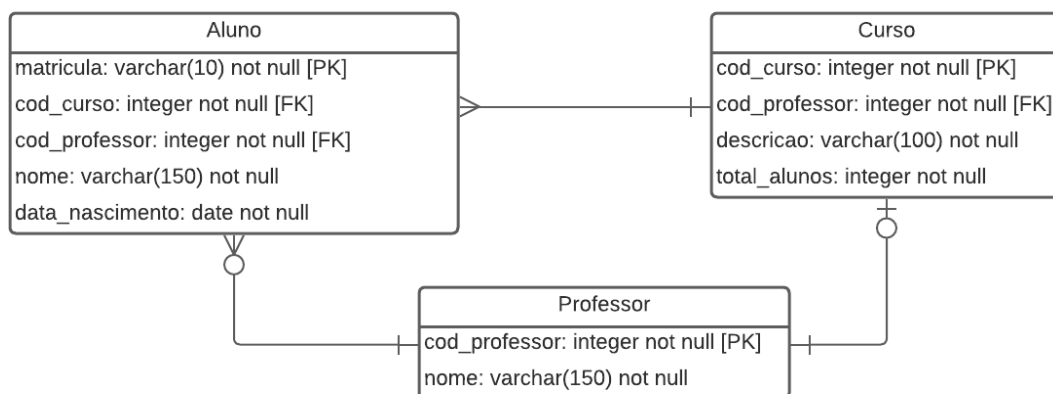
Conforme Elmasri et al. (2005) a tecnologia de banco de dados possibilitou a difusão do uso de sistemas computacionais em diversas áreas, comércio eletrônico, medicina, direito, engenharia e entre outros fazem uso dos princípios de banco de dados para representar alguns aspectos do mundo real, já que, para Heuser (1998) os modelos de banco de dados descrevem os tipos de informações sensíveis que serão armazenadas como “memória” para o sistema computacional.

Elmasri et al. (2005) explicam que a técnica de modelagem de dados entidade relacionamento (ER) é amplamente utilizada na construção de sistemas computacionais que

acaba sendo considerada um padrão para modelagem, nesta técnica apresenta-se o conceito de entidade que é um conjunto de objetos reais que podem ser modelados de forma genérica.

Machado (2004) afirma que utilizando os fundamentos do modelo ER se obtém esquemas fielmente idealizados sobre a finalidade do sistema, pois a ideologia deste modelo é retratada a partir do relacionamento entre entidades, a Figura 7 apresenta um exemplo de um diagrama de entidade relacionamento (DER).

Figura 7 – Exemplo de Diagrama Entidade Relacionamento.



Fonte: elaborada pelo autor.

O diagrama entidade relacionamento da Figura 7 é um fluxograma que ilustra as entidades Aluno, Professor e Curso que se relacionam de forma direta para reproduzir de forma digital, o processo de construção de um curso, onde cada curso deve possuir um professor e um número não definido de alunos, bem como um professor pode possuir um número indefinido de alunos e estar alocado em um curso específico.

3 SOFTWARE PARA PLANEJAMENTO DE REDES IOT

Neste Capítulo são apresentadas as etapas para o desenvolvimento do produto de *software*. A Seção 3.1 apresenta o contexto e os objetivos do *software*. A Seção 3.2 apresenta os requisitos levantados. Na Seção 3.3 apresenta-se a modelagem do projeto de banco de dados. Na Seção 3.4 são apresentadas as metodologias empregadas nas medições e análises dos modelos de propagação utilizados. Os resultados são apresentados na Seção 3.5.

3.1 Introdução

A idealização do *software* para planejamento de redes IoT, dá-se a partir da necessidade da redução de custos de implantação da infraestrutura de redes para provimento de sinal em diferentes tipos de meio ambiente, sendo este um desafio sem precedentes para regiões amazônicas, haja vista que sua composição climática é uma combinação de vários fatores, com predominância climática quente e úmida. Este *software* foi desenvolvido sobre o arcabouço das etapas de levantamento de requisitos, modelagem de banco de dados e estudos acerca dos modelos de propagação previamente selecionados para este propósito.

3.2 Requisitos

O levantamento de requisitos é a etapa onde ocorre a visualização das funcionalidades que devem estar presentes na solução implementada, para isso as necessidades dos especialistas de telecomunicações vinculados ao Laboratório de Computação e Telecomunicações (LCT) são consideradas. Dessa forma o Quadro 4 apresenta os requisitos funcionais do *software* organizados em ordem numérica além da descrição de seu serviço.

Quadro 4 – Requisitos funcionais do *software* de planejamento de redes IoT.

N#	Requisito
1	O <i>software</i> deve permitir ao usuário fazer seu cadastro através do email ou redes sociais.
2	O <i>software</i> deve permitir ao usuário a recuperação/alteração de sua senha de acesso.
3	O <i>software</i> deve permitir ao usuário fazer login com email.
4	O <i>software</i> deve permitir ao usuário fazer logout de sua conta.

5	O <i>software</i> deve permitir que o usuário defina a área a ser otimizada.
6	O <i>software</i> deve permitir que o usuário escolha o modelo de propagação <i>Floating Intercept</i> ou <i>Close In</i> .
7	O <i>software</i> deve permitir que o usuário escolha as configurações do equipamento LoRa, tais como potência de transmissão, fator de espalhamento
8	O <i>software</i> deve apresentar aos usuários um detalhamento técnico acerca dos parâmetros para facilitar a compreensão do uso por parte de usuários não especialistas.
9	O <i>software</i> deve permitir ao usuário a visualização dos resultados.
10	O <i>software</i> deve apresentar ao usuário um glossário de ajuda.
11	O <i>software</i> deve possuir uma comunidade para discussão das simulações entre usuários.

Fonte: elaborado pelo autor.

Já no Quadro 5, são apresentados os requisitos não funcionais do *software* que atendem as necessidades de uso da aplicação em relação a usabilidade, desempenho, segurança da informação, entre outros (MOUMANE & IDIR & ABRAN, 2016).

Quadro 5 – Requisitos não funcionais do *software* de planejamento de redes IoT.

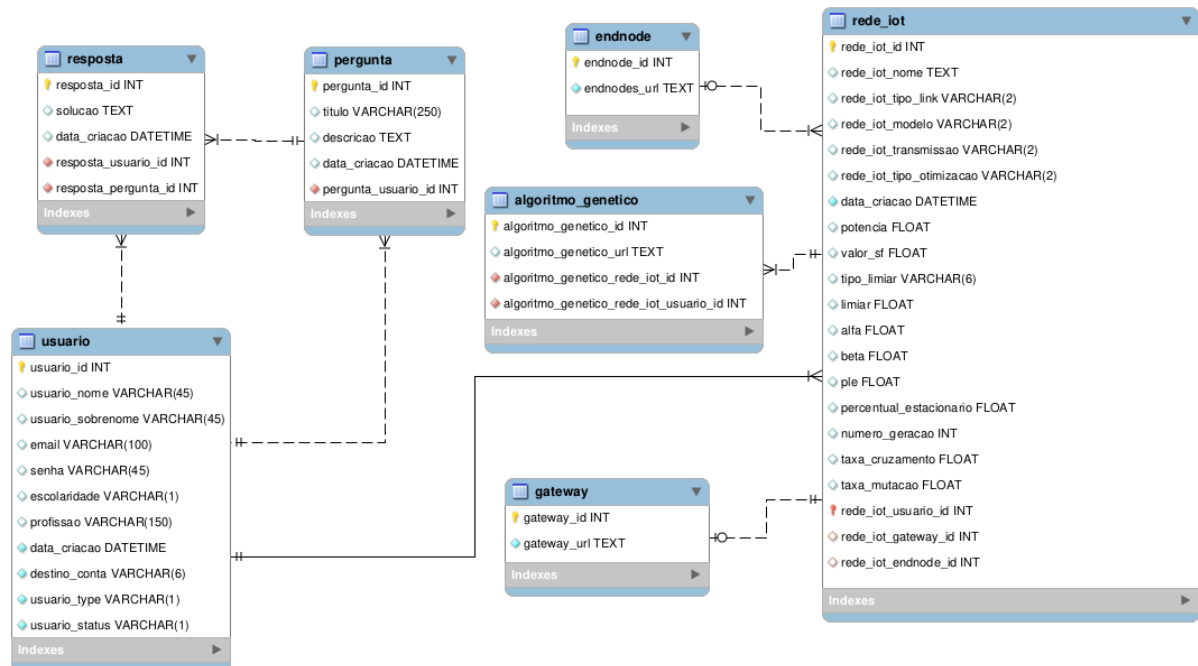
N#	Requisito
1	O <i>software</i> deve ser compatível com os navegadores Google Chrome e Mozilla Firefox.
2	O <i>software</i> deve ser responsivo e acessível para os usuários, permitindo acesso via dispositivos móveis.
3	O <i>software</i> deve possuir mecanismos de segurança para autenticação de usuário e controle de acesso de seu conteúdo.
4	O <i>software</i> deve possuir interface gráfica que proporcione interatividade, dinamicidade e agilidade para os usuários.
5	O <i>software</i> deve permitir que os dados sejam tratados e os resultados sejam coerentes. Para isso, os cálculos matemáticos executados devem ser íntegros e seus resultados devem ser revalidados.
6	O <i>software</i> deve ter alta disponibilidade (24/7), para que as informações sejam inseridas e acessadas a qualquer tempo.
7	O <i>software</i> deve proteger os dados de seus usuários, criptografando suas senhas e possuindo mecanismos de <i>anti hacking</i> .
8	O <i>software</i> deve ser implementado dentro da arquitetura web, utilizando PHP, Javascript e Python.

Fonte: elaborado pelo autor.

3.3 Modelagem do Banco de Dados

Após a definição dos requisitos, abstrai-se um modelo de banco de dados capaz de atender as necessidades do *software* a partir da identificação de seus indivíduos tangíveis e de suas respectivas ações no *software*, gerando informações essenciais para o garantir o propósito da aplicação. A Figura 8 ilustra a modelagem final do *software*, seguindo os paradigmas de programação orientada a objetos.

Figura 8 - Modelo lógico do banco de dados do *Software* de Planejamento de Redes IoT.



Fonte: elaborada pelo autor.

A tabela *usuario* é responsável por armazenar os dados dos usuários do *software*, sendo eles: o identificador único do usuário (*usuario_id*), primeiro nome do usuário (*usuario_nome*), sobrenome do usuário (*usuario_sobrenome*), email do usuário (*email*), senha de acesso do usuário (*senha*), nível de formação do usuário (*escolaridade*), data de criação do registro do usuário no banco de dados (*data_criacao*) e profissão (*profissao*). Além disso, a tabela *usuario* possui relacionamentos com as tabelas *pergunta*, *resposta* e *rede_iot*, com cardinalidade 1 para muitos (1:N).

Já a tabela *pergunta* é responsável em armazenar as perguntas realizadas durante o uso do fórum de perguntas e respostas da aplicação. Para isso, a tabela possui os seguintes atributos: o identificador único da pergunta (*pergunta_id*), título da pergunta (*titulo*), descrição da pergunta (*descricao*), data de criação do registro da pergunta no banco de dados (*data_criacao*), e a chave estrangeira do usuário que efetuou login e fez a pergunta (*pergunta_usuario_id*). Em relação ao

relacionamento, a tabela pergunta possui o relacionamento de muitos para 1 (N:1) com a tabela usuario, e 1 para muitos (1:N) com a tabela resposta.

A tabela resposta é responsável em armazenar as respostas obtidas a partir da interação entre usuários do fórum de discussão. Para isso, possui os seguintes atributos: o identificador único da resposta (`resposta_id`), o texto da resposta dada por um usuário (`solucao`), data de criação do registro da resposta no banco de dados (`data_criacao`), a chave estrangeira do usuário que efetuou login e sugeriu uma solução (`resposta_usuario_id`), e a chave estrangeira da pergunta em questão (`resposta_pergunta_id`). Tal tabela é um relacionamento de muitos para 1 (N:1) com as tabelas usuario e pergunta.

A tabela rede_iot é responsável por armazenar os dados gerais de configuração adotados em determinada simulação. Além disso, a tabela é responsável em guardar o relacionamento com o usuário que planejou e executou determinada simulação, o que permite a recuperação e nova realização de testes com os parâmetros configurados anteriormente. Para se criar um registro, a tabela rede_iot armazena as seguintes informações: o identificador único da rede configurada (`rede_iot_id`), o nome da rede planejada (`rede_iot_nome`), o tipo de link adotado para simulação (`rede_iot_tipo_link`), o modelo de propagação (`rede_iot_modelo`), o tipo de transmissão (`rede_iot_transmissao`), o tipo de otimização (`rede_iot_tipo_otimizacao`), a data de criação do registro no banco (`data_criacao`), potencia (`potencia`), valor de SF (`valor_sf`), tipo de limiar (`tipo_limiar`), valor do limiar (`limiar`), alfa (`alfa`), beta (`beta`), PLE (`ple`), percentual estacionário (`percentual_estacionario`), número geração (`numero_geracao`), taxa de cruzamento (`taxa_cruzamento`), taxa de mutação (`taxa_mutacao`), além das chaves estrangeiras do usuário que efetuou o cadastro da rede (`rede_iot_usuario_id`) e dos arquivos salvos no servidor que contém as informações de endnodes (`rede_iot_endnode_id`) e gateways (`rede_iot_gateway_id`) da rede. Além disso, a tabela rede_iot possui relacionamentos com as tabelas gateway e endnode com cardinalidades muitos para 1 (N:1).

