

**CENTRO FEDERAL DE EDUCAÇÃO
TECNOLÓGICA DE MINAS GERAIS – CAMPUS TIMÓTEO
Curso de Graduação em Engenharia de Computação**

Augusto César Castro Ribeiro

**UM OBJETO DE APRENDIZAGEM COMO INSTRUMENTO DE ENSINO DE
REDES DE SENSORES SEM FIO**

TIMÓTEO
2017

Augusto César Castro Ribeiro

**UM OBJETO DE APRENDIZAGEM COMO INSTRUMENTO DE ENSINO DE
REDES DE SENSORES SEM FIO**

Monografia apresentada ao Curso de Engenharia de Computação do Centro Federal de Educação Tecnologia de Minas Gerais, como requisito para obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Adilson Mendes Ricardo.

Co-orientadores: André Rodrigues da Cruz e Viviane Cota Silva

TIMÓTEO
2017

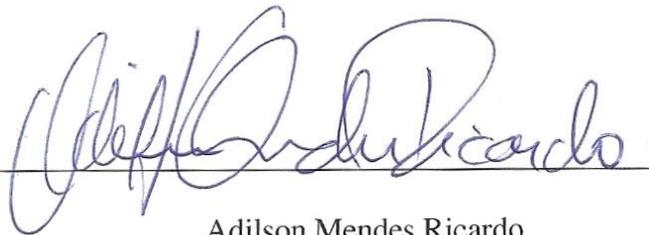
Augusto César Castro Ribeiro

**UM OBJETO DE APRENDIZAGEM COMO INSTRUMENTO DE ENSINO DE
REDES DE SENSORES SEM FIO**

Monografia apresentada ao Curso de Engenharia de Computação do Centro Federal de Educação Tecnologia de Minas Gerais, como requisito para obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Adilson Mendes Ricardo.

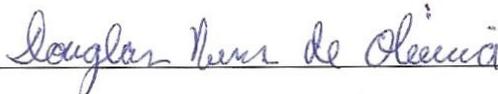
Co-orientadores: André Rodrigues da Cruz e Viviane Cota Silva



Adilson Mendes Ricardo



Viviane Cota Silva



Douglas Nunes Oliveira

Agradeço a Deus por ter colocado pessoas
incríveis no meu caminho.

AGRADECIMENTOS

Agradeço primeiramente a Deus, sem ele nada disso seria possível, foi ele quem esteve me ajudando nos momentos mais difíceis, sem Deus nada nesta jornada seria possível. Aos meus pais que estiveram em todos os momentos me dando forças para prosseguir, e ajudando para que não desistisse da caminhada.

Aos amigos que fiz na faculdade que pretendo levar por toda a vida, partilhamos momentos bons, momentos ruins, mas sempre juntos, conseguimos vencer mais este obstáculo, obrigado pelo apoio e que Deus abençoe a jornada de cada um que passou comigo durante este tempo.

As minhas irmãs Alane e Andressa que sempre estiveram presentes, me ouvindo e apoiando, agradeço a Deus por ter vocês na minha vida.

Ao meu orientador Adilson por me apoiar na mudança de projeto durante o percurso e sempre ser paciente e compassivo comigo em suas orientações e aos meus co-orientadores Viviane e André por terem me ajudado na escolha para este projeto e aberto minha mente para novos horizontes, obrigado por terem iluminado meu caminho, pelos momentos que estiveram comigo e pelas ideias que me ajudaram a conseguir finalizar mais um projeto.

Aos professores, que com cuidado e zelo garantiram meu aprendizado e me ensinaram a persistir em meus objetivos, a não desistir dos meus sonhos e a sempre prosseguir.

A todos que passaram pelo caminho neste projeto e forneceram contribuições para que o mesmo fosse realizado, fica a minha gratidão.

RESUMO

A busca por conhecimento na área de Redes de Sensores Sem Fio (RSSF) tem cada vez se tornado mais constante com a crescente busca por “*Internet of Things*”, e a fim de levar novos pesquisadores a conhecer conceitos básicos de RSSF e aumentarem seu interesse pela área, este trabalho propõe o desenvolvimento e análise de um objeto de aprendizagem para o ensino de Redes de Sensores Sem Fio (RSSF), utilizando-se da biblioteca gráfica OpenGL. Este trabalho vem apresentar uma nova forma de ensinar os conceitos de RSSF a estudantes da área, além de propiciar a estudantes e professores o conhecimento de conceitos empregados na construção do Objeto de aprendizagem. Cabe ao usuário configurar as posições dos nós sensores e as respectivas distâncias máximas de envio que serão colocados na demonstração, os dados que serão enviados e a posição do Access Point. No presente trabalho apresentou-se também uma avaliação do Objeto de aprendizagem gerado, através da aplicação do mesmo para o ensino do conteúdo em uma aula da disciplina de Redes II, e aplicação de questionários aos alunos participantes. Observou-se com esta pesquisa um crescente conhecimento e interesse pela área através da aplicação da aula com o objeto gerado. Através do trabalho pretende-se futuramente facilitar a geração de novos simuladores de RSSF gráficos, onde o usuário poderá verificar dados importantes na utilização de RSSF como consumo de energia dos nós sensores e melhores configurações de disposições dos dispositivos na rede.

Palavras-chave: Redes de Sensores Sem Fio, RSSF, redes, Network, Sensors, objetos de aprendizagem, computação gráfica, simulação.

ABSTRACT

The pursuit of knowledge in Wireless Sensor Networks (WSN) it's becoming more frequent through the increase of search for the Internet of Things, and with the purpose of making researchers know the basic concepts of WSN and increase their interest in it, this article suggests the development and analysis of a learning object, using the OpenGL graphics library. This paper proposes a new way to help the teachings of the WSN concepts to students. The method described in this work allows the user to configure the positions of sensor nodes e the respective maximum distances of transmission that will be put in the demonstration, the data that will be sent, and the position of the Access Point. Also, it will be shown in this article the evaluation of the learning object, through the application of it as a content to a class of the course "Redes II", and a test to the enrolled students. It was noted that with this research it happened an increase of interest for this area, after using the learning object in a class. Thus, it is planned to make it easier the generation of new WSN graphic simulators, where the user can verify important data, such as, energy consumption of the sensor nodes and make better adjustments of the settings of the networking devices.

Key words: Wireless Sensor Networks, sensors, teachings, learning object.

LISTA DE FIGURAS

FIGURA 1– Tendência de procuras por temas no <i>Google</i> segundo o <i>Google Trends</i>	4
FIGURA 2- <i>Internet of things</i> , uma tendência de procuras.....	4
FIGURA 3- Procura por sensores acompanha a procura por internet das coisas.	5
FIGURA 4 – RSSF plana e Hierárquica.	9
FIGURA 5 - Fluxo de informação.	14
FIGURA 6 - Menu inicial do programa.	35
FIGURA 7 - Ilustração do funcionamento do programa antes de utilizar os recursos.	36
FIGURA 8 - Ilustração do programa após a utilização de recursos.	37
FIGURA 9 - O funcionamento da rede com todos os nós.	38
FIGURA 10 - O funcionamento da rede após a exclusão de nós sensores.	39
FIGURA 11 - Caso onde não ocorre o roteamento ao Access Point	40
FIGURA 12 - Grau de conhecimento de computação dos entrevistados.	41
FIGURA 13 - Grau de conhecimento anterior em RSSF.	41
FIGURA 14- Grau de conhecimento em RSSF após a aula direcionada.	42
FIGURA 15 - Grau de interesse antes da utilização do software.	42
FIGURA 16 - Grau de interesse pela área após a aplicação do software.	43
FIGURA 17 - Conhecimento prévio dos termos de RSSF.	44
FIGURA 18 - Conhecimento dos termos após a apresentação do trabalho.	45

LISTA DE TABELAS

TABELA 1 – Configurações de RSSFs (Adaptado de RUIZ, 2003 e VERONA, 2010). 9

TABELA 2 - Ficha de contextualização do OA. 27

LISTA DESIGLAS

ABR - Associativity-Based Routing

ADV - Adaptive Distance Vector Routing

AODV – Ad Hoc On-demand Distance Vector Routing

AP – Access Point

APP – Área de Preservação Permanente

CBRP - Cluster Based Routing Protocol

CEDAR - Core-Extraction Distributed Ad hoc Routing

CEFET – Centro Federal de Educação Tecnológica

CGSR - Clusterhead Gateway Switch Routing

DDR - Distributed Dynamic Routing Algorithm

DREAM - Distance Routing Effect Algorithm for Mobility

DSDV - Destination Sequenced Distance Vector Routing

DSR - Dynamic Source Routing

EA – Elemento Autônômico

FORP - Flow Oriented Routing Protocol

FSLS - Fuzzy Sighted Link State Algorithms

FSR - Fisheye State Routing

GSR - Global State Routing

GPSR - Greedy Perimeter Stateless Routing

HSR - Hierarchical State Routing

INPE – Instituto Nacional de Pesquisas Espaciais

IoT – Internet of Things

LANMAR - Landmark Routing Protocol

LAR - Location Aided Routing

LMR - Lightweight Mobile Routing

OA – Objeto de Aprendizagem

OLSR - Optimized Link State Routing

OOP - Programação Orientada a Objetos

RREP - Route Reply

RREQ - Route Requests

RRER - Router Error

RSSF– Redes de Sensores Sem Fio

SSR - Signal Stability Routing

STAR - Source Tree Adaptive Routing

TBRPF - Topology Broadcast Based on Reverse Path Forwarding

TORA - Temporally Ordered Routing Algorithm

TTL – Time to live

WAR - Witness Aided Routing

WRP - Wireless Routing Protocol

ZRP - Zone Routing Protocol

SUMÁRIO

1. INTRODUÇÃO	3
1.1. Pergunta de Pesquisa	5
1.2 Objetivos	6
1.2.1 Objetivos específicos	6
1.3 Justificativa	6
1.4 Apresentação do trabalho	6
2. FUNDAMENTAÇÃO TEÓRICA	8
2.1 Redes de sensores sem fio	8
2.1.1 Fundamentos de RSSF	8
2.1.2 Protocolos de RSSF	14
2.1.3 Broadcast	16
2.1.4 Funcionamento dos protocolos de roteamento	16
2.1.4.1 DSR	16
2.1.4.2 AODV	16
2.1.4.3 DSDV	18
2.2 Computação Gráfica	19
2.2.1 C++	19
2.2.1.1 Orientação a objeto	19
2.2.1.2 Classe	20
2.2.1.3 Funções	21
2.2.2 OpenGL	22
2.3 Objetos de aprendizagem	24
2.3.1 Tipos de objetos de aprendizagem	24
2.3.2 Características de um Objeto de Aprendizagem	25
2.3.3 Metodologia INTERA	26
3. TRABALHOS EXISTENTES	28
3.1 Ambientes de simulação	28
3.1.1 Network Simulator 3 (NS-3)	28

3.1.2 TraceGraph	28
3.1.3 MannaSim	28
3.2 Trabalhos relacionados	29
3.3 Revisão de literatura	29
4. METODOLOGIA	32
5. RESULTADOS	35
3.3 Resultados do Software	35
5.2 Resultados de Pesquisa	40
6. CONCLUSÃO	48
7. REFERÊNCIA BIBLIOGRÁFICA	49

1. INTRODUÇÃO

Nos últimos anos um dos temas mais recorrentes em se tratando de redes são aquelas compostas por sensores sem fio, que a cada dia, vem sendo mais utilizadas para diversos objetivos, desde o monitoramento de áreas agrícolas até mesmo a supervisão de aquecimento em áreas industriais.

Dado a relevância do tema, decidiu-se, a princípio, utilizar um ambiente de simulação, capaz de prever o alastramento de queimadas em áreas de preservação permanente. Para isso, iniciou-se uma pesquisa sobre o funcionamento de Redes de Sensores Sem Fio (RSSF) e, durante este processo de aprendizagem, foi observado que um dos problemas de novas pesquisas de RSSF é a falta de material prático de ensino.

Uma das dificuldades enfrentadas na pesquisa do tema é a não compreensão de conceitos relacionados ao assunto, bem como as diferentes arquiteturas abordadas e as configurações de RSSF. Sendo assim, o entendimento dos temas e pesquisas relacionadas à área torna-se cada vez mais difícil.

Segundo pesquisas no *software Google Trends* durante os últimos anos o interesse pela área de RSSF tem se mantido estável, porém, a tendência é que ele venha a crescer com a forte influência do interesse em internet das coisas, que utiliza profundamente os conceitos de RSSF, conforme apresentam os gráficos gerados pelo *Google Trends*, gráficos estes que demonstram um comparativo entre o número de pesquisas feitas no site www.google.com em um período de tempo, como podemos ver na figura de número 1; o interesse pelo tema “internet das coisas” avaliado pelo número de pesquisas nos últimos dois anos tem aumentado consideravelmente:

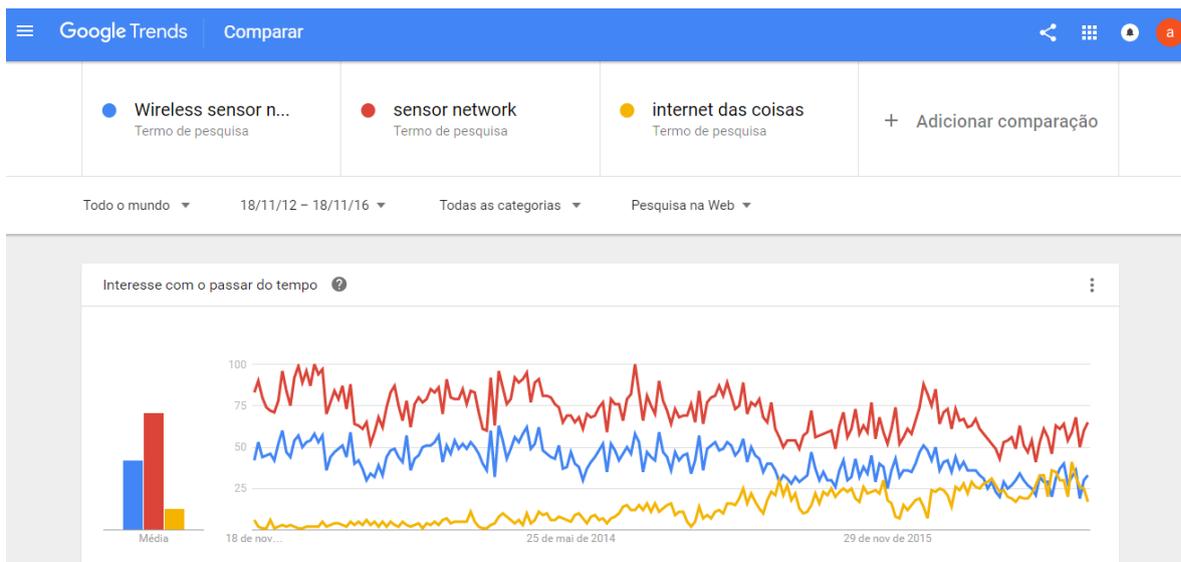


Figura 1. Tendência de procuras por temas no Google, segundo o Google Trends. Disponível no endereço eletrônico: <<https://trends.google.com/trends/>>.

É importante analisarmos também os gráficos gerados que demonstram o interesse por “*internet of things*” (internet das coisas). Segundo Bilinski “A Internet das coisas ou IoT, objetiva ligar todos os equipamentos eletrônicos que usamos no dia a dia, a internet ou bases de dados e com o uso de redes de sensores, processar essas informações e retornar benefícios aos usuários.” Podemos observar que o interesse pela área de *internet of things* tem aumentado consideravelmente nos últimos três anos, como podemos ver no gráfico da figura 2:

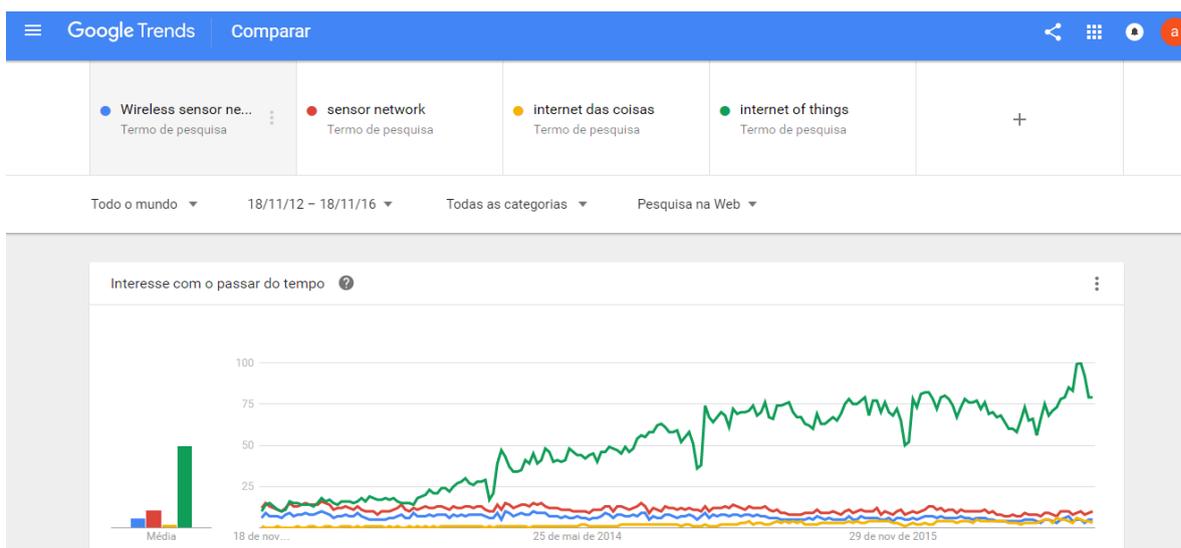


Figura 2. *Internet of things*, uma tendência de procuras. Disponível no endereço eletrônico: <<https://trends.google.com/trends/>>.

