



UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
TRABALHO DE CONCLUSÃO EM ENGENHARIA
ELÈTRICA



Estudo da Rede Sigfox para Aquisição de Dados em Aplicações de Internet das Coisas

Autor: Gabriel Marins da Costa

Orientador: Prof. Dr. Ivan Muller

Porto Alegre, 9 de novembro de 2020

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
TRABALHO DE CONCLUSÃO EM ENGENHARIA ELÈTRICA

Estudo da Rede Sigfox para Aquisição de Dados em Aplicações de Internet das Coisas

Gabriel Marins da Costa

Relatório final do Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul como requisito parcial requisito parcial para obtenção do título de Engenheiro Eletricista

Orientador: Prof. Dr. Ivan Müller

Porto Alegre, 9 de novembro de 2020

Sumário

Agradecimentos	iii
Resumo	v
Abstract	vii
Lista de Figuras	ix
Lista de Tabelas	xi
Lista de Abreviaturas e Siglas	xii
Lista de Símbolos	xiv
1 Introdução	1
1.1 Motivação	1
1.2 Objetivos	3
2 Fundamentação Teórica	5
2.1 <i>Internet of Things</i>	5
2.1.1 <i>Link Budget</i>	8
2.2 <i>Low-Power Wide Area Network</i>	9
2.2.1 <i>Narrow-Band</i>	13
2.2.2 <i>Spread spectrum</i>	13
2.2.3 Baseadas em OFDM	13
2.2.4 Acesso ao Espectro Sub-1GHz ISM	13
2.3 Sigfox	14
2.3.1 <i>Link-Budget</i> do Sigfox	18
2.3.2 DBPSK	19
2.3.3 GFSK	20
2.4 Casos de Uso da Rede Sigfox	22
3 Metodologia Experimental	25
3.1 Kit de Desenvolvimento EDK da uDEV	25
3.1.1 Microprocessador ARM Cortex M0+	27
3.2 Módulo Transceptor para Comunicação com a Rede	29
3.3 Acesso à Rede Sigfox	33
3.4 Aquisição e Processamento de Dados	34
3.4.1 <i>Callbacks</i>	34

3.4.2	Geolocalização	36
3.4.3	Processamento em Nuvem	38
4	Resultados e Discussões	41
4.1	Autenticação e Acesso à Rede a Partir de um Dispositivo	41
4.2	Aquisição de Dados Iniciais com <i>Callbacks</i> Tipo Data	43
4.3	Testes de Geolocalização	48
4.3.1	Comparação da Geolocalização com Dois Dispositivos Distintos	48
4.4	Integração com <i>Cloud</i> para Armazenamento e Processamento dos Dados Adquiridos	51
4.4.1	Configuração dos Conectores do Sigfox para AWS	52
4.4.2	Configuração do Banco de Dados	59
4.4.3	Configuração das Regras para Aquisição de Dados	64
4.4.4	Aquisição e Manipulação dos Dados	66
5	Conclusões	69
6	Referências	71
	Apêndices	75
A	Visão Geral do Amazon Identity and Access Management (IAM)	75
B	<i>Template</i> para Criação do <i>Stack</i> no AWS CloudFormation	76

Agradecimentos

Agradeço ao Prof. Dr. Ivan Müller pelo apoio e direcionamento ao longo do projeto desde a concepção.

Agradeço à uDev Soluções Tecnológicas pelos kits de desenvolvimento que foram de muita valia para o presente trabalho e continuaram sendo úteis para o desenvolvimento de trabalhos futuros em telecomunicações e Internet das Coisas.

Agradeço extensivamente a minha família - em especial, à minha mãe - pelo apoio incondicional no meu desenvolvimento acadêmico e profissional.

Resumo

O presente trabalho visa avaliar a viabilidade de uma solução de *IoT* baseada na rede Sigfox. Um sistema de desenvolvimento baseado em ARM com uma antena e modem Sigfox é utilizado para aquisição e transmissão de dados sem fio através de longas distâncias. Testes são feitos de forma a verificar a qualidade do sinal, confiabilidade da rede e aplicabilidade para uso como uma alternativa a ser considerada em projetos de *IoT*. Também são avaliadas as interfaces e funcionalidades do *backend* Sigfox para gerenciamento, armazenamento e *backup* dos dados adquiridos em um sistema de nuvem. Os resultados dos testes mostram que, apesar de haver espaço para melhoria em alguns aspectos da rede como o georreferenciamento, a rede se mostrou capaz de adquirir dados em diferentes localidades através de longas distâncias e integrá-los num sistema de processamento baseado em servidores em nuvem.

Abstract

This work intends to evaluate the viability of an IoT solution using Sigfox. An ARM-based development platform containing a Sigfox modem is used to acquire and transmit wireless data through long ranges. Tests are performed to verify the signal's quality, confiability and applicability of the network to validate its use as an alternative to be considered in IoT projects. The network's backend is also be evaluated to check its capabilities of data management, storage and backup in cloud computing. The results of the tests show that, even though there is space for improvement specially considering the georeferencing aspect of the network, the Sigfox network is capable to acquire data in different locations through long ranges and integrate them into a cloud computing system.

Lista de Figuras

1	Principais casos de uso de IoT globalmente	1
2	Valor adicionado de tecnologias IoT por setor	2
3	Camadas de Abstração dos Modelos de Rede para IoT	6
4	Exemplos de Redes Utilizadas para IoT Classificadas por Taxa de Transmissão Média e Alcance Médio	7
5	Representação Gráfica do Conceito de <i>Link Budget</i>	9
6	Projeções do Mercado de IoT nos Estados Unidos por Segmento de Tecnologia	10
7	Arquitetura Geral Ilustrativa de uma LPWAN	10
8	Bandas de Frequência e Suas Limitações de Potência e Ciclo de Trabalho Definidas pela União Europeia na Faixa entre 863-870 MHz	14
9	Estrutura dos Pacotes da Rede Sigfox para <i>Uplink</i> e <i>Downlink</i>	15
10	Relação entre a Pilha de Protocolos da Rede Sigfox e o Modelo OSI	15
11	Arquitetura da Rede Sigfox	16
12	Cobertura Territorial da Rede Sigfox (azul)	17
13	Cadeia de Valor do Ecossistema Sigfox	18
14	<i>Link-Budget</i> da Rede Sigfox para Um Exemplo de Transmissão a 100 bps	19
15	Diagrama Temporal de um Sinal DBPSK	19
16	Placa do Kit EDK	26
17	Posição dos Dispositivos Presentes no Kit EDK	26
18	Conjunto de Instruções <i>Assembly</i> para as Diferentes Famílias ARM e Aplicações Suportadas	28
19	Regiões de Cobertura Sigfox e Suas Respectivas Frequências e Potência dos Sinais Empregados	29
20	Vistas Superior e Inferior do Módulo Transceptor Wisol	30
21	Características Elétricas e de RF do Módulo Transceptor	31
22	Diagrama de Blocos do Módulo Transceptor	32
23	Implementação por Máquina de Estados Finita do Procedimento de Checagem do Estado Atual do Módulo Transceptor	33
24	Sumário dos Tipos de <i>Callbacks</i> Suportados pela Rede Sigfox	35
25	Comparativo entre as Principais Tecnologias de Geolocalização	37
26	Ilustração da Determinação da Geolocalização de um Dispositivo Sigfox	38
27	Demonstração de Alguns dos Serviços Disponibilizados do AWS e suas Soluções	39
28	Diagrama da Integração do AWS com a Rede Sigfox	40
29	Parâmetros de um Dispositivo com Suporte à Sigfox para Conexão à Rede	41
30	Página de Registro do Dispositivo Sigfox	42
31	Identificação Inicial do Dispositivo no <i>backend</i> da rede Sigfox	42
32	Identificações do Modem do Dispositivo Conectado à Rede Sigfox vistas pelo <i>backend</i>	43
33	Criação do <i>Callback</i> Customizado do tipo DATA	43

34	Resultado do Acionamento do <i>Callback</i>	44
35	Erro da Medida de Latitude e Longitude no <i>Callback</i> Tipo DATA	46
36	Análise da Significância da Cidade para a Qualidade do Sinal (RSSI) para um $\alpha = 0,05$	48
37	Análise da Significância do Kit Utilizado para o Erro de Geolocalização Medido Considerando um $\alpha = 0,05$	51
38	Ilustração da Criação de Um <i>Stack</i> pelo AWS CloudFormation	52
39	Seleção do Tipo de <i>Callback</i> no <i>Backend</i> Sigfox	54
40	Configuração do <i>Callback</i> AWS IoT	55
41	Definição do <i>Template</i> a ser Empregado na Formação do <i>Stack</i>	56
42	<i>Input</i> dos Dados do Dispositivo Sigfox e Definição de Parâmetros	57
43	Configurações de Parâmetros Adicionais do <i>Stack</i>	58
44	Geração do ARN Role ID para Comunicação com o <i>Backend</i> Sigfox	58
45	Comparação da Estrutura de Dados num Banco de Dados SQL e NoSQL	60
46	Serviços de Banco de Dados Disponíveis no AWS	61
47	Exemplos de Bancos de Dados DynamoDB Ilustrando Tabelas, Itens e Atributos	62
48	Criação de um Banco de Dados DynamoDB	63
49	Exemplos de Ações Disponíveis para Uma Regra AWS IoT	65
50	Configuração da Ação para a Regra AWS IoT	66
51	Conjunto de Dados Provenientes do <i>Backend</i> Sigfox Recebidos e Armazenados num Banco de Dados DynamoDB	67
52	Exemplo do Dado Provenientes do <i>Backend</i> Sigfox Recebidos e Armazenados num Banco de Dados DynamoDB	67
53	Diagrama Ilustrativo do Papel do IAM na Aplicação Desenvolvida	75
54	Exemplo do Papel do IAM numa Aplicação Multi-Usuário e com Mais de Um Serviço	76

Lista de Tabelas

1	Comparativo de redes de comunicação sem fio utilizadas para IoT	3
2	Visão Geral das Tecnologias LPWAN	12
3	Testes Iniciais de Aquisição de Dados por <i>Callbacks</i> feitos em São Paulo	44
4	Níveis de Qualidade do Enlace da Rede Sigfox para a Região Brasileira (RC2) com Base no RSSI do Sinal	47
5	Testes Iniciais de Aquisição de Dados por <i>Callbacks</i> feitos em Porto Alegre	47
6	Medidas de Geolocalização com o Uso da Rede Sigfox Utilizando Dois Kits de Desenvolvimento Distintos	50

Lista de Abreviaturas e Siglas

ANATEL	Agência Nacional de Telecomunicações	14, 26
APIs	<i>Application Programming Interfaces</i>	16, 69
DBPSK	<i>Differential Binary Phase-Shift Keying</i>	14, 19
EDK	<i>Embedded Development Kit</i>	25, 27
ERB	Estação Rádio Base	15, 16, 45
GFSK	<i>Gaussian Frequency Shift keying</i>	14, 20
IAM	<i>Identity and Access Management</i>	53, 75, 76
IoT	<i>Internet of Things</i>	1, 9, 14
JSON	<i>JavaScript Object Notation</i>	54, 55, 75
LPWAN	<i>Low-Power Wide Area Network</i>	2, 9, 10, 14
OFDM	<i>Orthogonal Frequency Division Multiplexing</i>	11, 13
RF	Radio Frequência	25, 30
RSSI	<i>Received Signal Strength Indicator</i>	37, 46, 47
SQL	<i>Structured Query Language</i>	59, 63, 64

1 Introdução

1.1 Motivação

Avanços recentes na microeletrônica tornaram possível o desenvolvimento de dispositivos menores, com maior capacidade de processamento e menor consumo energético. Esse contexto, aliado com os desenvolvimentos em telecomunicações permitiram a criação de enlaces de comunicação sem fio cada vez mais confiáveis e apropriados para os ambientes urbanos ou rurais. A junção dos avanços em eletrônica com as telecomunicações permitiram que diversos governos e empresas privadas fomentassem o que conhecemos como *Internet of Things* (IoT), cidades inteligentes (*smart cities*) e Indústria 4.0. Isso decorre do fato que é crescente o número de "coisas" conectadas à uma rede: seja a nível doméstico, com eletrodomésticos e eletroportáteis, ou a nível industrial ou governamental, com malhas de sensores que possam cobrir uma extensa área de infraestrutura, entre outras aplicações que podem ser vistas na Figura 1:

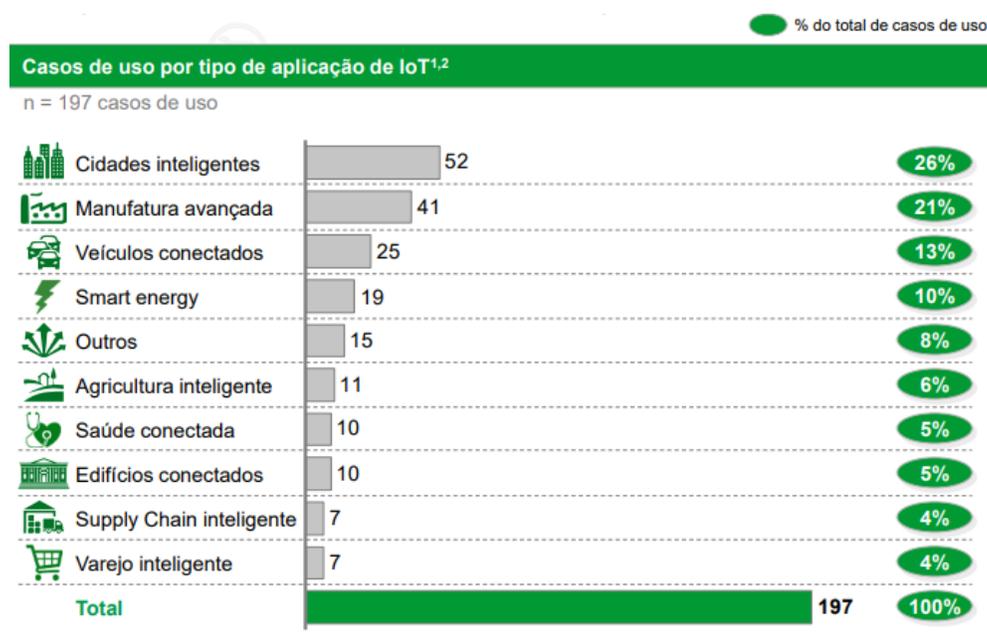


Figura 1: Principais casos de uso de IoT globalmente

Fonte: BNDES (2017)

Não só os usos das tecnologias IoT são variados, como o valor das aplicações vem crescendo substancialmente. A Figura 2 traz o valor adicionado de tecnologias IoT por setor, o que comprova a percepção de melhoria que a aplicação destas tecnologias realizam tanto em empresas, residências ou infraestrutura governamental.

É esperado que o crescimento vivido pelo setor de 44% ao ano em termos compostos (*Compound Annual Growth Rate* (CAGR)) venha a continuar em ritmo acelerado (BNDES, 2017). O

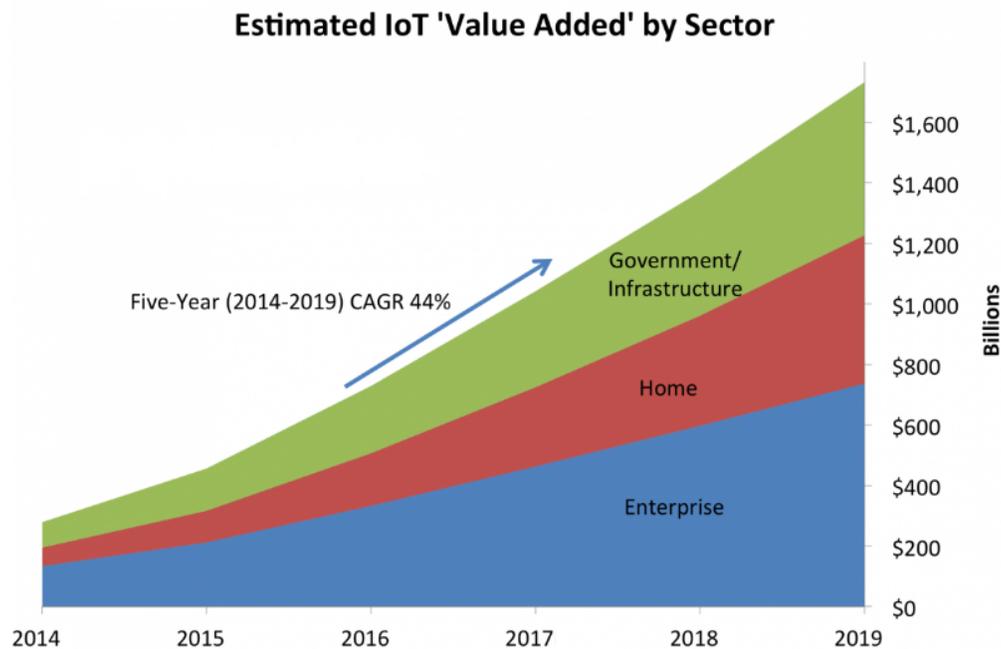


Figura 2: Valor adicionado de tecnologias IoT por setor

Fonte: BNDES (2017)

BNDES realizou, em 2017, um extenso estudo para definir o plano de ação nacional e as linhas de atuação de políticas públicas para IoT até 2022. Além disso, é estimado que a Internet das Coisas possa gerar economias entre 4 e 11 trilhões de dólares até 2025, o que pode ser equivalente a até 11% da economia mundial (Manyika et al., 2015).

Alcançar este impacto irá requerer certas condições que, notavelmente, incluem vencer barreiras técnicas, organizacionais e regulatórias. Em particular, empresas que usam tecnologias IoT terão um papel crítico em desenvolver os sistemas e processos corretos que maximizem a criação de valor. Além disso, IoT terá um enorme impacto em economias emergentes, podendo essas regiões capturar até 40% do valor gerado por IoT ao todo. (Manyika et al., 2015)

Nota-se, pelo mostrado na Figura 1, a variedade de usos possíveis das tecnologias baseadas em IoT. Uma primeira distinção que pode ser feita é quanto às distâncias envolvidas em cada possível tipo de aplicação. Há casos em que IoT é empregado com baixa distância entre os dispositivos, como em prédios ou dentro de residências. Nesses casos a rede empregada pode ser baseada em Wi-Fi ou Bluetooth. Há outros casos, porém, que requerem transmissão de dados em longas distâncias, como nos casos de cidades inteligentes ou agricultura inteligente. Neste último contexto são mais adequadas as redes *Low-Power Wide Area Network* (LPWAN). A Tabela 1 traz um comparativo entre diferentes redes comumente utilizadas no contexto de IoT

Tabela 1: Comparativo de redes de comunicação sem fio utilizadas para IoT

Característica	Wi-Fi	Bluetooth 5	Sigfox	LoraWAN	Zigbee
<i>Throughput</i> Máximo	72Mbps	2Mbps	100bps	50kbps	250kbps
Raio de Atuação Máximo	100m	750m	25km	10km	130m
Topologia	Estrela	Ponto-a-Ponto/Malha	Estrela	Estrela	Malha e estrela
Acesso à Internet no Nó	Sim	Não	Não	Não	Não
Sistema Operacional Móvel no Nó	Sim	Sim	Não	Não	Não

Fonte: Texas Instruments (2017)

Das redes mostradas na Tabela 1, tanto a Sigfox quanto a LoRAWAN são LPWANs. É notável a diferença de *Throughput* e distância entre as LPWANs e as demais redes da Tabela 1. Dadas as diversas aplicações para redes de longa distância no contexto de IoT vivido, é relevante o desenvolvimento de aplicações e sensores que se adaptem a esta nova realidade.

1.2 Objetivos

Este trabalho visa desenvolver uma aplicação de aquisição de dados de sensores utilizando a rede Sigfox através de uma plataforma de desenvolvimento compatível com a rede e com armazenamento dos dados obtidos em nuvem privada. A utilização de sensores para monitoramento remoto com baixo consumo permitem a redução da frequência de manutenções necessárias das estações de medição, resultando no decréscimo das despesas relacionadas a este tipo de atividade. Além disso, o armazenamento em nuvem reduz consideravelmente o risco de perda de um banco de dados, visto que os dados ficam descentralizados e podendo ser acessados por qualquer máquina. Dada a relevância dos aspectos de consumo energético, alcance e confiabilidade da comunicação, o trabalho irá apresentar ensaios experimentais com a aplicação desenvolvida para a análise dessas características.

2 Fundamentação Teórica

Neste capítulo são apresentados conceitos fundamentais para o entendimento de uma aplicação Sigfox, como o *IoT*. São também apresentados aspectos técnicos da rede Sigfox de forma a fundamentar algumas de suas características na camada física.

2.1 *Internet of Things*

A Internet das Coisas, ou IoT, se refere ao uso inteligente de dispositivos e sistemas conectados de forma a melhor explorar os dados coletados por sensores e atuadores embarcados em máquinas e outros objetos físicos (GSMA, 2014).