Por fim, as tabelas endnode, gateway e algoritmo_genetico são responsáveis em armazenar, respectivamente, os dados de geolocalização dos *end nodes*, *gateways* e os resultados simulados a partir do algoritmo genético. Para isso, a tabela endnode é utilizada para armazenar o identificador único de endnode (`endnode_id`) e a URL para acesso ao arquivo que contém os pontos de latitude e longitude (`endnode_url`), a tabela gateway é utilizada para armazenar o identificador único de *gateway* (`gateway_id`) e a URL para acesso ao arquivo que contém os pontos de latitude e longitude (`gateway_url`), na tabela algoritmo_genetico se armazena o identificador único de simulação pelo AG (`algoritmo_genetico_id`), a URL para

acesso ao arquivo com os resultados da simulação (`algoritmo_genetico_url`), o identificador do usuário que executou a simulação (`algoritmo_genetico_usuario_id`) e o identificador da simulação (`algoritmo_genetico_rede_iot_id`).

3.4 Medições e Modelagem dos Modelos de Propagação

Foram realizadas campanhas de medições *in loco* nas dependências do Campus da Universidade Federal do Pará (UFPA) – Belém/PA, um ambiente composto por prédios e árvores localizado às margens do rio Guamá, isto é um cenário tipicamente amazônico. O campus de Belém, da UFPA contém uma área edificada equivalente a 2.135.775 m² (Barbosa, 2012), por conta disso é dividido em 3 subáreas: básico, profissional e saúde. A partir dos estudos realizados com base nos dados coletados nas campanhas de medições, define-se o valor do limiar de corte que possibilita o cálculo do raio de cobertura do sinal no ambiente.

Para as medições, foram utilizados dois módulos de rádios Dragino modelo SX1276, conectados ao microcontrolador AT Mega modelo Arduino UNO, estando ambos equipados com antenas omnidirecional com ganho de 3 dBi. Cada dispositivo foi conectado a um Notebook, sendo utilizados como transmissor e receptor (TX-RX). No Quadro 6 são apresentados os parâmetros adotados durante as medições. As medições foram realizadas durante o período matutino e vespertino de dias ensolarados.

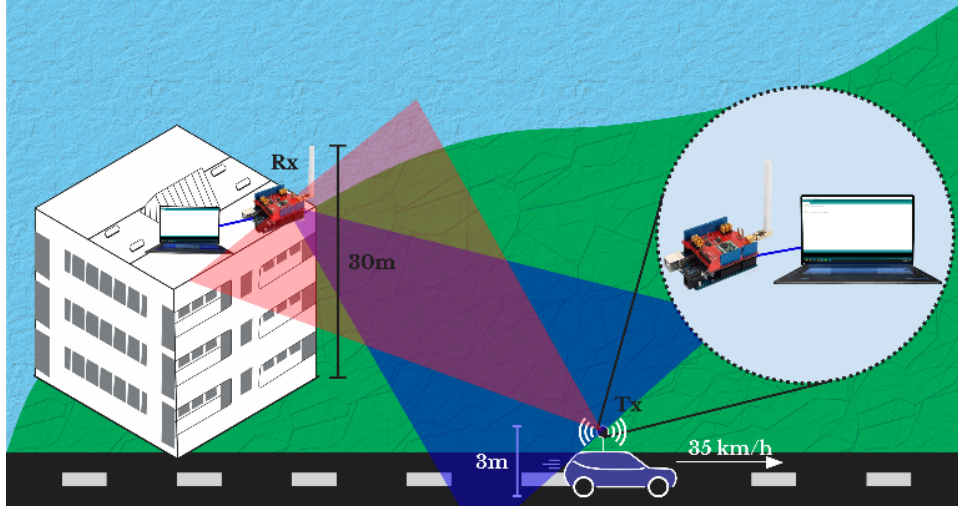
Quadro 6 – Parâmetros utilizados nas campanhas de medições.

Parâmetros	Valor e Unidade
Frequência de operação	915 MHz
Potência efetiva irradiada	20 dBm
<i>Spreading Factor</i>	7, 9, 12
Largura de Banda	125 KHz
<i>CodingRate</i>	5(4/5)

Fonte: elaborado pelo autor.

Um ponto fixo no espaço da UFPA foi definido, o maior prédio, o Mirante do Rio, localizado no setor Básico com a seguinte geolocalização: latitude -1.476941 e longitude -48.456547. Nele instalou-se um dos equipamentos ao topo do prédio com aproximadamente, 30 metros de altura do nível do mar e o outro equipamento foi instalado em um carro com 3 metros de altura, o qual se movimenta pela rota do ônibus circular, conforme apresentado na Figura 9.

Figura 9 - Esquema da metodologia de medição



Fonte: elaborada pelo autor.

Durante as medições foram coletados dados de latitude e longitude, RSSI e SNR do sinal propagado, que eram retransmitidas continuamente pelo percurso, a uma velocidade de 35 Km/h, considerando o sinal de *uplink* (do carro para o prédio) e *downlink* (do prédio para o carro). Estas campanhas se repetiram por 5 vezes para que fosse possível criar um banco de informações e a partir destes dados realizar tratamentos para obtenção de métricas importantes para o desenvolvimento do software, sendo: a distância entre o dispositivo TX e o dispositivo RX, e a perda de propagação.

A distância entre os dispositivos TX e RX foi definida a partir dos valores de latitude e longitude coletados, e calculada a partir da Equação (9) de Haversine, onde d é a distância entre dois pontos em metros, r é o raio da terra em metros (6371000 m); $\varphi_1, \varphi_2, \lambda_1$ e λ_2 , são respectivamente, a latitude do ponto 1 e a latitude do ponto 2, a longitude do ponto 1 e a longitude do ponto 2.

$$d = 2r \sin^{-1} \left(\sqrt{\sin^2 \left(\frac{\varphi_1 - \varphi_2}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left(\frac{\lambda_1 - \lambda_2}{2} \right)} \right) \quad (9)$$

A perda de propagação é calculada pelo nível de potência do sinal (SEYE et al., 2018), para isso se executa a soma entre os valores de RSSI e SNR conforme Equação (10) e então obtém-se a resultante através da Equação (11) de balanceamento de link.

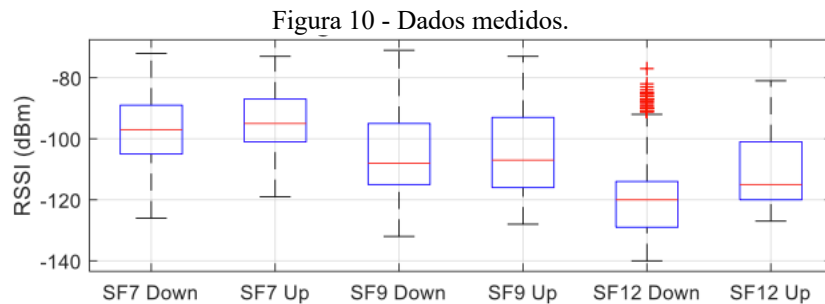
$$Pr = RSSI + SNR \quad (10)$$

$$PL = Pt + G - Pr \quad (11)$$

onde Pr é a potência recebida em determinado ponto em dBm; Pt é a potência transmitida pelo gateway em dBm; G é a soma dos ganhos de transmissão e recepção, geralmente em dBi e, por fim PL representa a perda de propagação do sinal em Db.

3.4.1 Análise de Dados

Durante a análise de dados, foram realizadas investigações considerando o valor de RSSI (dBm) em função da variação do SF. A Figura 10 ilustra o grau de espalhamento dos dados para os SFs verificados durante o experimento de medição. Neste caso, os resultados são apresentados na forma de *boxplot*, que por sua vez, leva em consideração o primeiro, segundo e terceiro quartis, assim como, os limites inferiores e superiores.



Fonte: elaborada pelo autor.

Conceitualmente na estatística descritiva, quartis são valores extraídos pela análise global do conjunto de dados, observado de forma ordenada e crescente, que dividem a distribuição destes em quatro partes iguais. O primeiro quartil (Q1) é correlacionado à amostra ordenada de forma crescente de 25% dos dados; o segundo quartil (Q2) é correlacionado mediana; e o terceiro quartil (Q3) é correlacionado a 75% dos dados organizados de forma crescente. A partir dessa investigação, se definiu que os valores mínimos de limiar que são utilizados para determinação do raio de cobertura do *gateway*.

A modelagem de perda de propagação para o modelo *Close-In* caracterizou-se pela determinação dos valores de “ n ” e para o modelo *Floating Intercept* pela determinação do α e β . Por conta destas variáveis representarem o comportamento médio da perda do sinal para o ambiente do campus.

O ajuste destas variáveis é calculado pelo método de Mínimos Quadrados Lineares (MQL) (MACCARTNEY & DENG, 2015), que por sua vez, objetiva reduzir a soma dos quadrados das diferenças entre os dados coletados e dos dados simulados, conforme apresentado na Equação 12.

$$f_{obj} = \sum_{i=1}^n (D_m - D_s)^2 \quad (12)$$

Dessa forma o ajuste destas variáveis é definido pelo sistema linear da Equação 13.

$$A = \begin{bmatrix} 1 & D_m \\ \cdot & \\ \cdot & \\ \cdot & \\ 1 & D_i \end{bmatrix} x = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} B = \begin{bmatrix} D_1 \\ D_i \end{bmatrix} \quad (13)$$

E a partir da Equação 13, isolamos a solução matricial de acordo com a Equação 14.

$$x = (A^T A)^{-1} A^T B \quad (14)$$

Deste modo, foram calculados os coeficientes n , α e β para o campus da UFPA, conforme apresentado no Quadro 7.

Quadro 7 – Coeficientes calculados.

Modelos				
	CI		FI	
	Down	Up	Down	Up
SF7	PLE = 3,48	PLE = 3,45	$\alpha = 134,32$ $\beta = 2,74$	$\alpha = 130,24$ $\beta = 2,28$
SF9	PLE = 3,59	PLE = 3,57	$\alpha = 139,17$ $\beta = 3,47$	$\alpha = 136,99$ $\beta = 2,98$
SF12	PLE = 3,7	PLE = 3,64	$\alpha = 142,64$ $\beta = 3,78$	$\alpha = 140,73$ $\beta = 3,51$

Fonte: elaborado pelo autor.

A partir destes coeficientes foi calculada a perda de propagação do sinal para o ambiente e o raio de cobertura do mesmo, adotando-se as equações dos respectivos modelos *Close-In* e *Floating Intercept*, isolando a variável distância, conforme apresentado nas Equações 15 e 16.

$$Distancia_{CI} (m) = 10^{\left[\frac{P_t(dBm) - Limiar(dB) - FSPL}{10 \times n} \right]} \quad (15)$$

$$Distancia_{FI} (Km) = 10^{\left[\frac{P_t(dBm) - Limiar(dB) - \alpha}{10 \times \beta} \right]} \quad (16)$$

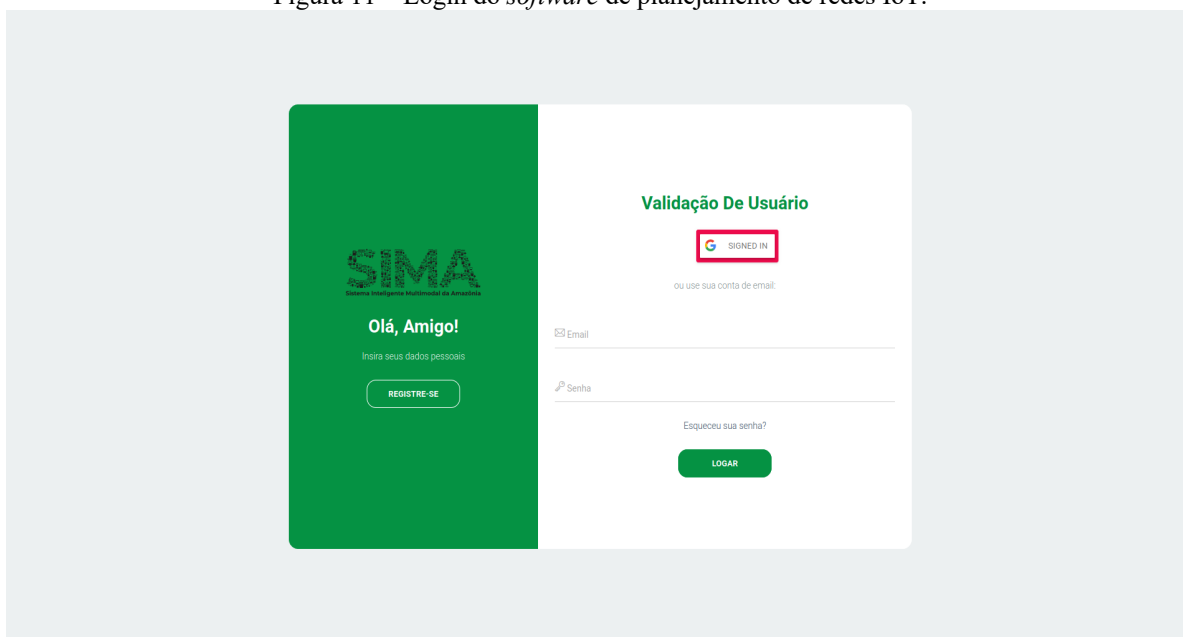
3.5 Resultados

Nesta subseção, serão apresentadas as principais funcionalidades e áreas de navegação presentes no *software*, assim como, os resultados de simulações. O *software* foi desenvolvido de acordo com os requisitos propostos utilizando as linguagens de programação PHP, Javascript e Python que respectivamente são responsáveis, pelo controle de requisições ao servidor, dinamicidade nas páginas de navegação e cálculos matemáticos do *software*. Para armazenamento de informações, foi utilizado o banco de dados MariaDB.