Podemos ver também que a busca pelo termo “*sensors*” acompanha a busca por “*internet of*

things” o que demonstra que os termos são diretamente ligados, e com a crescente busca por *internet of things* a tendência é que o interesse pela área de sensores também seja crescente conforme podemos ver no gráfico:

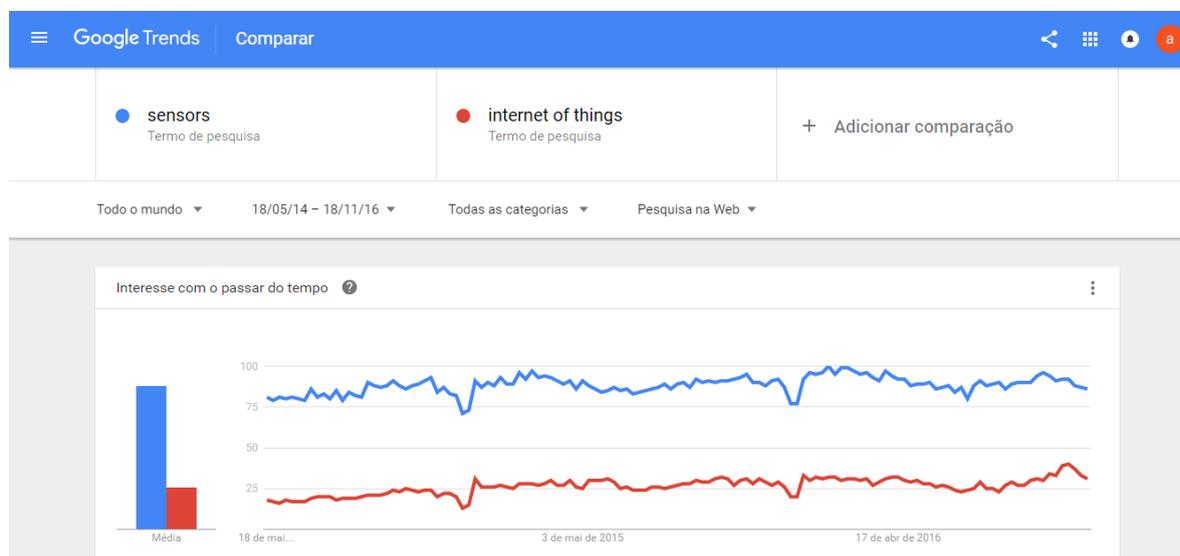


Figura 3. Procura por sensores acompanha a procura por internet das coisas. Disponível no endereço eletrônico: <<https://trends.google.com/trends/>>.

Devido à falta de material para a aprendizagem do conteúdo foi proposta então a criação de um objeto de aprendizagem (OA) capaz de demonstrar graficamente o funcionamento de uma RSSF no modelo *floodfill* com uma rede de topologia plana, conceitos que serão explicados posteriormente nos capítulos referentes à parte 2.1.

1.1. Pergunta de Pesquisa

Com este projeto pretende-se criar um OA capaz de fazer com que o sistema didático de ensino de RSSF nas disciplinas relacionadas a redes, se torne mais eficaz, fazendo com que os alunos ganhem um maior interesse pela área.

Pretende-se também criar um ambiente capaz de simular o envio de dados e o número de vezes em que um nó sensor é acessado dentro de um ambiente simulado, fazendo com que futuramente pessoas interessadas na área possam transformar este OA em um simulador real.

Além disso, futuramente, será possível que este simulador que possa vir a ser gerado possua dados como o consumo de energia de um nó sensor (tema de maior relevância na implementação de uma RSSF) que pode ser calculado de acordo com a quantidade de vezes

que o nó é acessado, a distancia cujo qual o mesmo envia o dado para o próximo nó sensor e especificações do sensor.

Para que isto ocorra pretende-se responder os seguintes questionamentos:

- Quais as maiores dificuldades na aprendizagem de RSSF?
- Como melhorar o ensino de RSSF?
- Como criar um OA capaz de gerar graficamente os dados passados pelo usuário?

1.2. Objetivo Geral

O principal objetivo deste projeto é criar um OA para que professores e alunos possam utilizar como recurso no ensino-aprendizagem de RSSF.

1.2.1. Objetivos Específicos

Especificamente pretende-se com este estudo:

- a) Suscitar um ambiente de simulação capaz de gerar diversos ambientes de posicionamento de nós sensores;
- b) Pesquisar e analisar técnicas e procedimentos existentes para criação de objeto de aprendizagem.
- c) Garantir que futuros estudantes possam implementar uma RSSF capaz de se adaptar à realidade apresentada da melhor maneira possível.
- d) Gerar um estudo completo capaz de ajudar futuros estudantes a avaliarem outras aplicações para RSSF.
- e) Despertar em novos alunos o interesse pela área.

1.3. Justificativa

Durante os últimos anos o interesse na área de RSSF vem aumentando e gerando um ganho contínuo de novas funcionalidades. Para que esse aumento no desenvolvimento da área seja cada vez maior é necessário levarmos novos estudantes ao aprendizado eficaz do funcionamento de uma RSSF.

1.4. Apresentação de Trabalho

Este trabalho visa a criação de um OA na área de RSSF, gerando assim, um maior conhecimento sobre o tema a partir de um processo que será dividido da seguinte maneira:

- Capítulo 2: serão apresentados os materiais teóricos necessários para compreensão do significado dos seguintes termos: Rede de sensores sem fio, Computação gráfica e objetos de aprendizagem.

- Capítulo 3: apresentará Ferramentas de simulação existentes e trabalhos relacionados.

- Capítulo 4: serão apresentados os métodos utilizados na pesquisa.

- Capítulo 5: serão discutidos os resultados obtidos.

- Capítulo 6: será dedicado a conclusões e trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

Para a fundamentação teórica utilizada no desenvolvimento do trabalho foram apresentados alguns conceitos bases dentro dos campos de conhecimentos em objetos de aprendizagem, RSSF e computação gráfica.

2.1. Redes de sensores sem fio

As RSSF surgiram com o objetivo de monitorar algum fenômeno, sendo muitas vezes utilizadas para monitorar dados em ambientes de áreas restritas, áreas de difícil acesso, ou áreas perigosas, sendo muito recorrentes em áreas como engenharia, controle ambiental, industrial, militar, entre outras.

2.1.1. Fundamentos de RSSF

Segundo Corson *et al.*, (1999 apud Verona 2010) “As RSSFs são consideradas uma subclasse das redes *ad-hoc*”. Verona (2010) ainda afirma que:

“As redes *ad-hoc* caracterizam-se pela descentralização do envio e recepção de dados entre os componentes atuantes na rede pela ausência de um ponto de coleta que centraliza os dados. Nesse tipo de rede qualquer terminal pode realizar roteamento de dados, agindo de forma colaborativa com as solicitações de envio e recepção de terminais vizinhos”.

Segundo Ruiz *et al.* (2004 apud Verona 2010) uma RSSF é composta por uma larga escala de nós sensores. Estes nós sensores são distribuídos nas diversas áreas para análise de dados, sendo essas extensões as mais diversas, podendo ser desde áreas abertas até locais onde é impossível que o ser humano pratique reparos. Os nós sensores consistem em componentes capazes de coletar dados em ambientes durante um tempo de vida e levar os dados a um ponto de acesso principal (*Access Point*), que irá processar os dados.

Segundo Verona (2010) existem diversos tipos de nós sensores, cada um com uma especificação diferente. Alguns nós sensores, por exemplo, são capazes de, antes de enviar informações, processá-las a fim de enviar as mais concisas para o *Access Point* (AP), como por exemplo, nós sensores que captam o índice pluviométrico transformando esse índice para mm por dia e só depois de processarem esta informação enviam para o AP. Outros nós sensores apenas enviam as informações direto para o AP e deixam com que o mesmo processe as informações.

Além dos diferentes tipos de nós sensores existem também diferentes topologias que podem ser utilizadas na distribuição dos nós sensores e AP. A primeira coisa a se levar em conta é a utilização de uma RSSF hierárquica ou plana. Segundo Ruiz (2003) e Braga (2006), as RSSF do tipo plana possuem nós sensores distribuídos pela área. Esses nós se conectam para levar os dados até o AP se conectando de forma direta para levar os dados. Sendo assim, todos os nós possuem a mesma preferência de envio. As RSSF do tipo hierárquica por sua vez elegem líderes dentro de grupos de nós sensores. Os nós sensores enviam os dados para os líderes de grupo que por sua vez levam os dados ao AP para que o mesmo faça o processamento das informações enviadas. Podemos ver melhor como ocorre este processo na figura 4:

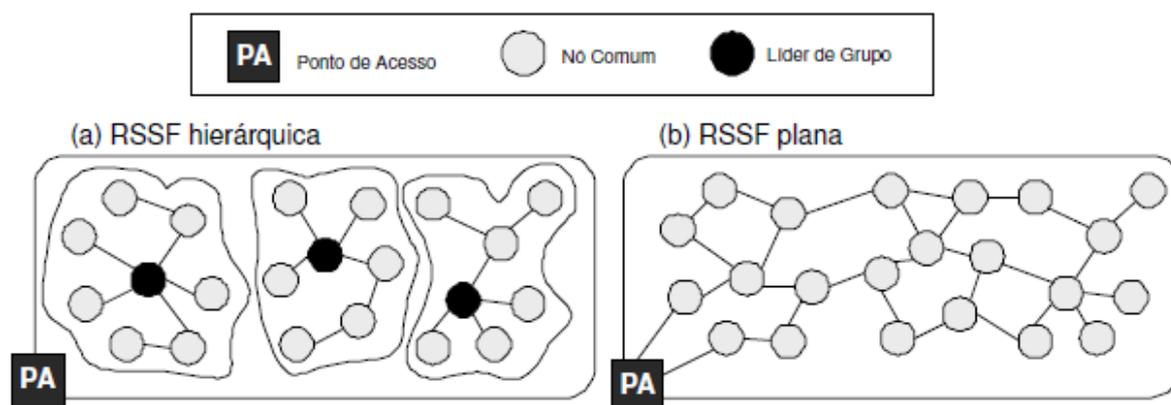


Figura 4. Redes de Sensores Sem Fio (Braga *et al.*, 2006)

Existem outras características que devem ser levadas em conta dentro de uma RSSF. Essas características são mostradas na tabela a seguir:

Tabela 1. Configurações de RSSFs (Adaptado de RUIZ, 2003 e VERONA, 2010)

Composição	Homogênea	Rede composta por dispositivos que apresentam a mesma capacidade de hardware. Eventualmente os nós podem executar software diferenciado.
	Heterogênea	Rede composta por nós com diferentes capacidades de hardware.
Mobilidade	Estacionária	Todos os dispositivos permanecem no local onde foram depositados durante todo o tempo de vida da rede.
	Móvel	Rede em que os dispositivos podem ser deslocados do local onde inicialmente foram depositados.
Densidade	Balanceada	Rede que apresenta uma concentração e distribuição de dispositivos por unidade de área considerada ideal para melhor funcionamento da rede.
	Densa	Rede que apresenta uma alta concentração de dispositivos

		por unidade de área.
	Esparsa	Rede que apresenta baixa concentração de dispositivos por unidade de área.
Distribuição	Irregular	Rede que apresenta uma distribuição não uniforme de nós sobre a área monitorada.
	Regular	Rede que apresenta uma distribuição uniforme de nós sobre a área monitorada.
Tamanho	Pequena	Rede composta de uma centena de elementos de rede.
	Média	Rede composta de centenas a mil elementos de rede.
	Grande	Rede composta por milhares de elementos de rede.
Coleta de dados	Periódica	Os dispositivos da rede coletam os dados de interesse em intervalos regulares de tempo;
	Contínua	Os dispositivos da rede coletam os dados de interesse em intervalos contínuos;
	Reativa	Os dispositivos da rede coletam os dados de interesse quando ocorre algum evento ou quando solicitado pelo observador
Tipos de conexão	Simétrica	Todos os dispositivos pertencentes à rede possuem a mesma capacidade ou distância de transmissão, com exceção do AP.
	Assimétrica	Os dispositivos pertencentes à rede possuem diferentes capacidades ou distâncias de transmissão.
Disseminação de dados	Programada	Se assemelha à coleta programada, enviando dados entre intervalos de tempo predeterminados;
	Contínua	Se assemelha à coleta contínua, enviando dados continuamente assim que são coletados;
	Sob demanda (sob eventos)	Se assemelha à coleta reativa, enviando dados somente quando solicitados pelo observador ou quando um evento predeterminado ocorre.
Transmissão	<i>Simplex</i>	Dispositivos de rede somente enviam dados, não podendo receber dados.
	<i>Half-Duplex</i>	Os Nós podem enviar e receber dados, desde que em instantes diferentes.

	<i>Full-Duplex</i>	Os dispositivos da rede podem enviar e receber dados de forma simultânea, não sendo necessário aguardar o término de um dos eventos para o outro ocorrer.
Fluxo de Informação:	<i>Flooding</i> (inundação)	Nesta topologia quando o dispositivo não possui alcance para o AP ele repassa os seus dados a todos os dispositivos ao redor dele (que possuam alcance de transmissão).
	<i>Unicast</i>	Neste tipo de rede, os dispositivos que não possuem alcance para comunicação direta com o AP encaminham o dado a apenas um dispositivo vizinho.
	<i>Multicast</i>	Neste tipo de rede quando um dispositivo não possui alcance para comunicação direta com o AP eles enviam para alguns de seus vizinhos esses dados, seus vizinhos tentam então enviar os dados para o AP.

Para entendermos um pouco mais sobre as características de cada rede de sensores sem fio iremos destrinchar um pouco mais sobre cada característica.

Composição: Segundo o conceito de Verona (2010) é de suma importância saber se a RSSF é heterogênea ou homogênea, pois, dispositivos homogêneos fazem com que todos os dispositivos tenham as mesmas características, sendo assim não é de tamanha importância observar onde cada nó é colocado visto que os nós são iguais. Diferente das RSSF homogêneas as RSSF heterogêneas possuem dispositivos com características diferentes, geralmente utilizados para RSSF que possuem diferentes características por onde são espalhados os nós, sendo necessário que alguns nós sejam mais robustos ou tenham uma distância de envio maior que outros.

Mobilidade: De acordo com o conceito de Verona (2010) redes estacionárias são geralmente as mais utilizadas, por exemplo, para simular uma RSSF e capaz de prevenir o alastramento de queimadas em áreas de preservação permanente (APP), visto que os nós, uma vez colocados dentro das APP, não irão mais se deslocar. As redes móveis por sua vez são utilizadas em casos mais específicos, em redes onde é necessário que os nós não sejam fixos.

Densidade: A partir das definições de Verona (2010) pode-se observar que redes densas são mais utilizadas em casos onde existe algum local com um maior índice do

fenômeno estudado. Neste local existe uma concentração maior de dispositivos, as redes esparsas possuem menos dispositivos, porém, geralmente os dispositivos são um pouco mais robustos. As redes balanceadas por sua vez são utilizadas quando é necessário uma consistência maior dos dados em relação a área, dados como produção por área, animais em determinadas áreas, e afins se tornam grandes candidatos a participarem deste tipo de topologia.

Distribuição: Pelas definições de Verona (2010) redes do tipo regular são redes com dispositivos distribuídos igualmente pela área e são geralmente utilizadas em locais que necessitam uma consistência maior dos dados por área, tal como as redes balanceadas. Redes irregulares possuem uma distribuição não uniforme pela área monitorada.

Tamanho: O tamanho de cada rede vai variar de acordo com a necessidade da área a ser monitorada, com a realidade dos equipamentos utilizados e com o tipo de densidade dos nós sensores.

Coleta de dados: Segundo as definições de Ruiz (2003) observa-se que dispositivos que efetuam coleta reativa são utilizados geralmente em locais que medem algum tipo de evento como, por exemplo, medição de gases atmosféricos quando um vulcão entra em erupção ou medição de terremotos. Os dispositivos efetuam a coleta apenas ao ocorrer o fenômeno, como por exemplo, na erupção do vulcão. A coleta de dados periódica geralmente é utilizada quando se quer medir um fenômeno poucas vezes ao dia e com uma boa precisão. Este é geralmente utilizado, pois com ele o gasto de energia é um pouco menor que no contínuo. A coleta de dados contínua é utilizada quando a rede de sensores não pode parar, ou seja, ela tem que em todo momento controlar alguma informação.