O IoT permite que objetos físicos possam interagir, compartilhar informações, atuar sobre outros dispositivos e tomar decisões de forma coordenada. É esperado que IoT tenha cada vez mais aplicações significativas, tanto para o contexto residencial quanto empresarial, e que venha a melhorar a qualidade de vida e a economia mundial (IEEE, 2018).

Da mesma forma, o número de objetos com conectividade à Internet já superou a população humana. Hoje em dia, IoT já conecta desde termostatos e casas inteligentes, até dispositivos relacionados à cuidados com a saúde, dispositivos de automação industrial e atuadores de grande porte em indústrias de energia.

Dada a relevância do IoT no contexto profissional e acadêmico, o IEEE já publicou uma série de padrões e normas que delimitam o escopo, as funcionalidades e o *framework* geral de IoT (IEEE, 2017). Entre eles, estão:

- IEEE P2413: desenvolve a arquitetura do *framework* para IoT, que inclui a descrição dos variados domínios de IoT, definições das abstrações dos domínios de IoT e interação entre domínios;
- IEEE 1451-99: focado em desenvolver um padrão para harmonização entre sistemas e dispositivos de IoT, incluindo métodos de compartilhamento de dados, interoperabilidade e segurança da rede;
- IEEE P2510: define medidas de qualidade, controles, parâmetros e definições para sensores de IoT de forma a garantir uma conformidade mínima de performance

Com o desenvolvimento contínuo de IoT, foi necessário desenvolver um novo modelo de abstração de rede, mais adequado do que os já existentes Modelo OSI e TCP-IP (Forouzan and Mosharraf, 2012). A maioria dos modelos iniciais incluíam uma arquitetura de três camadas, constituído das camadas de Percepção, Rede e Aplicação. Neste modelo inicial, os sensores adquiriam e digitalizavam os dados a serem transmitidos num canal seguro na camada de Percepção. Na sequência, a camada de Rede gerenciava a agregação, filtragem e roteamento dos dados para *hubs* IoT e dispositivos inteligentes. Finalmente, a camada de Aplicação entregava os dados ao dispositivo final. Modelos recentes, porém, adicionaram mais abstração à arquitetura da rede. O

resultado é um *framework* de cinco camadas. A evolução dos *frameworks* é ilustrada na Figura 3.

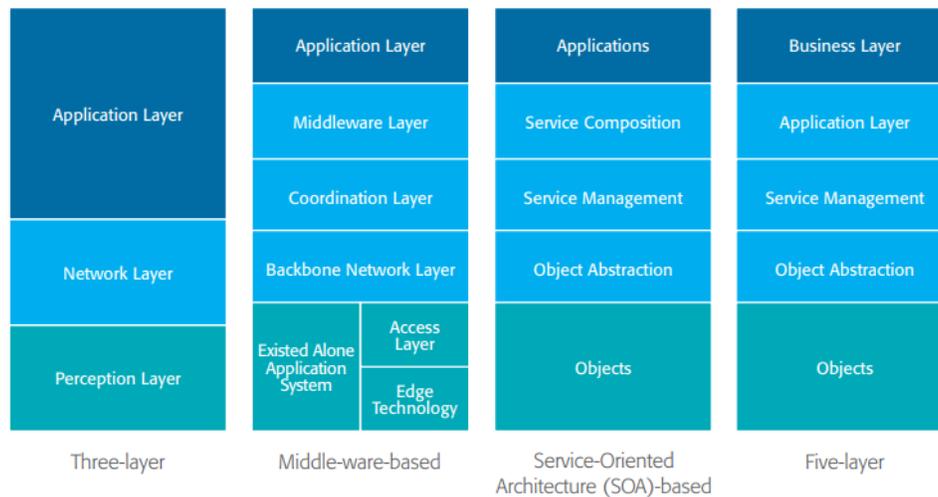


Figura 3: Camadas de Abstração dos Modelos de Rede para IoT

Fonte: IEEE (2018)

No *framework* de cinco camadas, suas funções podem ser divididas em:

- Camada de Objetos: coleta e processamento de dados dos sensores;
- Camada de Abstração de Objetos: transferência de dados produzidos na Camada de Objetos através de tecnologias como RFID, 3G, GSM, Wi-Fi, Sigfox, LoRaWAN para a Camada de Gerenciamento de Serviço;
- Camada de Gerenciamento de Serviço: pareamento de serviços com os *requests* baseados nos endereços de cada um; a camada também processa os dados recebidos, toma decisões e entrega os serviços através de outros protocolos de rede;
- Camada de Aplicação: camada que provê os serviços para as aplicações do usuário final, como casas inteligentes, automação industrial e outros;
- Camada de Negócios: camada de maior abstração, em que o sistema de IoT constrói gráficos, fluxogramas e outras visualizações dos dados recebidos na Camada de Aplicação; nesta camada também ocorre tomada de decisão baseada em análises de *big data*.

Outra importante contribuição da IEEE (2018) para o desenvolvimento do IEEE foi a definição dos Seis Elementos do IoT - que definem os requerimentos básicos para garantir as funcionalidades do IoT. São eles:

- Identificação: inclui os métodos de identificação, endereçamento e acesso múltiplo dos dispositivos;

- Controle e Sensoriamento: inclui a aquisição de dados a nível físico e controle através da rede;
- Comunicação: inclui as tecnologias de telecomunicações entre dispositivos e *hubs* de comunicação;
- Computação: inclui todo o processamento a nível de sistema operacional e de *Real Time Operating System* (RTOS);
- Serviços: inclui os serviços de identificação de dispositivos, agregação de informação, colaboração de priorização;
- Semântica: inclui o reconhecimento e análise de dados de diferentes fontes e formatos;

As tecnologias de redes utilizadas para IoT podem ser divididas em alguns grupos, sejam por velocidade, alcance, banda, entre outros. A Figura 4 traz um comparativo entre alcance médio e taxa de transferência de tecnologias de redes comumente utilizadas em IoT.

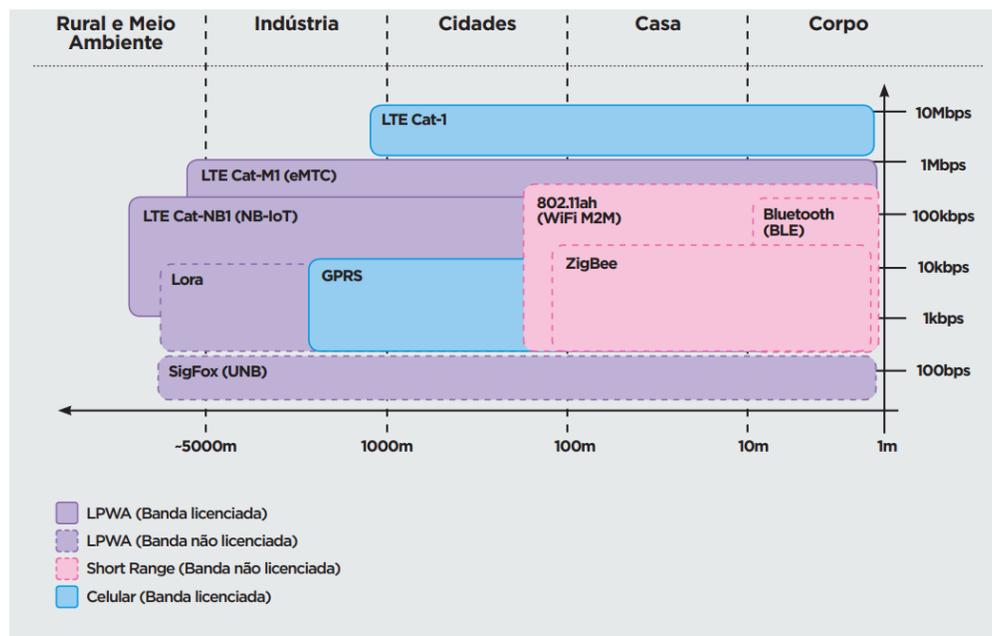


Figura 4: Exemplos de Redes Utilizadas para IoT Classificadas por Taxa de Transmissão Média e Alcance Médio

Fonte: Santos (2018)

Nota-se, que as LPWANs são as que conseguem operar com maior alcance, sendo especialmente adequadas para aplicações em ambiente rural ou que tenham requisições especiais de *link budget*, em troca de uma menor taxa de transferência de dados.

2.1.1 Link Budget

Um dos conceitos mais utilizados em discussões acerca da comparabilidade de uma tecnologia de rede sobre a outra é o conceito do *link budget*. Trata-se do cálculo da perda de potência de sinal entre um gateway e o dispositivo final e essa grandeza acaba definindo os limites de alcance, sensibilidade, perdas no canal e ganho de antena para a rede escolhida (Tabbane, 2017).

A equação da potência do sinal recebida é dada por:

$$P_{RX} = P_{TX} + G_{TX} - L_{TX} - L_{FS} - L_M + G_{RX} - L_{RX} \quad (1)$$

, em que as variáveis representam:

- P_{RX} : Potência recebida (dBm)
- P_{TX} : Potência de saída do transmissor (dBm)
- G_{TX} : Ganho da antena do transmissor (dBi)
- L_{TX} : Perdas internas no transmissor (dB)
- L_{FS} : Perdas no espaço livre (dB)
- L_M : Perdas variadas, como multicaminho (dB)
- G_{RX} : Ganho da antena do receptor (dBi)
- L_{RX} : Perdas internas no receptor (dB)

O receptor, portanto deve ter sensibilidade de recepção (S_{RX} (dBm)) de magnitude maior ou igual a P_{RX} .

A Figura 5 traz uma representação visual da Equação (1)

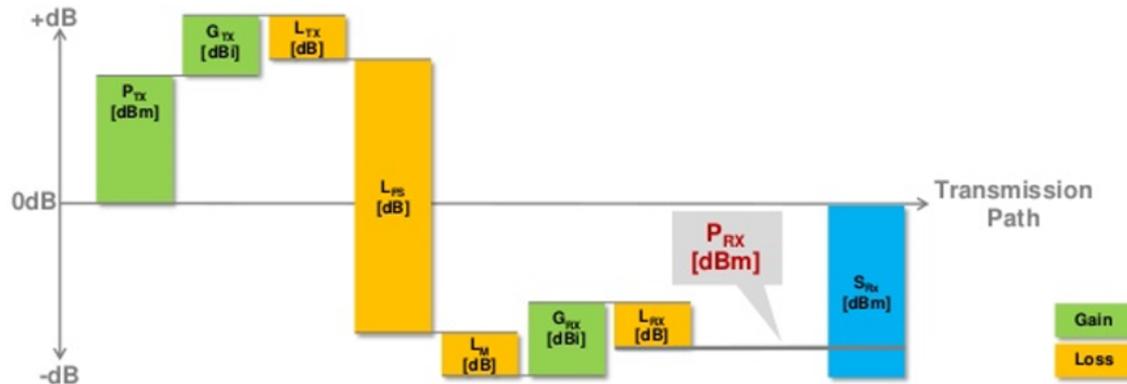


Figura 5: Representação Gráfica do Conceito de *Link Budget*

Fonte: Tabbane (2017)

O *link budget*, portanto, indica a perda máxima que o sinal é capaz de suportar ("orçamento do enlace") de forma a manter a integridade do símbolo transmitido dadas as características de sensibilidade receptor. Um *link budget* maior é desejável, portanto, de forma a poder suportar maiores perdas no canal sem fio de transmissão e com receptores mais baratos.

2.2 Low-Power Wide Area Network

As LPWAN são um dos tipos de rede que podem ser usadas para aplicações de IoT que necessitem de grande distância entre os nodos e os *hubs* da rede.

É esperado que o número de aplicações de IoT cresçam exponencialmente nos próximos anos. Atualmente, a maior parte das aplicações de IoT nos Estados Unidos ainda são baseadas em Wi-Fi ou outras tecnologias de curta distância (Walsh, 2018).

Tecnologias IoT de maior distância como *Long-Term Evolution* (LTE) ou LPWAN, segundo Walsh (2018), têm participação de aproximadamente 20% no número de aplicações. Porém, é esperado que o segmento LPWAN seja o que tenha maior crescimento esperado - de relevantes 58% ao ano - e é esperado que o número de aplicações LPWAN alcance 512 milhões em 2023.

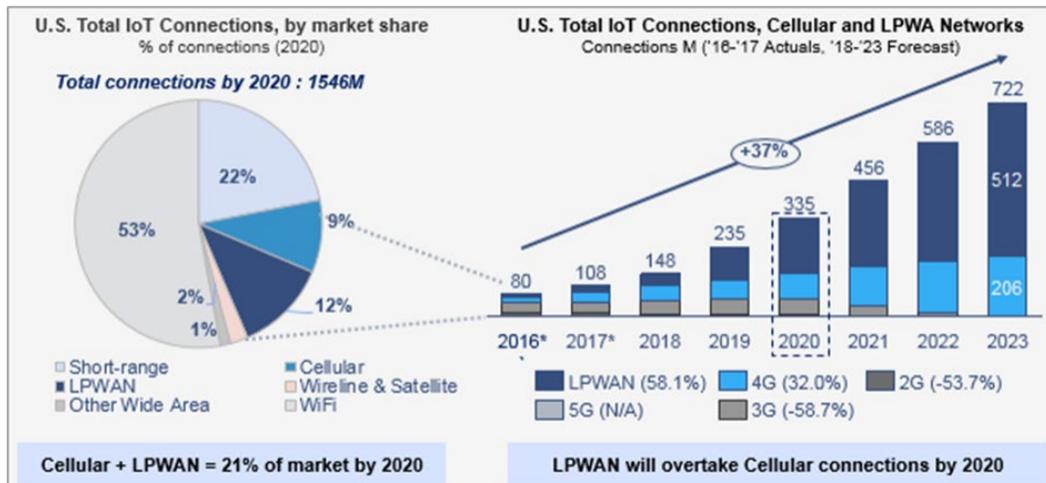


Figura 6: Projeções do Mercado de IoT nos Estados Unidos por Segmento de Tecnologia

Fonte: Walsh (2018)

Um modelo geral de uma rede LPWAN pode ser visto na Figura 7. Nota-se que mesmo as LPWANs utilizam outras redes para acesso a serviços e aplicações ligados à Internet (*Backhaul*), como a rede 4G, Internet cabeada (Ethernet) ou Wi-Fi.

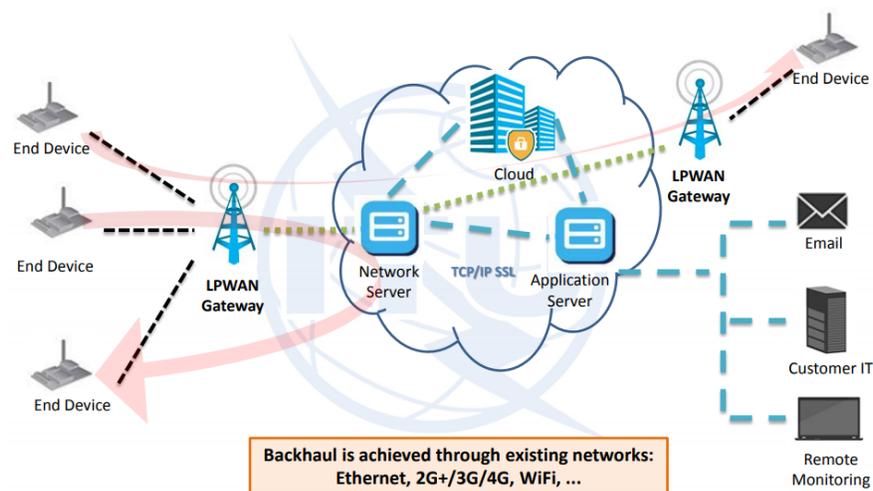


Figura 7: Arquitetura Geral Ilustrativa de uma LPWAN

Fonte: Tabbane (2017)

Mesmo que a maior distância permitida pelas tecnologias LPWAN sejam atrativas para diversos tipos de uso, elas também apresentam algumas desvantagens. Primeiramente, elas alcançam maior distância utilizando uma maior quantidade de energia por bit transmitido. A cobertura de

dispositivos LPWAN é aumentada com o uso de uma menor taxa de modulação, efetivamente colocando mais energia em cada bit (ou símbolo) transmitido e, portanto, resultando num maior *link budget*. Segundo, a operação em baixa potência (*low power*) é alcançada através de uma topologia simples em estrela, com um baixo ciclo de trabalho e utilizando um protocolo de acesso ao meio assíncrono e de baixa complexidade. O resultado disso é que a maioria dos dispositivos trabalhando numa LPWAN somente aguardam por mensagens de *downlink* espontaneamente e, na maioria dos casos, somente após uma transmissão de *uplink* (Ruckebusch et al., 2018).

As limitações mencionadas da LPWAN - especificamente a baixa taxa de dados e baixa complexidade no acesso ao meio - limitam o potencial para reconfiguração da rede a distância e atualização dos dispositivos LPWAN. Por essa razão, os desenvolvimentos atuais de LPWAN estão focados em permitir um *uplink* de milhares de sensores simples, permitindo que eles reportem suas leituras num intervalo relativamente curto.

Existem uma variedade de tecnologias LPWAN proprietárias e padronizadas que adotam diferentes modulações e codificações *Modulation and Coding Schemes* (MCS). De forma geral, as LPWANs podem ser agrupadas conforme seu MCS em (*ultra*) *narrow-band*, *spread spectrum* ou baseadas em Orthogonal Frequency Division Multiplexing (*OFDM*).

Tabela 2: Visão Geral das Tecnologias LPWAN

Tecnologia	Narrow-band		Spread-spectrum		OFDM-based	
	Sigfox	IEEE-802.15.4g	LoRa	IEEE-802.15.4g	IEEE-802.11ah	NB-IoT
MCS	UL: DBPSK DL: GFSK	MR-FSK	CSS	DSS MR-OQPSK	B/Q-PSK 16-QAM	B/Q-PSK 16/64/256-QAM
Largura de Banda	100 Hz	12,5 kHz	125 kHz	2MHz	200-1200 kHz	1-16MHz
Banda	sub-GHz ISM	sub-GHz & 2,4 GHz ISM	sub-GHz ISM	sub-GHz & 2,5 GHz ISM	sub-GHz & 2,4 GHz ISM	sub-GHz ISM
Unidade Máxima de Transmissão	UL: 12B DL: 8B	2kB	250B	2kB	2kB	7991B
Taxa de Dados (kb/s)	UL: 0,1 DL: 0,6	50-200	0,3-37,5	6,25-50	50-800	150-346666,7
						UL: 250 DL: 226,7

2.2.1 *Narrow-Band*

As técnicas de modulação *Narrow-Band* - ou banda estreita - codificam o sinal numa pequena largura de banda de forma a obter maior *link budget* já que o nível de ruído dentro de uma única banda estreita é mínimo. Decodificar o sinal, portanto, não requer um grande ganho através do *de-spreading*, resultando num projeto mais simples de transceptor. A eficiência espectral também é maior já que cada portadora ocupa apenas uma banda bastante estreita. O Sigfox é um exemplo de modulação em banda (ultra) estreita, utilizando uma largura de banda de apenas 100 Hz.

2.2.2 *Spread spectrum*

As técnicas de modulação de *Spread spectrum* - ou espectro espalhado - aumentam o *link budget* por espalhar um sinal de banda estreita numa maior banda de frequência mantendo a densidade de potência. A transmissão resultante é mais resistente à interferência, *eavesdropping* e *jamming*. Porém, a decodificação requer maior ganho o resultado do espalhamento resulta numa menor eficiência espectral. Diferentes variantes de técnicas de espalhamento espectral são utilizado. LoRa utiliza *Chirp Spread Spectrum (CSS)*, enquanto o IEEE-802.15.4g utiliza *Direct Sequence Spread Spectrum (DSSS)*

2.2.3 *Baseadas em OFDM*

As técnicas de modulação OFDM utilizadas no IEEE-802.15.4g e no IEEE-802.11ah priorizam maiores taxas de transmissão de dados em detrimento do *link budget*. O raio de transmissão é menor, especialmente quando consideradas as taxas de transmissão. O NB-IoT, padronizado pelo 3GPP no LTE Release 13 é uma tecnologia LPWAN que utiliza OFDM já que ela opera num espectro licenciado, ela pode alcançar altas taxas de dados enquanto mantém uma grande cobertura.

2.2.4 *Acesso ao Espectro Sub-1GHz ISM*

O espectro disponível não licenciado na faixa do sub-GHz é limitado. Segundo Ruckebusch et al. (2018), por exemplo, há seis bandas na faixa do sub-GHz disponibilizadas pela União Europeia para uso geral na faixa dos 868 MHz e duas bandas na faixa dos 433 MHz. Cada banda possui uma potência de transmissão máxima e um ciclo de trabalho por dispositivo especificados. A máxima potência permitida limita a máxima distância de transmissão alcançável. As limitações no ciclo de trabalho impõem limites na máxima quantidade de mensagens que um dispositivo pode enviar por hora - o que impacta o projeto de roteamento da rede e o protocolo *Medium Access Control (MAC)*.