3.5.1 Credenciamento e Primeiro acesso

A tela de *login* do *software*, ilustrada pela Figura 11 é responsável pelo primeiro contato entre o usuário e a aplicação, sendo que a partir do registro do usuário será possível realizar as simulações, armazenar os resultados das simulações, recuperar dados de configurações, obter resultados gráficos de simulações executadas, e participar do fórum técnico de discussão; para isso a autenticação do usuário poderá ser realizada via *Gmail* ou cadastro na plataforma.

Figura 11 – Login do *software* de planejamento de redes IoT.

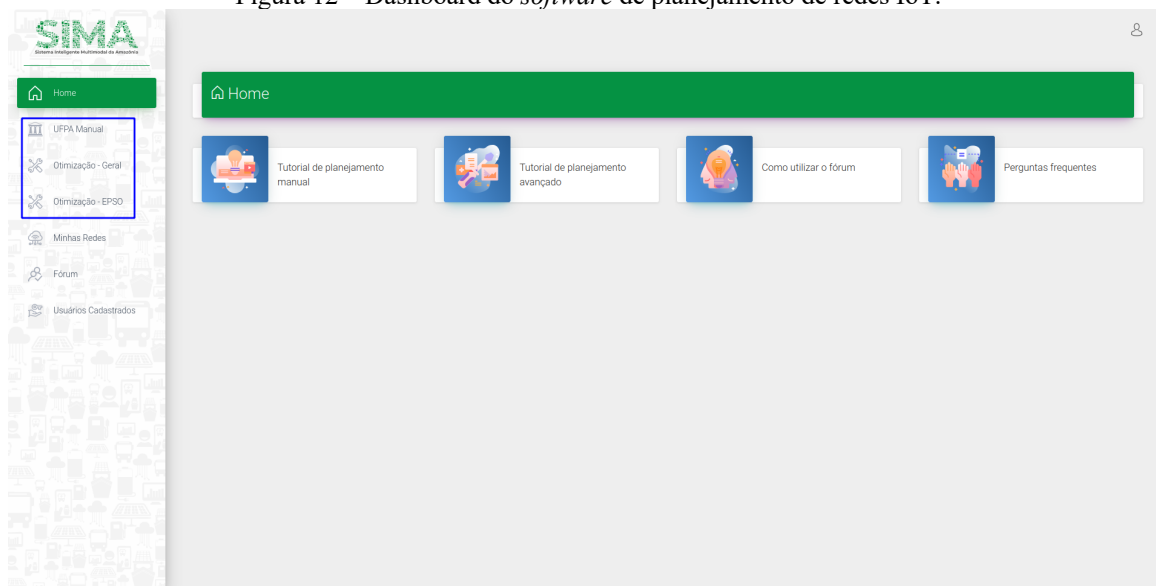


Fonte: elaborada pelo autor.

Após esse procedimento, o usuário é redirecionado para o *dashboard* principal do *software*, Figura 12, a qual é uma página que disponibiliza todas as funções que podem ser realizadas pelo usuário, destacando-se as funcionalidades de simulação, UFPA Manual, Otimização – Geral e Otimização – EPSO. Além destas, na área principal, estão disponíveis as

documentações que possuem como finalidade instruir e ajudar o usuário na utilização do *software*.

Figura 12 – Dashboard do *software* de planejamento de redes IoT.



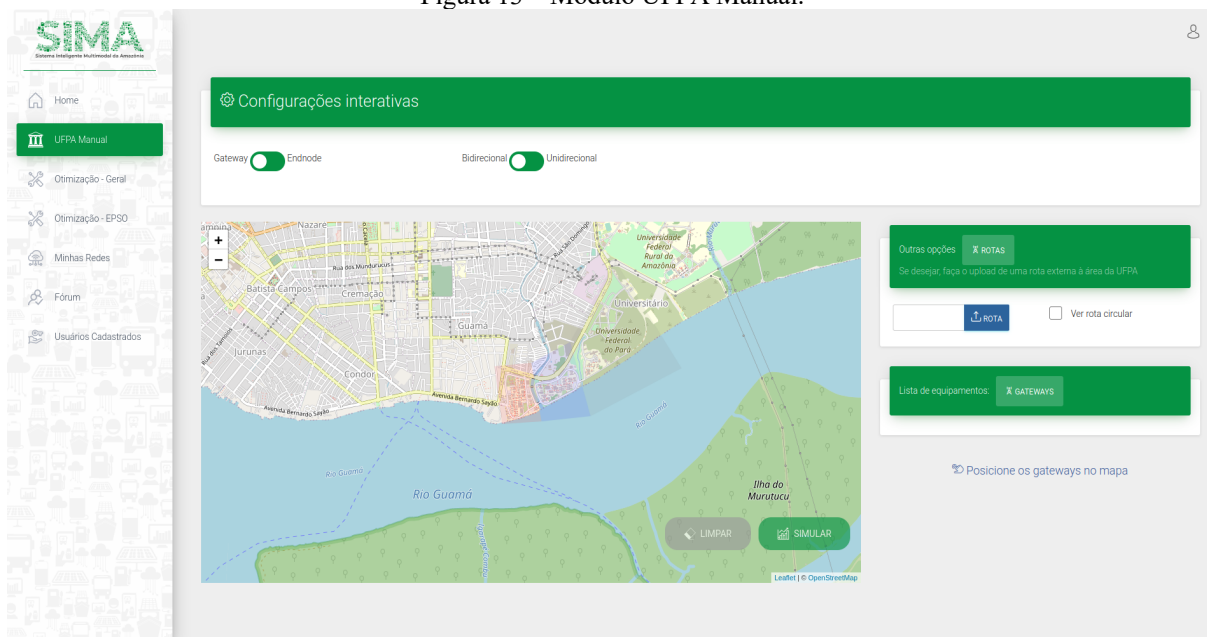
Fonte: elaborada pelo autor.

3.5.2 UFPA Manual

Na opção UFPA Manual, ilustrada na Figura 13 é carregado o mapa do ambiente, estando este centralizado no Campus Universitário da UFPA Belém. Para isso, foi utilizado a biblioteca *OpenStreetMap*. As dependências da UFPA foram destacadas em três cores: vermelho, azul e cinza que representam respectivamente os Campi Básico, Profissional e Saúde. As opções de configurações são apresentadas na parte superior e lateral direita do mapa.

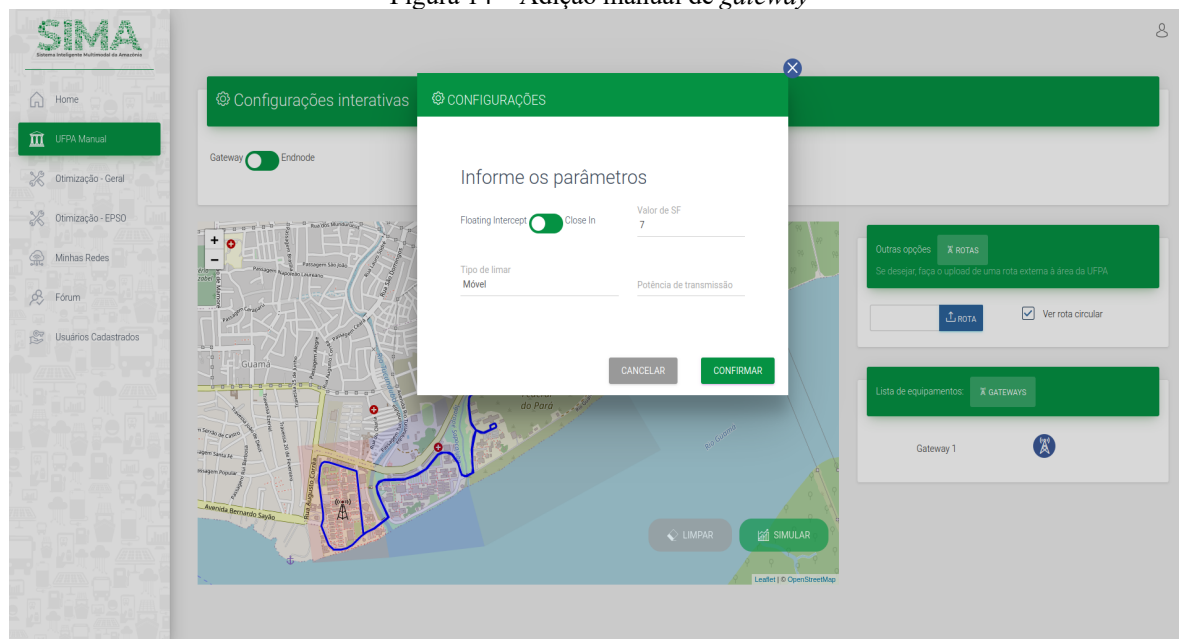
Na parte superior do mapa há a opção de configuração para o tipo de clique no mapa, *Gateway* ou *Endnode*, que controla o evento do mouse em que se efetua o posicionamento do dispositivo selecionado. Além da determinação do tipo de comunicação, Bidirecional ou Unidirecional, que controla uma variável que estipula um valor *default* a ser utilizado nos cálculos dos modelos de propagação propostos.

Figura 13 – Módulo UFPA Manual.



Fonte: elaborada pelo autor.

No menu lateral direito do mapa, o usuário pode realizar o *upload* de um arquivo do tipo texto, para carregar as posições (latitude e longitude) de *end nodes* ou habilitar a visualização dos pontos de geolocalização que correspondem a rota do ônibus Circular no Campus Belém, com isso, e após o usuário efetuar o posicionamento de um *gateway* no mapa o *software* automaticamente apresenta uma área para configuração do equipamento a partir do preenchimento dos seguintes campos: modelo de propagação, valor de SF, tipo de limiar e potência de transmissão, assim adicionando-se à lateral direita do mapa uma lista de componentes, onde se pode revisar ou editar a configuração manual de cada *gateway*, conforme ilustrado na Figura 14.

Figura 14 – Adição manual de *gateway*

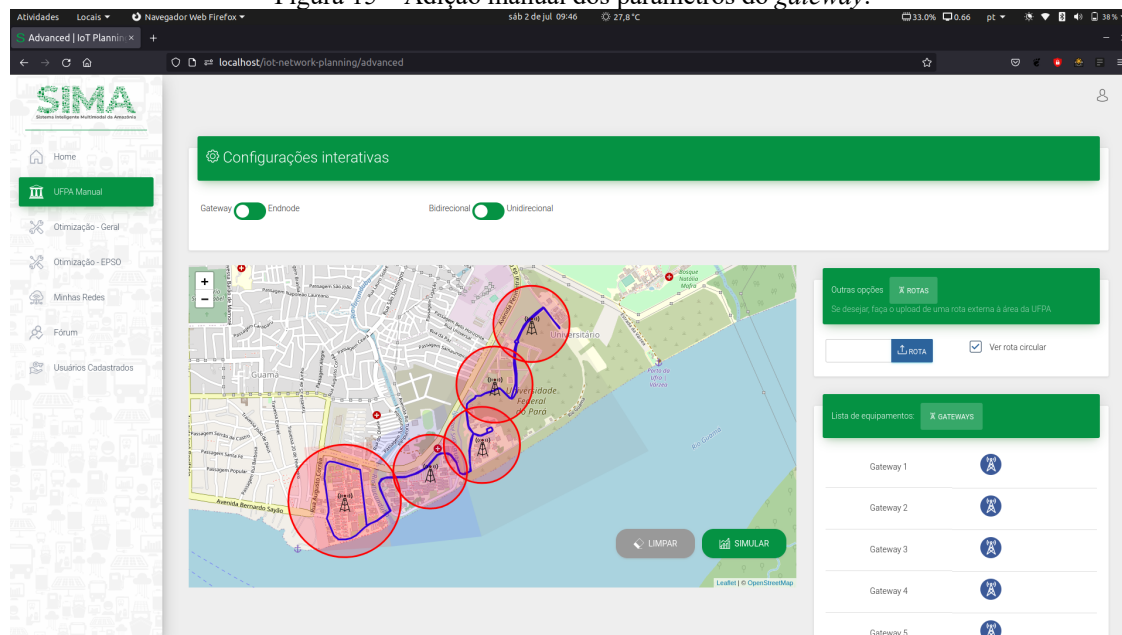
Fonte: elaborada pelo autor.