Tipos de conexão: A partir dos conceitos de Verona (2010) observamos que o tipo de conexão diz respeito também às características da rede ser ou não homogênea. Redes simétricas possuem dispositivos com mesma capacidade de transmissão e distâncias com exceção do AP. Já as redes assimétricas possuem dispositivos com capacidades e distâncias de transmissão diferentes.

Transmissão:

Segundo Verona (2010) pode-se dizer que os dispositivos de rede do tipo *simplex* somente podem enviar dados sem a possibilidade de recepção, o que faz com que as redes

com esse tipo de enlace não tenham capacidade de encaminhar dados de dispositivos vizinhos. Verona (2010) ainda afirma que os dispositivos *Half-Duplex* são aqueles que podem enviar e receber dados desde que em instantes diferentes, o que possibilita que dados sejam encaminhados entre dispositivos de rede até a chegada ao AP, resolvendo casos onde o dispositivo não tem alcance direto com o AP. Dispositivos *Full-Duplex* por sua vez, segundo Verona (2010) podem enviar e receber dados de forma simultânea, não sendo necessário aguardar o término de um dos eventos para que o outro ocorra, isso faz com que o desempenho da rede melhore e reduza o atraso entre a coleta de dados e a sua chegada ao AP.

A transmissão dos dados influi diretamente em diversas outras características da rede e ela diz respeito à característica dos nós. Se os nós tiverem capacidade de enviar informações a rede é *simplex*. Dispositivos *simplex* podem ser utilizados em redes hierárquicas com dispositivos líderes sendo *Full* ou *Half duplex*. Redes planas com características de *simplex* fazem com que dados sejam transmitidos direto para AP. As transmissões do tipo *Half-duplex* não permitem que os dados sejam enviados e recebidos ao mesmo tempo, porém os nós têm capacidade para enviar e receber dados. As transmissões do tipo *Full-duplex* permitem que os dados sejam enviados e recebidos simultaneamente, o que pode aumentar consideravelmente o tempo de coleta de dados (VERONA, 2010).

Disseminação de dados: De acordo com as definições de Verona (2010), pode-se dizer que é de suma importância conhecer a forma de disseminação de dados, visto que isso será crucial para o consumo de energia de um nó sensor.

Fluxo de informação:

Segundo Verona (2010), no funcionamento das redes do tipo *Flooding* quando um dispositivo que não possui alcance até o AP vai enviar um dado, ele repassa o dado a todos os dispositivos que possuem alcance na transmissão. Ao receberem os dados eles repassam a todos os dispositivos que conseguem comunicar até que o dado chegue ao AP.

Segundo Verona (2010), “essa abordagem reage bem em redes com topologias dinâmicas, que sofrem alterações constantes, porém ocasiona sobrecarga na rede pelo excesso de encaminhamentos do dado, o que gera um consumo elevado de energia, reduzindo o tempo de vida da rede.”.

Segundo Verona (2010) nas redes do tipo *Unicast* ao enviar um dado, um dispositivo que não possui alcance direto com o AP, encaminha o dado apenas a um dispositivo vizinho. Esse tipo de conexão é conhecida também como ponto a ponto. Esta forma é muito utilizada ao se

conhecer a rota do dispositivo à origem, o que faz com que necessite uma pequena quantidade de encaminhamentos até a chegada do dado, e isto diminui o tempo para o observador utilizar o dado.

Segundo Verona (2010) as Redes do tipo *Multicast* reúnem características dos métodos *unicaste flooding*. Nesta rede os dispositivos enviam os dados para alguns vizinhos e não a todos que possuem alcance. Este conjunto é responsável por enviar o dado ao *AP*, ou aos próximos vizinhos, até que o dado chegue ao *AP*. Neste método a quantidade de dados trafegando pela rede é reduzida, o que faz com que ele apresente melhores resultados que o *flooding*.

Para exemplificar melhor a diferença entre o fluxo de informação do tipo *Flooding*, *Unicast* e *Multicast*, têm-se a figura número 5, adaptada de Verona (2010):

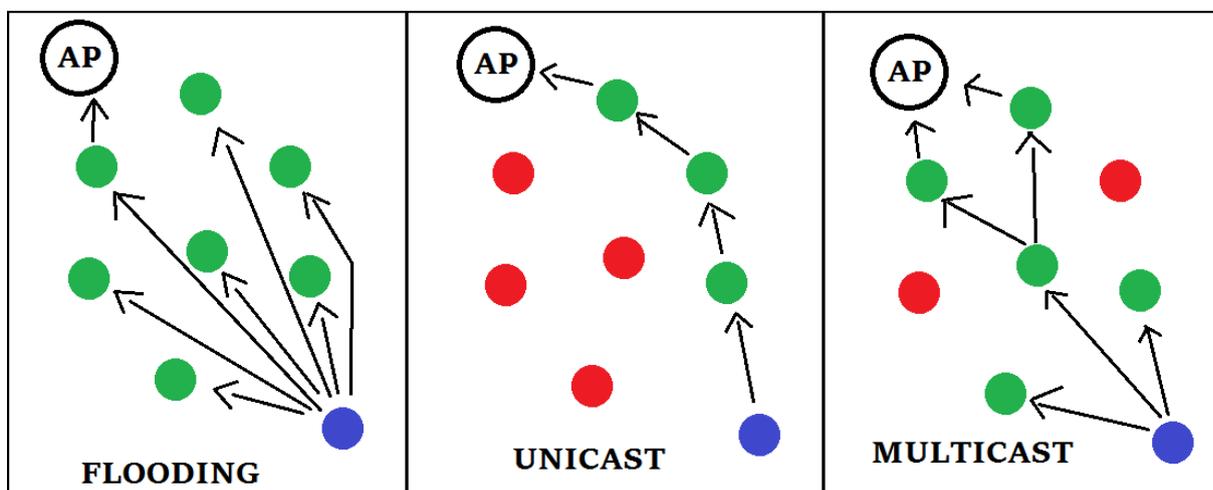


Figura 5 - Funcionamento do fluxo de informação

2.1.2. Protocolos de RSSF

Como já foi visto anteriormente na seção 2.1.1, uma das tarefas mais difíceis dentro de uma RSSF é a de definir as suas características como: se os nós farão processamento de informações, quantos *AP* possuem na rede, se influenciam diretamente no consumo de energia, atraso médio e dados perdidos.

Determinar para onde um dispositivo de uma RSSF deve enviar um dado até que ele chegue ao *AP* é uma tarefa que pode ter grande influência no consumo de energia, no atraso desde a coleta do dado até sua chegada ao *AP*, e no número de dados que vão se perder não chegando ao *AP*. Essa função é desempenhada nas redes *ad-hoc* pelos protocolos de roteamento, que são classificados como reativos, proativos e híbridos (VERONA, 2010).

Os protocolos do tipo reativo possuem, como o próprio nome já diz, o tipo de coleta de dados reativo, ou seja, os dados são coletados de acordo com a necessidade do usuário, ou no momento que ocorrer algum evento como exemplificado anteriormente. Segundo Herek (2011) “entre os protocolos de roteamento para redes *ad-hoc*, alguns exemplos que utilizam o mecanismo de manutenção de rotas sob demanda são: ABR, CEDAR, DREAM1, DSR, FORP, GEDIR, LAR, SSR, WAR [...]”.

Os protocolos do tipo Proativos por sua vez são protocolos que trabalham para manter as informações de roteamento atualizadas o tempo todo. Basicamente os protocolos que utilizam esta estratégia são divididos em duas classes: protocolos orientados a eventos e protocolos de atualizações regulares (HEREK, 2011).

Sendo assim os protocolos Pró-ativos trabalham gerenciando informações do tipo periódica e reativa que ocorrem quando algum evento ocorre. O autor Herek (2011) ainda afirma que “entre os protocolos de roteamento para redes *ad hoc*, alguns exemplos que utilizam o mecanismo de manutenção de rotas sob demanda, orientada a eventos são: CBRP, CGSR, GSR, LMR, TORA, WRP e [...] o DSDV.” O autor continua dizendo que “entre os protocolos que utilizam atualizações regulares estão: DDR, FSLs, FSR, GPSR, LANMAR, OLSR, STAR e TBRPF”.

Os protocolos híbridos são protocolos que se utilizam de parte dos algoritmos Proativos e parte dos algoritmos Reativos. Entre os protocolos que utilizam essa estratégia estão: ADV, Termino de *Routing* e ZRP. As principais características que irão diferenciar os protocolos segundo Herek (2011) são: A força do sinal onde em alguns algoritmos como o ABR e o SSR os pacotes de dados possuem um sinal com forte intensidade para serem roteados. A estabilidade de link (conexões) que é presada em alguns algoritmos como o DST e FORP, o que garante um protocolo com conexões mais estáveis. O direcionamento de roteamento, que é utilizado principalmente nos protocolos DREAM, GEDIR, GPSR, LAR, sendo esses algoritmos que presam que os dados sempre devem ser enviados na direção do destino (seja ele um AP ou um líder de grupo). O autor ainda afirma que “*Link Reversal Rounting* – Utilizada pelos protocolos LMR e TORA é baseada em fluxos em um gráfico.”

Distância do vetor é utilizada a partir de uma contagem de saltos, geralmente é a mais utilizada, sendo utilizada em protocolos como o AODV, DSDV, DSR, ADV, WRP e o ZRP. Menor caminho ou Links estáveis baseia-se em selecionar o menor caminho a partir de algumas métricas, usado em protocolos como CEDAR, DDR, FSR, GSR, HSR, LANMAR, OLSR, STAR, TBRPF (HEREK, 2011).

2.1.3. *Broadcast*

Outro conceito muito importante para se conhecer uma RSSF é o de *broadcast*. Basicamente é o conceito que envia o sinal para os outros nós a fim de criar um roteamento e enviar os dados. O uso de mensagens *broadcast* em uma rede sem fio *ad-hoc* pode ser aplicada de três formas: *broadcast* completo, *broadcast* local e *broadcast* limitado. No *broadcast* completo as mensagens são destinadas para todos os nós da rede e passam por nós intermediários até atingir os nós mais distantes. Por outro lado, no *broadcast* local, as mensagens são destinadas apenas ao nó no raio de alcance da fonte, e não são retransmitidos aos outros nós fora do alcance da fonte. Os casos em que as mensagens *broadcast* possuem uma contagem máxima de saltos, e é limitada conforme a configuração do protocolo são classificadas como *broadcast* limitado (LANG, 2003 e HEREK, 2011).

Basicamente as mensagens de Broadcast funcionam da seguinte forma, quando um nó precisa enviar algum pacote a um destino, caso ele ainda não tenha uma rota conhecida, ele primeiramente envia um *Route requests* (RREQ) para toda a rede requerendo uma rota. Se o RREQ chega a um nó destino este envia um *Route Reply* (RREP) ao nó de origem e então a rota é traçada. Caso ainda o *Request* não tenha chegado ao destino os nós intermediários enviam RREP para a origem e o nó de origem envia novamente um RREQ até que a rota seja traçada. Ao traçar a rota essa passa a ser utilizada. Caso algum erro ocorra em algum nó uma mensagem *Router Error* (RERR) é enviada à origem e uma nova rota é traçada (HEREK, 2011).

2.1.4. Funcionamento dos protocolos de roteamento

2.1.4.1. DSR

Segundo afirmação do autor Bastos (2011):

“O DSR (*Dynamic Source Routing*) (14) é um algoritmo de roteamento reativo para redes *ad-hoc* no qual a origem determina por qual caminho o pacote deve passar para chegar ao seu destino, tendo como característica um melhor desempenho em ambientes onde a velocidade de mobilidade dos nós é baixa. Diferente dos protocolos existentes em redes *ad-hoc*, o DSR não atualiza as informações sobre rotas frequentemente e tem como objetivo oferecer vantagens em relação aos modelos tradicionais de roteamento para redes cabeadas, mas possuindo algumas desvantagens em situações de grande mobilidade e um volume grande de dados na rede”.

O protocolo DSR em resumo trabalha da seguinte forma, o nó de origem envia o dado a outro nó. Ao enviar o dado a outro nó o nó de origem especifica qual será a ordem em que os nós serão visitados até chegar ao nó de origem (seja ele um *AP* ou um nó líder de grupo). Neste protocolo as informações de qual será a rota do pacote de dados é anexada ao cabeçalho dos pacotes enviados. Quando há mais de uma rota disponível para chegar ao destino o algoritmo escolhe a rota com menor número de cascatas possível. Caso o algoritmo já tendo rodado para determinado nó, a rota tomada por ele fica armazenada em uma cache. Este nó, portanto, não precisa mais calcular a rota, ele vai escolher a rota que foi calculada anteriormente. Caso o nó nunca tenha sido calculado ou a rota não esteja disponível ele irá calcular a rota usando novamente o algoritmo (BASTOS, 2011).

2.1.4.2. AODV

Segundo Herek (2011) “O protocolo *Ad-hoc On-demand Distance Vector (AODV)* é um protocolo reativo derivado do protocolo DSDV, descrito na RFC 3561. É um protocolo frequentemente discutido em pesquisas e muitas vezes comparado com outros protocolos”

Neste protocolo é enviada uma mensagem de *broadcast RREQ* a partir do nó de origem com um pequeno tempo de vida, conhecido com *time to live (TTL)*. Assim, é iniciada a descoberta da rota, se a rota for descoberta, chegando ao nó de destino ou a um nó intermediário que já possui uma rota bem definida. Então a mensagem é enviada e a rota é gravada. Caso contrário uma nova mensagem de RREQ é enviada com um TTL maior até que uma rota seja traçada.

De acordo com Herek (2011) o cabeçalho para entrada na tabela de armazenamento é composto de:

- Endereço IP de destino;
- Número de sequência do destino;
- Estado de validade do número sequencial do destino;
- Outros estados e controle de roteamento (válido, inválido, reparável);
- Número de saltos para alcançar o destino;
- Próximo salto;

- Lista de precursores;
- Tempo de vida.

Assim como vimos anteriormente, na sessão de *broadcast*, no protocolo AODV, caso em alguma rota ocorra falha uma mensagem de erro RERR é enviada à origem e a mesma precisa recalculer a sua rota.

Cada entrada na tabela de roteamento recebe um tempo de vida, e este tempo é atualizado a cada vez que a rota é utilizada. O campo tempo de vida na tabela de roteamento tem dupla função: para as rotas ativas significa o tempo de expiração e para as rotas inválidas o tempo de eliminação (HEREK, 2011).

2.1.4.3. DSDV

Neste protocolo as informações de roteamento são atualizadas não no momento em que a mensagem é enviada como em outros casos como vistos anteriormente. Neste protocolo a tabela de informações de roteamento é atualizada periodicamente.

Em resumo Herek (2011) diz que as entradas da tabela de roteamento do protocolo DSDV possuem os seguintes campos:

- O endereço IP do destino;
- O endereço IP do próximo salto;
- O número de saltos até o destino;
- O número sequencial das informações recebidas sobre o destino.

Para se evitar um grande tráfego de rede, que essas atualizações podem gerar, são aplicadas dois tipos de atualizações: completa e incremental. Na atualização completa, também chamada de *full dump*, todas as informações de roteamento disponíveis são transmitidas e podem exigir o envio de múltiplos *network protocol data units* (NPDUs). Quando existem poucas mudanças de topologia, as atualizações completas são menos frequentes. Por outro lado, as atualizações incrementais utilizam necessariamente, por determinação do projeto do protocolo, apenas uma NPDU. As atualizações incrementais são utilizadas para trafegar as mudanças ocorridas depois da última transmissão completa e ajudam a diminuir o tráfego gerado (HEREK, 2011).

2.2. Computação Gráfica

Para criação do OA foi utilizado a linguagem C++ juntamente com a biblioteca gráfica *OpenGL*. A biblioteca gráfica *OpenGL* é uma biblioteca livre utilizada para o desenvolvimento de ambiente gráficos, e foi utilizada com o intuito de mostrar os nós sensores cadastrados na tela. Já o C++ foi utilizado para o controle dos nós sensores e *Access Point*, além de cálculos necessários.

2.2.1. C++

Os principais conceitos utilizados da linguagem C++ são o conceito de orientação a Objeto, classes, Criação de funções e leituras de arquivos de texto.

2.2.1.1. Orientação a objeto

A linguagem C++ é uma linguagem orientada a objetos, sendo assim ela utiliza de modelos de classes para a criação de objetos reais que devem ser manipulados durante o projeto. Cada objeto é, portanto, onde armazenam-se os valores de cada atributo de uma classe.

Segundo Schildt (2002) “No centro de C++ encontra-se a programação orientada a objetos (OOP). Como foi explicado a OOP foi o impulso para a criação de C++. Por causa disto, é útil entender os princípios básicos da OOP antes de escrever um único programa em C++”.

Segundo Schildt (2002) “A programação orientada a objetos pegou as melhores ideias da programação estruturada e as combinou com diversos novos conceitos. O resultado foi uma maneira melhor e diferente de organizar um programa.”. Ainda segundo Schildt (2002) a programação OOP faz com que um programa possa ser organizado em torno de seu código, ou em torno de seus dados.