Algumas limitações feitas pelas legislações europeias, e que são seguidas para a maioria dos projetos de LPWAN no resto do mundo, devem incluir pelo menos uma das seguintes:

- Ciclo de trabalho: dispositivos numa banda de frequência têm a permissão de somente estabelecer conexão durante um determinado período de tempo. A legislação europeia define este intervalo como uma hora e o ciclo de trabalho pode variar entre 0,1% e 10% deste período.
- Acesso moderado ao espectro: dispositivos são limitados a um tempo de acesso contínuo ao

espectro de 100s por hora para cada 200 kHz de largura de banda num espectro. Há duas diretrizes a serem seguidas - *Listen Before Talk (LBT)* e *Adaptive Frequency Agility (AFA)*. Os dispositivos devem checar se o meio está livre antes de uma transmissão (LBT); se o meio está ocupado, então o dispositivo deve aguardar ou checar outra frequência (AFA).

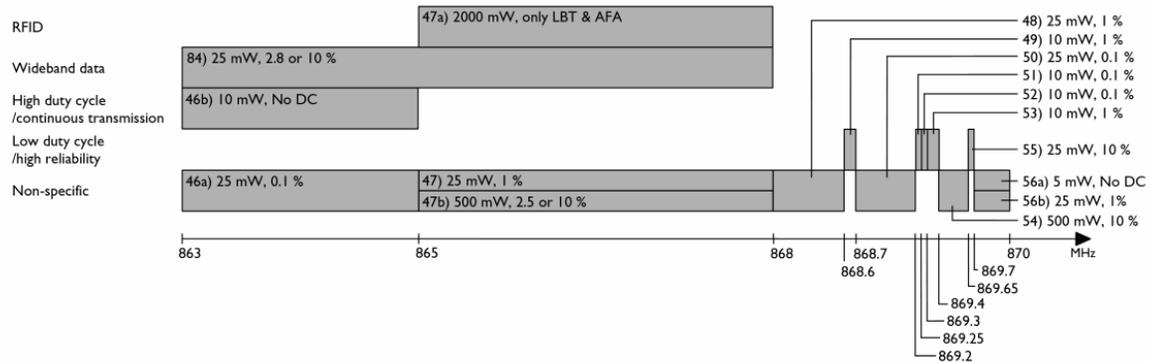


Figura 8: Bandas de Frequência e Suas Limitações de Potência e Ciclo de Trabalho Definidas pela União Europeia na Faixa entre 863-870 MHz

Fonte: Ruckebusch et al. (2018)

A legislação brasileira, dada pela Resolução nº 506 e pela Resolução nº680 da Agência Nacional de Telecomunicações (ANATEL), traz os mesmos requisitos; porém, a frequência pública que pode ser utilizada no Sub-1GHz *Industrial, Scientific and Medical (ISM)* é de 915-940 MHz no Brasil, que é compatível com os sensores e módulos produzidos nos Estados Unidos, cujo espectro sub-1GHz é compatível com o brasileiro (ANATEL, 2017).

2.3 Sigfox

Sigfox é uma companhia francesa que opera globalmente uma rede de mesmo nome cujo objetivo é ser a solução definitiva para IoT de longa distância sem a necessidade de estabelecer e manter estruturas de rede fixas com *gateways*. Essa rede permite conectar em *broadcast* milhares de sensores de baixo consumo que podem continuamente realizar medições e transmiti-las através da rede celular Sigfox.

Sua tecnologia de banda ultra-estreita emprega modulações *Differential Binary Phase-Shift Keying (DBPSK)* e *Gaussian Frequency Shift keying (GFSK)* que transmitem os dados na banda conhecida como *Industrial, Scientific and Medical (ISM)*, que corresponde a uma faixa de 868MHz a 902MHz. O fato da banda ser ultra-estreita permite que o sinal possa atravessar diversas barreiras sólidas, sendo essa característica necessária para o uso urbano, e requer baixa potência, sendo portanto uma LPWAN.

A rede Sigfox é classificada como sendo de Ultra Narrow-Band e de baixo *throughput*. O

uplink emprega modulação DBPSK e cada pacote pode ter entre 0 a 12 bytes de *payload*, com um preâmbulo fixo de 12 bytes que contém o ID do dispositivo e outros metadados. Cada transmissão leva em torno de 2 segundos e contém 3 dos pacotes, cada um transmitido em uma frequência pseudoaleatória dentro da banda disponível. Isso é feito para fins de redundância do sistema e da arquitetura "*fire and forget*", em que as mensagens enviadas pelo dispositivo não recebem um *acknowledgement* da Estação Rádio Base (ERB) (Sigfox, 2018).

A Figura 9 mostra a estrutura de pacotes da rede no *Uplink* e no *Downlink*

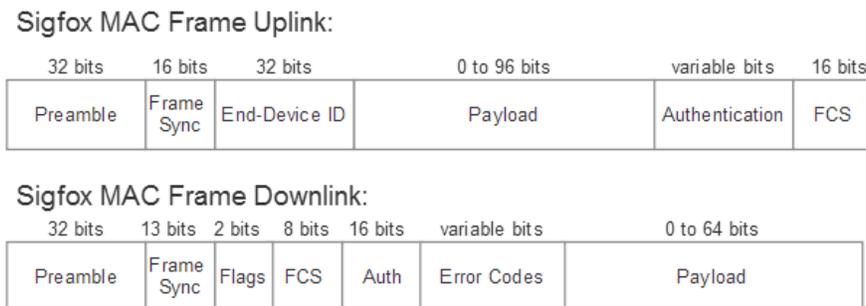


Figura 9: Estrutura dos Pacotes da Rede Sigfox para *Uplink* e *Downlink*

Fonte: Sigfox (2018)

A pilha de protocolos da rede Sigfox pode ser relacionada ao tradicional modelo OSI conforme a Figura 10:

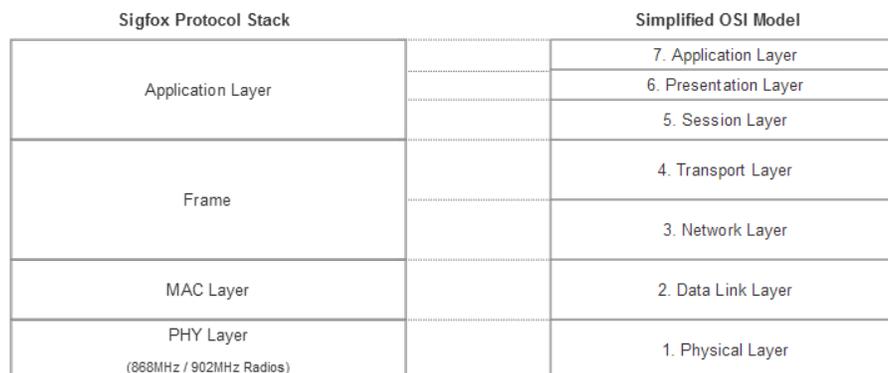


Figura 10: Relação entre a Pilha de Protocolos da Rede Sigfox e o Modelo OSI

Fonte: Sigfox (2018)

- Camada PHY: sintetiza e modula os sinais utilizando DBPSK ou GFSK, dependendo se é *uplink* ou *downlink*;

- Camada MAC: adiciona os campos de identificação e autenticação dos dispositivos e adiciona o código corretor de erros;
- Camada de Frame: Gera o *frame* de rádio a partir dos dados da aplicação e sistematicamente relaciona um *sequence number* ao frame;

A rede Sigfox pode chegar a uma densidade de até 1 milhão de nós por ERB. A densidade depende apenas do número de mensagens que trafegam pela malha da rede de cada ERB. Todos os nós dentro da cobertura de uma ERB formam uma rede em topologia estrela.

Todos os dados são gerenciados pelo *backend* da rede Sigfox. Todas as mensagens que chegam a uma estação de rádio Sigfox devem chegar ao servidor de *backend* através da Internet, que ficarão armazenadas em nuvem. O *backend* armazena e, caso necessite transmitir uma mensagem para um *endpoint*, irá selecionar o *gateway* com melhor conectividade e encaminhar a mensagem no *downlink*.

O *backend* já possui a identificação dos dispositivos de interesse através de seus IDs. Na arquitetura Sigfox, um dispositivo não pode ser acessado diretamente e tampouco pode o *backend* se conectar diretamente a um dispositivo.

A nuvem Sigfox roteia os dados para o destino final. Esse serviço de nuvem oferece *Application Programming Interfaces* (APIs) para integrar as funções da nuvem Sigfox a plataformas de desenvolvimento produzidas por terceiros. O acesso aos dados é feito por comandos de *callbacks* a partir das plataformas desenvolvidas por terceiros.

Para assegurar a integridade dos dados no modelo *fire and forget*, múltiplos *gateways* podem receber uma transmissão de um nó; todas as mensagens são encaminhadas ao *backend* Sigfox e duplicatas são removidas.

A Figura 11 ilustra a arquitetura da rede.

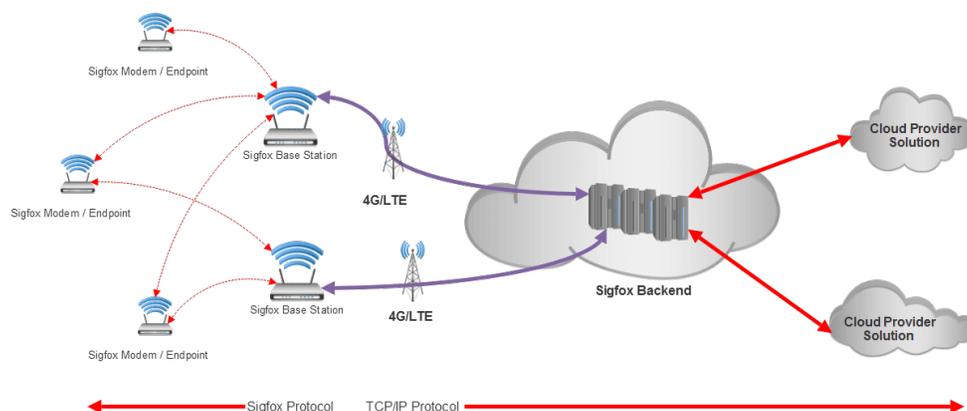


Figura 11: Arquitetura da Rede Sigfox

Fonte: Sigfox (2018)

A Figura 12 mostra os países que já possuem cobertura da rede Sigfox. É esperado que essa cobertura cresça consideravelmente nos próximos anos (Synchrologix, 2018).

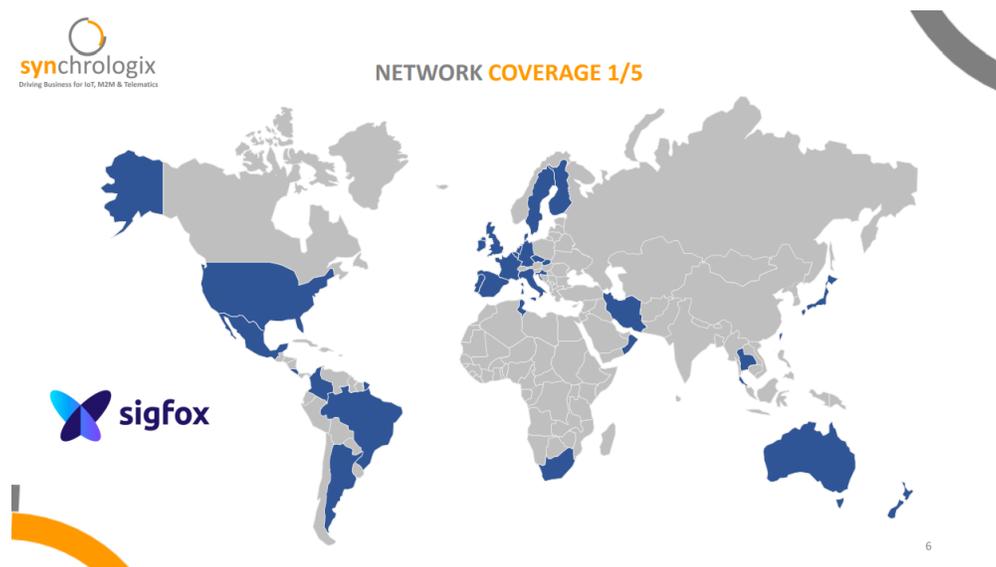


Figura 12: Cobertura Territorial da Rede Sigfox (azul)

Fonte: Synchrologix (2018)

Além de simplesmente uma rede ou um protocolo, a Sigfox pretende criar um ecossistema de desenvolvimento integrados numa cadeia de valor que vai de produtores de módulos, desenvolvedores de kits ou soluções, provedores de plataformas e usuários finais.

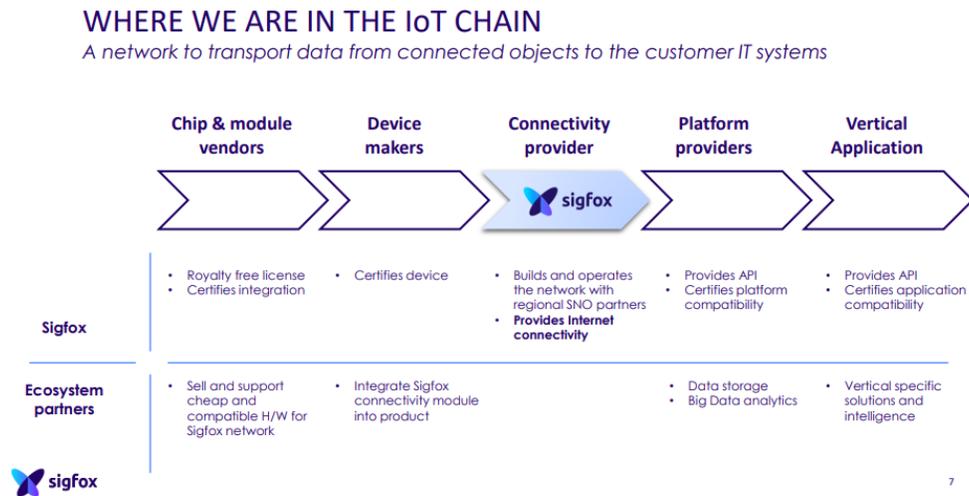


Figura 13: Cadeia de Valor do Ecosistema Sigfox

Fonte: Zuñiga (2017)

2.3.1 Link-Budget do Sigfox

A combinação do uso da modulação DBPSK e GFSK com a baixa taxa de transmissão de dados da rede resultam numa alta sensibilidade do receptor.

O receptor é capaz de adaptar sua sensibilidade de acordo com a taxa de transmissão de dados. Pode ser -142 dBm para uma 100 bps ou pode ser -134 dBm para 600 bps (Sigfox, 2018).

A Figura 14 ilustra um o *link-budget* para uma transmissão de 100bps. Numa situação ideal, o dispositivo pode irradiar até 16,15 dBm. Na ERB, o ganho da antena é 5,15 dBi e a sensibilidade do receptor é -142 dBm. Neste caso, o *link-budget* teórico seria 163,3 dB.

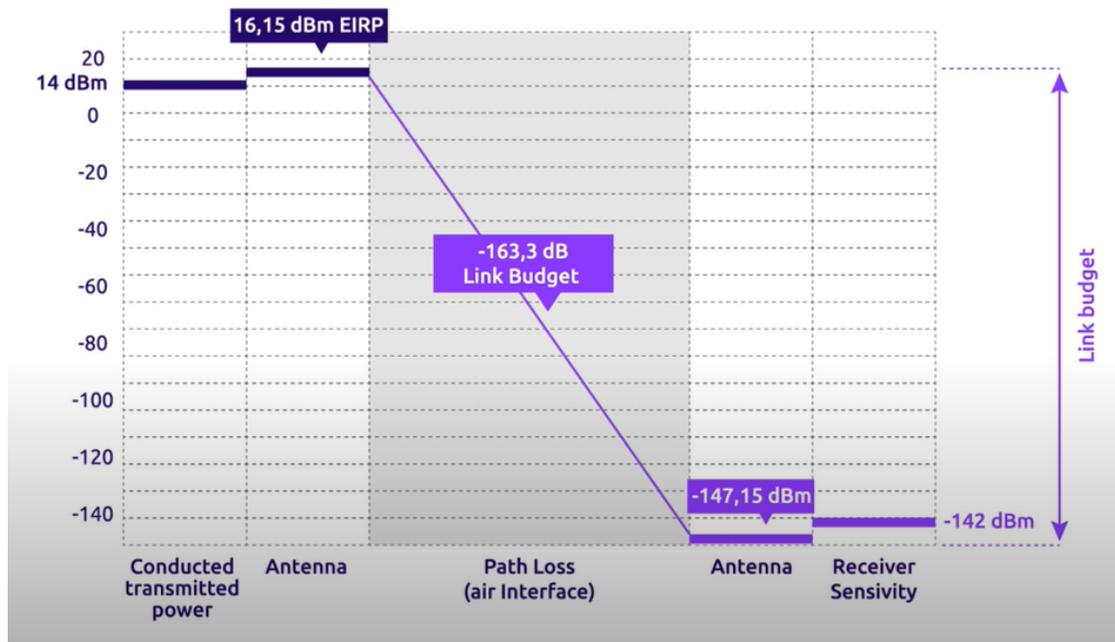


Figura 14: *Link-Budget* da Rede Sigfox para Um Exemplo de Transmissão a 100 bps

Fonte: Sigfox (2018)

Por ser um mercado competitivo, o consórcio Sigfox não torna público maiores detalhes técnicos sobre o link-budget, mas a informação disponível já mostra a robustez do sistema.

2.3.2 DBPSK

O DBPSK, ou DPSK binário, é um tipo de modulação em que, em vez da informação codificar uma determinada fase instantânea, a codificação define uma mudança instantânea na fase, o que é especialmente útil em meios de transmissão que possam rotacionar a constelação do sinal por algum motivo (Proakis and Salehi, 2008).

Por exemplo, no DBPSK um símbolo "1" pode ser transmitido pela adição de 180 à fase instantânea de um sinal, qualquer que ela seja, enquanto um símbolo "0" pode ser transmitido, neste mesmo exemplo, pela adição de 0, como exemplificado na Figura 15

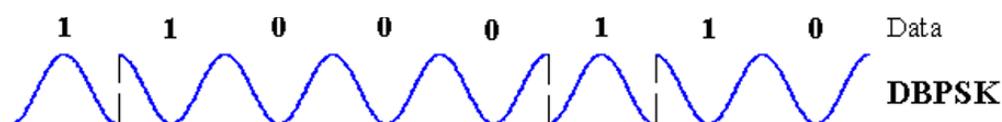


Figura 15: Diagrama Temporal de um Sinal DBPSK

Fonte: Proakis and Salehi (2008)

2.3.3 GFSK

O GFSK é um método de modulação para comunicações digitais encontrado em muitos padrões como o Bluetooth, DECT e Wavenis. O método é uma variante do FM (*frequency modulation*) de uma portadora de frequência ω_c^2 . O *Frequency shift keying* (FSK) converte informação através da variação da frequência da portadora para um símbolo "0" e aumentando a frequência para um símbolo "1" por uma duração de tempo estabelecida. Se a esse processo for aplicado um filtro Gaussiano para o sinal de onda-quadrada que desloca a portadora, surge o GFSK (Gerez, 2016).

O sinal transmitido $s(t)$ pode ser descrito como um cosseno de frequência intermediária ω_{IF} como uma fase dependente do tempo:

$$s(t) = A \cos(\omega_{IF}t + \phi(t)) \quad (2)$$

Na Equação (2), A é a amplitude do sinal que é constante. A fase $\phi(t)$ é derivada dos bits que são transmitidos:

$$\phi(t) = h\pi \int_{-\infty}^t \sum_i \alpha_i \gamma(\tau - iT) d\tau \quad (3)$$

, em que h é o índice de modulação (quanto maior o valor, maior é a largura de banda ocupada pela portadora). Um valor típico para o GFSK é $h = 0,5$. Para o caso $h = 0$ não há modulação da portadora (Gerez, 2016).

α_i é o número sequencial: $+1$ se o i -ésimo bit é "1" e -1 se o bit é "0".

$\gamma(t)$ é o pulso de frequência. Se não há filtro Gaussiano aplicado (FSK em vez de GFSK) o pulso de frequência seria retangular: $\frac{1}{T_S}$ no intervalo $[0, T_S]$ e "0" fora deste intervalo, onde T_S é a duração de um símbolo. Assim, ignorando-se α_i , a contribuição de fase de um símbolo seria $h\pi$. Neste mesmo raciocínio, transmitir uma série contínua de "1"s durante 1 segundo levaria um deslocamento total de fase de $\frac{h\pi}{T_S}$, que resultaria neste mesmo deslocamento de frequência angular. O deslocamento em frequência, portanto, dividindo-se a expressão por 2π , seria de $\frac{h}{2T_S}$. Isso significa que a frequência instantânea no sinal FSK é tanto $f_{IF} - \frac{h}{2T_S}$ ou $f_{IF} + \frac{h}{2T_S}$, para $f_{IF} = \frac{\omega_{IF}}{2\pi}$.