3.5.2.1 Cenário de Aplicação

Para exemplificar o funcionamento do módulo UFPA Manual, foi realizado um planejamento de rede IoT considerando a rota percorrida pelo ônibus circular da UFPA, que por sua vez é formada por 902 pontos de latitude e longitude, conforme apresentado no Apêndice.

Após o carregamento da rota, a partir do clique na caixa de seleção “ver rota circular” na área lateral direita, foi definida no menu superior a opção *gateway*, assim permitindo a adição via clique do usuário, dessa forma o usuário informa para o *software* de forma visual os pontos de latitude e longitude para instalação do *gateway* e após configuração dos parâmetros de transmissão o *software* indica de forma “instantânea”, a área de cobertura do mesmo, caso o usuário não fique satisfeito com a área de cobertura estimada pelo sistema, é possível realizar novamente o processo de adição de *gateway*, possibilitando que a arquitetura da rede seja estimada a partir de múltiplas tentativas. O que traz dinamicidade ao processo de planejamento de redes IoT, evitando que novas campanhas de medição sejam realizadas.

Figura 15 – Adição manual dos parâmetros do gateway.

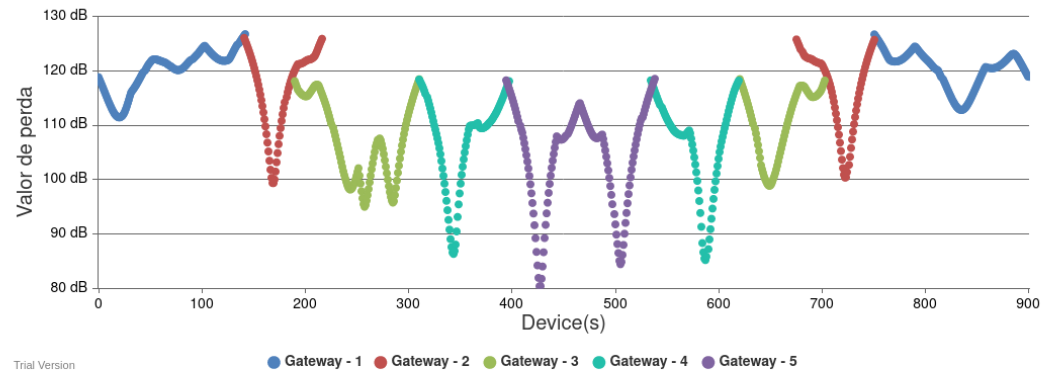


Fonte: elaborada pelo autor.

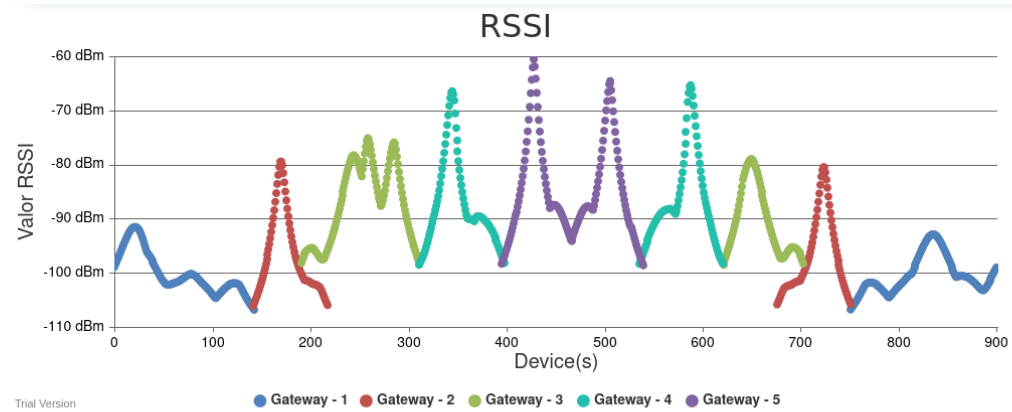
Assim que satisfeito com a arquitetura final, conforme apresentada na Figura 15, o usuário pode solicitar que o *software* realize os cálculos de perda de sinal para cada ponto da rota estabelecida clicando no botão “simular”, presente sob o mapa. Os resultados, da perda do sinal é calculado em função dos *gateways* posicionados e a disposição dos *end nodes* adicionados no ambiente. Em seguida, os resultados são apresentados ao usuário na área abaixo do mapa.

A Figura 16 (a) ilustra o gráfico de perda de sinal para cada posição de *end node* disposto na arquitetura da rede apresentada na Figura 15. É possível analisar que o *gateway* 1 estabelece comunicação com os *end nodes* de [0-141] e [751-902]. O *gateway* 2 estabelece comunicação com os *end nodes* de [141-217] e [676-751]. O *gateway* 3 estabelece comunicação com os *end nodes* de [190-311] e [621-703]. O *gateway* 4 estabelece comunicação com os *end nodes* de [311-398] e [535-690]. Por fim, o *gateway* 5 estabelece comunicação com os *end nodes* de [395-539]. Além disso é possível perceber que, há casos, em que um grupo de *end nodes* foram cobertos pelos *gateways* 2 e 3 indicando que neste caso é mais eficiente para o *end node* se conectar com o *gateway* 2 por este ter um menor valor de perda associado.

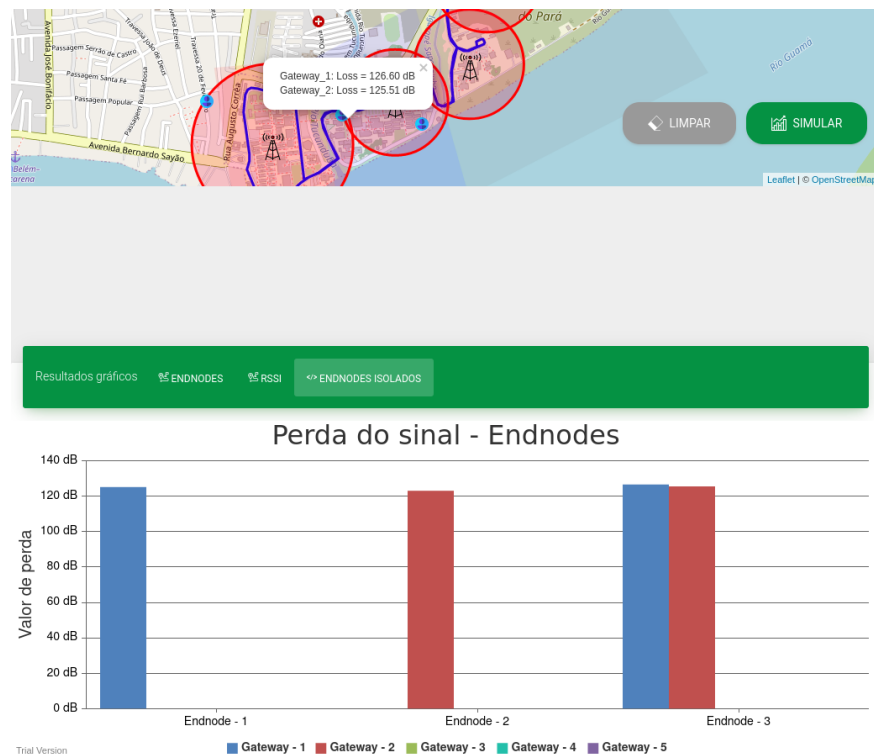
Figura 16 – Gráficos resultantes. (a) Perda de sinal. (b) RSSI. (c) Dispositivos *end nodes* discretos



(a)



(b)



(c)

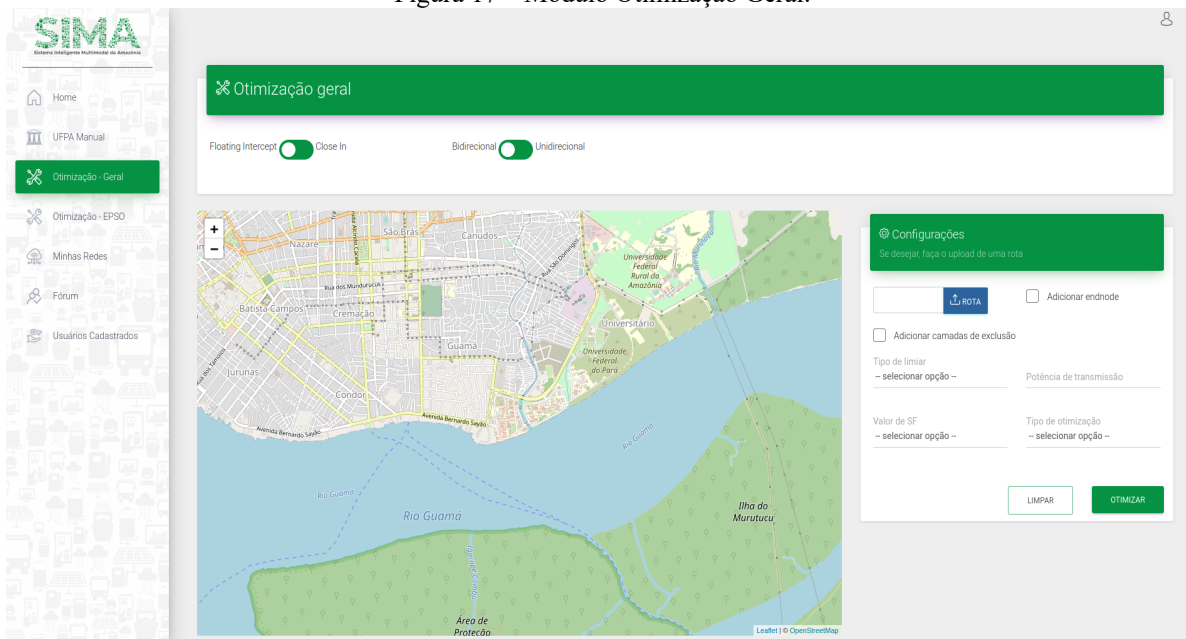
Fonte: elaborada pelo autor.

A Figura 16 (b), corresponde ao gráfico dos valores de RSSI calculados, que evidenciam que quanto mais próximo do centro do raio de cobertura do *gateway* melhor é a qualidade do sinal no dispositivo. Já a Figura 16 (c), apresenta, um gráfico correspondente a perda do sinal para um ou mais pontos discretos definidos pelo usuário no mapa, quando este modifica o tipo de clique no mapa, selecionando a opção *Endnode* na parte superior de configuração. Para cada clique efetuado pelo usuário é possível observar o valor de perda no ponto de geolocalização definido, assim caso o ponto adicionado esteja localizado em uma área de intersecção de coberturas apresenta-se os valores associados para cada *gateway*. Isso garante que novas prospecções sejam realizadas caso haja necessidade de no futuro, se adicionar mais *end nodes* à arquitetura de rede IoT.

3.5.3 Otimização Geral

O planejamento via otimização pode ser realizado a partir das opções Otimização – Geral (Algoritmo Genético e de Múltiplas Tentativas) e Otimização – EPSO, a Figura 17 apresenta a seleção de otimização geral. Que disponibiliza o mapa para as interações do usuário, além das opções de configurações na parte superior e lateral direita do mapa. As configurações superiores alternam-se entre os modelos de propagação *Floating Intercept* ou *Close In* e a determinação do tipo de comunicação, sendo elas Bidirecional ou Unidirecional.

Figura 17 – Módulo Otimização Geral.