Neste projeto cada nó sensor é tratado como um Objeto separado da Classe “Nós”, tendo os atributos mais utilizados:

- **x:** Armazena a posição x do nó sensor.
- **y:** Armazena a posição y do nó sensor.
- **width:** Armazena o valor da largura da caixa que será mostrada na tela.
- **height:** Armazena o valor da Altura da caixa que será mostrada na tela.

- **distanciaAP:** Armazena a distância do um nó ao Access Point.
- **distanciaDadoNo:** Armazena a distância de dado até o nó.

Além disso, o conjunto de dados a serem enviados são tratados como objetos da Classe “Dado” tendo cada um como atributos mais utilizados:

- **x:** Armazena a posição x do dado a ser enviado.
- **y:** Armazena a posição y do dado a ser enviado.
- **width:** Armazena o valor da largura da caixa que será mostrada na tela.
- **height:** Armazena o valor da Altura da caixa que será mostrada na tela.
- **dadoInicial:** Armazena um objeto da classe “Nos” que será o dado inicial.
- **dadoFinal:** Armazena um objeto da classe “Nos” que será o dado final, passando a ser o dado Inicial em um próximo passo do encaminhamento do dado ao Access Point.
- **MapaDistanciaNos:** Contém um mapa com todos os valores das distancias do nó inicial aos outros nós existentes, sendo utilizado para calcular o próximo nó a ser atingido.
- **DistanciaAPAnterior:** Armazena a distância do Access Point em que o dado se encontra, fazendo com que nenhum dado seja enviado a nós mais distantes do AP que os atuais.

Temos também o *Access Point* tratado como um objeto da Classe “*AccessPoint*”.

2.2.1.2. Classe

Segundo Vicenzi (2004) “uma classe é uma entidade estática que engloba atributos (ou dados) e métodos (ou funções membro) que representam operações que podem ser realizadas sobre os dados”. O autor traz também a definição de objeto como sendo uma instância de uma classe criada em tempo de execução. Cada objeto tem uma cópia dos dados definidos na classe e encapsula estado e comportamento.

As classes, portanto, são utilizadas para dizer ao programa quais são os atributos dos

Objetos. As classes criadas neste projeto são:

- **Nós:** Classe que possui os atributos dos nós sensores que serão cadastrados
- **Dado:** Classe que possui os atributos desejados dos dados que serão enviados
- **AccessPoint:** Classe que possui os atributos desejados do *Access Point* a serem cadastrados.

2.2.1.3. Funções

Segundo Horstmann (2005) “Funções são um bloco fundamental da construção de programas C++. Uma função encapsula uma computação em uma forma que possa ser facilmente entendida e reutilizada.”.

Schildt (2002) diz que uma função é uma sub-rotina que contém uma ou mais instruções de C++ e realiza uma tarefa específica. Uma função em C++ possui blocos chamados blocos de construção C++, pois, um programa é uma coleção de funções. Todas as instruções que fazem alguma ação são encontradas nas funções, assim sendo, uma função contém as instruções que você considera como sendo parte executável do programa.

Funções são utilizadas para controlar o fluxo do programa, fazendo desde o controle de dados dos objetos, até o controle de eventos como o pressionamento de teclas do teclado. Além das funções padrões do *OpenGL* utilizadas, temos como principais funções deste projeto:

- **Circulo(intxC, intyC, int r, int color):** Desenha um círculo na tela utilizando algoritmo de Breseham.
- **ControleDadoFinal(intij):** A partir dos dados iniciais, controla quais serão os dados finais no programa, além de desenhar na tela os dados iniciais, uma linha ligando o dado inicial e o final, um círculo que mostra a distância máxima de envio do nó sensor, e uma linha que mostra a distância atual do nó sensor até o Access Point.
- **DesenhaEnvio():** Desenha na tela o mapa de dados e nós.
- **lerArquivoAP(char* nomeArquivo):** Lê o arquivo de texto do *Access Point* e armazena os valores lidos na classe “*AccessPoint*”.

- **lerArquivoDado (char* nomeArquivo):** Lê o arquivo de texto do Access Point e armazena os valores lidos na classe “Dado”, gerando assim o mapa de dados.
- **lerArquivoFase(char* nomeArquivo):** Lê o arquivo de texto do Access Point e armazena os valores lidos na classe “Nos”, gerando assim o mapa de nós.
- **Linha(int x0,int y0, int x1, int y1):** Desenha na tela uma linha ligando dois pontos passados como parâmetro utilizando o algoritmo de bresenham.
- **setaDadosFinais():** Armazena os dados finais no mapa de dados.
- **setaDadosIniciais():** Armazena os dados iniciais no mapa de dados.

2.2.2. OpenGL

O *OpenGL* é uma biblioteca gráfica livre que possui pacotes para desenvolvimento de aplicativos gráficos, jogos, cenários 3D, entre outros.

Segundo Manssour (2005) “*OpenGL* é definida como "um programa de interface para *hardware* gráfico". Na verdade, *OpenGL* é uma biblioteca de rotinas gráficas e de modelagem, bi (2D) e tridimensional (3D), extremamente portátil e rápida.”

Ainda segundo Manssour (2005) “*OpenGL* não é uma linguagem de programação, é uma poderosa e sofisticada API (*Application Programming Interface*) para criação de aplicações gráficas 2D e 3D. Seu funcionamento é semelhante ao de uma biblioteca C, uma vez que fornece uma série de funcionalidades.”

As principais aplicações do *OpenGL* são:

- ***glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB):*** De acordo com Manssour (2005):

“Avisa a GLUT que tipo de modo de exibição deve ser usado quando a janela é criada. Neste caso os argumentos indicam a criação de uma janela *single-buffered* (*GLUT_SINGLE*) com o modo de cores RGBA (*GLUT_RGB*). O primeiro significa que todos os comandos de desenho são feitos na janela de exibição. Uma alternativa é uma janela *double-buffered*, onde os comandos de desenho são executados para criar uma cena fora da tela para depois rapidamente colocá-la na *view* (ou janela de visualização). Este método é geralmente utilizado para produzir efeitos de animação. O modo de cores RGBA significa que as cores são

especificadas através do fornecimento de intensidades dos componentes *red*, *green* e *blue* separadas.”

- ***glutCreateWindow("RSSF")***: é um comando utilizado para criar uma janela com o nome “RSSF”, sendo este nome correspondente a legenda no título da janela (MANSSOUR, 2005).
- ***glutInitWindowSize(500,500)***: é o comando que especifica o tamanho de exibição inicial da tela em pixels (MANSSOUR, 2005).
- ***glutInitWindowPosition(0,0)***: é o comando que aponta a localização inicial da janela GLUT sendo este o canto superior esquerdo da tela do computador (MANSSOUR, 2005).
- ***glutReshapeFunc(AlteraTamanhoJanela)***: é a função que estabelece “AlteraTamanhoJanela” como uma função *callback* que será chamada toda vez que ocorrer uma alteração no tamanho da janela, alterando o sistema de coordenadas (MANSSOUR, 2005).
- ***glBegin();... glEnd()***: é uma função utilizada para desenhar um polígono na tela a partir de vários vértices, o OpenGL mapeia as coordenadas dos vértices segundo posição atual da janela definida na função *callback* “AlteraTamanhoJanela” (MANSSOUR, 2005).
- ***glutTimerFunc(FPS, Timer, 0)***: Manssour (2005) estabelece que:

“a função *Timer* previamente definida como a função *callback* de animação. Seu protótipo é: `void glutTimerFunc(unsigned int msec, void (*func)(int value), int value)`;. Esta função faz a GLUT esperar *msec* milissegundos antes de chamar a função *func*. É possível passar um valor definido pelo usuário no parâmetro *value*. Como esta função é "disparada" apenas uma vez, para se ter uma animação contínua é necessário reinicializar o *timer* novamente na função *Timer*.”

- ***glutKeyboardFunc (GerenciaTeclado)***: Manssour (2005) estabelece como *callback* a função *GerenciaTeclado* que será chamada ao ocorrer o pressionamento de uma tecla que gera um código ASCII, além de gerar a posição (x,y) do mouse. A função tem como parâmetros (*unsigned char key, int x, int y*).
- ***glutSpecialFunc(TeclasEspeciais)***: é uma função que define como *callback* a função *TeclasEspeciais* que será chamada ao ocorrer o pressionamento de uma tecla que não gere um código ASCII quando pressionada, como F1, F2, PgUp, PgDn, e afins, além

disso a posição (c,y) do mouse será retornada quando a tecla for pressionada (MANSSOUR, 2005).

- **glutMouseFunc(GerenciaMouse):** é uma função que define como *callback* a função GerenciaMouse, que será chamada ao ocorrer um evento de movimentação no mouse sendo seus parâmetros: (int button, int state, int x, int y) (MANSSOUR, 2005).
- **glutDisplayFunc(Desenha):** é uma função que define Desenha como *callback* de exibição, ou seja, sempre que a janela precisar ser redenhada a função Desenha será chamada, seja ao redimensionar a tela ou desenhar algum objeto nela (MANSSOUR, 2005).
- **glutMotionFunc(MoveMouseBotaoPressionado):** é uma função que define como *callback* a função MoveMouseBotaoPressionado que será chamada toda vez que ocorrer o evento de movimento do mouse com o botão pressionado (MANSSOUR, 2005).

2.3. Objetos de aprendizagem

Segundo Braga, (2014, p.20) “(...) os objetos de aprendizagem podem ser vistos como componentes ou unidades, catalogados e disponibilizados em repositórios na Internet. Assim, podem ser utilizados em diversos contextos de aprendizagem, de acordo com o projeto instrucional.”.

Segundo Braga (2014, p.21 apud Wiley, 2000) o autor David Wiley sugeriu uma definição menos ampla onde OA pode ser definido como qualquer recurso digital que possa ser reutilizado para apoiar a aprendizagem.

Os objetos de aprendizagem são, portanto, ferramentas que auxiliam no processo de aprendizagem, fazendo com que os alunos consigam compreender os conteúdos propostos de uma maneira mais simples e eficaz.

2.3.1. Tipos de objetos de aprendizagem

Os OAs podem ser divididos em algumas categorias sendo algumas dessas citadas a seguir:

- Imagem

- Áudio
- Vídeo
- Animações
- Simulação
- Hipertexto
- Softwares

Sendo dentre esses o principal foco neste artigo a produção de um OA que englobe as categorias de animações e simulações dentro de um *software*.

2.3.2. Características de um Objeto de Aprendizagem

As características dos objetos de aprendizagem possuem duas perspectivas: a pedagógica e a técnica.

As características relacionadas à dimensão pedagógica fazem referência à concepção de objetos que facilitem o trabalho de professores e alunos, visando à aquisição do conhecimento (DIAS et al., 2009). Nesse contexto, são considerados aspectos pedagógicos importantes segundo Galafassi *et al.*, (2014):

- **Interatividade:** indica se há suporte às consolidações e ações mentais, requerendo que o aluno interaja com o conteúdo do OA de alguma forma, podendo ver, escutar ou responder algo.
- **Autonomia:** indica se os objetos de aprendizagem apoiam a iniciativa e tomada de decisão.
- **Cooperação:** indica se há suporte para os alunos trocarem opiniões e trabalhar coletivamente sobre o conceito apresentado.
- **Cognição:** refere-se às sobrecargas cognitivas alocadas na memória do aluno durante o processo de ensino-aprendizagem.
- **Afetividade:** refere-se aos sentimentos e motivações do aluno com sua aprendizagem e durante a interação com o OA.

Segundo Braga (2014) é necessário ainda para que a perspectiva técnica atinja um nível adequado se atentar às seguintes questões:

- Disponibilidade
- Acessibilidade
- Confiabilidade
- Portabilidade
- Facilidade de instalação
- Interoperabilidade
- Usabilidade
- Manutenibilidade
- Granulidade
- Agregação
- Durabilidade
- Reusabilidade

2.3.3. Metodologia INTERA

A metodologia utilizada para o desenvolvimento do projeto foi baseada na metodologia INTERA. Segundo Braga *et. al.*, (2013) a metodologia INTERA é uma metodologia interativa, que sugere o “ir e vir” em suas etapas. A metodologia INTERA considera o processo de desenvolvimento de OAs como sendo um projeto e por isso ela também contempla que o ciclo de vida de um OA define as fases que conectam o início de um projeto de desenvolvimento de OA ao final. Essa consideração é baseada no guia do conjunto de conhecimentos em gerenciamento de Projetos (PMBOK, 2008), um dos pilares no qual a metodologia INTERA se apoia.

Os componentes avaliados pela metodologia INTERA são as fases, papéis, etapas e artefatos.

Os principais papéis segundo Braga *et. al.*, (2013) são:

- Analista
- Conteudista

- Demandista
- Gerente de projetos
- Designer de projetos
- Designer Instrucional
- Designer de Interface
- Designer Técnico
- Equipe de Desenvolvimento
- Equipe de Teste.

A ficha de contextualização desse objeto de aprendizagem gerada neste trabalho pode ser vista na tabela 2:

Tabela 2: Ficha de contextualização do OA

Ficha de Contextualização do OA	
Categoria do AO	Prático / Software
Objetivos pedagógicos	Proporcionar o aluno aprendizado de um RSSF.
Área de conhecimento	RSSF
Disciplina Principal	Redes de Computadores
Ementa seguida	Conceitos iniciais RSSF
Tópicos da Ementa	1. Nós sensores e Access Point 2. Tipos de RSSF 3. Configurações de uma RSSF
Descrição breve do OA	Propõe o aprendizado de conceitos iniciais de RSSF através de um software que simula uma rede com funcionamento <i>floodfill</i> direcionados para o público de ensino superior na área de Tecnologia de Informação e Redes.
Público Alvo	Alunos de ensino superior na área de Tecnologia de Informação e Redes.
Conhecimento prévio esperado	Conceitos iniciais de Redes.
Grau de Acessibilidade	Página computador.
Fluência tecnológica exigida	Computação Básica
Solução esperada	Contribuir para o aprendizado de RSSF.

3. TRABALHOS EXISTENTES

O principal objetivo deste capítulo é apresentar os trabalhos utilizados no processo de aprendizagem do tema e trabalhos que tenham intuito parecido com o deste trabalho, e logo após contextualizar os trabalhos utilizados.

3.1. Ambientes de simulação.

A simulação de sistema é de suma importância a fim de analisar quais os melhores procedimentos a serem feitos dentro de um sistema, um dos principais ambientes de simulação existe é o *Network Simulator 3* (NS-3).

3.1.1. *Network Simulator 3* (NS-3)

Segundo o NETWORK SIMULATOR 3o NS-3“é um simulador de redes baseado em eventos discretos desenvolvido especialmente para pesquisa e uso educacional. O projeto ns-3 iniciou em 2006 e tem seu código aberto.”

3.1.2. *TraceGraph*

O *Network Simulator* gera arquivos no formato trace que serão analisados pelo *TraceGraph* que é uma ferramenta para análise de arquivos de trace dos tipos *wired*, *satellite* e *newtrace* gerados pela ferramenta NS-2 que funciona tanto em plataforma Windows como em Linux e possui como pré-requisito para instalação a ferramenta Matlab (MATLAB, 2009; VERONA, 2010; LUKOIT, *et al.*, 2009).

3.1.3. *MannaSim*

O *MannaSim* é um módulo de *software* que foi desenvolvido por pesquisadores da Universidade Federal de Minas Gerais durante o desenvolvimento da tese proposta por Ruiz (2003) para ser integrado a ferramenta de simulação de redes NS-2 com o objetivo de suprir a falta de funcionalidades e parâmetros para simulação das RSSFs, integrando, por exemplo, novos protocolos, como o LEACH (MANNASIM, 2009).

O *MannaSim* compreende um conjunto de classes que podem ser utilizadas pelos pesquisadores na simulação das RSSFs, sendo estas classes as responsáveis por estender as funções da ferramenta NS-2 para as RSSFs (VERONA, 2010).

3.2. Trabalhos Relacionados

Para procurar artigos científicos que falassem sobre o tema foi efetuada uma pesquisa dentro dos periódicos da capes, sendo o mesmo um site de pesquisa de artigos acadêmicos. Primeiramente utilizou-se como pesquisa a fim de saber se existia algum artigo que tivesse feito algo parecido anteriormente. A primeira pesquisa efetuada foi por artigos que contivessem as palavras “redes de sensores” e “Objeto de aprendizagem” ao mesmo tempo não encontrando nenhum artigo que contivessem as duas palavras ao mesmo tempo nos últimos 20 anos. A segunda pesquisa foi por artigos que contivesse “redes de sensores sem fio”, foram encontrados 25 artigos relacionados, não possuindo nenhum artigo relacionado que fizesse trabalho parecido com tal nos últimos 20 anos. A terceira pesquisa foi por artigos que contivessem “*sensors*” e “*Learning objects*” ao mesmo tempo, foram encontrados 3 artigos relacionados, não possuindo nenhum artigo relacionado que fizesse trabalho parecido como tal nos últimos 20 anos. A fim de validar a pesquisa foi feito uma última busca por artigos que contivessem as palavras “*Wireless sensor networks*” e “*Learning objects*” e nenhum artigo foi encontrado nos últimos 20 anos.