O filtro Gaussiano suaviza o formato do pulso de frequência e o torna mais largo que o período de um símbolo (o que causa interferência intersimbólica). O objetivo é evitar altas frequências causadas pelas harmônicas de um pulso quadrado. Quando a sequência a ser transmitida contém múltiplos bits iguais, o efeito da filtragem desaparece e as frequências instantâneas derivadas para o FSK são alcançadas. Se isso não acontece, porém, a variação de frequência em torno de ω_{IF} são menores.

O filtro Gaussiano é descrito por:

$$g(t) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\left(\frac{t}{\sigma}\right)^2} \quad (4)$$

,em que σ está relacionada à largura de banda B em $-3dB$:

$$\sigma = \frac{\sqrt{\ln 2}}{2\pi B} \quad (5)$$

Nota-se que a resposta ao impulso do filtro Gaussiano vai de $-\infty$ a $+\infty$. Por razões práticas, esse *span* tem de ser limitado quando implementado.

O método mais comum utilizado na recepção do sinal consiste em deslocar o sinal para a banda base - isto é, reduzir a frequência da portadora para zero - filtrar o sinal e, então, analisar as componentes em fase e em quadratura do sinal.

O sinal descrito na Equação (2) pode ser reescrito como:

$$s(t) = A \cos((\omega_{IF} + \omega_d)t) \quad (6)$$

,em que ω_d é o deslocamento instantâneo de frequência devido à modulação (já que $\omega_d = \frac{d}{dt}\phi(t)$). Assim, se for possível conhecer ω_d por um período de símbolo, é capaz de saber o valor do bit pelo sinal de ω_d : $\omega_d > 0$ significaria que "1" foi recebido, enquanto $\omega_d < 0$ significa que "0" foi recebido.

Para eliminar ω_{IF} , pode-se multiplicar $s(t)$ por um seno e cosseno não modulados.

$$i_m(t) = s(t) \cdot \cos(\omega_{IF}t) \quad (7)$$

$$q_m(t) = s(t) \cdot -\sin(\omega_{IF}t) \quad (8)$$

Esse processo é conhecido como mixagem, e os sinais $i_m(t)$ e $q_m(t)$ são chamados de componentes "em fase" e "em quadratura" do novo sinal.

Com a aplicação de identidades trigonométricas, tem-se:

$$i_m(t) = \frac{A}{2}(\cos((2\omega_{IF} + \omega_d)t) + \cos(\omega_d t)) \quad (9)$$

$$q_m(t) = \frac{A}{2}(-\sin((2\omega_{IF} + \omega_d)t) + \sin(\omega_d t)) \quad (10)$$

As Equações (9) e (10) indicam que os sinais conterão o sinal original em duplicidade: uma vez ao redor da frequência central $2\omega_{IF}$ e uma vez ao redor da frequência 0. O componente em $2\omega_{IF}$ pode ser filtrado com um filtro passa-baixas de forma a se obter:

$$i_l(t) = \frac{A}{2} \cos(\omega_d t) \quad (11)$$

$$q_l(t) = \frac{A}{2} \sin(\omega_d t) \quad (12)$$

A operação de atraso e multiplicação (*delay and multiply*, em inglês) é uma técnica clássica de demodulação FM (Rappaport, 2009) que consiste em realizar a seguinte operação descrita na Equação (13):

$$d(t) = q_l(t) \cdot i_l(t - \Delta T) - i_l(t) \cdot q_l(t - \Delta T) \quad (13)$$

Aplicando-se identidades trigonométricas, obtém-se:

$$d(t) = \frac{A^2}{4} \sin(\omega_d \Delta T) \quad (14)$$

Se $\Delta T = T_s$ (que pode ser escolhido arbitrariamente) e com $h = 0,5$, o valor máximo de $d(t)$ será $\frac{A^2}{4} \sin \frac{\pi}{2} = \frac{A^2}{4}$. De forma similar, o menor valor de $d(t)$ será $-\frac{A^2}{4}$. A paridade de sinais se mantém após as operações matemáticas, de forma que $d(t) > 0$ para $\omega_d > 0$ e $d(t) < 0$ para $\omega_d < 0$. Isso significa que a sequência de mixagem, filtragem passa-baixas e *delay and multiply* resultam num sinal em que a sequência original de bits pode ser extraída facilmente por amostragem.

2.4 Casos de Uso da Rede Sigfox

A rede Sigfox já vem tendo seu uso comprovado em diversos setores da economia e por empresas de grande porte em termos de faturamento. Isto serve como um indicador da confiabilidade da rede e das soluções sendo implementadas (Sigfox, 2017).

No setor de energia, aplicações Sigfox são utilizadas para adquirir dados ao longo de linhas de transmissão e sistemas de distribuição de forma a prover os clientes com visibilidade e dados em tempo real. Algumas das aplicações da rede são em prevenção de falhas, monitoramento de níveis de potência, monitoramento de estoques e monitoramento do nível de água em reservatórios. Alguns dos clientes que utilizam a ferramentas são grandes empresas do setor de energia, como a Engie e a Daikin (Sigfox, 2017).

No setor de agricultura, a Bayer utiliza aplicações Sigfox para coleta de dados ao longo de grandes talhões de plantações de forma a consolidá-los num painel de controle e aplicar algoritmos de forma a prever o tempo de forma hiperlocalizada para otimizar a aplicação de fertilizantes. A empresa também utiliza a rede para monitorar os produtos ao longo da vasta cadeia de suprimentos, até o consumidor final (Sigfox, 2017).

No setor de alimentos, a Nestlé utiliza aplicações Sigfox para automatizar protocolos de segurança alimentar e monitorar o nível de demanda e estoques nos distribuidores e redes varejistas.

Isso contribui para a companhia ter maior visibilidade do planejamento produtivo (Sigfox, 2017).

A diversidade de setores e o uso por diversas grandes e reconhecidas empresas também mostra os altos padrões de segurança e conformidade com protocolos que a rede possui, visto que essas empresas possuem políticas rígidas quanto a Tecnologia da Informação.

3 Metodologia Experimental

O presente capítulo visa apresentar os equipamentos e sistemas utilizados para os testes propostos com o sistema Sigfox.

3.1 Kit de Desenvolvimento EDK da uDEV

A forma mais indicada de se trabalhar e desenvolver aplicações Sigfox é por meio de kits de desenvolvimento certificados pela Sigfox.

O kit escolhido para o presente trabalho é o *Embedded Development Kit* (EDK) da empresa brasileira uDev, parceira certificada da Sigfox.

O processo de desenvolvimento de aplicações, portanto, se torna muito mais ágil, visto que os parceiros Sigfox tratam de desenvolver o *hardware* básico necessário para acesso à rede, como os circuitos de Radio Frequência (RF), antenas, fonte de alimentação, interface de depuração, sensores básicos, ambiente de desenvolvimento, bibliotecas de baixo nível, entre outros.

Algumas das funcionalidades do EDK são:

- Processador ARM Cortex-M0+ ATSAMD21G18A-M, compatível com depuração via Atmel Studio
- Sensor de umidade e temperatura Si7006-A20 e DS18B20
- Sensor de luz ambiente Si1133
- Conectores para sensores analógicos
- *RTC* (Real Time Clock) integrado
- Modem Sigfox integrado
- Antena integrada
- Conector Micro USB
- Conector de bateria externa, com sistema integrado de gerenciamento de bateria

A Figura 16 mostra a vista superior do kit utilizado.



Figura 16: Placa do Kit EDK

Fonte: uDev Soluções Tecnológicas Ltda. (2018)

A Figura 17 mostra o posicionamento dos principais dispositivos na placa do kit EDK.

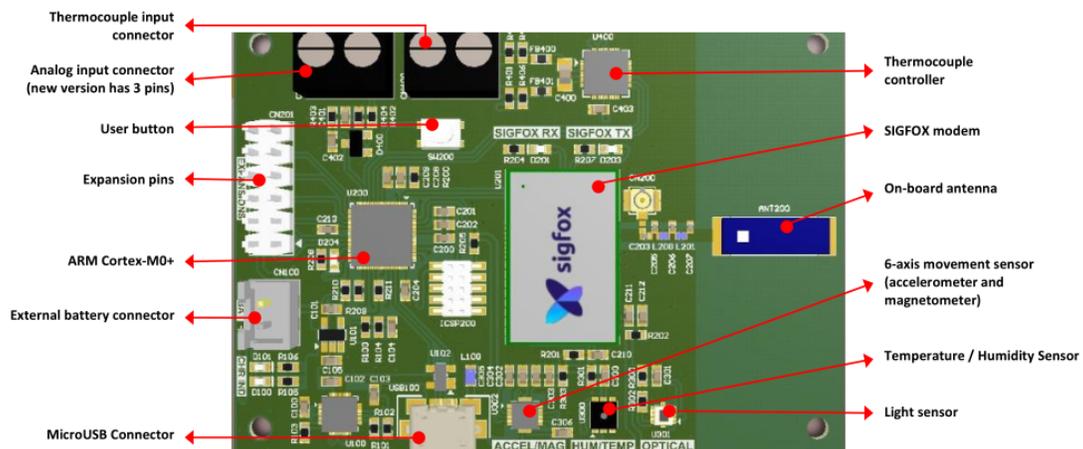


Figura 17: Posição dos Dispositivos Presentes no Kit EDK

Fonte: uDev Soluções Tecnológicas Ltda. (2018)

Além das características do kit serem satisfatórias para a proposta do presente projeto, a preferência a esse kit também se dá pelo fato de que a empresa desenvolvedora é brasileira, o que facilita bastante questões de suporte, resoluções de possíveis problemas e logística. Outro ponto importante é que o kit já está preparado para o padrão da rede Sigfox utilizado no Brasil, bem como o kit é certificado pela ANATEL.

Nota-se também, que um outro ponto importante para desenvolvimento de aplicações em

sistemas depurados é a interface de depuração e de carregamento dos códigos. O kit EDKse destaca por possuir suporte ao ambiente Atmel Studio, que é de extensa documentação e apresenta suporte robusto ao sistema operacional Windows. O kit também suporta depuração e carregamento de códigos pelo depurador oficial da Atmel - o Atmel ICE - ou pela mais comumente encontrada interface Micro USB.

3.1.1 *Microprocessador ARM Cortex M0+*

O microprocessador utilizado no kit de desenvolvimento é da família ARM Cortex-M, que são microprocessadores de 32-bit RISC com arquitetura ARM. Esses microprocessadores são otimizados para arquiteturas de baixo custo e com alta necessidade de eficiência energética. Tais dispositivos ganharam tanta popularidade que são embarcados em mais de dezenas de bilhões de dispositivos atualmente, desde sistemas embarcados simples até *smartphones* e aplicações de alta tecnologia (ARM Holdings, 2020).

Um dos motivos que tal família de microprocessadores é tão utilizada é devido ao licenciamento; a ARM Holdings não produz nem vende microprocessadores, mas provê o licenciamento de sua arquitetura para partes interessadas, cobrando *royalties*. Esse sistema provê agilidade no desenvolvimento e proteção de propriedade intelectual e foi uma das grandes razões que contribuiu com a popularidade da arquitetura (ARM Holdings, 2020).

As diversas famílias ARM possuem diferenças principalmente quanto ao conjunto de comandos *assembly* nativos e suas aplicações recomendadas, como pode ser visto na Figura 18.

Cortex-M Instruction Set

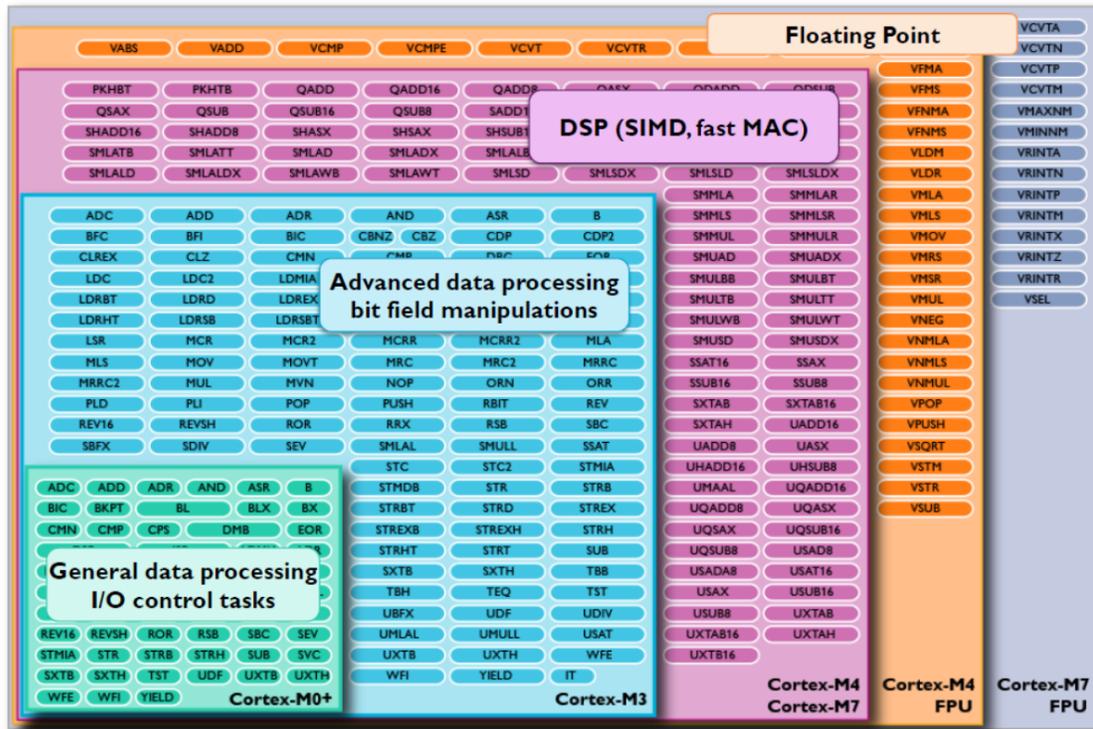


Figura 18: Conjunto de Instruções Assembly para as Diferentes Famílias ARM e Aplicações Suportadas

Fonte: ARM Holdings (2020)

A família utilizada no kit de desenvolvimento escolhido é a Cortex-M0+, uma versão otimizada do Cortex-M0. O Cortex-M0+ possui retrocompatibilidade total com o Cortex-M0, podendo utilizar as mesmas ferramentas de compilação e depuração. O *pipeline* do Cortex-M0+ foi reduzido de 3 para 2 estágios, o que contribui significativamente para a eficiência energética, sendo uma das poucas famílias ARM com essa característica, o que faz com que seja utilizado até hoje (Holdings, 2020).

O microprocessador da família ARM Cortex M0+ utilizado foi o ATSAMD21G18A-M, produzido pela Microchip com foco em automação residencial e aplicações finais de medições (Microchip, 2020).

Algumas de suas principais características são:

- 256KB de memória flash e 32KB de SRAM
- Até 48 MHz de frequência de operação

- Seis módulos de comunicação serial configuráveis como UART/USART, SPI ou I2C
- Cinco *Timers/Counters* de 16-bit e *Real-Time Clock* de 32-bit
- 20 Canais PWM
- ADC de 12-bits com 14 canais de acesso
- DAC de 10-bits
- Alimentação entre 1,62V e 3,63V
- Suporte ao ambiente de desenvolvimento Atmel Studio e MPLAB X IDE

3.2 Módulo Transceptor para Comunicação com a Rede

O módulo utilizado para conexão do kit escolhido com a rede Sigfox é o Wisol/WSSFM11R2DAT. Este é um módulo coreano fabricado pela empresa Wisol para utilização nas zonas 2 e 4 (RC2 e RC4) da cobertura Sigfox.

A Figura 19 ilustra as regiões de cobertura Sigfox e a respectiva característica de frequência e potência dos sinais.

Radio zones

- RZ1: Europe, MEA
868 MHz, +14 dBm
- RZ2: North America, Brazil
902 MHz, +22 dBm
- RZ3: Japan, Korea
920 MHz, +14 dBm, LBT
- RZ4: South America
920 MHz, +22 dBm



Figura 19: Regiões de Cobertura Sigfox e Suas Respectivas Frequências e Potência dos Sinais Empregados

Fonte: (Sigfox, 2018)

As vistas superior e inferior do módulo utilizado podem ser vistas na Figura 20. O módulo

possui 13mm de comprimento por 20mm de largura e 2,21mm de altura.

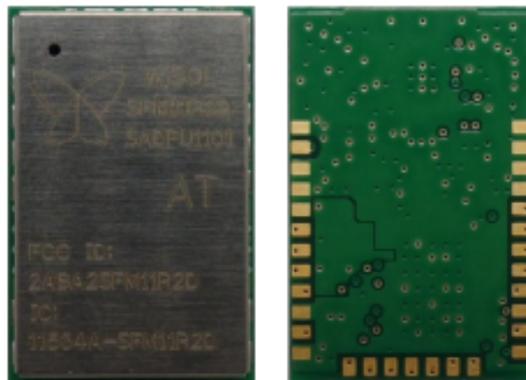


Figura 20: Vistas Superior e Inferior do Módulo Transceptor Wisol

Fonte: (Wisol, 2017)

As especificações elétricas e RF do módulo estão descritas na Figura 21. Nota-se que as frequências de operação para ambas as regiões, e tanto para transmissão ("Tx") quanto para recepção ("Rx") estão em linha com o que é definido pela rede Sigfox conforme a Figura 19.

Conditions: VCC=3.3V, Temp=25°C

Parameter		Min	Typ.	Max	Unit
RF Characteristics					
RF Frequency	RC2	Tx		902.2	MHz
		Rx		905.2	MHz
	RC4	Tx		920.8	MHz
		Rx		922.3	MHz
Tx output power(at "24" setting)		-	22.5	-	dBm
Frequency Error Tolerance(+25°C)		-2.5	-	+2.5	ppm
2 nd Harmonics(conducted)		-	-44	-	dBm
3 rd Harmonics(conducted)		-	-47	-	dBm
Rx Sensitivity(@600bps, GFSK)		-	-129	-	dBm
Rx Spurious Emission(30MHz~12.75GHz)				-54	dBm

Figura 21: Características Elétricas e de RF do Módulo Transceptor

Fonte: (Wisol, 2017)

O módulo conta com um total de 31 pinos de conexão, sendo 10 pinos GPIO com funcionalidades diversas (SPI, ADC, $\Sigma - \Delta$ DAC, além das funções de entrada e saída), 2 pinos de comunicação UART, um pino de entrada e saída RF, além de outros pinos para LEDs de indicação, alimentação e resets.

O diagrama de blocos do módulo é ilustrado na Figura 22

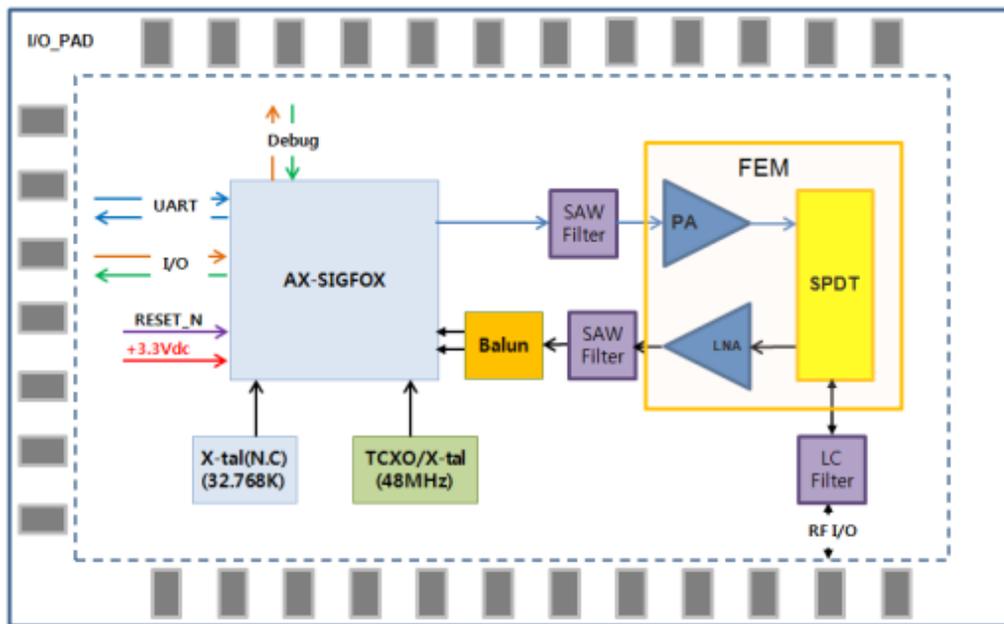


Figura 22: Diagrama de Blocos do Módulo Transceptor

Fonte: (Wisol, 2017)

Nota-se pela Figura 22 que, apesar de se tratar de uma rede de comunicação moderna e atual, o *front-end* de RF não destoa dos projetos usuais, sendo composto por:

- Filtro SAW (*Surface Acoustic Wave*): Filtro eletromecânico que utiliza materiais piezoelétricos e com amplo uso em aplicações de RF por sua eficiência;
- *Power Amplifier*: Amplificador de potência comumente utilizado em circuitos de RF para condicionar os sinais antes da transmissão;
- Chave RF SPDT (*Single-Pole Double-Throw*): Chave com uma entrada conectada a dois caminhos selecionáveis específica para aplicações RF; utilizada para comutar entre um sinal transmitido ou enviado;
- *Low-Noise Amplifier*: Amplificador de baixo-ruído para condicionar o sinal recebido;
- Balun (*balanced to unbalanced*): Utilizado para balanceamento/acoplamento do sinal considerando o casamento de impedâncias do restante do circuito.