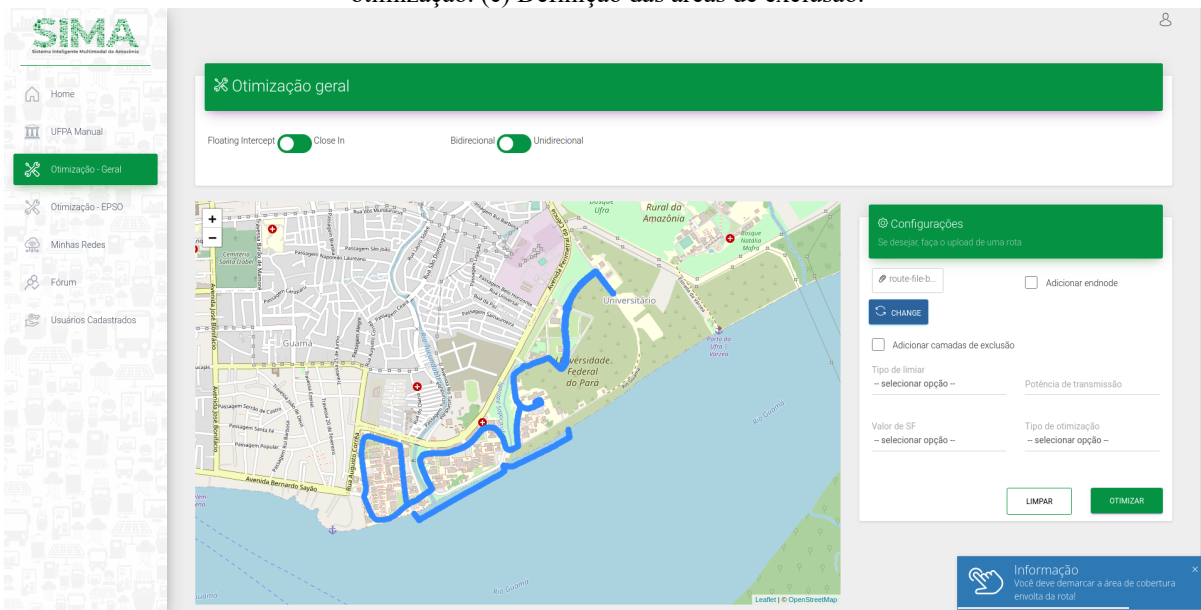


Fonte: elaborada pelo autor.

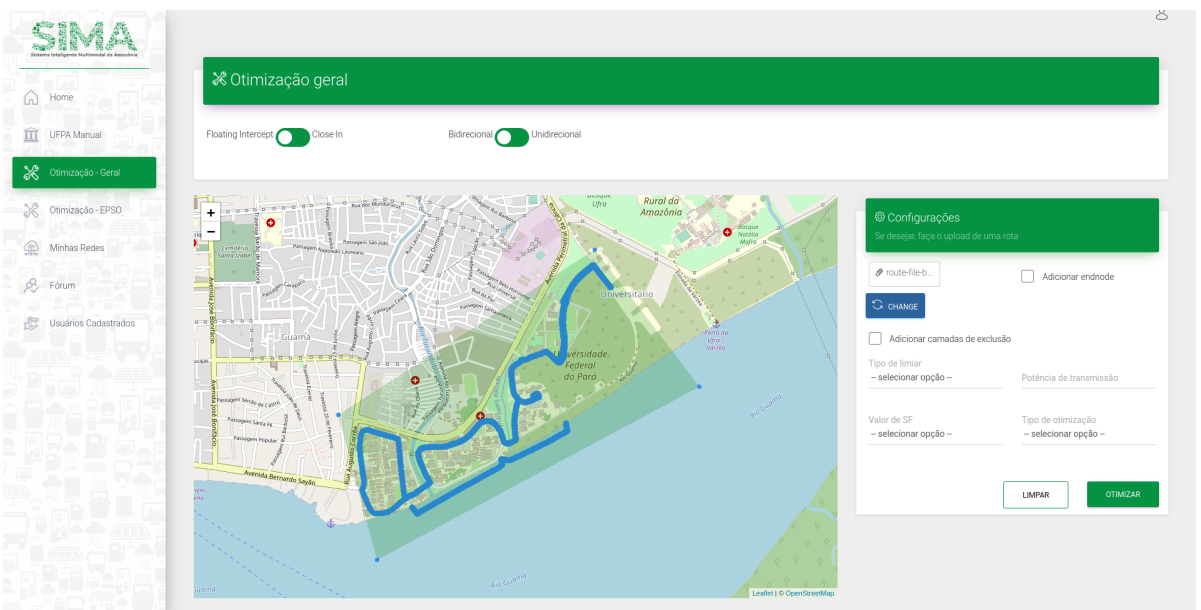
Na lateral direita do mapa, o usuário pode realizar a configuração da simulação selecionando entre as opções: algoritmo de Múltiplas Tentativas ou Algoritmo Genético (AG),

o pseudocódigo destes algoritmos é apresentado nos Apêndices desta dissertação, além disso, o usuário pode adicionar *end nodes* de forma automática, a partir do upload de um arquivo com as suas respectivas localizações ou de forma manual, a partir da seleção do campo “Adicionar Endnode” habilitando, neste caso, um evento do mouse para posicionamento, quando efetuado um clique no mapa, ressalta-se que também há a possibilidade de o usuário realizar esse procedimento utilizando as duas formas, realizando sequencialmente o processo automático e manual. Ainda é permitido adicionar áreas de exclusão, ao ambiente, que representam áreas restritas de localizações.

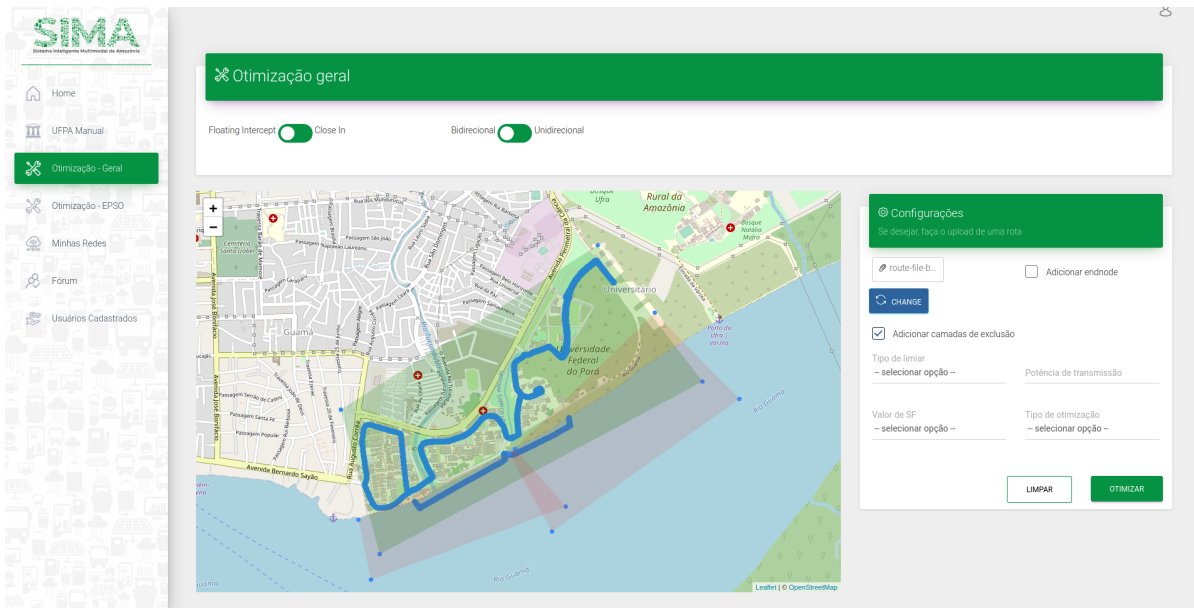
Figura 18 – Processo de adição de entradas do sistema. (a) Carregamento de rota. (b) Definição da área de otimização. (c) Definição das áreas de exclusão.



(a)



(b)

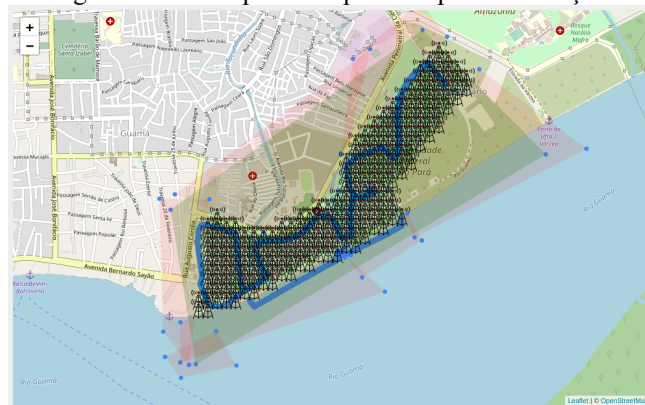


(c)

Fonte: elaborada pelo autor.

Após a realização do *upload* do arquivo de coordenadas, 1208 pontos geográficos (ver no apêndice), os *end nodes* são apresentados em tela, neste caso, compõem a rota do ônibus Circular e Barco Elétrico que trafegam no campus da universidade, além disso, um menu informativo é exibido solicitando que o usuário defina a área de otimização, conforme a Figura 18 (a). Para realizar esse passo, o usuário deve realizar o posicionamento de quatro pontos (vértices) que englobem todos os pontos de *end nodes* que deverão ser atendidos, conforme Figura 18 (b). Em seguida, o usuário poderá definir áreas de restrição, onde o *gateway* não pode ser instalado, a Figura 18 (c) apresenta as áreas de restrição em vermelho, sobrepostas ao rio, indicando regiões onde não se pode adicionar *gateways* por questões econômicas ou técnicas. Assim, após adição dos parâmetros do *gateway* e seleção do algoritmo, o *software* preenche a área disponível para instalação de equipamentos com um conjunto de *gateways* candidatos para construção da rede, conforme a Figura 19.

Figura 19 – Exemplo de arquitetura para otimização.



Fonte: elaborada pelo autor.

3.5.4 Resultados Matemáticos

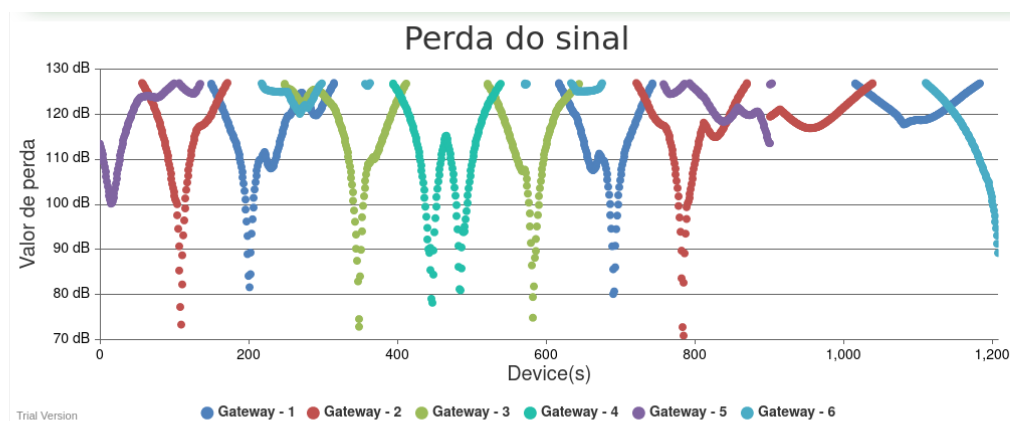
Para validar o *software*, foram realizadas simulações de planejamento de rede IoT no ambiente do *smart campus* do Guamá de acordo com a arquitetura de rede definida na Figura 19. Os resultados da otimização dependem do algoritmo selecionado, haja vista que, o conjunto de possibilidades gerado para o posicionamento de *gateways*, no caso do algoritmo de múltiplas tentativas, as soluções indicadas pelo software estão contidas nesse conjunto grande e finito; já no caso do algoritmo genético as soluções indicadas serão modificadas a cada iteração por meio dos operadores de cruzamento e mutação.

A Figura 20 apresenta os resultados alcançados pela execução do algoritmo de múltiplas tentativas, com os seguintes parâmetros de entrada: modelo de propagação *Floating Intercept*, *link* bidirecional, limiar móvel, potência de transmissão de 20dBm e SF 9. Estes resultados apresentam visualmente a eficácia do método, indicando que um número total de 6 gateways seria necessário para garantir a cobertura de todos os *end nodes* dispostos na área de interesse, respeitando as restrições impostos pelo usuário, conforme apresentado na Figura 20 (a).