Para estudos foram procurados artigos com a palavra-chave “redes de sensores sem fio”. No título foram encontrados nove artigos. Os artigos utilizados serão descritos de maneira breve nos próximos parágrafos. Além disso, alguns outros artigos foram encontrados em fontes alternativas como pesquisas no site do Google Acadêmico e indicações do professor orientador.

3.3. Revisão de literatura

Ao longo do tempo, cada vez mais surge o interesse de pesquisadores em contribuir com a sociedade acadêmica na expansão do conhecimento de Redes de Sensores Sem Fio. Para que seja possível criar meios de facilitar a compreensão do tema foram pesquisados alguns artigos sobre o conteúdo.

Verona (2010) propõe a análise e simulação de um ambiente de RSSF que seja a mais adequada a uma área de viticultura, em que seja possível controlar situações meteorológicas adversas. Para fazer este ambiente de simulação Verona utiliza de um simulador conhecido como *Network Simulator*, além de diversas bibliotecas que convertem os dados em gráficos. Neste trabalho Verona encontra o melhor ambiente para controlar situações meteorológicas em uma área de viticultura situada na cidade de Marinalva no Paraná, cidade que se destaca pela produção de uvas.

Bastos (2011), em seu trabalho, descreve o funcionamento de um protocolo NSINKS originado do protocolo DESRO. Além disso, Bastos compara os dois protocolos verificando que o protocolo NSINKS obteve melhores taxas e maior autonomia. Em seu trabalho, descreve também o funcionamento de uma Rede de sensores sem fio, mostrando as camadas de arquitetura, requisitos e afins.

Herek (2011) faz uma análise comparativa entre diversos protocolos de redes móveis ADHOC, apresentando o funcionamento de redes MANET, descrevendo o funcionamento de redes AODV, DSR e DSDV comparando os protocolos e ilustrando através de gráficos comparativos o funcionamento dos mesmos, gerando assim um excelente material de estudo sobre os protocolos utilizados.

NETWORK SIMULATOR 3 (acesso em 2015) possui um documento que demonstra como utilizar a ferramenta *Network Simulator 3*, mostrando desde a visão conceitual até códigos para construção de topologias e construções das redes deste modelo.

Ruiz (2003) apresenta informações funcionais sobre as diferentes arquiteturas e características de uma RSSF, além de apresentar informações sobre gerenciamento das arquiteturas, suas restrições quanto a energia e poder de computação, além de mostrar suas dimensões, outras características e ilustrá-las.

Braga (2006) propõe a criação de um Elemento Autônomo (EA) que seja capaz de executar em ambientes com rigorosas restrições. Em seu trabalho Braga verifica que aplicar computação autônoma em RSSF pode reduzir o consumo de energia e diminuir a porcentagem e pacotes perdidos além de melhorar a qualidade do serviço em diversos parâmetros. Braga mostra também o funcionamento de um nó sensor sem fio, comparativo entre diferentes plataformas de nós sensores entre outros conceitos de suma importância para o aprendizado de RSSF.

Cada vez mais pesquisadores tem demonstrado também interesse na área de Objetos de aprendizagem a fim de contribuir com a sociedade acadêmica. Para compreender melhor sobre o tema foi pesquisado o livro “Objetos de Aprendizagem Volume 1 - Introdução e Fundamentos” escrito por Braga (2014). Este livro introduz os fundamentos sobre Objetos de aprendizagem, estratégias pedagógicas, repositórios, acessibilidade, Jogos educacionais, reuso e temas afins. Utilizou-se também outros textos da autora como o que fala sobre a metodologia INTERA na produção de OAs para produção de OAs com qualidade e que organizem o processo de produção, além de “Desafios para o Desenvolvimento de Objetos de Aprendizagem Reutilizáveis e de Qualidade” escrito por Braga; Dotta; Pimentel; Stransky

(2012), que fala sobre como aumentar o reuso e qualidade de objetos de aprendizagem.

Além disso, utilizou-se também o texto de Vicenzi (2004) durante o estudo das definições de computação gráfica. Este trabalho mostra testes e validações dentro de um contexto, tendo uma vasta revisão bibliográfica sobre Orientação a Objeto, conceito muito utilizado durante a implementação do OA.

Bilinski (s/d) Tem como objetivo descrever a proposta do projeto escolhido pelo aluno sobre “Internet das coisas”, tratando a definição para “internet das coisas” utilizada neste artigo.

Schildt (2002) é um livro que contém os conceitos básicos sobre programação em C++, tendo como título “C++ guia para iniciantes”. Horstmann (2005) por sua vez escreve sobre conceitos essenciais de computação em seu livro “Conceitos de computação com o essencial de C++”.

4. METODOLOGIA

O método utilizado para o desenvolvimento do trabalho foi baseado na metodologia INTERA, como citado no material teórico. Primeiramente avaliou-se a necessidade da criação de um objeto de aprendizagem do conteúdo através de pesquisas de materiais atuais. Observou-se então a dificuldade de encontrar bons materiais que possibilitassem novos alunos ao aprendizado do conteúdo. Para isso propôs-se então a criação deste objeto de aprendizagem.

Após a definição do tema, foi elaborada uma pesquisa do conteúdo de RSSF com a finalidade de promover o melhor aprendizado. Em seguida realizou-se o desenvolvimento do projeto, orientado por um professor que leciona o conteúdo. Através de algumas reuniões foi-se definindo durante o desenvolvimento maneiras mais adequadas de atingirmos um objetivo.

Foi definido para a criação do OA a utilização da linguagem C++, utilizando-se da biblioteca gráfica OpenGL para a criação gráfica do conteúdo. Para ser feita a programação utilizou-se o Visual Studio 2015. Segundo a documentação recomendada pelo Visual Studio (2008):

"Visual Studio é um conjunto completo de ferramentas para construir aplicativos desktop e aplicativos Web empresariais desenvolvidos por equipes. Além de criar aplicativos individuais de alto desempenho, você pode usar as poderosas ferramentas de desenvolvimento baseadas em componentes do Visual Studio e outras tecnologias para simplificar o trabalho em equipe no projeto, desenvolvimento e implantação de soluções empresariais."

Segundo Manssour (2005) *OpenGL* é definida como "um programa de interface para *hardware* gráfico". Na verdade, *OpenGL* é uma biblioteca de rotinas gráficas e de modelagem, bi (2D) e tridimensional (3D), extremamente portátil e rápida."

Ainda segundo Manssour (2005) "*OpenGL* não é uma linguagem de programação, é uma poderosa e sofisticada *API* (*Application Programming Interface*) para criação de aplicações gráficas 2D e 3D. Seu funcionamento é semelhante ao de uma biblioteca C, uma vez que fornece uma série de funcionalidades."

Ficou definido que o projeto contaria inicialmente com a demonstração na tela de dados que iriam ser exibidos na tela com valores entre 0 e 500 unidades de medida (tomando-se como base Metros), sendo assim os valores colocados na tela seriam representações de uma área de 250.000 m². Além disso, definiu-se também que ao clicar na tela o nó mais próximo seria excluído, permitindo ao usuário verificar como seria o novo roteamento do dado após a

exclusão do mesmo.

Após a primeira versão do projeto pronta o mesmo foi submetido a uma análise de quais as possibilidades de acréscimo poderiam ser feitas no projeto, implementando-se então novas funcionalidades ao programa, como a funcionalidade de zoom, mostrar na tela o caminho percorrido e máximas distâncias de envio, o que permitiu que dados que estejam além de 250.000 m² inicialmente definidos possam ser vistos, e possibilitou também que sejam verificados casos em que o nó sensor não encontre a possibilidade de enviar o dado a um nó posterior, visto que nenhum dos nós se aproxima do AP.

Com o projeto concluído, o mesmo passou por uma fase de testes, detectando alguns erros que passaram a ser corrigidos como a possibilidade de colocar em arquivo de texto números de dados maiores que os alocados, entre outros problemas corridos.

Logo após o programa pronto, a metodologia de avaliação escolhida foi uma pesquisa qualitativa baseada em estudos bibliográficos, observações e realização de entrevista através de questionário. O questionário foi gerado através de um aplicativo conhecido como *Google Forms*, aplicativo que gera resultados através de gráficos das respostas obtidas pela internet. A pesquisa qualitativa vale-se de procedimentos de coleta de dados os mais variados. O processo de análise e interpretação pode, naturalmente, envolver diferentes modelos de análise. Todavia, é natural admitir que a análise dos dados seja de natureza predominantemente qualitativa (GIL, 2008, p.141).

Foram sujeitos da investigação os alunos da turma Redes de computadores II e alunos convidados da engenharia de computação que se dispuseram a comparecer. Esta turma em sua grande maioria é composta por alunos que em alguma vez durante sua vida acadêmica se deparou com o estudo de RSSF. A turma era composta por 27 alunos de engenharia de computação, e foi explicado a eles a necessidade da assinatura do termo de consentimento livre e esclarecido. Todos os alunos presentes, e por sua vez participantes, assinaram um termo de consentimento livre e esclarecido para participação da pesquisa e divulgação dos resultados. O termo de consentimento livre e esclarecido pode ser visto no anexo 1.

O objetivo da pesquisa foi coletar dados que pudessem comprovar a importância do aplicativo criado e sua aplicabilidade prática. Optou-se pela utilização da pesquisa participante após uma aula expositiva do conteúdo utilizando o objeto de aprendizagem, pois esta propicia que o pesquisador participe e interaja junto aos sujeitos da pesquisa. A pesquisa participante é apresentada como aquela em que o pesquisador, para realizar a observação, sendo dos fenômenos ou da situação problema a ser investigada, compartilha a vivência dos

sujeitos pesquisados, participando, de forma sistemática e permanente, ao longo do tempo da pesquisa, das suas atividades.

Além da observação e inserção, realizou-se também uma entrevista através de questionários contendo perguntas abertas as quais foram respondidas pelos alunos que se dispuseram a contribuir com esta investigação. As questões estiveram focadas no tema de utilização do OA e graus de aprendizagem na disciplina. As entrevistas foram realizadas utilizando-se de uma plataforma virtual, onde cada aluno respondeu os questionários separadamente. Diante dos dados coletados através de ambos os instrumentos, organizou-se o presente trabalho buscando analisá-los a luz do referencial teórico.

5. RESULTADOS

Neste trabalho iremos separar os resultados obtidos em dois momentos, sendo eles, Resultados do software e Resultados de pesquisa.

5.1.Resultados do Software

Após a realização do projeto foi possível criar um software que é capaz de, através da leitura de arquivos de texto, gerar um modelo visual do funcionamento de uma RSSF. Este software contém um menu inicial “simples e intuitivo” segundo alunos entrevistados, para que qualquer usuário, desde o menor conhecimento em computação até o maior conhecimento em computação, consiga utilizar, como demonstra a figura 6:

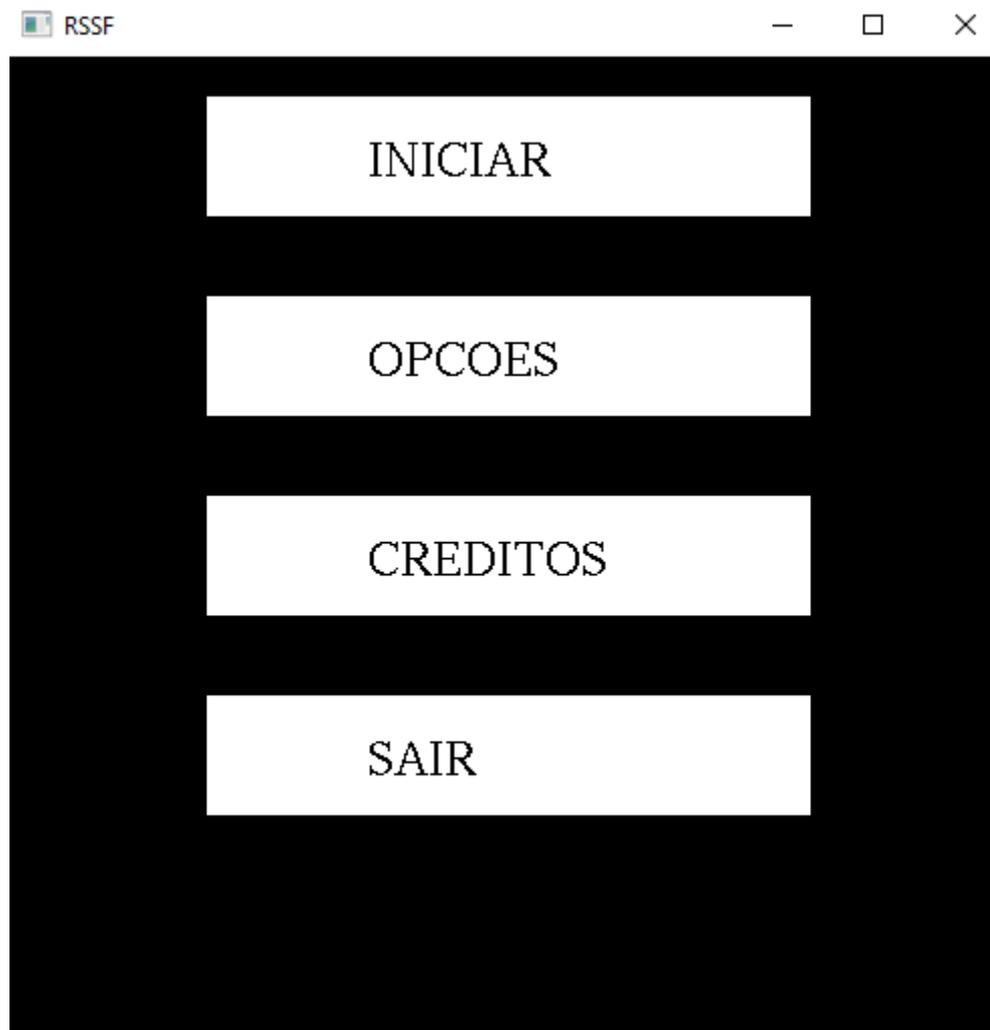


Figura 6 - Menu inicial do programa

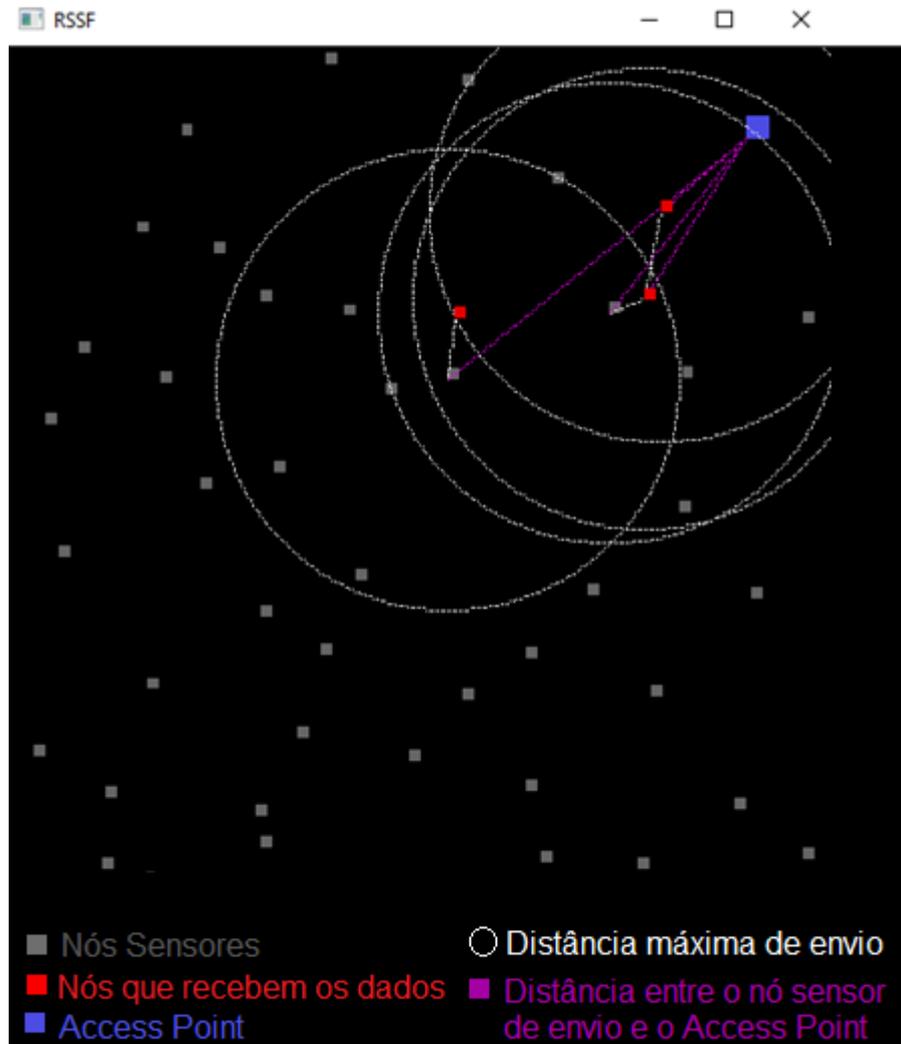


Figura 8 – Ilustração do programa após a utilização de recursos

Foi possível também demonstrar no software a distância máxima de envio, a distância do dado até o Access Point e a distância de roteamento do dado, foi possibilitado ao usuário a exclusão de um nó sensor da rede, fazendo com que ele consiga observar o funcionamento de um algoritmo de roteamento ao não encontrar o mesmo nó, como podemos ver nas figuras de números 9 e 10:

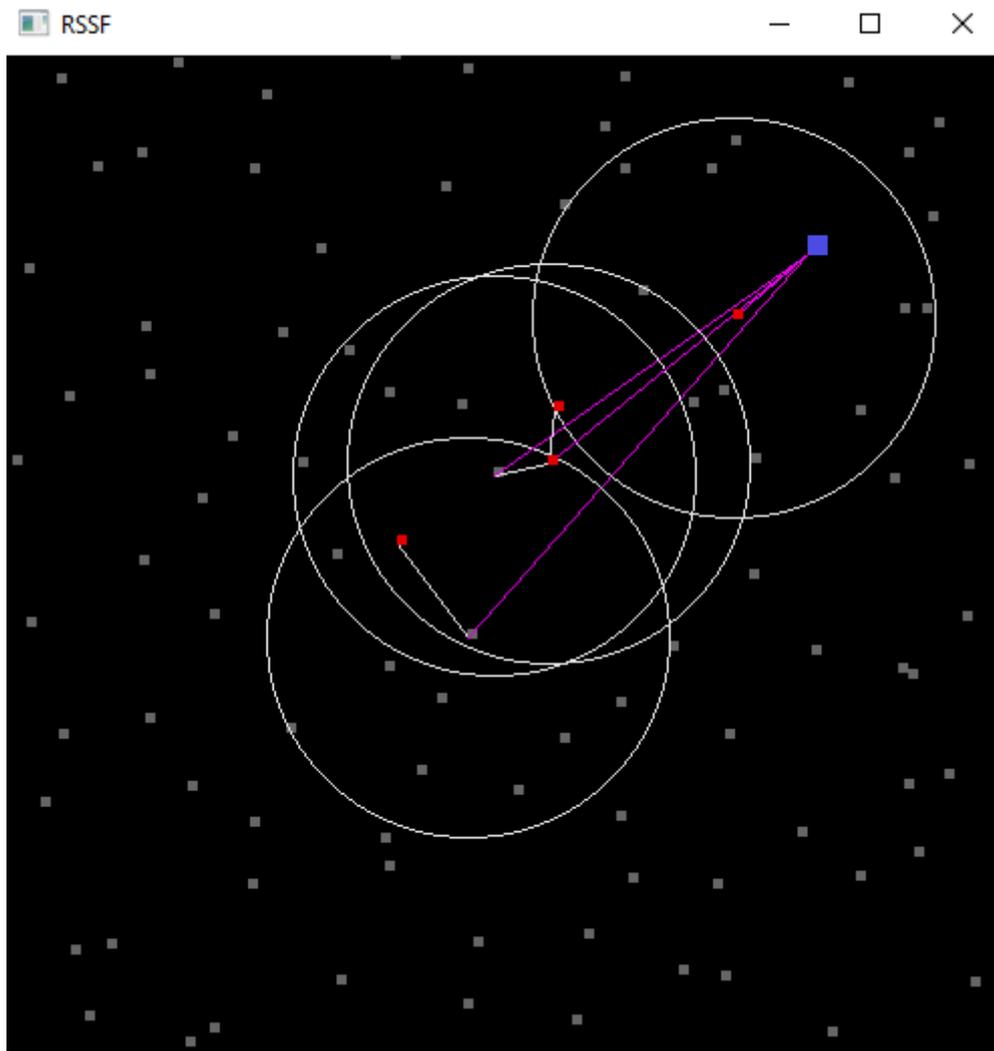


Figura 9 – O funcionamento da rede com todos os nós

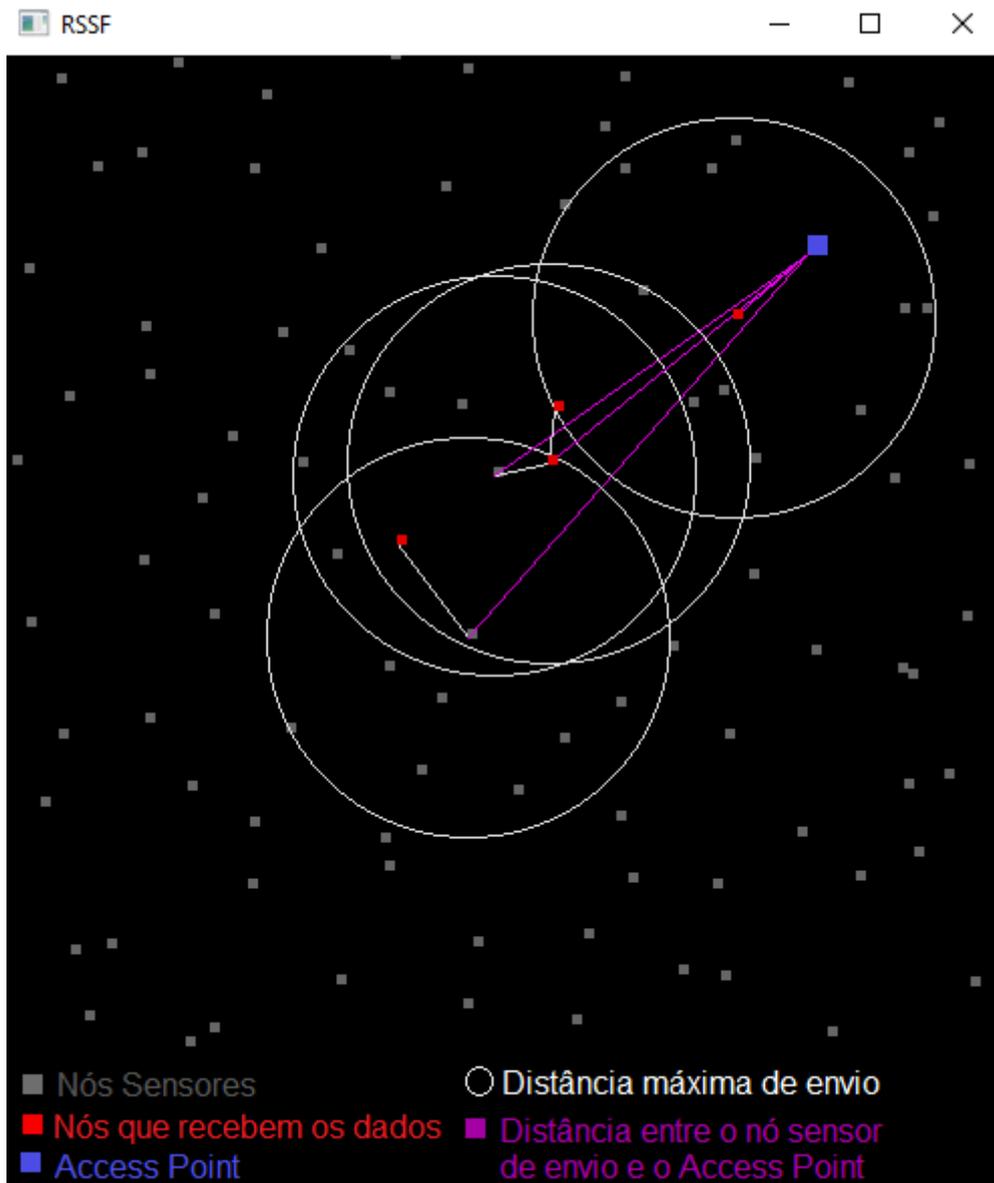


Figura 10 – O funcionamento da rede após a exclusão de nós sensores

O programa permitiu também aos usuários observarem casos onde o dado não consegue atingir o *Access Point* devido a falta de sensores disponíveis para o envio do dado, fazendo com que o dado permaneça sem ser roteado ao encontrar tal caso, como podemos ver na figura 11:

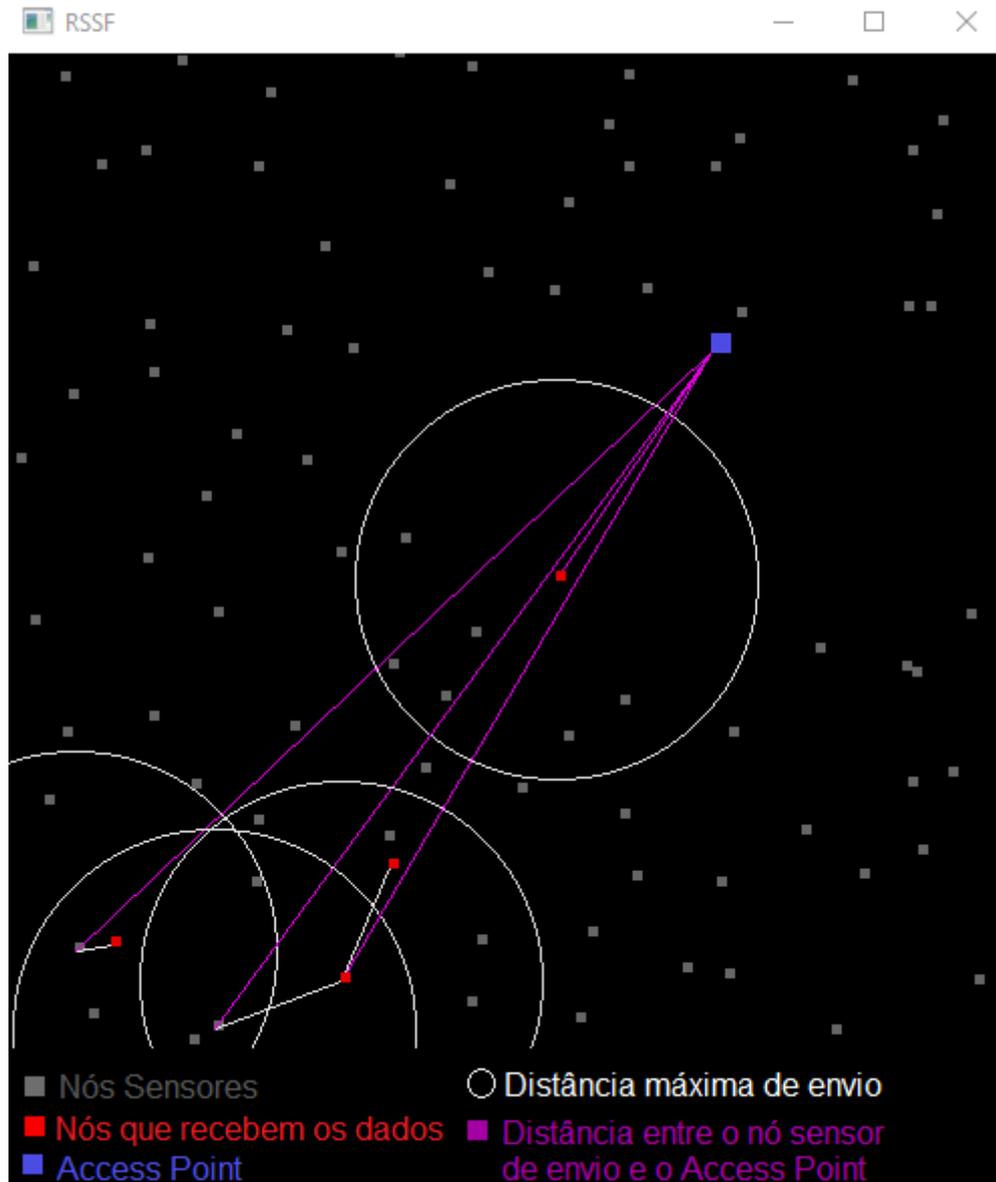


Figura 11 – Caso onde não ocorre o roteamento ao Access Point

Além disso, foi gerado junto ao projeto um tutorial de como utilizar o programa. Para mais informações, ler o anexo 2.

5.2.Resultados de Pesquisa

Após a conclusão do *software* o mesmo foi aplicado em uma turma contendo 26 alunos de engenharia de computação do CEFET-MG e foram disponibilizados aos mesmos dois questionários virtuais contendo questões sobre o ensino, aprendizagem e a utilização do programa (para visualizar os questionários ver anexos 3 e 4).

Destes alunos 22 alunos responderam o questionário sobre os seus conhecimentos prévios em Redes de Sensores Sem Fio, e 21 responderam sobre os seus conhecimentos posteriores.

Das 22 respostas ao primeiro questionário 18 respostas foram nominais, já no segundo questionário 19 respostas foram nominais.

Os resultados demonstram que os alunos entrevistados possuem um grau de alto conhecimento em computação. A média de conhecimento da turma em um grau de 1 a 5 é de aproximadamente 3.95 conforme mostra o gráfico na figura número 12:

Qual o seu grau de conhecimento de computação? (22 responses)

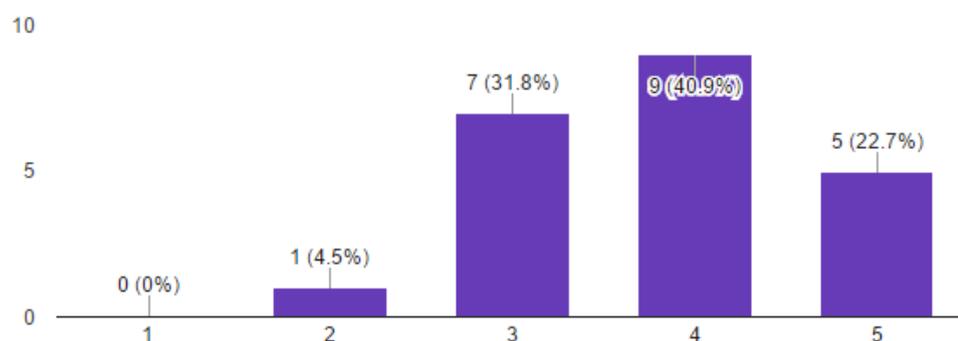


Figura 12 - Grau de conhecimento de computação dos entrevistados

Ao serem perguntados sobre o grau de conhecimento em RSSF antes da aplicação do programa e após a aplicação do programa, foi observado que o grau de conhecimento dos alunos na área subiu de 4.68 para 6.61 (cálculo feito através de média ponderada), segundo os mesmos, o que representa um acréscimo de quase 20% conforme podemos observar na figura de número 13 e 14:

Qual o seu grau de conhecimento sobre o tema Redes de Sensores Sem Fio (RSSF)?

(22 responses)

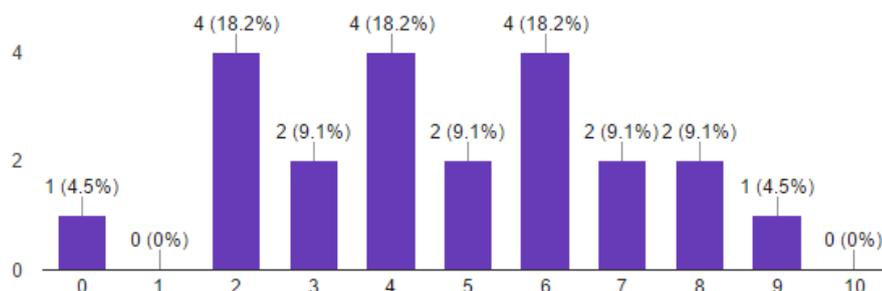


Figura 13 – Grau de conhecimento anterior em RSSF

Após a aula direcionada qual o seu grau de conhecimento sobre o tema Redes de Sensores Sem Fio (RSSF)?

(21 responses)

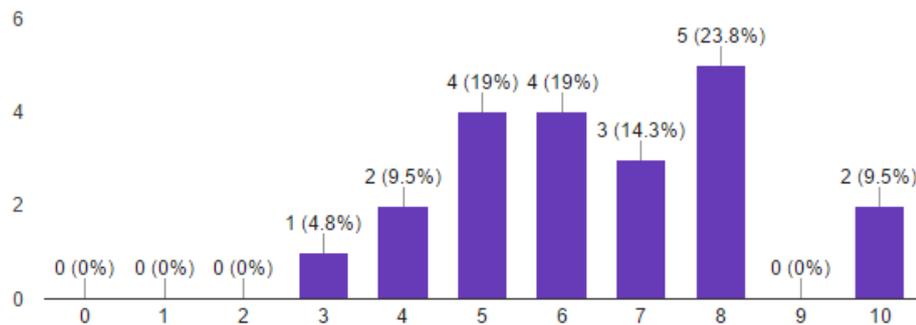


Figura 14 Grau de conhecimento em RSSF após a aula direcionada

Observou-se também que o grau de interesse pela área teve um acréscimo de pouco mais de 10%, isso demonstra que o software é determinante para que os alunos tenham o interesse pela área despertado, enquanto alunos que não possuem interesse nenhum pela área continuaram a não ter muita afinidade. Com estes gráficos observamos também que muitos alunos aumentaram o grau de afinidade com a área ao máximo, demonstrando interesse 5 em uma escala de 1 a 5, o que não havia acontecido antes da utilização do *software*, conforme vemos nas figuras de números 15 e 16:

Qual o seu grau de interesse pela área de RSSF? (22 responses)

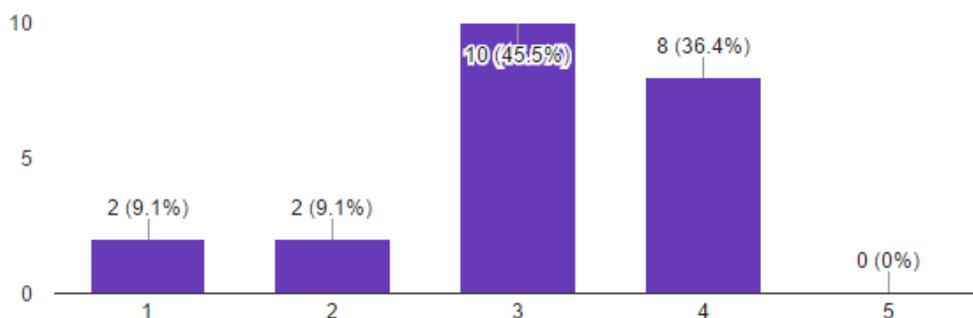


Figura 15 - Grau de interesse antes da utilização do software

Após a aula direcionada através da utilização do objeto de aprendizagem qual passou a ser o seu grau de interesse pela área de RSSF?