O modo de operação do transceptor é implementado via uma máquina de estados finita. O fabricante não fornece muitos detalhes sobre os muitos modos de operação, mas fornece um exemplo de operação da máquina para checagem do estado atual, que pode ser visto na Figura 23.

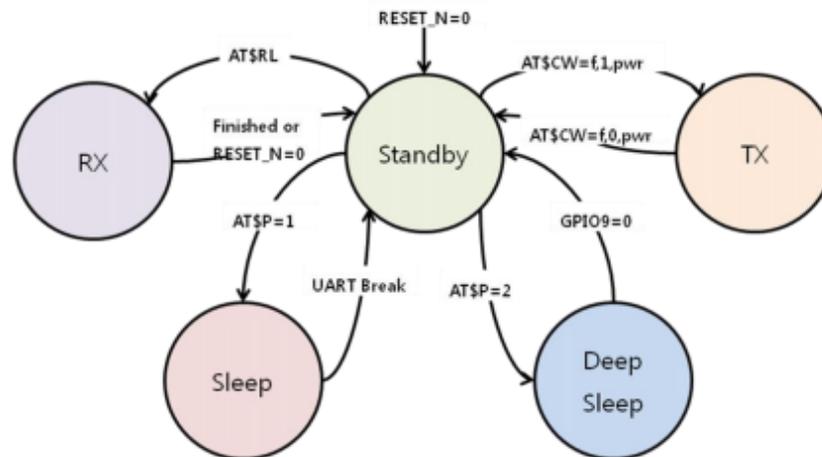


Figura 23: Implementação por Máquina de Estados Finita do Procedimento de Checagem do Estado Atual do Módulo Transceptor

Fonte: (Wisol, 2017)

3.3 Acesso à Rede Sigfox

O acesso à rede Sigfox se dá por meio de operadores locais licenciados da rede. No Brasil, a rede é operada e provida pela empresa WND Networks, que também é responsável por operar a rede no Reino Unido.

Cada módulo Sigfox compõe um nó da rede e é identificado por uma ID de 32 bits que está associada a um *Porting Authorization Code* (PAC) de 8 bytes, que serve como *checksum* necessário para autenticar e registrar o nó no backend da rede. Uma chave 128-AES única para cada nó é utilizada para encriptar as mensagens que é decriptada na estação rádio base.

A operadora local da rede cobra pelo acesso e o valor depende do número de mensagens por objetos (nós) ativados. No Brasil, a WND provê os seguintes planos, descritos para um nó da rede:

- Plano Ultra: até 140 mensagens de *uplink* por dia e 4 mensagens de *downlink* por dia
- Plano Plus: até 70 mensagens de *uplink* por dia e 2 mensagens de *downlink* por dia
- Plano Basic: até 2 mensagens de *uplink* por dia e 1 mensagem de *downlink* por semana

É importante para viabilidade e adoção do sistema que o custo da conectividade possa ser baixo. Hoje em dia a WND opera a rede a um custo que pode variar entre 1 dólar/mês/dispositivo até 1 dólar/ano/dispositivo. É esperado que este custo diminua com o passar do tempo (Zuñiga, 2017).

3.4 Aquisição e Processamento de Dados

3.4.1 Callbacks

Um dos elementos principais da arquitetura do *backend* da rede Sigfox são os *Callbacks*.

Basicamente, um *Callback* é um *request* http customizado contendo dados do dispositivo Sigfox enviados para um determinado servidor ou plataforma.

Os *callbacks* podem ser acionados por diferentes eventos - recebimento de uma mensagem, perda de conexão, quando um dispositivo é reiniciado, entre outros.

Não só os *callbacks* podem ser inteiramente configurados pelo usuário a partir de um servidor HTTP("Hypertext Transfer Protocol") padrão, como o próprio *backend* Sigfox já provê suporte para *callbacks* diretos para os seguintes serviços de *cloud*:

- AWS IoT;
- AWS Kinesis;
- Microsoft Azure Event Hub;
- Microsoft Azure IoT Hub.

Nota-se, que os serviços de *cloud computing* da Amazon e da Microsoft estão entre os mais utilizados atualmente (Barr, 2019). e já são nativamente suportados pela rede Sigfox.

Há três tipos de *callbacks* suportados pela rede Sigfox:

- *DATA*: opção em que se pode obter dados do dispositivo *sequence number*
- *SERVICE*: opção mais avançada em que é possível configurar o dispositivo para um modo de operação; possui encapsulada as funções da opção "Data";
- *ERROR*: opção para o caso em que há perda de comunicação.

Cada tipo de *callback* bem como as informações obtidas em cada um serão exploradas na seção 4.1.

O *callback* pode ser enviado por dois meios:

- HTTP: é possível enviar os dados direto para um servidor HTTP tanto via método GET ou POST; fácil interface com outras plataformas;
- E-mail: é possível enviar os dados diretamente para um endereço de e-mail ou protocolo SMTP; fácil uso, mas pouca interface automática com outras plataformas.

A rede também possui uma série de eventos padrões que podem ser usados adicionalmente para

acionar um *callback*:

- Mensagem fora de sequência;
- Recebimento da primeira mensagem de teste;
- Recebimento da primeira mensagem de *downlink*;
- Alteração do contrato com o provedor
- Pausa/retomada da aplicação;
- Chave de autenticação inválida;
- Recebimento de mensagem repetida;
- Sobre-utilização da rede;
- Reinicialização do dispositivo.

Nota-se, que essas são opções de eventos já suportadas nativamente pelo *backend* da rede; de toda forma, é possível que o usuário, a nível da aplicação, acione *callbacks* de forma totalmente personalizada em qualquer evento desejado.

A Figura 24 traz um sumário dos tipos de *callbacks* suportados pela rede.

Callback type		Description
DATA	UPLINK	Send uplink messages to a customer platform.
	BIDIR	Send uplink messages to a customer platform and waits for a downlink message from the same platform.
SERVICE	STATUS	To retrieve device battery and temperature information provided by service messages (e.g. keep-alive messages).
	ACKNOWLEDGE	To retrieve the network acknowledgement, confirming the downlink message transmission.
	DATA_ADVANCED	To retrieve optional data, such as geolocation, as well as metadata information. The list of available variables is displayed on the backend upon creation. Some variables are computed over the different base stations which received the messages. This callback is therefore delayed by approximately 30s.
ERROR		In case of communication failure, it informs on whether it is a device issue (based on the keep-alive value defined in the device edition page) or a network issue
EVENT		To be alerted upon device event occurrence. This can be configured at the Group, Device Type or Device level.

Figura 24: Sumário dos Tipos de *Callbacks* Suportados pela Rede Sigfox

Fonte: Sigfox (2017)

3.4.2 Geolocalização

Uma das aplicações que a rede oferece e que será testada experimentalmente é o serviço Sigfox de geolocalização.

Além das tecnologias de geolocalização já presentes, como o GPS e o Wi-Fi (pontos de acesso como âncoras num sistema de localização baseado em trilateração), que podem estar presentes nas plataformas de desenvolvimento, a rede Sigfox possui um serviço próprio de geolocalização baseado unicamente na rede - também conhecido como Atlas.

O Sigfox Geolocation é um serviço de baixo consumo energético e baixo custo disponível a todos dispositivos equipados com módulos Sigfox, mesmo os que não possuem GPS. O serviço é ideal para uso em casos em que grande precisão métrica não é necessária, como em rastreamento de cargas em múltiplas localizações ou rastreamento de frotas (Sigfox, 2017).

O serviço provê coordenadas em tempo real através de um *callback* e é baseado num algoritmo proprietário de cálculo probabilístico da localização mais provável do dispositivo. Esta abordagem permitiu com que o primeiro *release* da tecnologia tivesse uma acurácia de 1 a 10km para 80% dos dispositivos - o que deve melhorar ainda mais nos próximos anos.

Outra proposta do serviço é servir como uma tecnologia de geolocalização complementar às existentes. Por exemplo, a tecnologia pode ter bom uso complementar ao GPS, já que pode rastrear dispositivos *indoor* e também pode ser complementar ao Wi-fi, já que possui cobertura em vastas áreas rurais. Logo, o serviço tem boa complementariedade às soluções existentes e se mostra como mais um diferencial competitivo às outras LPWANs (Sigfox, 2017).

A Figura 26 mostra visualmente um comparativo entre as principais tecnologias de geolocalização existentes, incluindo as RTLS (*Real Time Location Systems*, que englobam RFID e Bluetooth).

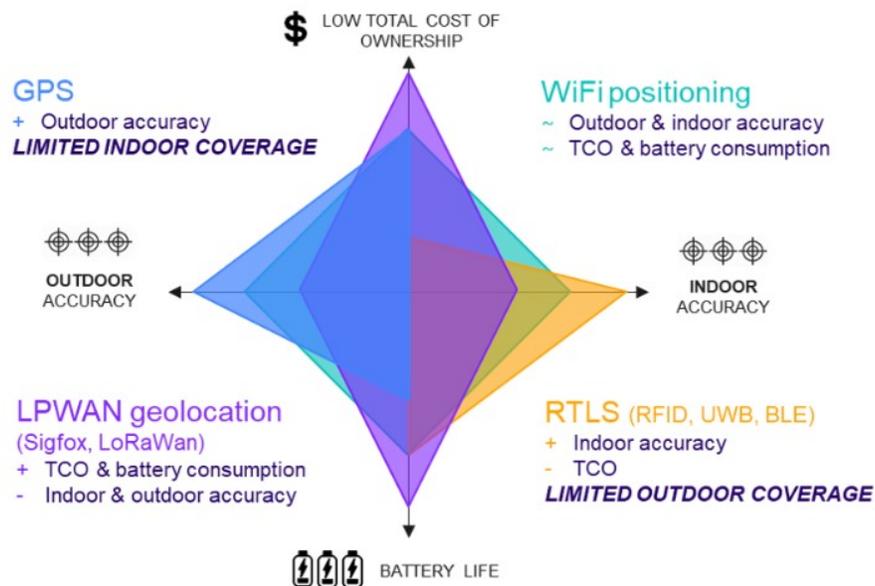


Figura 25: Comparativo entre as Principais Tecnologias de Geolocalização

Fonte: Sigfox (2017)

O serviço funciona baseado num cálculo probabilístico da localização mais provável do dispositivo graças ao *Received Signal Strength Indicator* (RSSI) obtido das mensagens enviadas e recebidas pela rede Sigfox e, portanto, não depende de métodos como *Time Difference Of Arrival* (TDOA) ou do Efeito Doppler do sinal. O funcionamento pode ser descrito por:

- O dispositivo envia uma mensagem qualquer, em que o RSSI é parte do metadado da mensagem;
- Diversas estações Sigfox recebem a mensagem - quanto mais estações a recebem, mais preciso o cálculo da localização será feito;
- Com os identificadores das estações Sigfox, a rede pode determinar as localizações prováveis e selecionar aquela com maior probabilidade.

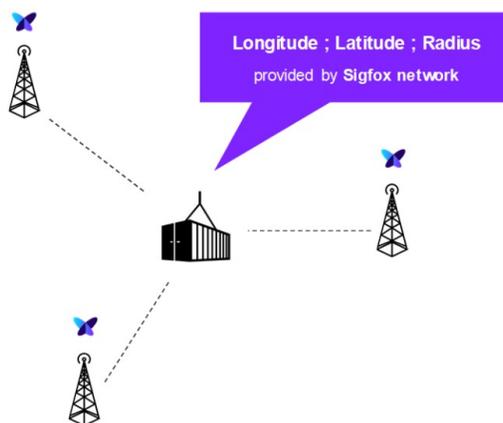


Figura 26: Ilustração da Determinação da Geolocalização de um Dispositivo Sigfox

Fonte: Sigfox (2017)

O serviço é muito eficiente por não requerer hardwares adicionais e pela eficiência energética, já que o *payload* da mensagem pode simplesmente estar vazio.

A acurácia do serviço será testada para verificar se para as regiões e kits testado ela está dentro dos parâmetros propostos pela rede. O serviço também será comparado com a geolocalização via GPS e Wi-fi para verificar sua complementaridade.

3.4.3 Processamento em Nuvem

Uma das principais propostas da rede Sigfox é a integração completa com serviços comerciais de processamento em nuvem existentes, como o Google Cloud e o Amazon AWS. Devido a fatores de preço e documentação existente, o sistema terá sua conectividade testada com a nuvem o Amazon AWS. Além disso, o AWS é o serviço de computação em nuvem mais utilizado, possuindo 34% deste mercado (Rama, 2017).

Computação em nuvem, ou *cloud computing* é a entrega de recursos de tecnologia da informação sob demanda por meio da Internet com definição de preço conforme a utilização. Em vez de comprar, ter e manter infraestrutura própria de *datacenters* e servidores físicas, é possível acessar serviços de tecnologia, como capacidade computacional, armazenamento e bancos de dados, conforme a necessidade, utilizando um provedor de nuvem (Amazon, 2020).

A Amazon Web Services (AWS) é a plataforma de *cloud computing* mais adotada e abrangente do mundo, oferecendo mais de 175 serviços completos de *datacenters* em todo o mundo. A AWS possui 70 zonas de disponibilidade em 22 regiões geográficas em todo o mundo.

O AWS é baseado em serviços - funcionalidades em que se deseja trabalhar com a plataforma - que podem variar desde terminais virtuais executados num servidor remoto até complexas infraestruturas sob demanda para aplicações críticas. A Figura 27 mostra um exemplo de alguns desses

serviços. Por sua vez, cada serviço possui diversas soluções disponíveis, desde as mais genéricas até soluções específicas.

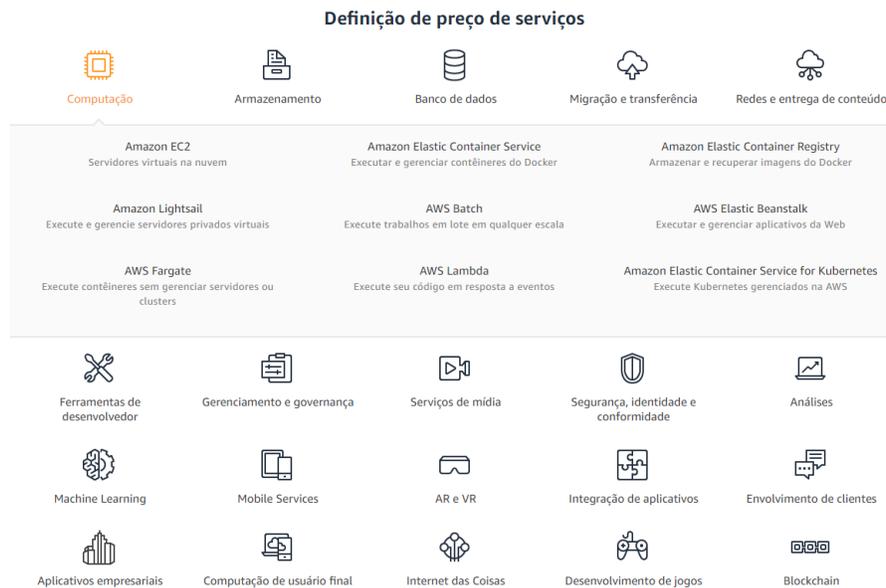


Figura 27: Demonstração de Alguns dos Serviços Disponibilizados do AWS e suas Soluções

Fonte: Amazon (2020)

A cobrança do uso é feita com base na utilização. Há três planos - um plano gratuito, um plano "Pro", com custo de US\$ 25 por mês e um plano empresarial, com custo de US\$ 100 por mês. As funcionalidades disponíveis em cada plano dependem do volume de dados e recursos utilizados. Porém, devida à ampla escala do serviço, o pacote gratuito já possui uma franquia considerável de uso de dados. Para um serviço de armazenamento simples o serviço é gratuito para até 25GB e 30.000 requisições GET num protocolo HTTP.

A integração dos dados do *backend* Sigfox num serviço de *cloud computing* é especialmente interessante para permitir ao usuário um número elevado de funcionalidades e opções de armazenamento, processamento e tomada de decisões com base nas medidas feitas com seu dispositivo. Além disso, há um ganho considerável no tempo de desenvolvimento visto que as plataformas de *cloud* já possuem documentação e ambientes próprios para desenvolvimento de diversas soluções.

A Figura 28 mostra como um dado obtido através da rede Sigfox se comunica com a infraestrutura AWS.

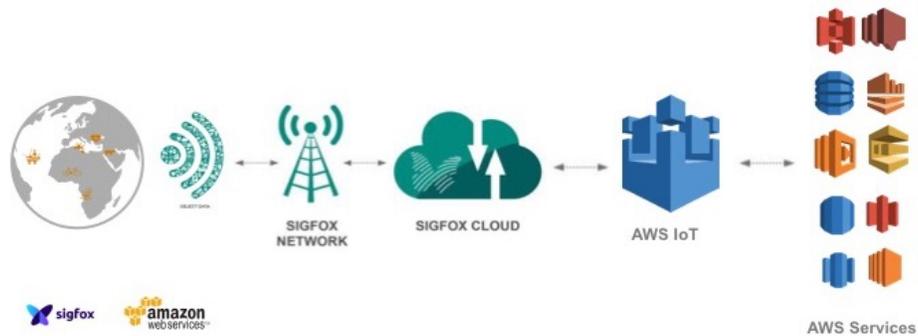


Figura 28: Diagrama da Integração do AWS com a Rede Sigfox

Fonte: Amazon (2020)

De forma a implementar esta arquitetura, serão seguidas e documentadas as seguintes etapas:

- Configurar os conectores de IoT do AWS com o *backend* Sigfox;
- Configurar os recursos AWS de forma com que o Sigfox possa enviar os dados através dos serviços AWS específicos para estas funções;
- Configurar uma estrutura de banco de dados para armazenar os dados recebidos.

4 Resultados e Discussões

Na presente seção são descritos os procedimentos e testes realizados de forma a avaliar as características, funcionalidades e operação da rede Sigfox no contexto de avaliação para seu uso em uma aplicação.

4.1 Autenticação e Acesso à Rede a Partir de um Dispositivo

Cada dispositivo autorizado a se conectar à rede Sigfox e que possua um modem, necessitará de dois parâmetros, únicos para cada dispositivo, para que possa ser estabelecida sua conexão com a rede - são eles o "ID" (*Device ID*, ou identificação do dispositivo) e o "PAC" (*Porting Authorization Code*), como ilustrado na figura 29



Figura 29: Parâmetros de um Dispositivo com Suporte à Sigfox para Conexão à Rede

Fonte: uDev Soluções Tecnológicas Ltda. (2018)

A autenticação e conexão à rede é feita numa interface *web* amigável e de fácil operação. Basta possuir um cadastro do site Sigfox e registrar o dispositivo na conta do usuário, como ilustrado na Figura 30.

Figura 30: Página de Registro do Dispositivo Sigfox

Fonte: uDev Soluções Tecnológicas Ltda. (2018)

A partir daí, o dispositivo estará autenticado e habilitado a acessar a rede Sigfox de acordo com o plano e tarifas de utilização escolhidas. A forma mais comum de gerenciamento do dispositivo dentro do ambiente Sigfox é por meio do *backend* provido pelo próprio consórcio na forma de uma página *web*. Por meio dele é possível verificar a localização de cada dispositivo e o histórico de cada uma das trocas de mensagens/*callbacks*. Essa arquitetura se mostra bastante robusta e, ao mesmo tempo, amigável ao usuário, já que é funcional e permite uma visão abrangente dos dispositivos e dados trocados independente da aplicação sendo executada no sistema embarcado e sem precisar acessar as interfaces físicas de comunicação dos dispositivos. A Figura 31 mostra a identificação do dispositivo logo após a conexão ser estabelecida.

Time	Data / Decoding	LQI	Callbacks	Location
2020-02-23 20:59:28	082a173e0ccd0000			

Figura 31: Identificação Inicial do Dispositivo no *backend* da rede Sigfox

Fonte: Autor

A Figura 32 mostra as informações identificadas pelo *backend* do dispositivo cadastrado. Nota-se que o enlace respeita o *link budget* proposto.