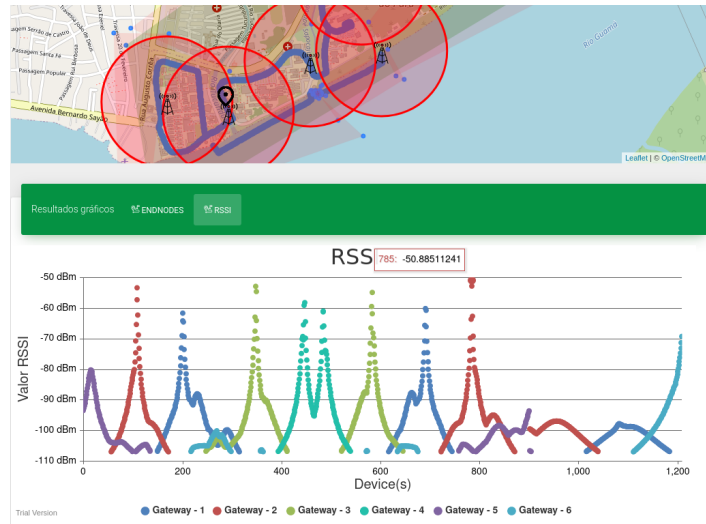
Figura 20 – Resultados da otimização por múltiplas tentativas. (a) Posicionamento de gateways. (b) Valores de perda calculados. (c) Valores de RSSI calculados.



(a)



(b)



(c)

Fonte: elaborada pelo autor.

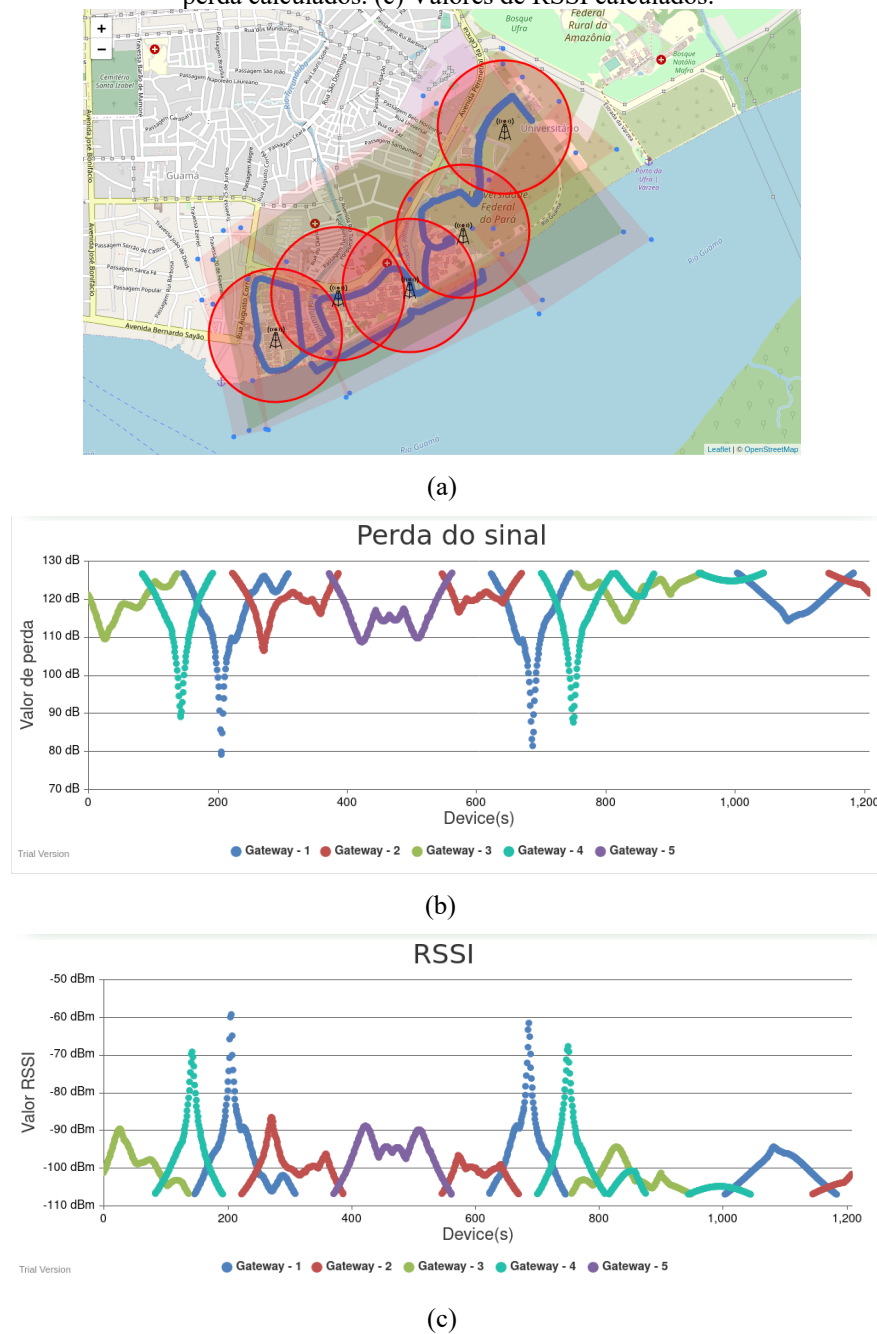
Na Figura 20 (b) é possível analisar que o *gateway* 1 estabelece comunicação com os *end nodes* de [0-134], [758-899] e [902], o *gateway* 2 estabelece comunicação com os *end nodes* de [56-168], [722-870] e [902-1038], o *gateway* 3 estabelece comunicação com os *end nodes* de [248-409] e [521-645], o *gateway* 4 estabelece comunicação com os *end nodes* de [217-298], [357-362], [394-536], [571] e [634-675], o *gateway* 5 estabelece comunicação com os *end nodes* de [758-899] e [902] e o *gateway* 6 estabelece comunicação com os *end nodes* de [634-675] e [1111-1209]. Além disso é possível perceber que há casos, em que um grupo de *endnodes* são cobertos por mais de um *gateway* indicando que neste caso é mais eficiente para o *end node* se conectar com o *gateway* que possua menor valor de perda associado.

Na Figura 20 (c) é possível analisar a configuração arquitetural da rede posposta a partir do nível de potência RSSI calculado para cada ponto que o *end node* pode estar localizado, estes resultados nos indicam que há uma excelente qualidade de sinal nos dispositivos que estiverem mais próximo dos centros dos raios de cobertura do *gateway*.

Na simulação com algoritmo genético foram definidos os seguintes parâmetros de entrada: modelo de propagação *Floating Intercept*, *link* bidirecional, limiar móvel, potência de transmissão de 20dBm, SF 9, 1000 gerações, 85% de percentual estacionário, 0.8 de taxa de cruzamento, e 0.05 de taxa de mutação.

A Figura 21 apresenta os resultados alcançados a partir da execução do AG, os resultados também demonstram a eficácia do método, desta vez indicando que um número total de 5 de *gateways* seria necessário para garantir a cobertura de todos os *end nodes* presentes na área de interesse, conforme Figura 21 (a), além disso, este algoritmo também obedeceu às restrições impostas pelo usuário.

Figura 21 – Resultados da otimização por algoritmo genético. (a) Posicionamento de gateways. (b) Valores de perda calculados. (c) Valores de RSSI calculados.

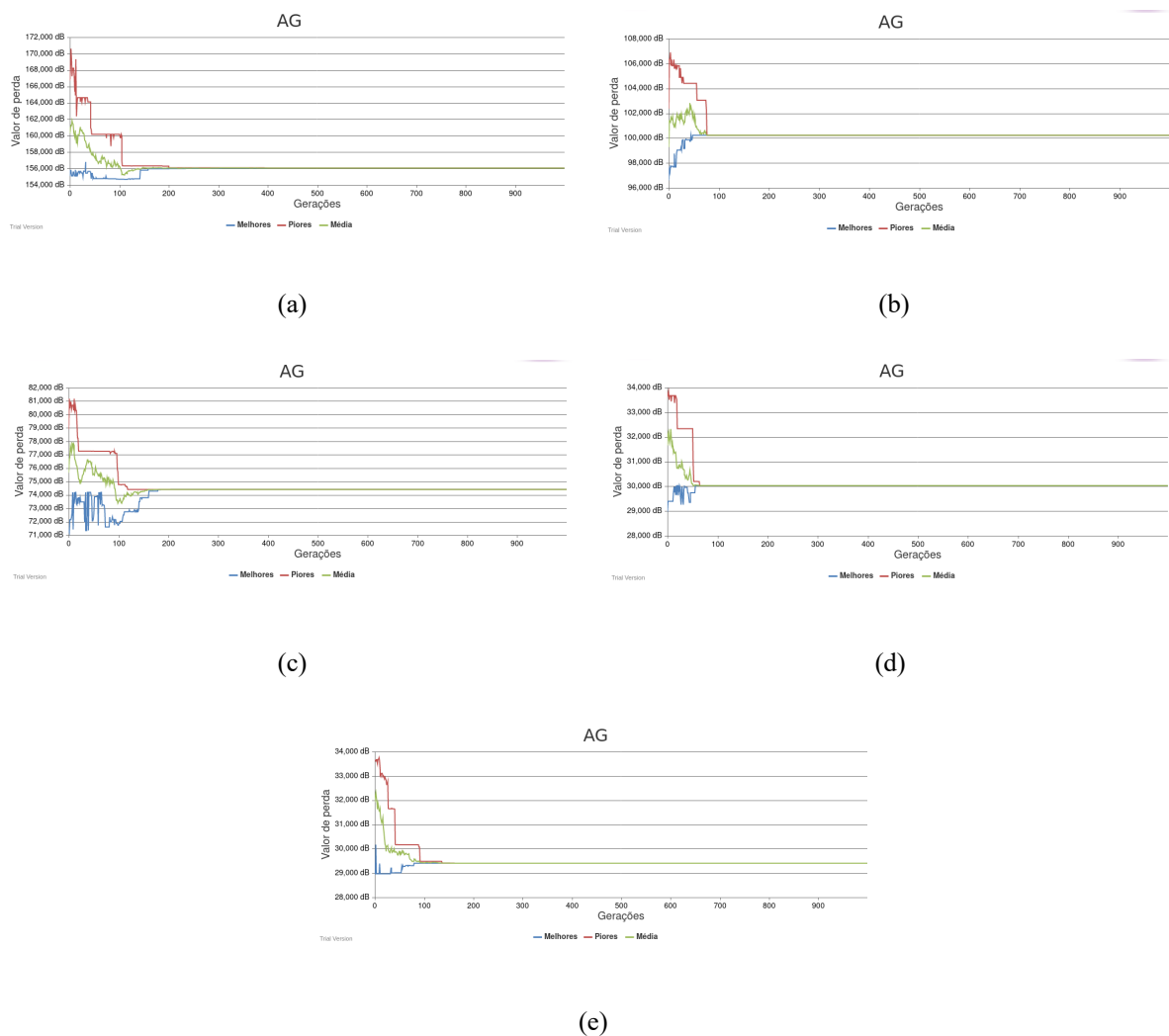


Fonte: elaborada pelo autor.

A Figura 21 (b) ilustra os resultados de perda calculados e associados para cada dispositivo alocado no espaço de interesse, é possível perceber que o *gateway* 1 estabelece comunicação com os *end nodes* de [147-307], [623-744] e [1004-1180], o *gateway* 2 estabelece comunicação com os *end nodes* de [223-384], [547-668] e [1145-1208], o *gateway* 3 estabelece comunicação com os *end nodes* de [0-137] e [756-947], o *gateway* 4 estabelece comunicação com os *end nodes* de [83-193], [701-872] e [945-1043] e o *gateway* 5 estabelece comunicação com os *end nodes* de [372-561], assim se conclui que para um determinado grupo de

dispositivos que realizam comunicação com um ou mais *gateways* é preferível que este opte se comunicar com o equipamento que provem maior perda, além disso, percebe-se o comportamento estocástico da estratégia evolutiva com base na disposição dos *gateways*, haja vista que, diferentemente do algoritmo de múltiplas tentativas o posicionamento do *gateway* não seguiu a ordem cronológica dos dispositivos (*end nodes*), indicando que houve um certo “passar” entre o espaço de soluções. Na Figura 21 (c) são apresentados os valores de RSSI calculados para os *end nodes*, validando a análise realizada para o estudo de caso do algoritmo de múltiplas tentativas.

Figura 22 – Resultados de convergência do algoritmo genético.



Fonte: elaborada pelo autor.

No que diz respeito a convergência do algoritmo genético implementado, pode se analisar a partir dos gráficos apresentados na Figura 22, que essa estratégia de minimização é mais eficiente, pois houve uma diminuição na quantidade de equipamentos necessários para realizar a cobertura completa da rede IoT.

Ressalta-se que o número de gráficos apresentado é igual ao número de *gateways* sugeridos para solução, haja vista que o posicionamento deste é definido a partir da execução de um processo evolutivo, que a cada execução decrementa o valor de perda total da rede, de forma proporcional ao número de *end nodes* que ainda não foram atendidos pelo *gateway* sugerido, isto é, na Figura 22 (a) o valor máximo de perda calculado para os piores indivíduos da população foi de aproximadamente 171.000db, enquanto que na Figura 22 (b) este valor foi calculado aproximada à 107.000db, já na Figura 22 (c) este valor novamente teve uma queda chegando à um valor aproximado de 81.000db, na Figura 22 (d) este valor novamente foi decrementado chegando a resultante de 34.000db e na Figura 22 (e) o valor final máximo para os piores indivíduos da população manteve-se na faixa de 34.000db. Isso implica, que a cada convergência do algoritmo o número de dispositivos não atendidos era reduzido de forma proporcional.