(21 responses)

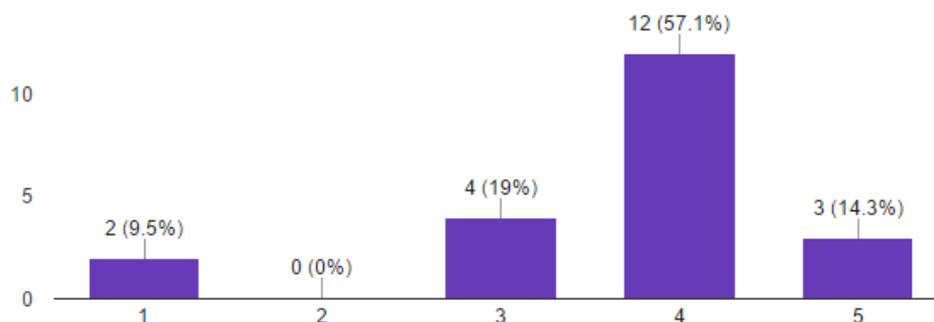


Figura 16 Grau de interesse pela área após a aplicação do software

Outro ponto relevante encontrado na pesquisa foi o acréscimo de conhecimento de termos utilizados em RSSF, que foi de uma média de 10.12 em 25 (40.48%) termos conhecidos por aluno no primeiro questionário para o valor de 18.16 em 25 (72,46%), o que condiz com o primeiro questionário sobre o grau de conhecimento em RSSF antes e após, visto que o acréscimo na pesquisa anterior foi de quase 20% e neste momento o acréscimo foi de aproximadamente 32% conforme as figuras de números 17 e 18:

Quais dos seguintes termos e definições é de seu conhecimento sobre redes de sensores sem fio?

(21 responses)

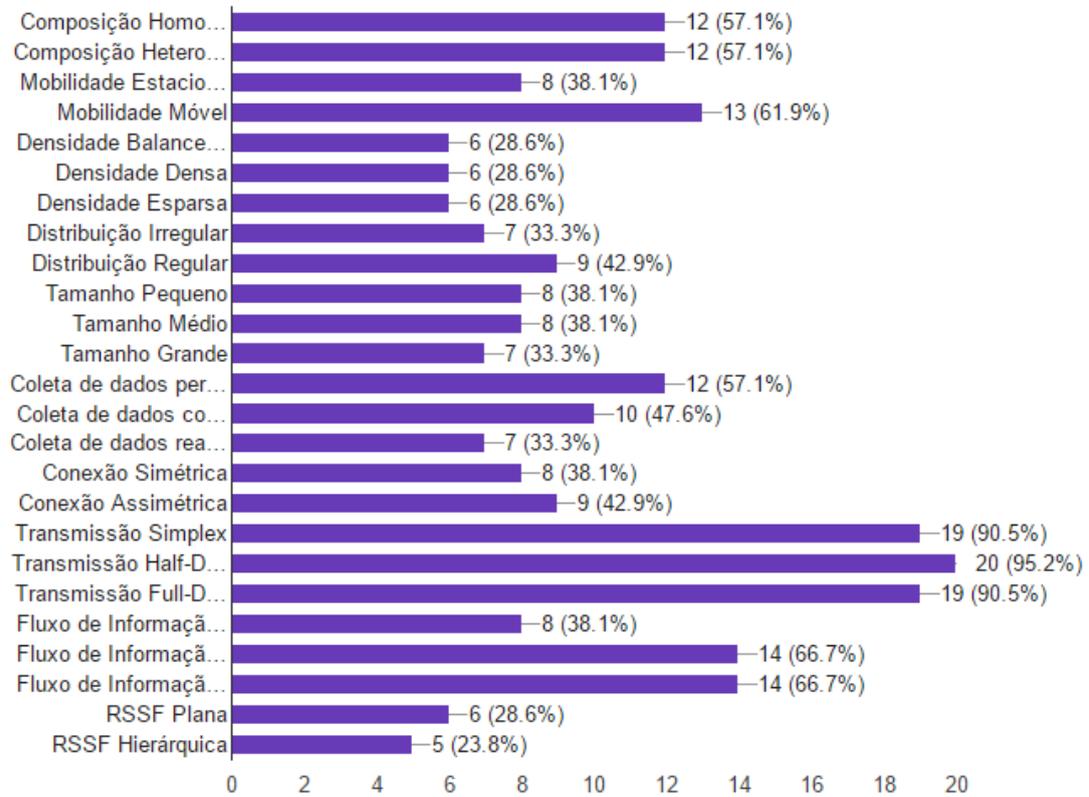


Figura 17 - Conhecimento prévio dos termos de RSSF

Quais dos seguintes termos e definições você consegue definir depois da aula sobre redes de sensores sem fio?

(21 responses)

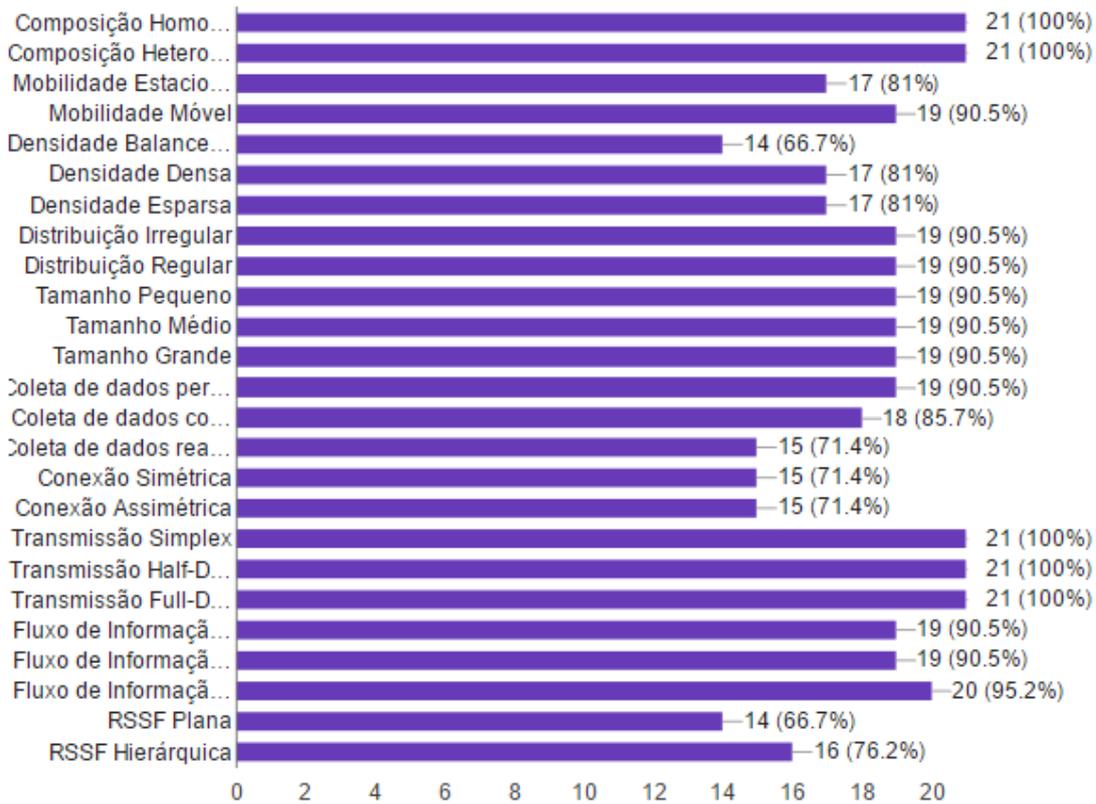


Figura 18 – Conhecimento dos termos após a apresentação do trabalho

Conforme visto nos gráficos, existiam termos em que a turma possuía pouco conhecimento como termo RSSF hierárquica, que teve um acréscimo de mais de 50% de conhecimento após a utilização do OA. Além disto, vimos que o objeto possui uma grande importância para o entendimento do conceito de fluxo de informação *flooding*, visto que o mesmo foi o utilizado durante o projeto e demonstrado através do OA.

Ao serem perguntados sobre as facilidades que o programa trouxe os alunos demonstraram que o programa facilitou a compreensão lógica do roteamento como podemos ver nas citações feitas durante a entrevista:

- “Permitiu observar com clareza a execução dos algoritmos de roteamento utilizados pela rede de sensores sem fio. Facilitou a ilustração dos conceitos necessários”
- “Permitiu simular o funcionamento da rede e do seu algoritmo de roteamento e

verificar o que ocorre quando um nó se inativa.”

- “O programa permitiu visualizar de forma prática o funcionamento de uma RSSF”
- “O programa me permitiu observar visualmente o significado dos termos apresentados, e entender com clareza processo de roteamento implementado pelo aluno”.
- “Uma representação mais ampla de como funciona o sistema de monitoramento e funcionamento dos "nós" e o "ACC"”
- “Permitiu ver com mais clareza como atuam os sensores durante o processo. Visto que esses temas nas aulas são bastante teóricos e cansativos, mais dinâmico”
- “Visualizar melhor a coleta de informações por parte dos sensores e seu envio para áreas desejadas”

Ao serem perguntados sobre as dificuldades em utilizar o programa, os alunos demonstraram que o programa é bastante intuitivo, nenhum aluno apresentou dificuldade para utilizar o programa.

Ao serem perguntados sobre os pontos positivos do programa, grande parte dos alunos citou que o programa é fácil e intuitivo, como observamos nas citações feitas nas entrevistas:

- “Intuitivo e de fácil compreensão do fluxo de dados.”
- “Intuitividade e facilidade”
- “Me auxiliou na compreensão de diversos novos termos e funcionamento de diversos equipamentos e novas lógicas”
- “O programa permitiu observar com clareza o processo de roteamento entre os nodos da rede sem fio e o *Access Point*.”
- “Uma representação simples onde qualquer pessoa leiga na área consiga identificar o funcionamento de uma RSSF”
- “Simplicidade e Objetividade.”

- “Interface bem didática”

Ao serem perguntados sobre os pontos negativos do programa os únicos pontos negativos levantados pelos alunos foram:

- “A rede simulada no programa era pequena e simples, didaticamente adequada, porém não foram apresentados todos os métodos de roteamento.”
- “Interface um pouco reduzida”
- “Uma interface de configuração”
- “Programa muito simples, poderia ter mais funções de algoritmos usados nestes tipos de sistemas.”

6. CONCLUSÃO E TRABALHOS FUTUROS

Com este trabalho conseguiu-se gerar um objeto de aprendizagem simples e eficaz, capaz de ajudar alunos na compreensão do funcionamento de uma RSSF *floodfill*. Além disso, foi observado que alunos passaram a ter um interesse maior e conhecimento na área com a utilização deste modelo.

Gerou-se também um material de apoio pedagógico de conhecimento teórico na área através da junção de conceitos feitos no mesmo. Além disso, o projeto possibilita obter dados como quais as distâncias percorridas por um dado até chegar em um nó sensor ou quais os nós sensores mais acessados, e com isso futuramente o projeto é capaz de se transformar em um simulador gráfico real, com o acréscimo de dados como gasto de bateria e afins, para então facilitar a expansão da área de conhecimento.

Observou-se também que para uma melhor aprendizagem dos conteúdos, seria necessária a implementação de outros tipos de protocolos de rede, como redes hierárquicas, redes que enviam dados aos nós mais longínquos possíveis, ou protocolos baseados em protocolos reais como o AODV e o DSR por exemplo.

Com este trabalho conseguimos também obter uma melhoria no entendimento de um algoritmo de roteamento, o que facilita futuros pesquisadores a investirem cada dia mais em algoritmos melhores e mais eficazes.

Sugere-se como trabalhos futuros a transformação deste OA em um simulador real para determinados tipos de implementação, além da criação de novos modelos didáticos de ensino, além da melhoria de interface e possíveis otimizações possibilitando assim aos alunos uma melhoria no ensino da área.

7. REFERÊNCIAS BIBLIOGRÁFICAS

BASTOS, Alex Vidigal. **NSINKS: Protocolo de roteamento para redes de sensores sem fio em redes com vários sorvedouros**. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal de Viçosa, 2011.

BENTO NETO, A. **LCMS para a construção de Objetos de Aprendizagem baseados em modelos**. Dissertação (Graduação) – Informática na Educação, Universidade do Vale do Itajaí, Itajaí, 2010. Disponível em: <<http://siaibib01.univali.br/pdf/Antonio%20Bento%20Neto.pdf> >. Acesso em: 22 Dez. 2016.

BILINSKI, Fernando Rodrigo. Internet das coisas. **CEP**, v. 81531, p. 980.

BRAGA, Thais; R. M. SILVA, Fabrício A.; RUIZ, Linnyer B.; NOGUEIRA, José Marcos S.; LOUREIRO, Antonio A. F. **Um Elemento Autônomo para Redes de Sensores Sem Fio**. Ce-resd/sbc, v. 4, n. 9, p.1-16, jan. 2006.

BRAGA, Juliana Cristina; DOTTA, Silvia; PIMENTEL, Edson; STRANSKY, Beatriz. **Desafios para o Desenvolvimento de Objetos de Aprendizagem Reutilizáveis e de Qualidade**. Workshop de Desafios da Computação Aplicada à Educação (desafie). Universidade Federal do ABC – UFABC, 2012.

BRAGA, Juliana Braga; PIMENTEL, Edson; DOTTA, Silvia. **Metodologia INTERA para o desenvolvimento de Objetos de Aprendizagem**. II Congresso Brasileiro de Informática na Educação. XXIV Simpósio Brasileiro de Informática na Educação. Centro de Matemática, Computação e Cognição – Universidade Federal do ABC (UFABC) – Grupo de pesquisa INTERA, 2013.

BRAGA, Juliana Cristina. **Objetos de aprendizagem, volume 1: introdução e fundamentos**. Santo André: Editora da UFABC, 2014.

GIL, A.C. **Como elaborar projetos de pesquisa**. São Paulo: Atlas, 2008.

HEREK, Thiago Alexandre. **Análise comparativa de desempenho dos protocolos de roteamento em redes móveis ad hoc**. Monografia (Especialização) - Curso de Teleinformática e Redes de Computadores, Universidade Tecnológica Federal do Paraná,

Curitiba, 2011.

HORSTMANN, Cay. **Conceitos de Computação com o Essencial de C++**. 3ª edição. Porto Alegre: Bookman, 2005.

LOUREIRO, Antonio A.F.; NOGUEIRA, Jose Marcos S.; RUIZ, Linnyer Beatrys; MINI, Raquel Aparecida de Freitas; NAKAMURA, Eduardo Freire; FIGUEIREDO, Carlos Maurício Serodio. **Redes de Sensores Sem Fio**. Departamento de Ciência da Computação Universidade Federal de Minas Gerais, Belo Horizonte, s/d.

MANSSOUR, Isabel Harb. **Introdução à OpenGL**. Agosto de 2005. Disponível em: <<http://www.inf.pucrs.br/~manssour/OpenGL/Tutorial.html>>. Acesso em: 09 de setembro de 2016.

MICROSOFT. **Visual Studio 2008**. 2008. Disponível em: <[https://msdn.microsoft.com/pt-br/library/52f3sw5c\(v=vs.90\).aspx](https://msdn.microsoft.com/pt-br/library/52f3sw5c(v=vs.90).aspx)>. Acesso em: 08 de março de 2017.

NETWORK SIMULATOR 3 (Org.). **Ns-3 Rastreamento Versão ns-3.23**. Disponível em: <<https://www.nsnam.org/docs/release/3.23/tutorial-pt-br/ns-3-tutorial.pdf>>. Acesso em: 06 julho 2016.

ORTIZ, Maurício Soares. **Plataforma baseada em sensores sem fio para o monitoramento de parâmetros meteorológicos**. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Faculdade de Engenharia da Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2013.

RUIZ, L. B. **MANNA: A Management Architecture for Wireless Sensor Networks**. Phd. thesis, Federal University of Minas Gerais, 2003.

SCHILDT, Herbert. **C++ - Guia Para Iniciantes**. 1ª edição. Rio de Janeiro: Ciência Moderna, 2002.