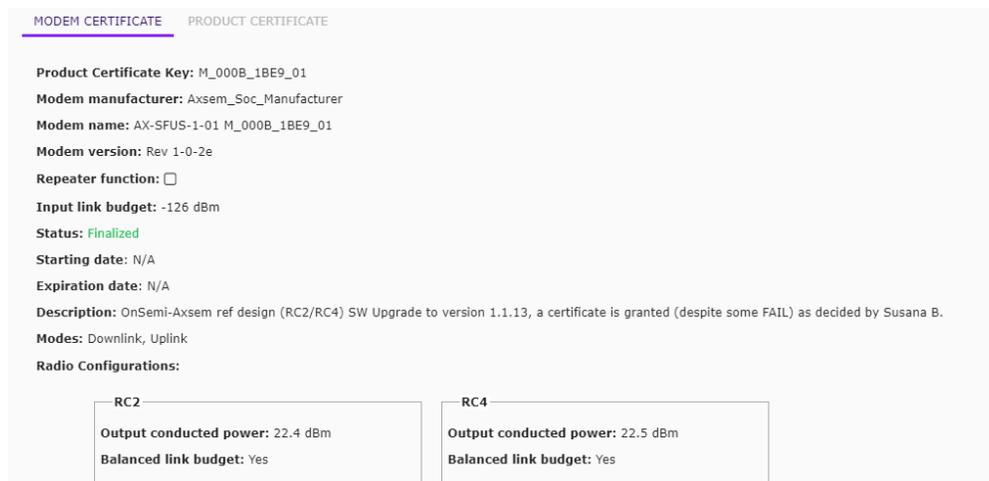


Figura 32: Identificações do Modem do Dispositivo Conectado à Rede Sigfox vistas pelo *backend*

Fonte: Autor

4.2 Aquisição de Dados Iniciais com *Callbacks* Tipo Data

A fim de auferir a boa conexão com a rede e as funcionalidades básicas do dispositivo, foi configurado um *callback* de dados para aquisição dos dados da placa.

Para isso, foi criado um *callback* customizado no *backend* Sigfox, conforme a Figura 33

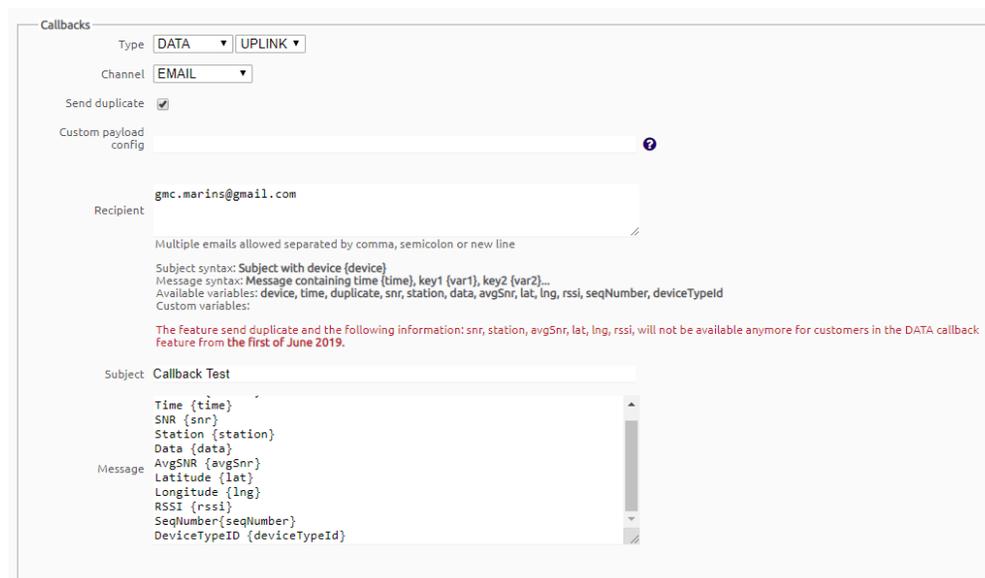


Figura 33: Criação do *Callback* Customizado do tipo DATA

Fonte: Autor

Tendo o *callback* sido configurado, ele será acionado toda vez que o dispositivo realizar uma requisição (*uplink*) ou toda vez que o dispositivo for acessado pela rede (*downlink*). Como o dispositivo envia uma mensagem toda vez que ele é ligado ("ping"), o *callback* pode ser acionado simplesmente ao reiniciar o dispositivo.

O resultado do *callback* pode ser visto na Figura 34

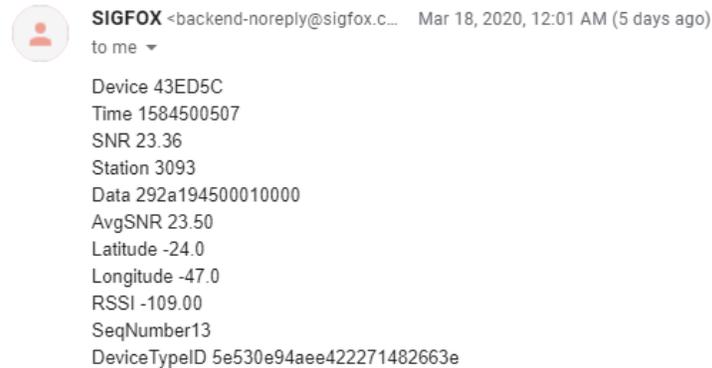


Figura 34: Resultado do Acionamento do *Callback*

Fonte: Autor

Objetivando testar a funcionalidade e operação desta funcionalidade, foram feitos 10 *callbacks*, espaçados de 1 minuto cada, na cidade de São Paulo. Os dados adquiridos podem ser vistos na Tabela 3

Tabela 3: Testes Iniciais de Aquisição de Dados por *Callbacks* feitos em São Paulo

Data	SNR (dB)	Estação	Latitude	Longitude	RSSI(dBm)
18-03-2020 12:01	23,36	3093	-24,0	-47,0	-109,00
18-03-2020 12:02	13,37	3033	-24,0	-47,0	-125,00
18-03-2020 12:03	11,53	307D	-24,0	-47,0	-117,00
18-03-2020 12:04	14,11	6C48	-24,0	-47,0	-112,00
18-03-2020 12:05	15,14	7A59	-24,0	-47,0	-114,00
18-03-2020 12:06	12,66	3033	-24,0	-47,0	-125,00
18-03-2020 12:07	12,53	8E3E	-24,0	-47,0	-122,00
18-03-2020 12:08	23,17	90D2	-24,0	-47,0	-111,00
18-03-2020 12:09	22,22	3093	-24,0	-47,0	-110,00
18-03-2020 12:10	22,84	90D2	-24,0	-47,0	-110,00

Na Tabela 3 foram removidos os dados de *Device* (dispositivo do usuário), *SeqNumber*, *DeviceTypeID*, *Time* e *Data*, já que eram os mesmos para todos os *callbacks*. Os dados de *Device* e

DeviceTypeID são os mesmos já que todos os *callbacks* foram feitos com o mesmo dispositivo. O dado de *SeqNumber* é relacionado ao dígito sequencial utilizado na comunicação para autenticação das mensagens e é reiniciado a cada sessão, mantendo-se inalterado durante a sessão ou até que o usuário reinicie-o manualmente. Já os dados de *Data* e *Time* são relativos ao *payload* requisitado ou enviado no *callback*; como este campo foi deixado em branco para os testes, os dados são espúrios.

Os dados dos campos *Latitude* e *Longitude* também foram adquiridos. Nota-se que o dado não possui muitas casas decimais, o que torna inviável seu uso para a identificação precisa da localidade do dispositivo, dada a natureza da medida. Porém, isso não impossibilita o uso do Sigfox para geolocalização. A identificação da posição do dispositivo a partir da rede Sigfox depende tanto do tipo de *callback*, do plano contratado e do *hardware* do dispositivo e o tipo de *callback* testado, de fato, não é o mais adequado para esta utilização específica.

O dado do campo *station* traz o identificador de uma das ERB Sigfox utilizadas para transmissão da mensagem. Como podem haver mais de uma estação envolvida na transmissão, apenas uma acaba sendo mostrada. Ainda não há documentação detalhada sobre as características deste campo, o que indica que pode ser uma funcionalidade a ser desenvolvida no futuro.

Como o tipo de *callback* requisitado foi do tipo DATA - o mais simples - a medida foi de fato simplória. No Google Maps é possível ver a qual localidade a medida de localização se refere, conforme a Figura 35

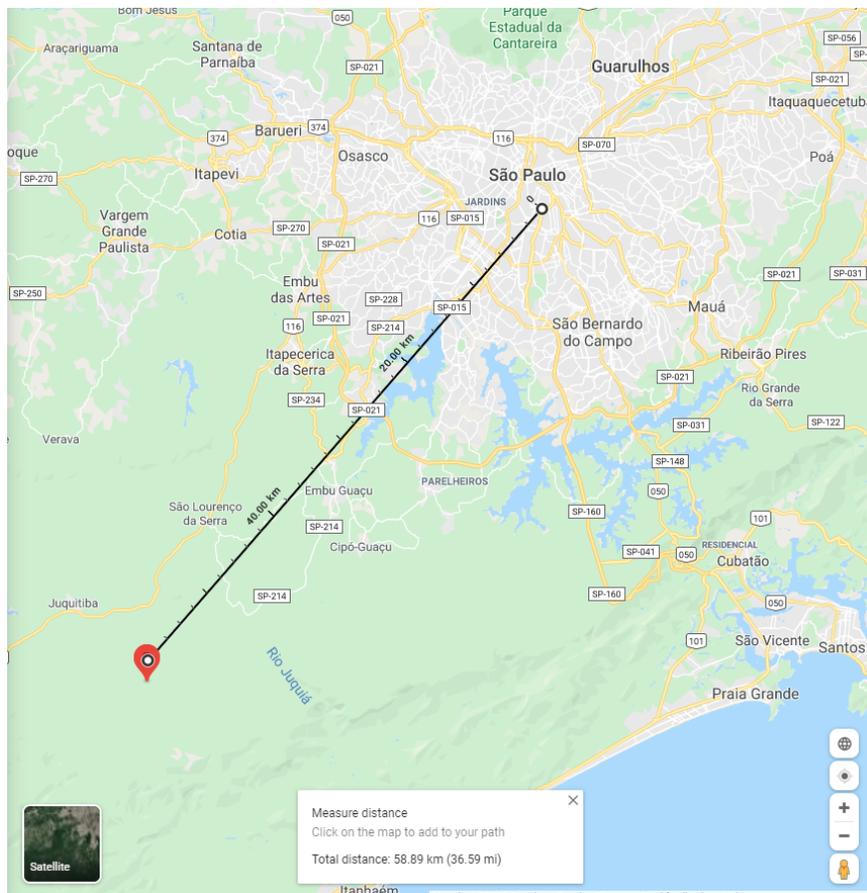


Figura 35: Erro da Medida de Latitude e Longitude no *Callback* Tipo DATA

Fonte: Autor

Nota-se um erro elevado, de quase 60km, o que não é adequado para uma medida precisa. Porém, nota-se que o tipo de *callback*, bem como a natureza da própria medida de geolocalização que depende de muitas casas decimais para que seja satisfatória, contribuíram negativamente para este resultado.

Ademais, nota-se nos resultados da Tabela 3 que a cada *callback* os dados eram enviados por uma estação rádio-base distinta, de forma que a região é bem coberta pela rede, não dependendo de apenas poucas estações. A relação sinal-ruído também se manteve acima dos 10dB para todas as mensagens e a RSSI também foi satisfatória, sendo classificadas como "Excelente" 5 das 10 mensagens e "Boa" as demais conforme os padrões do fabricante que podem ser vistos na Tabela 4 (considerando que há cobertura de 3 ou mais estações na região testada).

Tabela 4: Níveis de Qualidade do Enlace da Rede Sigfox para a Região Brasileira (RC2) com Base no RSSI do Sinal

RSSI	Número de Estações	Indicação de Qualidade do Enlace
-114dBm <RSSI	3	Excelente
-127dBm <RSSI ≤ 114dBm	3	Boa
-114dBm <RSSI	1 ou 2	Boa
-127dBm ≤ RSSI ≤ 114dBm	1 ou 2	Média
RSSI ≤ -127 dBm	qualquer	Limite

O fabricante, além disso, não indica o uso do SNR como indicador de qualidade do sinal, e informa para considerar o RSSI como um indicador adequado para isso (Sigfox, 2018).

Posteriormente, uma aquisição de dados similar, com um *callback* de mesmo tipo e com os mesmos dados, foi feita na cidade de Porto Alegre. Os dados adquiridos podem ser vistos na Tabela 5.

Tabela 5: Testes Iniciais de Aquisição de Dados por *Callbacks* feitos em Porto Alegre

Data	SNR (dB)	Estação	Latitude	Longitude	RSSI(dBm)
23-03-2020 23:03	19,17	7681	-30,0	-51,0	-121,00
23-03-2020 23:04	17,96	76A3	-30,0	-51,0	-114,00
23-03-2020 23:05	21,29	2FBB	-30,0	-51,0	-112,00
23-03-2020 23:06	12,08	6C48	-30,0	-51,0	-131,00
23-03-2020 23:07	11,41	7A59	-30,0	-51,0	-126,00
23-03-2020 23:08	23,51	3033	-30,0	-51,0	-114,00
23-03-2020 23:09	23,71	8E3E	-30,0	-51,0	-109,00
23-03-2020 23:10	12,88	90D2	-30,0	-51,0	-130,00
23-03-2020 23:11	23,07	3093	-30,0	-51,0	-111,00
23-03-2020 23:12	21,45	90D2	-30,0	-51,0	-113,00

Nota-se que, assim como São Paulo, Porto Alegre não é coberta por somente uma estação.

É possível, com os dados das Tabelas 3 e 5 avaliar se a infraestrutura de telecomunicações de uma determinada cidade é um fator significativo na qualidade do sinal, medida pelo RSSI, através de uma análise de variância considerando as 10 repetições de cada teste. O resultado da análise, feita no Microsoft Excel 2016, pode ser visto na Figura 36

Anova: Single Factor					
SUMMARY					
Groups	Count	Sum	Average	Variance	
São Paulo	10	-1155	-115.5	40.27778	
Porto Alegre	10	-1181	-118.1	67.65556	

ANOVA						
Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	33.8	1	33.8	0.626313	0.439016	4.413873
Within Groups	971.4	18	54.0			
Total	1005.2	19				

Figura 36: Análise da Significância da Cidade para a Qualidade do Sinal (RSSI) para um $\alpha = 0,05$

Fonte: Autor

Como pode ser visto na Figura 36, $F < F_{Crit}$, logo, não é possível afirmar que a infraestrutura de telecomunicações de uma cidade é um fator significativo para a qualidade do sinal - o que indica, para as duas cidades analisadas, a isonomia na qualidade da cobertura da rede Sigfox.

4.3 Testes de Geolocalização

Foi realizada uma série de aquisições de dados de geolocalização, utilizando *callbacks* tipo *SERVICE* com dois kits distintos e em duas cidades de forma a avaliar a precisão e a aplicabilidade de um dispositivo Sigfox para este uso.

4.3.1 Comparação da Geolocalização com Dois Dispositivos Distintos

Neste primeiro teste foram utilizados dois kits de desenvolvimento e configurado um *callback* para obtenção da latitude, longitude e raio utilizando apenas o serviço próprio Sigfox de geolocalização.

O teste foi feito com dois dispositivos distintos no mesmo recinto e cada aquisição de dados foi feita simultaneamente com os dois kits, de forma a avaliar o dado obtido. O recinto utilizado (*indoor*) foi próximo a uma janela num apartamento residencial de dezoito andares.

Os dados obtidos foram comparados com a localização de referência obtida pelo Google Maps para cálculo do erro. O erro é calculado utilizando a Fórmula de Haversine para a menor distância de dois pontos (latitude, longitude) na superfície de uma esfera. O procedimento é descrito pela Equação (15), considerando o raio médio da terra de 6371 km e as diferenças angulares em radianos (Van Brummelen, 2012).

$$\begin{aligned}d &= 2r \arcsin \left(\sqrt{\text{hav}(\varphi_2 - \varphi_1) + \cos(\varphi_1) \cos(\varphi_2) \text{hav}(\lambda_2 - \lambda_1)} \right) \\ &= 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right)\end{aligned}\quad (15)$$

, em que:

- φ_1, φ_2 são a latitude do ponto 1 e do ponto 2, em radianos, respectivamente
- λ_1, λ_2 são a longitude do ponto 1 e do ponto 2, em radianos, respectivamente
- $\text{hav}(\theta)$ é a Função Haversine para um ângulo central θ entre a diferença de dois pontos numa esfera

Os dados obtidos são mostrados na Tabela 6. Os erros foram calculados considerando a referência da localização de latitude $-30,0458809^\circ$ e longitude $-51,2187468^\circ$

Tabela 6: Medidas de Geolocalização com o Uso da Rede Sigfox Utilizando Dois Kits de Desenvolvimento Distintos

Kit	Latitude (°)	Longitude (°)	Raio de tolerância (km)	RSSI (dBm)	Erro (km)
Kit 1	-30,0348657	-51,2144737	10,229	-118,00	1,292
	-30,0364333	-51,2096484	11,517	-118,00	1,368
	-30,0377178	-51,1911913	9,786	-116,00	2,803
	-30,0373810	-51,2036971	12,219	-118,00	1,730
	-30,0396132	-51,1919959	9,957	-116,00	2,668
	-30,0354301	-51,2135960	10,175	-117,00	1,263
	-30,0365813	-51,2041347	12,564	-118,00	1,746
	-30,0379603	-51,2009294	11,997	-116,00	1,928
	-30,0358714	-51,2050465	12,510	-116,00	1,726
	-30,0362135	-51,2066144	11,508	-118,00	1,587
	-30,0345937	-51,2135195	11,229	-122,00	1,352
	-30,0361470	-51,2025411	12,283	-119,00	1,899
	-30,0370929	-51,1937782	9,984	-120,00	2,594
	-30,0374443	-51,1979966	12,587	-118,00	2,207
	-30,0364325	-51,1999475	12,280	-118,00	2,092
	Kit 2	-30,0351987	-51,2140875	10,162	-107,00
-30,0348616		-51,2028098	12,201	-112,00	1,963
-30,0344159		-51,2114657	11,201	-123,00	1,455
-30,0360814		-51,1916132	10,055	-122,00	2,830
-30,0361117		-51,1976331	12,513	-120,00	2,304
-30,0358882		-51,1938153	10,087	-122,00	2,645
-30,0376120		-51,2052331	10,302	-121,00	1,593
-30,0347492		-51,2142980	11,159	-121,00	1,310
-30,0381714		-51,2040372	9,997	-118,00	1,655
-30,0358737		-51,2002946	12,716	-121,00	2,096
-30,0374564		-51,2068127	10,194	-120,00	1,482
-30,0366450		-51,1950822	10,363	-122,00	2,499
-30,0365883	-51,1847501	10,110	-123,00	3,432	
-30,0352490	-51,2092294	12,318	-122,00	1,496	
-30,0377447	-51,1838897	10,309	-121,00	3,475	

Várias observações podem ser feitas a partir dos dados medidos.

Primeiramente, nota-se o conservadorismo na medida do raio de tolerância calculada pela rede, isto é, o raio, medido a partir do ponto de latitude e longitude medidos, em que a rede garante, com 99% de confiabilidade, que o dispositivo está localizado. Em todas as medidas, o raio de tolerância se manteve próximo de 10km (média de 11,388 km para o Kit 1 e 10,912 para o Kit 2) - que é o raio de tolerância especificado pelo fabricante. O erro calculado, porém, ficou bem abaixo desse

raio de tolerância, sendo o erro médio do Kit 1 de 1,883 km e do Kit 2 de 2,100 km.

Realizando-se uma análise de variância, conclui-se, como esperado, que não há diferença significativa para o erro entre os Kits 1 e 2 considerando-se um $\alpha = 0,05$. A análise pode ser vista na Figura 37. Considerando a inovação do sistema de localização, é importante assegurar a repetibilidade deste tipo de medida, o que pode ser auferido com este teste.

SUMMARY						
Groups	Count	Sum	Average	Variance		
Kit 1	15	28.255	1.883667	0.253885		
Kit 2	15	31.505	2.100333	0.546087		
ANOVA						
Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	0.352083	1	0.352083	0.88024	0.356159	4.195972
Within Groups	11.1996	28	0.399986			
Total	11.55168	29				

Figura 37: Análise da Significância do Kit Utilizado para o Erro de Geolocalização Medido Considerando um $\alpha = 0,05$

Fonte: Autor

Sobre as medidas em si, nota-se que o erro medido, para ambos os kits, é da ordem de quilômetros. Conclui-se, portanto, que o sistema de geolocalização próprio da rede Sigfox, não pode ser utilizado para aplicações que necessitem de precisão da ordem de metros.