Em ambos os casos, otimização por algoritmo de múltiplas ou algoritmo genético, é possível se emitir um relatório final com as principais informações e conclusões acerca da simulação executada, um exemplo deste relatório é apresentado no Apêndice desta dissertação. Neste documento é possível constatar qual o posicionamento ideal dos *gateways* necessários para implantação da rede IoT, a partir dos seus respectivos pontos geográficos (latitude e longitude).

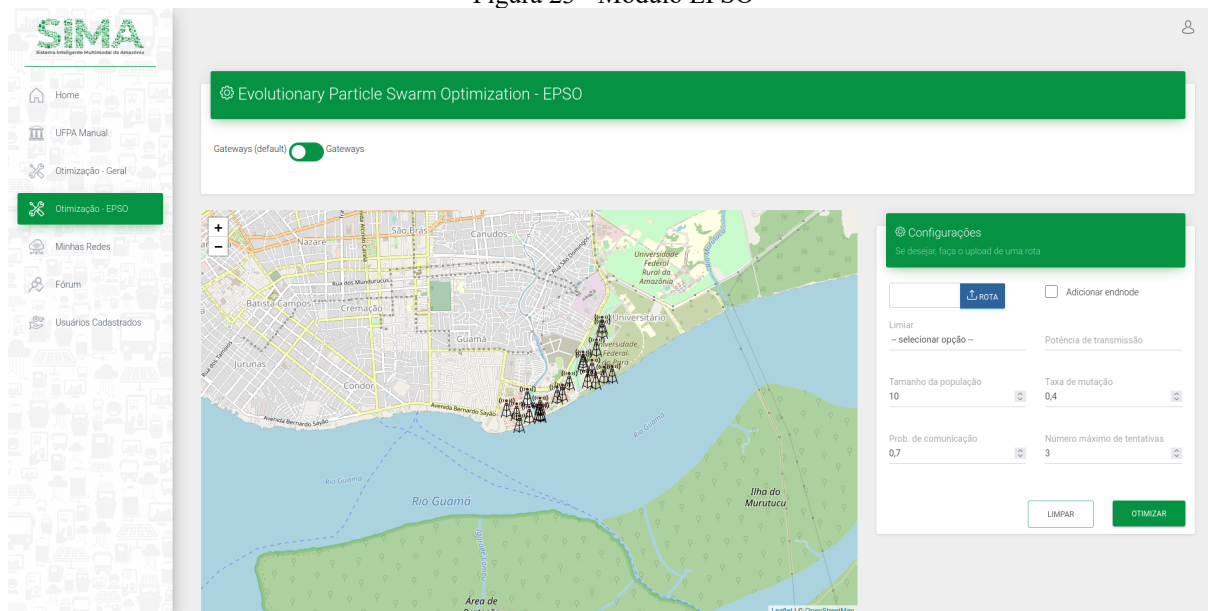
Comparando os resultados obtidos pelo algoritmo genético e de múltiplas tentativas, é possível observar que o algoritmo genético retornou melhores resultados, pois o número de *gateways* para cobrir a mesma área foi reduzido em um, o que impacta diretamente no custo de implantação da solução. Embora o de múltiplas tentativas tenha dado um resultado realístico, a aplicação do algoritmo genético também demonstrou que é possível se obter a mesma qualidade de sinal, mas com menos infraestrutura e equipamentos, o que reforça a aplicação de soluções baseadas em otimizadores para planejamento de redes IoT.

3.5.5 Otimização EPSO

O *software* também disponibiliza um módulo de simulação por EPSO, a Figura 23, apresenta a otimização por meio do algoritmo no qual, na área de configuração superior o usuário pode selecionar o posicionamento de *gateways*, entre os posicionamentos previamente indicados ou posicionamento manual, realizado por meio do clique no mapa, o algoritmo de EPSO realiza os cálculos de simulação com base no modelo de propagação proposto por Souza (2010) executando uma análise combinatória, que, no final indicará a quantidade de *gateways*

que obtiveram melhor desempenho quanto à comunicação dos *end nodes*, assim como seus respectivos pontos geográficos. O pseudocódigo deste algoritmo é apresentado nos anexos desta dissertação.

Figura 23 - Módulo EPSO



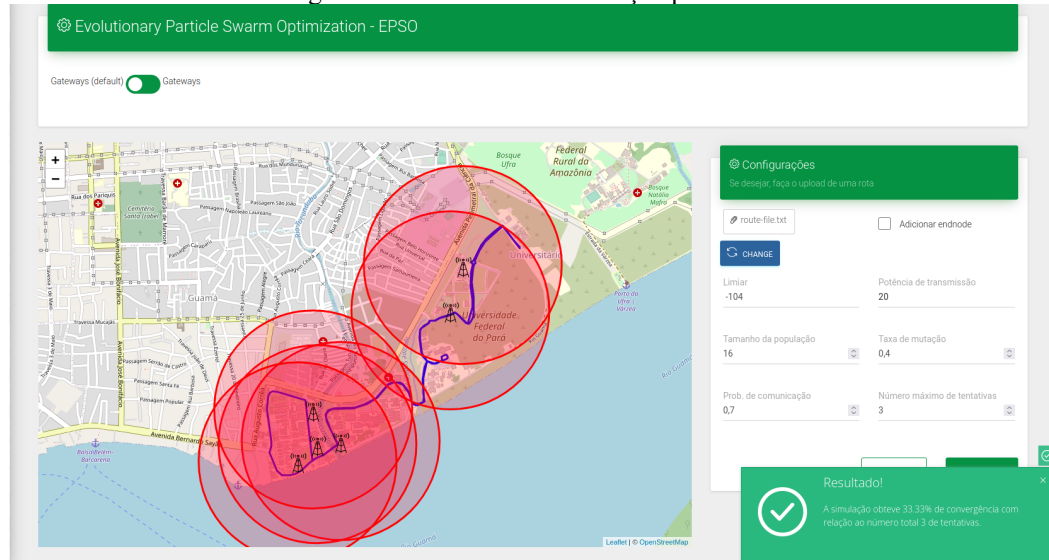
Fonte: elaborada pelo autor.

Conforme apresentado na Figura 20, na área de configuração à direita do mapa, é possível que o usuário realize o *upload* de um arquivo com as posições geográficas dos *end nodes* a serem atendidos pela rede, assim como, estas localizações podem ser inseridas de forma manual pelo usuário a partir da seleção do campo “Adicionar endnode”. Além disso, o usuário deve selecionar o valor de limiar a ser utilizado, bem como definir a potência de transmissão da rede, o tamanho da população de possíveis soluções, a taxa de mutação, probabilidade de comunicação e número máximo de tentativas que o algoritmo deve ser executado para estimar os resultados da simulação.

3.5.5.1 Cenário de Aplicação

Para exemplificação do seu funcionamento, foi realizado o *upload* do arquivo que geolocalizações referente à rota realizada pelo ônibus circular da UFPA, utilizando como candidatos para solução *gateways* previamente definidos para testes, e como parâmetros para simulação foram definidos o valor de limiar -104, potência de transmissão 20db, tamanho da população 16 indivíduos, taxa de mutação 0.4, probabilidade de comunicação 0.7 e número de tentativas para convergência igual a 3.

Figura 24 - Resultado da simulação por EPSO



Fonte: elaborada pelo autor.

A Figura 23 apresenta a localização dos *gateways* que o algoritmo EPSO considerou como pontos viáveis para instalação, além disso, o *software* também apresenta o *tooltip* indicando que a simulação obteve 33,33% de convergência, ressaltando que das três tentativas de simulação apenas uma conseguiu estabelecer um resultado consideravelmente adequado. Isso se dá por conta da finalidade do algoritmo de EPSO que busca minimizar os parâmetros de configuração da rede IoT, com relação ao número de *gateways* candidatos a solução. Dessa forma a cada iteração do algoritmo o resultado é equivalente a Figura 24.

Figura 25 - Exemplo de solução/indivíduo EPSO

Número de soluções indicadas

6	2	7	21	10	5	11	9	15	12	3
---	---	---	----	----	---	----	---	----	----	---

Índices das soluções

Fonte: elaborada pelo autor.

Isto é, um vetor que na primeira posição indica o número de *gateways* considerados aptos para instalação, e na sequência a organização dos índices de *gateways* da população, neste caso, o primeiro índice nos indica que apenas 6 *gateways* podem ser escolhidos para compor a rede IoT.

Após a simulação via EPSO, os resultados apresentaram um conjunto de *gateways* candidatos, assim permitindo ao especialista tomar a decisão final de onde realizar a

implantação. Deste modo, também levando em consideração as restrições presentes nos locais candidatos para instalação dos equipamentos.

4 CONCLUSÃO

Inúmeras pesquisas e projetos são realizados a fim que seja possível disponibilizar uma arquitetura de comunicação segura, consistente e de baixo custo operacional e financeiro; neste trabalho apresentamos uma ferramenta fundamentada em algoritmos de otimização visando facilitar o processo de planejamento e construção de redes IoT. De acordo com o melhor do nosso conhecimento, esta dissertação de mestrado é o primeiro esforço em se garantir um *software* composto por um conjunto de soluções baseadas em conhecimento do especialista, medições realísticas e otimizadores com o intuito de auxiliar no processo de planejamento e implantação de redes IoT.

Para isso, foram realizadas medições no *smart campus* da UFPA, com finalidade de se analisar e inferir parâmetros de limiares adequados para cada modelo de propagação integrado ao *software* apresentado, além disso, para sua implementação foram realizadas reuniões técnicas com os integrantes do projeto a fim de que fossem estabelecidos os requisitos fundamentais do *software*. E então, foi definido o modelo lógico do banco de dados que o software necessitaria, no processo de desenvolvimento do *software* foram implementados algoritmos de otimização baseados em estratégias evolutivas e múltiplas tentativas, que foram testados a partir do painel *web* desenvolvido.

Os resultados obtidos demonstram que uma etapa de planejamento bem consolidada, constituída das etapas: de aferições numéricas; análise preliminar dos dados obtidos e simulações que obedeçam a especificidade do ambiente de teste, podem ponderar uma solução que atenda todos os objetivos de comunicação esperados.

Com isso, é possível concluir que o *software* proposto é uma ferramenta útil para o planejamento de redes IoT, a partir da criação de cenários de simulação presentes em uma interface gráfica que permitem a configuração da área de interesse para posicionamento do *gateway*, determinação do posicionamento de *end nodes*, e criação de áreas de exclusão onde o *gateway* não poderá ser instalado. Sendo este, o primeiro *software* capaz de se planejar visualmente a posição dos equipamentos de comunicação LoRa em determinada área.

Além disso, é possível concluir que dados medidos no campo e aplicados aos modelos de propagação podem ser replicados como ferramentas de simulação computacional, sendo retornados valores realísticos e aderentes ao cenário sob investigação.

Também é possível concluir que os algoritmos de múltiplas tentativas, genético e EPSO são capazes de minimizar o custo da perda de sinal, ocasionando na determinação da menor quantidade de *gateways* e melhor posicionamento dos mesmos dentro da área de interesse, isto é, respeitando a minimização no número de *gateways* e reduzindo os custos associados a perda de sinal.

É possível concluir que não há apenas uma forma de se planejar as redes IoT, com isso, se faz necessário oferecer diferentes alternativas para os planejadores, visto que para cada situação pode existir diferentes restrições como limitação nos pontos para instalação dos *gateways*, ou oriundas da natureza a partir da não permissão de construção em áreas de mata ou rios. Deste modo, sendo útil uma ferramenta flexível para a tomada de decisão do especialista.

Como trabalhos futuros, espera-se aprimorar o simulador a partir da adição de mais modelos de propagação, assim como, a adição de novos algoritmos de otimização.

REFERÊNCIAS

AFZAL, Bilal et al. Enabling IoT platforms for social IoT applications: Vision, feature mapping, and challenges. *Future Generation Computer Systems*, v. 92, p. 718-731, 2019.

BANDEIRA, Lucilene Klenia Rodrigues; ARAÚJO NETO, Mario de Sousa. SMART CAMPUS NO BRASIL: A PERCEPÇÃO DOS GESTORES DAS IFES. *Perspectivas em Gestão & Conhecimento*; v. 10, n. 3 (2020); 189-204, v. 24, n. 2, p. 204-189.

BARBOSA, Jacilino Estumano et al. O uso do solo na cidade universitária Prof. José da Silveira Netto: contribuições para a gestão de espaços institucionais. 2012.