VERONA, André Barbosa. **Simulação e análise de redes de sensores sem fio aplicadas à viticultura**. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Estadual de Maringá, 2010.

VINCENZI, Auri Marcelo Rizzo. **Orientação a objeto: definição, implementação e análise de recursos de teste e validação**. Diss. Universidade de São Paulo (USP). Instituto de

Ciências Matemáticas e de Computação de São Carlos, 2004.

ANEXO 1: TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Você está sendo convidado(a) a participar, como voluntário(a), do estudo/pesquisa intitulado(a) “Um objeto de aprendizagem como instrumento de ensino de redes de sensores sem fio”, conduzida por Augusto César Castro Ribeiro. Este estudo tem por objetivo gerar um Objeto de Aprendizagem (OA) para que professores e/ou alunos possam utilizá-lo no ensino de Redes de Sensores Sem Fio (RSSF), além de gerar um simulador capaz de mostrar acesso aos nós sensores e distância percorrida no envio de dados. Você foi selecionado(a) por critério de seleção dos sujeitos do estudo, esclarecido de forma acessível. Sua participação não é obrigatória. A qualquer momento, você poderá desistir de participar e retirar seu consentimento. Sua recusa, desistência ou retirada de consentimento não acarretará prejuízo.

Sua participação no estudo não acarretará riscos ou danos, podendo ser utilizados dados e informações consentidas na pesquisa para fins acadêmicos. As eventuais despesas de participação são de inteira responsabilidade do pesquisador responsável. Sua participação nesta pesquisa consistirá em responder aos questionários aplicados pelo pesquisador responsável e participar de aula explicativa, além de utilizar o OA para fins acadêmicos. Os dados obtidos por meio desta pesquisa serão confidenciais e não serão divulgados em nível individual, visando assegurar o sigilo de sua participação. O pesquisador responsável se compromete a tornar públicos nos meios acadêmicos e científicos os resultados obtidos de forma consolidada sem qualquer identificação de indivíduos ou instituições participantes. Caso você concorde em participar desta pesquisa, assine ao final deste documento que deverá ser entregue ao pesquisador responsável/coordenador da pesquisa.

Declaro que entendi os objetivos e benefícios de minha participação na pesquisa, e que concordo em participar.

Timóteo, 01 de março de 2017

Assinatura do(a) participante

ANEXO 2: Tutorial de como usar:

Como usar

O Objeto de Aprendizagem de RSSF foi desenvolvido com o intuito de ajudar professores no ensino do conteúdo, mostrando graficamente aos alunos os conceitos de distância máxima de envio, re-roteamento, Access Point, nós sensores entre outros.

Para utilizar o OA é necessário seguir os seguintes passos:

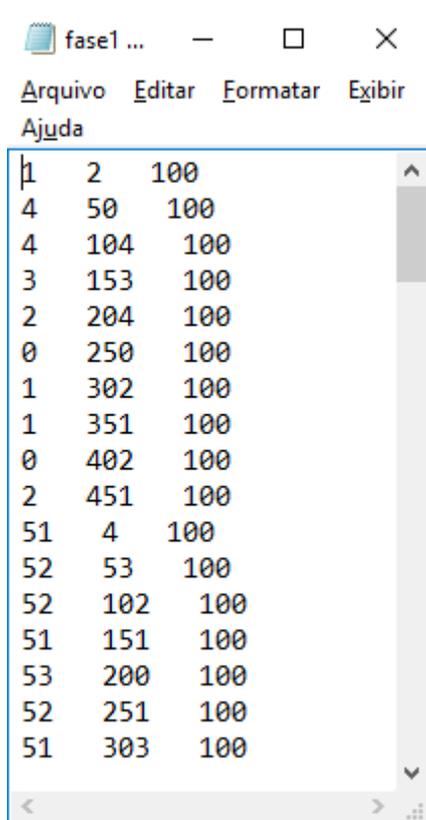
1. Cadastrar o conjunto de nós sensores no arquivo de texto “fase1.txt” disponível dentro da pasta, para isso seguir o seguinte padrão:

1ª coluna - Posição X do nó sensor

2ª coluna - Posição Y do nó sensor

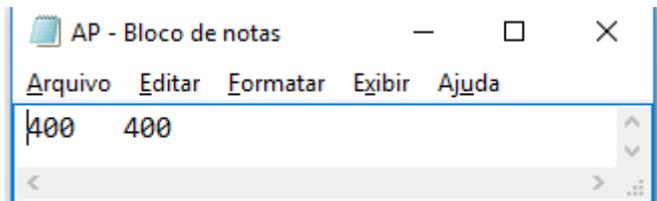
3ª coluna - Distância Máxima de envio do nó sensor

Como mostra o exemplo a seguir:

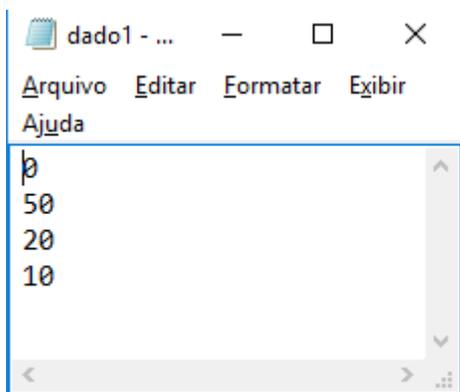


Foram disponibilizados para o usuário 4 (quatro) modelos de teste diferentes, sendo o modelo fase1.1 o com o menor número de nós sensores e o modelo fase1.4 o com o maior número de nós sensores.

2. Cadastrar a posição X e Y do Access Point como mostrado no exemplo:



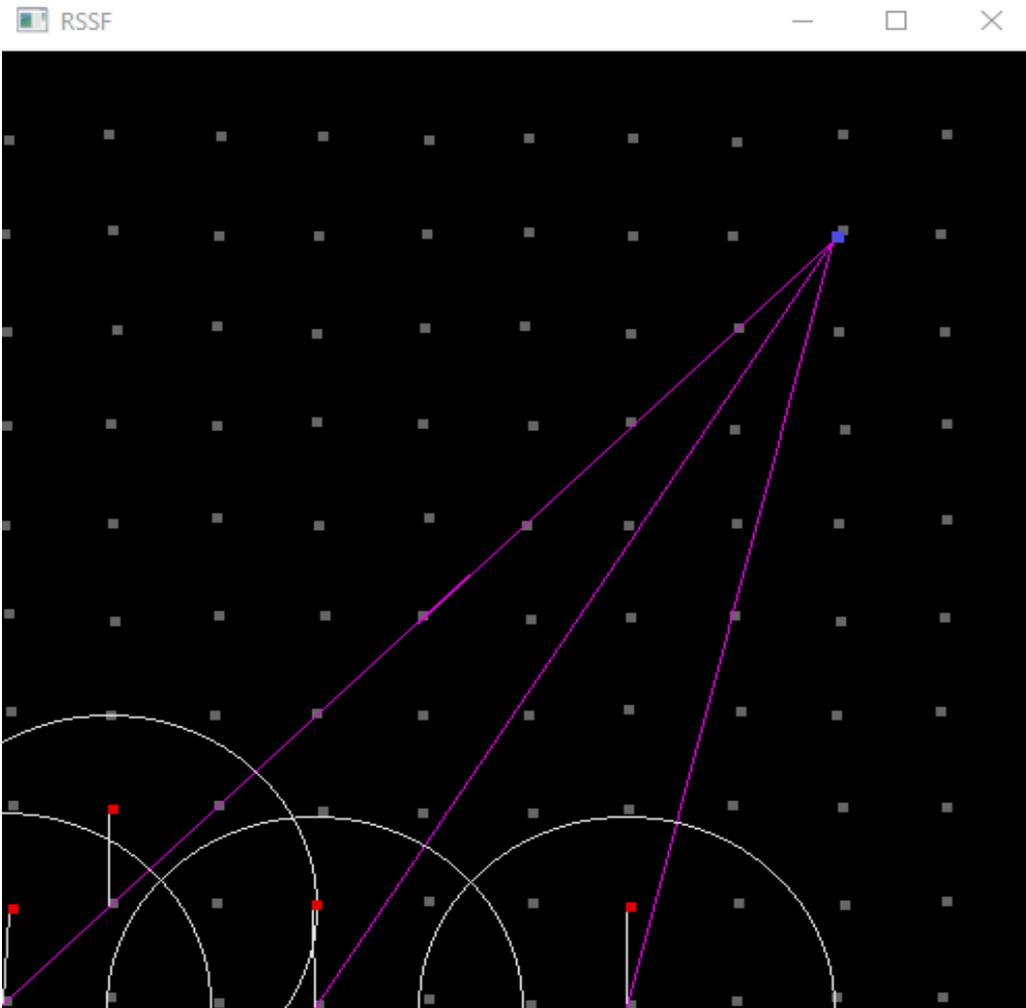
3. Cadastrar o conjunto de dados que serão enviados a partir dos nós sensores no arquivo de texto "dado1.txt" disponível dentro da pasta, para isso seguir o seguinte padrão de cadastrar os dados entre 0 e o valor do máximo de nós cadastrados, sendo 0 o nó sensor cadastrado na primeira linha do arquivo de texto "fase1.txt" e sendo o valor da última linha cadastrada o maior valor disponível conforme o exemplo a seguir:



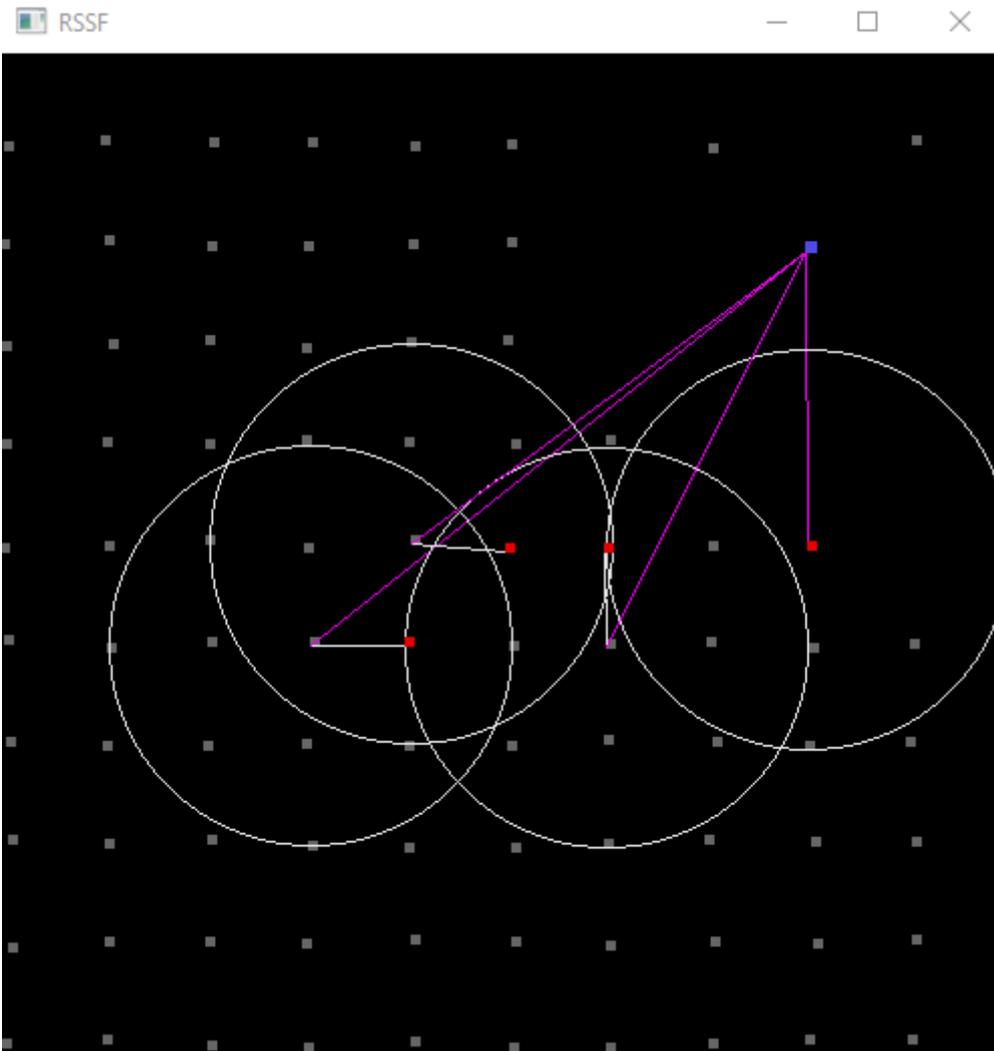
4. Feito isso, executar o programa. Ao executar você verá a seguinte tela:



5. Clique em iniciar, ao clicar você verá uma tela como esta:



6. A partir desse momento o algoritmo começará a mostrar como funciona o roteamento, mostrando o caminho que os dados fazem até chegar ao Access Point, ao finalizar todos os caminhos de envio de dados, o algoritmo começa o envio de dados novamente.
7. Ao clicar próximo a um nó sensor o algoritmo detecta o nó sensor mais próximo e retira o mesmo da lista de nós sensores disponíveis, recalculando assim o roteamento que será feito, fazendo com que o professor possa simular casos como a queima de um nó sensor ou algo próximo a isto, além disso esta opção permite ao professor demonstrar a finalização do processo de envio ao não encontrar um nó sensor mais próximo como vemos na seguinte imagem:



Os pontos em vermelho são os nós que estão recebendo os dados enviados pelo nó que está no centro do círculo e ligado por uma linha roxa, a linha em roxo mostra a distância do nó de envio até o Access Point e o círculo mostra a distância máxima de envio do nó, o Ponto em Azul representa o Access Point e as linhas em branco a distância de roteamento do nó de envio ao nó receptor.

Para uma melhor visualização, o algoritmo permite ao usuário utilizar as teclas “w” “a” “s” e ”d” para navegar pelos nós sensores, alterando assim a posição visualizada pelo usuário, além de permitir a utilização das setas superior e inferior do teclado para dar um zoom ou diminuir um zoom.

ANEXO 3: Formulário de entrevistas prévias:

Conhecimentos previos sobre Redes de sensores sem fio

* Required

Nome completo

Your answer

Qual o seu grau de conhecimento de computação? *

	1	2	3	4	5	
Pouco ou nenhum	<input type="radio"/>	Avançado				

Qual o seu grau de interesse pela área de RSSF? *

	1	2	3	4	5	
Pouco ou nenhum	<input type="radio"/>	Muito interesse				

Qual o seu grau de conhecimento sobre o tema Redes de Sensores Sem Fio (RSSF)?

0	1	2	3	4	5	6	7	8	9	10
<input type="radio"/>										

Quais dos seguintes termos e definições é de seu conhecimento sobre redes de sensores sem fio?

- Composição Homogênea
 - Composição Heterogênea
 - Mobilidade Estacionária
 - Mobilidade Móvel
 - Densidade Balanceada
 - Densidade Densa
 - Densidade Esparsa
 - Distribuição Irregular
 - Distribuição Regular
 - Tamanho Pequeno
 - Tamanho Médio
 - Tamanho Grande
-

- Coleta de dados periódico
- Coleta de dados contínuo
- Coleta de dados reativo
- Conexão Simétrica
- Conexão Assimétrica
- Transmissão Simplex
- Transmissão Half-Duplex
- Transmissão Full-Duplex
- Fluxo de Informação flooding
- Fluxo de Informação unicast
- Fluxo de Informação Multicast
- RSSF Plana
- RSSF Hierárquica

ANEXO 4: Formulário de entrevistas posteriores:

Conhecimentos posteriores sobre Redes de Sensores Sem Fio (RSSF)

* Required

Nome completo

Your answer

Após a aula direcionada através da utilização do objeto de aprendizagem qual passou a ser o seu grau de interesse pela área de RSSF?

	1	2	3	4	5	
Pouco ou nenhum	<input type="radio"/>	Muito interesse				

Após a aula direcionada qual o seu grau de conhecimento sobre o tema Redes de Sensores Sem Fio (RSSF)?

0	1	2	3	4	5	6	7	8	9	10
<input type="radio"/>										

Quais dos seguintes termos e definições você consegue definir depois da aula sobre redes de sensores sem fio? *

- Composição Homogênea
- Composição Heterogênea
- Mobilidade Estacionária
- Mobilidade Móvel
- Densidade Balanceada
- Densidade Densa
- Densidade Esparsa
- Distribuição Irregular
- Distribuição Regular
- Tamanho Pequeno
- Tamanho Médio
- Tamanho Grande

- Coleta de dados periódico
- Coleta de dados contínuo
- Coleta de dados reativo
- Conexão Simétrica
- Conexão Assimétrica
- Transmissão Simplex
- Transmissão Half-Duplex
- Transmissão Full-Duplex
- Fluxo de Informação flooding
- Fluxo de Informação unicast
- Fluxo de Informação Multicast
- RSSF Plana
- RSSF Hierárquica

Quais as facilidades o programa permitiu na área de conhecimento sobre o tema Redes de Sensores Sem Fio (RSSF)?

Your answer

Quais as dificuldades você teve em utilizar o programa?

Your answer

Quais foram os pontos positivos do programa?

Your answer

Quais foram os pontos negativos do programa?

Your answer