Considerando a aplicação de uma LPWAN, porém, o resultado é satisfatório já que permite ao usuário um dado de geolocalização útil para aplicações de longas distâncias usuais, ou como uma validação para uma medida feita utilizando outra tecnologia. Sabe-se que essa precisão melhorará com o tempo, principalmente devido ao planejamento de se aumentar a cobertura da rede nos próximos anos (Sigfox, 2017).

Reitera-se ainda a complementariedade da tecnologia de geolocalização Sigfox com o GPS e Wi-Fi, que pode ser possível combinar a precisão dessas com a cobertura de longas distâncias da rede Sigfox.

4.4 Integração com *Cloud* para Armazenamento e Processamento dos Dados Adquiridos

Mesmo que uma solução utilizando um kit de desenvolvimento Sigfox e as funcionalidades básicas do seu *backend* já seja satisfatória para muitos casos, numa aplicação prática de IoT é necessário uma infraestrutura mais robusta para aquisição, armazenamento e manipulação dos

dados.

O método básico para aquisição dos dados na rede Sigfox via *callbacks* - armazenando-os num servidor HTTP ou via E-mail (SMTP) - pode não ser o ideal em muitos casos visto que, atualmente, as soluções profissionais de IoT costumam ser baesadas em nuvem.

Foi explorada a integração da rede Sigfox com o serviço AWS de forma a expandir consideravelmente o potencial de uma aplicação a ser desenvolvida com Sigfox.

4.4.1 Configuração dos Conectores do Sigfox para AWS

O primeiro passo para a integração entre sistemas utilizando *cloud computing* é o provisionamento da infraestrutura necessária para tal conexão.

O serviço AWS responsável por tal funcionalidade é o AWS CloudFormation, que permite a criação e gerenciamento de recursos de infraestrutura *cloud*. A modelagem da infraestrutura e recursos a serem utilizados pela aplicação podem ser feitos em infraestrutura web, arquivo de texto ou linguagens de programação. Ele também provisiona os recursos de forma segura e encarrega de determinar qual são as operações mais adequadas e sua ordem de prioridade no caso de uma integração com outro sistema.

O AWS CloudFormation, na aplicação proposta, atuará como a ligação entre o sistema Sigfox e o *stack* AWS. O *stack* é a coleção de recursos AWS cujo acesso é gerenciado pelo CloudFormation.

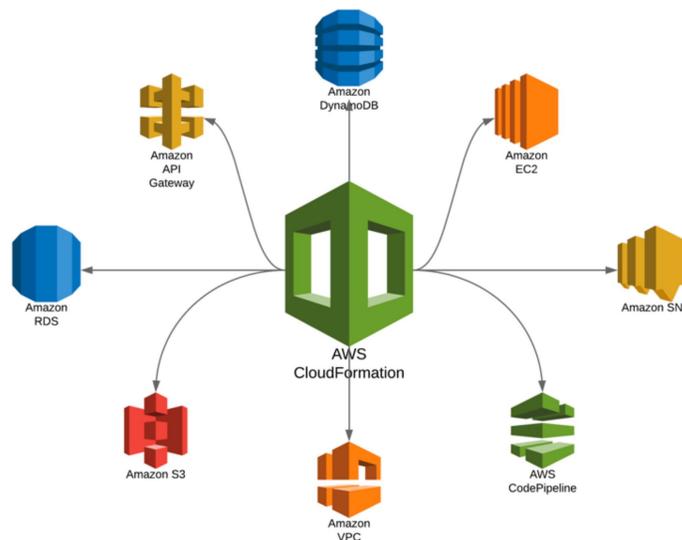


Figura 38: Ilustração da Criação de Um *Stack* pelo AWS CloudFormation

Fonte: AWS

A Figura 38 ilustra o papel do CloudFormation na aplicação proposta - o de criar, portanto, um *stack* AWS gerenciando a infraestrutura necessária para receber os dados de um sistema externo -

realizando assim a integração em nuvem com o Sigfox.

O primeiro passo para criar um *stack* no AWS CloudFormation é definir um modelo ("*Template*") para integração com um sistema externo. Isso é deixado a cargo do usuário, que pode definir quais campos, região de conexão e chaves de autenticação podem ser usadas para a integração. O padrão da rede Sigfox utiliza dois atributos para realizar a integração com o AWS, sendo eles:

- *External ID*
- *Amazon Resource Name (ARN) Role ID*

O External ID é uma chave hexadecimal única de 12 bytes gerada pelo *backend* Sigfox que identifica o dispositivo para uma conexão com um sistema de nuvem.

O ARN Role ID é uma chave única gerada pelo AWS CloudFormation que identifica o *stack* AWS a ser utilizado. Ela possui os atributos

```
arn:partição:serviço:região:identificação_aws:identificação_recurso
```

Os atributos do formato são:

- *partition*: identifica a partição que o recurso está localizado. Para regiões padrão do AWS a partição é "aws";
- *service*: identifica o serviço de identificação e acesso AWS a ser utilizado - neste caso o *Identity and Access Management (IAM)*;
- *region*: região em que o recurso está localizado - pode ser deixado em branco para o caso atual que se utiliza o IAM;
- *account*: é o número de identificação da conta AWS que irá gerenciar os recursos - cada conta possui um identificador numérico único;
- *resource*: é o nome do recurso criado no AWS CloudFormation;

Pelo fato deste método de integração utilizar dois atributos - um originado no *backend* Sigfox e outro originado no AWS CloudFormation - este é um método conhecido como *Cross-Account*, ou "Contas Cruzadas".

Dados os dois atributos que serão utilizados para comunicação, é necessário construir o *template* para configuração do *stack*. Este modelo consistirá basicamente na configuração das entradas e serviços ou funções a serem executadas pelo CloudFormation na criação do *stack*. Para a aplicação proposta, este modelo consistirá em:

1. Receber do usuário o External ID do dispositivo Sigfox;
2. Definir a região AWS a ser utilizada e o nome que irá identificar a pilha;
3. Receber o número de identificação da conta AWS;
4. Definir o recurso utilizado para integração (neste caso o IAM);

5. Definir a política de integração (neste caso uma política de "Contas Cruzadas");
6. Gerar a partir das entradas o ARN ID a ser empregado para comunicação com o *stack*;

A configuração do modelo é feita num formato *JavaScript Object Notation* (JSON) e o modelo criado pode ser visto no Apêndice B.

As Figuras 39 - 44 mostram o processo de configuração do *callback* e do *stack*.

Primeiramente, no *backend* Sigfox configura-se um novo *callback* - do tipo AWS IoT, conforme a Figura 39

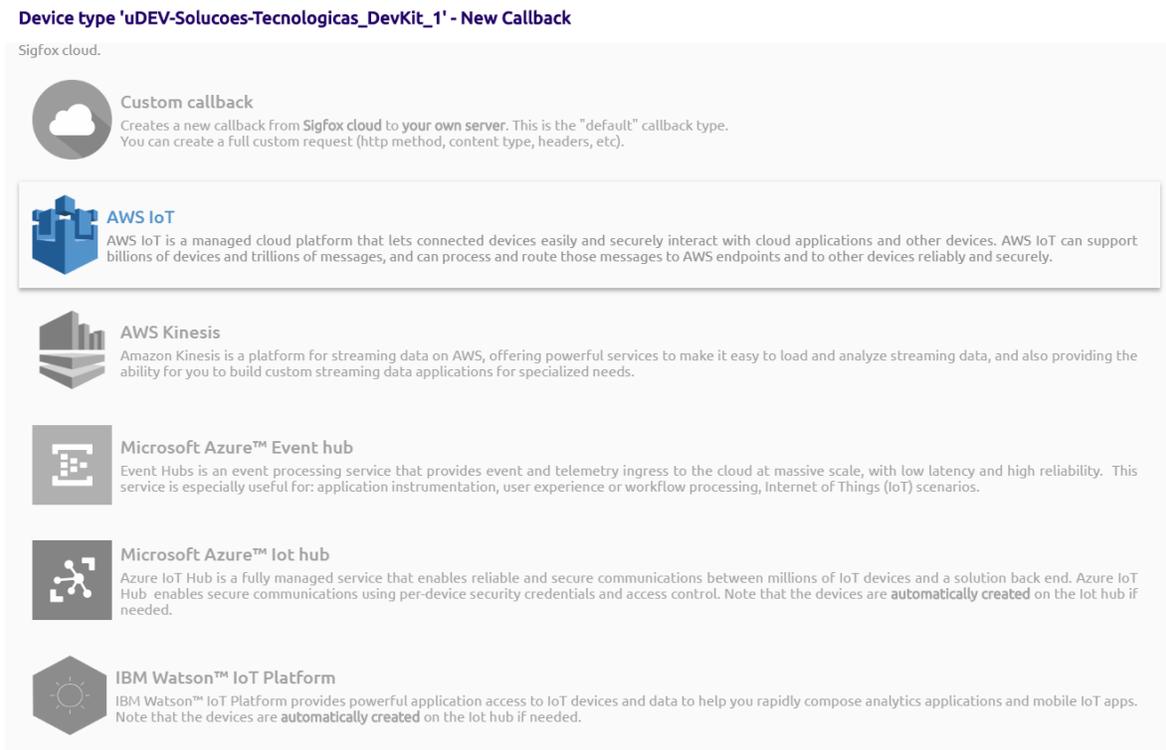


Figura 39: Seleção do Tipo de *Callback* no *Backend* Sigfox

Fonte: Autor

Na sequência, há uma tela similar àquela de configuração de um *callback* customizado como na Figura 33. É selecionado um *callback* do tipo *Service - Data Advanced*. Nota-se um campo com o External ID do dispositivo e campos para entrada das informações específicas do *stack* AWS. O campo com os dados a serem enviados pode ser preenchido com as variáveis padrão do tipo do *callback* e serão as informações a serem transmitidas para a nuvem.

Device type uDEV-Solucoes-Tecnologicas_DevKit_1 - Callback edition

You can find complete documentation about AWS IoT following this [link](#).
Click on ? buttons to display help relative to a particular field.

Callbacks

Type: SERVICE | DATA_ADVANCED

The DATA ADVANCED callback is delivered with a delay of approximately 30 seconds.

Config method: CROSS_ACCOUNT Launch Stack ?

External Id: XXXXXXXXXX ?

ARN Role: Paste your ARN role id here ?

Topic: Enter your topic here ?

Region: US East (Ohio) ?

Custom payload config: ?

Json Body

```
{
  "device": "{device}",
  "time": "{time}",
  "data": "{data}",
  "lqi": "{lqi}",
  "linkQuality": "{linkQuality}",
  "fixedLat": "{fixedLat}",
  "fixedLng": "{fixedLng}",
  "operatorName": "{operatorName}",
  "countryCode": "{countryCode}",
  "seqNumber": "{seqNumber}"
}
```

Available variables: device, time, data, seqNumber, lqi, linkQuality, fixedLat, fixedLng, operatorName, countryCode, deviceTypeId
Additional body variables: ? computedLocation
Custom variables:

Figura 40: Configuração do *Callback* AWS IoT

Fonte: Autor

Como alguns dos inputs (ARN Role ID e Topic Name) dependem da configuração do *stack* no próprio *backend* Sigfox há um link para o AWS. Neste ponto é escolhido o sistema a ser utilizado para realizar a integração - neste caso o CloudFormation.

O processo de criação do *stack* tendo o *Template* pronto pode ser feito em três etapas.

A primeira etapa consiste em selecionar o *Template* a ser utilizado pelo CloudFormation para realizar a conexão com o sistema Sigfox - o que é definido pelo arquivo JSON com os campos e operações a serem executadas.

The screenshot shows the 'Create stack' wizard in the AWS CloudFormation console. The breadcrumb navigation is 'CloudFormation > Stacks > Create stack'. The left sidebar shows four steps: Step 1 (Specify template), Step 2 (Specify stack details), Step 3 (Configure stack options), and Step 4 (Review). The main content area is titled 'Create stack' and is divided into two sections: 'Prerequisite - Prepare template' and 'Specify template'. In the 'Prerequisite' section, there are three radio buttons: 'Template is ready' (selected), 'Use a sample template', and 'Create template in Designer'. The 'Specify template' section explains that a template is a JSON or YAML file. It has two main options: 'Amazon S3 URL' and 'Upload a template file' (selected). Under 'Upload a template file', there is a 'Choose file' button and the filename 'SigfoxConnectorTemplate.json' is displayed. Below this, it says 'JSON or YAML formatted file'. At the bottom of this section, an 'S3 URL' is shown as 'https://s3.us-east-2.amazonaws.com/cf-templates-1anzl7q12ygd-us-east-2/2020138513-SigfoxConnectorTemplate.json' and a 'View in Designer' button is present. At the bottom right of the entire form, there are 'Cancel' and 'Next' buttons.

Figura 41: Definição do *Template* a ser Empregado na Formação do *Stack*

Fonte: Autor

Na sequência é feita a entrada do External ID do dispositivo Sigfox a ser integrado e a identificação da conta AWS do usuário que irá gerenciar o *stack*. Também é selecionado o nome global da aplicação (*Topic Name*) que servirá como identificador global dentro da infraestrutura AWS e dentro do *Sigfox* para identificar o conector. Também é possível definir a região AWS que o sistema irá utilizar.

The screenshot shows the 'Specify stack details' step in the AWS CloudFormation console. The breadcrumb navigation is 'CloudFormation > Stacks > Create stack'. On the left, a sidebar shows four steps: Step 1 'Specify template', Step 2 'Specify stack details' (which is active), Step 3 'Configure stack options', and Step 4 'Review'. The main content area is titled 'Specify stack details' and contains several input fields:

- Stack name:** A text box containing 'SigfoxIoTConnector'. Below it, a note states: 'Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-)'.
- Parameters:** A section header with a sub-note: 'Parameters are defined in your template and allow you to input custom values when you create or update a stack.'
- AWSAccountId:** A text box containing a redacted AWS Account ID. The instruction above it is: 'Input your AWS Account ID. http://docs.aws.amazon.com/IAM/latest/UserGuide/console_account-alias.html'.
- ExternalId:** A text box containing a redacted External ID. The instruction above it is: 'Please copy/paste the External Id assigned to you in the Sigfox console.'
- Region:** A text box containing 'us-east-1'. The instruction above it is: 'Input the AWS Region you wish to use.'
- TopicName:** A text box containing 'sigfox'. The instruction above it is: 'Input the name of the topic Sigfox will use to publish data.'

At the bottom right of the form, there are three buttons: 'Cancel', 'Previous', and 'Next'.

Figura 42: *Input dos Dados do Dispositivo Sigfox e Definição de Parâmetros*

Fonte: Autor

Por último, é possível personalizar algumas configurações, entre elas a criação de variáveis globais (do tipo dicionário chave-valor), políticas de acesso (usuário único ou múltiplos usuários) e regras de notificação.

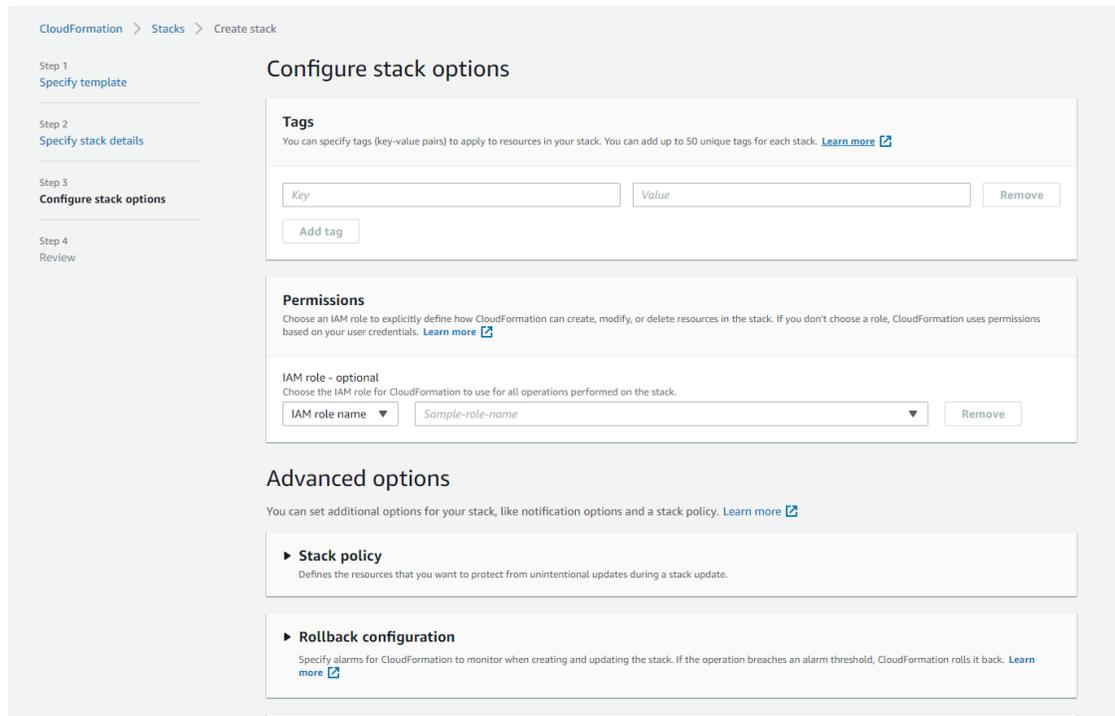


Figura 43: Configurações de Parâmetros Adicionais do *Stack*

Fonte: Autor

Com isso, o CloudFormation criará o *stack* executando as informações e operações descritas no *Template*. O output deste processo é o ARN Role ID a ser utilizado na outra ponta da Cross-Account (juntamente com o External ID provido pela rede Sigfox).

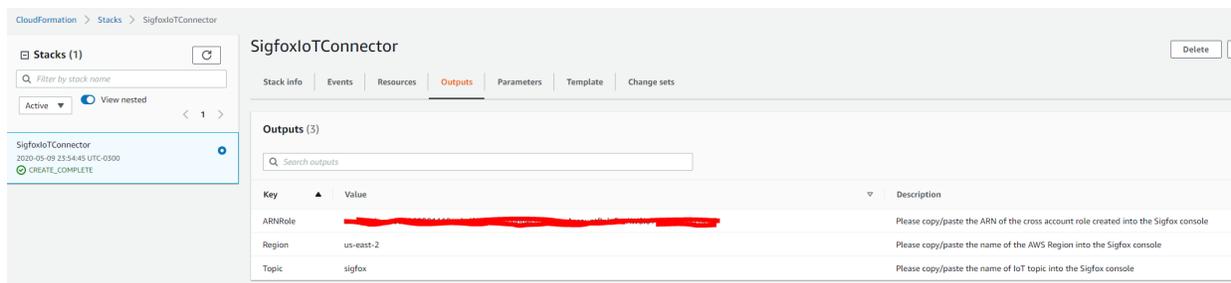


Figura 44: Geração do ARN Role ID para Comunicação com o *Backend* Sigfox

Fonte: Autor

O ARN Role ID e o *Topic Name* são então colocados na configuração de *callback* indicada na Figura 40. Neste ponto a integração do sistemas está feita e a infraestrutura necessária para

transmissão das informações fica provisionada e gerenciada pelo CloudFormation enquanto que o *backend* Sigfox gerencia os *callbacks* acionados pelo dispositivo.

4.4.2 Configuração do Banco de Dados

Mesmo que a conexão entre o *backend* Sigfox e o ambiente *cloud* do AWS tenha sido realizada, ainda é necessário uma estrutura para armazenar e permitir a manipulação dos dados obtidos. Para isso se torna necessária a criação de um banco de dados adequado para uma aplicação de IoT.

O padrão *de facto* para bancos de dados em aplicações de IoT são aqueles baseados em NoSQL (Souad Amghar, 2018).

O NoSQL representa uma mudança de paradigma frente ao padrão anterior - o *Structured Query Language* (SQL) - tendo este tido enorme sucesso diversos bancos de dados que continuam sendo utilizados até hoje, como o Microsoft SQL Server, Oracle DB, PostgreSQL e MySQL. O SQL foi criada para gerenciar dados num modelo de banco de dados relacional - isto é, adequado para armazenar e manipular dados estruturados.

A própria natureza do SQL o torna adequado para muitas aplicações em que o dado a ser recebido é estruturado e haja uma expectativa que ele assim se mantenha. Ele é organizado de forma muito similar a uma tabela - primeiramente as colunas de dados e seu tipo de dado (texto, inteiro, ponto flutuante) são definidas e, a partir daí, os dados são recebidos.

Porém, uma aplicação de IoT pode ser dinâmica e, portanto, nem sempre os dados a serem transferidos possuem um padrão fixo; não só a quantidade de dados sendo armazenados em cada requisição ("*query*") pode variar, como também o seu tipo. Neste contexto, se tornam adequadas as bases de dados não relacionais - NoSQL - que, de formas distintas, conseguem lidar com dados não estruturados de uma forma única.