BERNARDES, Wellington Maycon Santos. Algoritmo enxame de partículas discreto para coordenação de relés direcionais de sobrecorrente em sistemas elétricos de potência. 2013. Tese de Doutorado. Universidade de São Paulo.

BOR, Martin; VIDLER, John Edward; ROEDIG, Utz. LoRa for the Internet of Things. 2016.

CASTRO, Bruno Souza Lyra. Modelo de propagação para redes sem fio fixas na banda de 5,8 GHz em cidades típicas da região amazônica. 2010

CISCO. O que é WiFi. Disponível em: https://www.cisco.com/c/pt_br/products/wireless/what-is-wifi.html. Acesso em: 14 de jun. 2022.

CLÁUDIO DA SILVA, P. et al. Implementação de uma rede IoT LoRaWAN para integração com o núcleo 5G. In: *Anais da IX Escola Regional de Informática de Goiás*. SBC, 2021. p. 185-194

COSENZA, Vivian Barreiros. Estudo da cobertura do sinal de 14 GHz em ambientes fechados. 2017.

CURSO ARDUINO. Disponível em: <https://ro.onlinecheapbest.ru/content?c=pinos%20i%20o%20do%20arduino%20tipo%20de%20sinais&id=9>. Acesso em 03 de jun de 2022.

D. Ugrenovic and G. Gardasevic, "CoAP protocol for Web-based monitoring in IoT healthcare applications," 2015 23rd Telecommunications Forum Telfor (TELFOR), 2015, pp. 79-82, doi: 10.1109/TELFOR.2015.7377418.

DA PORCIÚNCULA, Cleber B. et al. Constrained application protocol (coap) no arduino uno r3: Uma análise pratica. In: **Anais do V Workshop Pré-IETF**. SBC, 2018.

DE MENDONÇA, Ricardo Augusto Ribeiro. Levantamento de requisitos no desenvolvimento ágil de software. Semana da Ciência e Tecnologia da PUC Goiás, p. 12, 2014

DODSON, Sean. The internet of things: A tiny microchip is set to replace the barcode on all retail items but opposition is growing to its use. The Guardian, v. 9, p. 2003, 2003.

DRAGINO SHIELD. Disponível em <https://www.dragino.com/products/lora/item/102-lora-shield.html>. Acesso em 04 de jun de 2022.

ELMASRI, Ramez et al. Sistemas de banco de dados. 2005.

EMBARCADOS, Protocolos de Rede sem fio de IoT. Disponível em: <https://www.embarcados.com.br/protocolos-de-rede-sem-fio-de-iot/#Ecossistema-IoT>. Acesso em: 3 jun. 2022

ESTEVAM, S. O.; GUEDES, M. CARACTERÍSTICAS E VANTAGENS DO CABEAMENTO IRRADIANTE. Interação - Revista de Ensino, Pesquisa e Extensão, v. 16, n. 16, p. 61 - 78, 15 fev. 2019.

FERREIRA, F. H. C.; ARAÚJO, R. M. Campus Inteligentes: Conceitos, aplicações, tecnologias e desafios. **RelaTe-DIA**, [S. l.], v. 11, n. 1, 2018. Disponível em: <http://seer.unirio.br/monografiasppgi/article/view/7147>. Acesso em: 20 ago. 2022.

G. R. Maccartney, T. S. Rappaport, S. Sun and S. Deng, "Indoor Office Wideband Millimeter-Wave Propagation Measurements and Channel Models at 28 and 73 GHz for Ultra-Dense 5G Wireless Networks," in IEEE Access, vol. 3, pp. 2388-2424, 2015, doi: 10.1109/ACCESS.2015.2486778

HEUSER, Carlos Alberto. Projeto de Banco de Dados, 6ª edição. Instituto de Informática. Editora Sagra, 1998.

JIN, Jiong et al. An information framework for creating a smart city through internet of things. IEEE Internet of Things journal, v. 1, n. 2, p. 112-121, 2014

KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: Proc. IEEE International Conference on Neural Networks. Perth, Australia: [s.n.], 1995

LEITE, JR Emiliano; MARTINS, Paulo S.; URSINI, Edson L. A Internet das Coisas (IoT): Tecnologias e Aplicações. School of Technology, University of Campinas (UNICAMP), 2017.

LOPES, Amanda Ingrid. Estudo de área de cobertura de dispositivo LORA para ambientes de Smart Campus. 2019. Trabalho de Conclusão de Curso. Universidade Federal do Rio Grande do Norte.

LORA ALLIANCE TECHNICAL MARKETING WORKGROUP. What is LoRaWAN? Disponível em: <https://lora-alliance.org/wp-content/uploads/2020/11/what-is-lorawan.pdf>, 2015. Acesso em 15 de jun de 2022.

MACHADO, Felipe Nery Rodrigues; DE ABREU, Mauricio Pereira. Projeto de banco de dados: uma visão prática. Saraiva Educação SA, 2004.

MADAKAM, Somayya et al. Internet of Things (IoT): A literature review. Journal of Computer and Communications, v. 3, n. 05, p. 164, 2015.

MATNI, Nagib et al. LoRaWAN gateway placement model for dynamic Internet of Things scenarios. Sensors, v. 20, n. 15, p. 4336, 2020.

MENEZES, João Tribouillet Marcial de; COSTA, Pedro Henrique Lira da. Desenvolvimento de modelo hierárquico de middlewares com aplicação de Edge Computing para redes IoT. 2019

MOUMANE, Karima; IDRI, Ali; ABRAN, Alain. Usability evaluation of mobile applications using ISO 9241 and ISO 25062 standards. SpringerPlus, v. 5, n. 1, p. 1-15, 2016.

NEVES, Ana Régia de M.; SARMANHO, Kaê U.; JR., Francisco C. Nascimento; MEIGUINS, Bianchi S. Iniciativa Smart Campus: um estudo de caso em progresso na Universidade Federal do Pará. In: WORKSHOP DE COMPUTAÇÃO URBANA (COURB), 1. 2017, Belém. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação, 2017. ISSN 2595-2706

NORTE ENERGIA. Norte Energia e UFPA implantam primeiro “corredor verde” de mobilidade elétrica do Pará. Disponível em: <https://www.norteenergiasa.com.br/pt->

br/imprensa/releases/norte-energia-e-ufpa-implantam-primeiro-corredor-verde-de-mobilidade-eletrica-do-para-101072. Acesso em: 30 de jun. 2022.

PACHECO, André. Otimização por enxame de partículas – PSO. Disponível em: <http://computacaointeligente.com.br/algoritmos/otimizacao-por-enxame-de-particulas/>.

Acesso em: 07 de jun. 2022

Pinto. Gustavo. Middleware de IoT: a ponte para a digitalização. Disponível em: <https://v2com.com/2022/03/22/middleware-de-iot-a-ponte-para-a-digitalizacao/>. Acesso em: 07 de jun. 2022

PROENÇA, Matheus Hipólito. Arquitetura IoT para aplicação em Smart Campus. 2022.

Projeto de Hardware, como escolher qual plataforma utilizar. ANA, Arles Sant, 2016. Disponível em: <https://www.embarcados.com.br/projeto-de-hardware-como-escolher-qual-plataforma-utilizar/>. Acesso em: 07 jun. 2022

RODRIGUES, Raphael Augusto Nascimento; CARVALHAES, Millys Fabrielle Araújo. Análise e Complexidade de Algoritmos: Backtracking e Força Bruta. Revista Ada Lovelace, v. 1, p. 45-48, 2017.

ROTTA, Giovanni; CHARÃO, Andrea; DANTAS, Mario. Um Estudo sobre Protocolos de Comunicação para Ambientes de Internet das Coisas. In: ESCOLA REGIONAL DE ALTO DESEMPENHO DA REGIÃO SUL (ERAD-RS), 2017, Ijuí. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2017. ISSN 2595-4164.

S. Arvind and V. A. Narayanan, "An Overview of Security in CoAP: Attack and Analysis," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), 2019, pp. 655-660, doi: 10.1109/ICACCS.2019.8728533.

SANTO, Walter do Espírito; ORDOÑEZ, Edward; RIBEIRO, Admilson. Uma revisão sistemática sobre a Segurança nos Protocolos de Comunicação para Internet das Coisas. Journal on Advances in Theoretical and Applied Informatics, [S.l.], v. 4, n. 1, p. 1-9, oct. 2018. ISSN 2447-5033. Available at: <https://revista.univem.edu.br/jadi/article/view/2482>. Date accessed: 03 june 2022. doi: <https://doi.org/10.26729/jadi.v4i1.2482>.

SCHILLER, Jochen H. Mobile communications. Pearson education, 2003.

SEYE, M. R. et al. A study of lora coverage: Range evaluation and channel attenuation model. In: 2018 1st International Conference on Smart Cities and Communities (SCCIC). [S.l.: s.n.], 2018. p. 1–4.

SILVA, Deivison Feitosa et al. Desafios na implementação de projetos de internet das coisas e ações para superá-los. 2018.

SOARES, Afonso Januário Pujoni et al. Avaliação da propagação do sinal LoRa e desenvolvimento de um método para auxiliar o planejamento de redes IoT usando otimização do modelo de Hata. 2019

SOMMERVILLE, Ian. Engenharia de Software. Oitava Edição. 2007

TANENBAUM, A. S. – Redes de Computadores – 4ª Ed., Editora Campus (Elsevier), 2003

TUTUI, MARIA FERNANDA SILVA. Utilização da tecnologia LoRaWAN para manutenção preditiva. 2019.

VENCESLAU, Franklin Arthur Mendes et al. Sistema para suporte à implantação e monitoramento de aplicações de campus inteligente baseado em protocolos de IoT. 2021

VINICIUS ANGELO DE O.MUNIZ., L. L. B. M. M. R. F. O. A. PROTOCOLOS DE COMUNICAÇÃO PARA INTERNET OF THINGS (IOT). Intercursos Revista Científica, [S. l.], v. 17, n. 1, 2019. Disponível em: <https://revista.uemg.br/index.php/intercursosrevistacientifica/article/view/3712>. Acesso em: 3 jun. 2022.

Apêndice

Pontos geográficos rota ônibus circular

Exemplo de relatório final exportado do *software*

Pontos geográficos rotas ônibus e barco circular

Disponíveis em: https://1drv.ms/u/s!AsuuwVz7jkr_joIaITFRFxebQ4zYwg?e=4eP0Uz

Pseudocódigo algoritmo de múltiplas tentativas

```
1  Entrada:
2      Matriz de pontos a serem cobertos
3      Lista de posições dos gateways
4      Parâmetros de configuração do gateway
5
6
7  Processamento:
8      for i = 0 até numeroGateways:
9          numeroEndNodes = tamanho(endNodes)
10
11         while haver endNodes sem cobertura faça:
12             raioGateway = calcula raio de cobertura do modelo de propagação
13
14             for i = 0 até numeroEndNodes:
15                 distancia = calculaDistanciaEntreEndnodeGateway
16
17                 if distancia > raioGateway:
18                     dispositivosForaDaAreaDeCobertura ← endNode
19
20             endNodes = dispositivosForaDaAreaDeCobertura
21
22         resultado ← posicaoGateway
23  Fim
```

Pseudocódigo algoritmo genético

```
1  Entrada:
2      Matriz de pontos a serem cobertos
3      Áreas a serem cobertas
4      Lista de posições dos gateways
5      Parâmetros de configuração do gateway
6
7  Processamento:
8      passo 1:
9          if primeira iteração:
10             gerar população de cromossomos por gateways candidatos distribuídos na área a ser coberta
11
12             calcular o valor de perda em dBm para todos os gateways candidatos em relação a todos os pontos de endnodes
13
14      passo 2:
15          if número de gerações alcançado:
16             selecionar o gateway candidato com a mínima perda a partir dos gateways candidatos
17             ir para o passo 3
18          else:
19             atualizar a população de cromossomos (cruzamento e mutação)
20             repetir os passos de 1 até 2
21
22      passo 3:
23          if todos os endnodes são cobertos pelo gateway candidato então:
24             Guardar o gateway candidato na matriz de gateways selecionados
25             Fim
26          else:
27             Podar a matriz de pontos a serem cobertos mantendo somente os pontos que não foram atendidos pelo gateway candidato
28             Guardar o gateway candidato na matriz de gateways selecionados
29             Atualizar os cromossomos (cruzamento e mutação)
30             Repetir os passos até o Fim
31  Fim
```

Pseudocódigo algoritmo EPSO



```
1  Entrada:
2      Matriz de pontos a serem cobertos
3      Lista de posições dos gateways
4      Parâmetros de configuração do gateway
5
6  Processamento:
7
8      NP = defineFrentePareto
9
10     for i = 0 até numeroGeracoes:
11         for j = 0 até NP:
12             resultado ← executaEPSO
13
14     calculaPercentualResultado
15  Fim
```