Isto não quer dizer que os dados em IoT sejam necessariamente desorganizados ou não possuam padrão. O fato é que o paradigma NoSQL consegue se adaptar a diferentes tipos de entradas de dados, não estando limitado a uma estrutura fixa.

A Figura 45 ilustra as diferenças entre os padrões SQL e NoSQL quanto à estrutura dos dados armazenados.

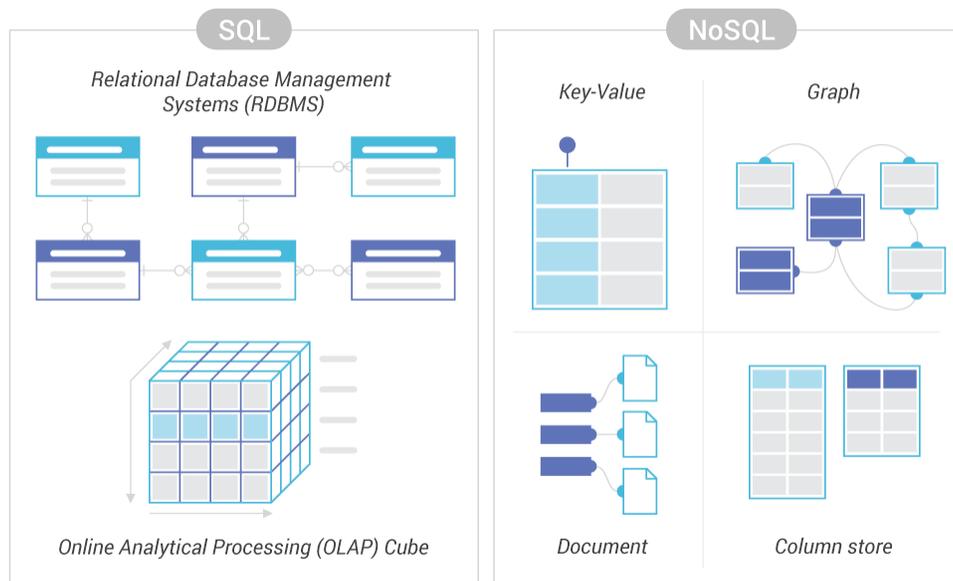


Figura 45: Comparação da Estrutura de Dados num Banco de Dados SQL e NoSQL

Fonte: (Souad Amghar, 2018)

Alguns dos atributos que caracterizam o padrão NoSQL são:

- **Heterogeneidade de dados:** os dados recebidos podem possuir formatos, modelos (desestruturados, semi-estruturados ou estruturados) e conteúdo. Na mesma aplicação IoT podem existir dados de texto, localização, temperatura, mídia e outros. A variabilidade de dados impõe um desafio para os sistemas de armazenamento
- **Interoperabilidade semântica:** a interoperabilidade significa permitir que diferentes serviços e aplicações possam trocar informações e dados de forma simples - permitindo amplo acesso como num contexto de *cloud*. A interoperabilidade pode ser alcançada através da unificação de formatos e linguagens de *query*
- **Escalabilidade:** é a habilidade de um sistema de executar grandes requisições de dados com baixo tempo de resposta independente do *hardware*, considerando o alto *throughput* de dados num ambiente IoT.
- **Processamento em tempo real:** a maioria das aplicações IoT recebem ou enviam dados que poderão - na maioria das aplicações - precisar serem acessados ou monitorados em tempo real de forma responsiva e com boa performance.
- **Segurança:** é condição necessária para uma rede IoT ser segura e, da mesma forma, o sistema de armazenamento e gestão dos dados precisa possuir camadas de criptografia e autenticação - de forma a apenas permitir dispositivos autorizados a transferir determinados dados com diferentes níveis de permissão

- Agregação de dados: é necessário que o sistema de banco de dados possua a habilidade de combinar dados advindos de diferentes fontes e que possa eliminar eventuais redundâncias

Existem quatro categorias de bancos de dados NoSQL .

- Bancos de dados "*Key-value*": dados são armazenados como pares de chave-valor e endereçados por uma chave única. O dado é composto por duas partes, uma chave (formato *string*) e um valor - que é o dado propriamente dito - que pode ser um tipo de dado usual (*string*, *float*, *double*), um objeto ou um documento inteiro. Sendo a categoria mais simples, são considerados aqueles com melhor desempenho (Souad Amghar, 2018)
- Bancos de dados "*Column-oriented*": dados são armazenados como linhas e colunas. Cada chave pode ser associada com uma ou mais colunas - que podem estar associadas de formas diferentes - e nem toda linha pode possuir as mesmas colunas.
- Bancos de dados "*Document-oriented*": dados são armazenados como conjuntos de documentos e endereçados via formatos de arquivos como XML e JSON. As estruturas dos documentos não precisam ser similares, ou seja, cada documento pode conter conjuntos diferentes de dados
- Bancos de dados baseados em grafos: dados são armazenados como grafos, em que os nodos atuam como os objetos e as conexões entre nodos relacionam os objetos entre si.

O ambiente AWS possui várias implementações de bancos de dados - relacionais ou não-relacionais - de diferentes categorias, como pode ser visto na Figura 46

Database type	Use cases	AWS service
Relational	Traditional applications, ERP, CRM, e-commerce	 Amazon Aurora  Amazon RDS  Amazon Redshift
Key-value	High-traffic web apps, e-commerce systems, gaming applications	 Amazon DynamoDB
In-memory	Caching, session management, gaming leaderboards, geospatial applications	 Amazon ElastiCache for Memcached  Amazon ElastiCache for Redis
Document	Content management, catalogs, user profiles	 Amazon DocumentDB (with MongoDB compatibility)
Wide column	High scale industrial apps for equipment maintenance, fleet management, and route optimization	 Amazon Keyspaces (for Apache Cassandra)
Graph	Fraud detection, social networking, recommendation engines	 Amazon Neptune
Time series	IoT applications, DevOps, industrial telemetry	 Amazon Timestream
Ledger	Systems of record, supply chain, registrations, banking transactions	 Amazon QLDB

Figura 46: Serviços de Banco de Dados Disponíveis no AWS

Fonte: AWS

De forma a garantir a simplicidade na manipulação de dados - visto que, idealmente, eles serão posteriormente transferidos para outros serviços - e dada a potencial variabilidade no formato ou quantidade dos dados sendo transferidos a cada *callback*, optou-se por utilizar o sistema de banco



Figura 47: Exemplos de Bancos de Dados DynamoDB Ilustrando Tabelas, Itens e Atributos

Fonte: AWS

de dados NoSQL do tipo chave-valor Amazon DynamoDB.

O Amazon DynamoDB é um banco de dados NoSQL que suporta o tipo "chave-valor" ou pode ser orientado por documentos. Ele se destaca pela alta performance e escalabilidade, suportando mais de 10 trilhões de requisições por dia e possui performance de pico de até 20 milhões de requisições por segundo. É utilizado por grandes empresas de tecnologia como Lyft, Airbnb, Samsung e Toyota para aplicações críticas (Amazon, 2020).

O DynamoDB possui três componentes básicos:

- **Tabelas:** similar a outros sistemas de bancos de dados, o DynamoDB armazena os dados em tabelas. Tabelas são conjuntos de dados, como, por exemplo, numa tabela *Carros* que armazena informações sobre veículos;
- **Itens:** cada tabela contém zero ou mais itens. Itens são grupo de atributos que são unicamente identificáveis entre todos os outros itens, como um item *Veículo A* na tabela *Carros*;
- **Atributos:** cada item é composto por um ou mais atributos - nem todos os itens precisam possuir os mesmos atributos. Cada atributo é a parcela fundamental dos elementos de dados, como, por exemplo, um atributo *proprietário* do item *Veículo A*

A criação do banco de dados DynamoDB é feita de forma simples, sendo necessários apenas alguns parâmetros, como ilustrado na Figura 48.

Criar tabela do DynamoDB Tutorial ?

O DynamoDB é um banco de dados sem esquema que requer somente o nome de uma tabela e a chave primária. A chave primária da tabela é constituída de um ou dois atributos que identificam itens, particionam os dados, e classificam os dados dentro da partição de maneira exclusiva.

Nome da tabela* ⓘ

Chave primária* Chave de partição

ⓘ

Adicionar chave de classificação

ⓘ

Configurações da tabela

As configurações padrão são a forma mais rápida de começar a usar sua tabela. Você poderá modificar essas configurações padrão agora ou depois que a tabela for criada.

Usar configurações padrão

Figura 48: Criação de um Banco de Dados DynamoDB

Fonte: Autor

O processo de criação do banco de dados NoSQL é bem mais simples do que um SQL - em que previamente devem ser definidas cada coluna e seus respectivos tipos de dados a serem definidos. Na maioria dos bancos de dados que não são baseados em *cloud* também acaba sendo necessário construir e provisionar toda a infraestrutura de rede caso seja necessário que haja recebimento de dados da Internet. Neste caso, toda a infraestrutura já é provida pelo AWS.

Nota-se o uso de dois parâmetros para a construção do banco de dados que são mostrados na Figura 48: uma chave de partição ("*Partition key*") e uma chave de classificação ("*Sort key*") - . Esses parâmetros servem como apoio de identificação das entradas de dados; como os dados podem vir sem um formato definido, é necessário alguma forma de identificar cada um deles - papel que seria de uma coluna de identificação numérica num padrão SQL clássico.

A chave primária ("*Primary Key*") é o elemento que identifica unicamente cada item na tabela - de forma com que dois itens não podem possuir a mesma chave. O DynamoDB suporta dois tipos de chave primária:

- Chave de Partição ("*Partition Key*"): uma chave primária simples composta de um único atributo. O DynamoDB utiliza o valor da chave como entrada para uma função *hash* cuja saída determina a partição (armazenamento físico interno do DynamoDB) em que o item será salvo. No caso da tabela possuir seguir este modelo, nenhum item pode compartilhar a mesma chave com outro - é o exemplo da tabela *People* na Figura 47 (cuja chave está em negrito);
- Chave de Partição e Chave de Classificação ("*Partition Key and Sort Key*"): chave composta em que um o DynamoDB utiliza uma das chaves para definir a partição. Todos os itens com a

mesma chave de partição serão armazenados juntos e classificados pela chave de classificação. Este modelo permite que um item compartilhe a mesma chave de partição com outro. É o caso da tabela *Music* na Figura 47 (cujas chaves estão em negrito).

Na aplicação feita foi escolhida como chave de partição a ID do dispositivo - assim pode-se armazenar todos os dados do mesmo dispositivo numa única partição - e, de forma a não sobrepor os dados, foi escolhida como chave de classificação o *timestamp* - gerado por um temporizador interno do AWS.

4.4.3 Configuração das Regras para Aquisição de Dados

Preparada a infraestrutura necessária para receber os dados, é necessário, por fim, configurar as regras que permitirão com que um dado transmitido possa ser armazenado no banco de dados criado. Para isso é utilizado o painel de IoT do AWS - o AWS IoT Core. Este é o serviço AWS em que pode ser feita a gestão e monitoramento de todos os dispositivos ou requisições de IoT. Para a aplicação proposta - será simplesmente criada uma regra em que, cada vez que o dispositivo Sigfox realizar um *callback* AWS - os dados da mensagem ("*payload*") será armazenado na tabela DynamoDB criada.

A regra possui uma instrução de consulta (a origem das mensagens a serem processadas), as suas ações decorrentes.

Neste caso a instrução de consulta foi simplesmente selecionar todos os dados presentes na infraestrutura CloudFormation provisionada. A instrução SQL para isso é simples:

```
SELECT * FROM 'sigfox'
```

Ela significa, em linguagem SQL, para selecionar todo o conteúdo (designado pelo símbolo "*") presente em 'sigfox' - que era o nome da nossa infraestrutura provisionada CloudFormation.

Além da instrução, é necessário definir a ação a ser realizada para cada consulta. Há diversas opções de ações a serem realizadas - algumas são mostradas na Figura 49

<input type="radio"/>		Inserir uma mensagem em uma tabela do DynamoDB DYNAMODB
<input type="radio"/>		Dividir mensagem em várias colunas de uma tabela de banco de dados (DynamoDBv2) DYNAMODBv2
<input type="radio"/>		Chamar uma função Lambda transmitindo a mensagem de dados LAMBDA
<input type="radio"/>		Enviar uma mensagem como uma notificação por push SNS SNS
<input type="radio"/>		Enviar uma mensagem para uma fila do SQS SQS
<input type="radio"/>		Enviar mensagens para um Amazon Kinesis Stream AMAZON KINESIS
<input type="radio"/>		Publicar novamente mensagens em um tópico do AWS IoT NOVA PUBLICAÇÃO DO AWS IOT
<input type="radio"/>		Armazenar mensagens em um bucket do Amazon S3 S3
<input type="radio"/>		Enviar mensagens para um stream do Amazon Kinesis Firehose AMAZON KINESIS FIREHOSE
<input type="radio"/>		Enviar dados da mensagem para métrica do CloudWatch MÉTRICAS DO CLOUDWATCH

Figura 49: Exemplos de Ações Disponíveis para Uma Regra AWS IoT

Fonte: AWS

A ação selecionada foi a de "Inserir uma mensagem em uma tabela do DynamoDB". Para isso é necessário definir a tabela a receber os dados e os atributos que servirão como chave. A Figura 50 mostra que foi escolhido o *deviceid* e o *timestamp* como chaves de partição e de classificação, respectivamente. Basicamente o AWS buscará por um atributo *device* dentro do corpo da mensagem para torná-lo a chave de partição e executará uma função interna do AWS *timestamp()* para gerar uma marcação temporal da mensagem de forma a distinguir um registro do outro. O dado a ser gravado é simplesmente o *payload* completo recebido pelo *backend* Sigfox.

Configurar ação

Inserir uma mensagem em uma tabela do DynamoDB

A tabela deve conter chaves hash e de intervalo.

*Nome da tabela
sigfox ↻ **Criar um novo recurso**

*Chave de partição	*Tipo de chave hash	*Valor da chave hash
deviceid	STRING	\${device}
Chave de intervalo	Tipo de chave de intervalo	Valor da chave de intervalo
timestamp	STRING	\${timestamp()}

Gravar dados da mensagem nessa coluna
payload

Operação ?
Insert

Figura 50: Configuração da Ação para a Regra AWS IoT

Fonte: Autor

Assim, cada vez que o dispositivo Sigfox realizar um *callback* este será detectado pela infraestrutura CloudFormation e seu conteúdo ("*payload*") será armazenado no banco de dados.

4.4.4 Aquisição e Manipulação dos Dados

Com a infraestrutura provisionada, banco de dados criado e as regras configuradas, é possível observar os dados provenientes de um *callback* podem ser recebidos e processados dentro do mais robusto ambiente AWS.

As Figuras 51 e 52 mostram o resultado da integração bem sucedida entre Sigfox e Amazon AWS.

deviceid	timestamp	payload
43ED5C	1589079751582	{"countryCode": {"S": "76"}, "data": {"S": "2e2a164400070000"}, "devi...
43ED5C	1589079811575	{"countryCode": {"S": "76"}, "data": {"S": "2029164400020000"}, "devi...
43ED5C	1589079855438	{"countryCode": {"S": "76"}, "data": {"S": "202a164400030000"}, "devi...
43ED5C	1589079901592	{"countryCode": {"S": "76"}, "data": {"S": "202a164400010000"}, "devi...
43ED5C	1589079905760	{"countryCode": {"S": "76"}, "data": {"S": "2a2a164400030000"}, "devi...
43ED5C	1589079941798	{"countryCode": {"S": "76"}, "data": {"S": "2c2a164400020000"}, "devi...
43ED5C	1589081195545	{"countryCode": {"S": "76"}, "data": {"S": "2a2a164400030000"}, "devi...
43ED5C	1589081143091	{"countryCode": {"S": "76"}, "data": {"S": "2e2a164400060000"}, "devi...
43ED5C	1589081149609	{"countryCode": {"S": "76"}, "data": {"S": "202a164500020000"}, "devi...
43ED5C	1589081167560	{"countryCode": {"S": "76"}, "data": {"S": "202a164400070000"}, "devi...
43ED5C	1589086458892	{"countryCode": {"S": "76"}, "data": {"S": "082a164400080000"}, "devi...
43ED5C	1589086465592	{"countryCode": {"S": "76"}, "data": {"S": "2029164400030000"}, "devi...
43ED5C	1589088257590	{"countryCode": {"S": "76"}, "data": {"S": "0a29164400030000"}, "devi...
43ED5C	1589089005549	{"countryCode": {"S": "76"}, "data": {"S": "2e29164400030000"}, "devi...

Figura 51: Conjunto de Dados Provenientes do *Backend* Sigfox Recebidos e Armazenados num Banco de Dados DynamoDB

Fonte: Autor

```

{
  "deviceid": "43ED5C",
  "payload": {
    "countryCode": "76",
    "data": "2e2a164400070000",
    "device": "43ED5C",
    "fixedLat": "0.0",
    "fixedLng": "0.0",
    "linkQuality": "2",
    "lqi": "Good",
    "operatorName": "SIGFOX_Brazil_WND",
    "seqNumber": "143",
    "time": "1589079728",
    "timestamp": "1589079751582"
  }
}

```

Figura 52: Exemplo do Dado Provenientes do *Backend* Sigfox Recebidos e Armazenados num Banco de Dados DynamoDB

Fonte: Autor

5 Conclusões

Com a metodologia proposta e os testes desenvolvidos foi possível determinar que a rede Sigfox é uma boa alternativa para aquisição de dados em longas distâncias.

A rede Sigfox se mostrou adequada como um meio de transmissão de dados de um kit de desenvolvimento para longas distâncias. Independente dos dados que podem ser adquiridos em um kit de desenvolvimento - que depende dos sensores utilizados - a rede se mostrou capaz de transmitir um *payload* em diferentes localidades sem prejuízo no conteúdo das mensagens, mesmo quando a qualidade do sinal não estava excelente.

O sistema proprietário de geolocalização - mesmo tendo a proposta de ser um serviço auxiliar ao GPS ou outros existentes - possui potencial de melhoria - reconhecido pelo próprio consórcio. O erro apresenta ordem de quilômetros - o que, mesmo sendo o suficiente para algumas aplicações de larga escala ou sobre grandes distâncias territoriais - pode não ser adequado para aplicações residenciais ou até industriais.

Também deve ser levado em conta aspectos qualitativos como a forma com que o consórcio Sigfox se preocupou em tornar a interface do *backend* amigável e as APIs de fácil integração com serviços de *cloud computing* já estabelecidos, necessitando o usuário apenas configurar a infraestrutura IoT e o banco de dados. Foi possível realizar uma integração bem sucedida entre o *backend* Sigfox e o Amazon AWS - o que aumenta consideravelmente o potencial da aplicação devido a todos os serviços AWS disponíveis.

Há espaço considerável para novas pesquisas e desenvolvimentos baseados na rede Sigfox. Trabalhos futuros podem, por exemplo, avaliar as especificidades da rede e das aplicações em situações *indoor* ou *outdoor*, bem como avaliar o consumo energético destas e destacar potenciais pontos de melhorias a níveis de projeto dos kits de desenvolvimento. Também podem ser explorados novos desenvolvimentos a nível da aplicação, avaliando outros sistemas de nuvem em que a infraestrutura de *cloud* possa ser construída, de forma a aumentar a integração da rede com outras plataformas amplamente utilizadas comercialmente.

6 Referências

- Amazon, 2020. **Computação em Nuvem com Amazon Web Services**. Disponível em: <https://go.aws/2JUmwRA>, Acesso: 04 Abr 2020.
- ANATEL, 2017. **Resolução N° 680**. Disponível em: www.anatel.gov.br/legislacao/resolucoes/2017/936-resolucao-680, Acesso: 24 Out 2019.
- ARM Holdings, 2020. **Arm Processors for the Widest Range of Devices—from Sensors to Servers**. Disponível em: <https://www.arm.com/products/silicon-ip-cpu>, Acesso: 04 Abr 2020.
- Barr, J., 2019. **AWS Named as a Leader in Gartner’s Infrastrucure as a Service (IaaS) Magic Quadrant for the 9th Consecutive Year**. Disponível em: <https://amzn.to/35ejni4>, Acesso: 08 Set 2020.
- BNDES, 2017. **Benchmark de Iniciativas e Políticas Públicas** Disponível em: <https://bit.ly/2mqOLpb>, Acesso: 21 Set 2019.
- Forouzan, B., Mosharraf, F., 2012. **Computer Networks: A Top-down Approach**. McGraw-Hill.
- Gerez, S.H., 2016. **Implementation of Digital Signal Processing: Some Background on GFSK Modulation**. Disponível em: <https://bit.ly/2r0Awdb>, Acesso: 17 Nov 2019.
- GSMA, 2014. **Understanding the Internet of Things (IoT)**. Disponível em: <https://bit.ly/2vRDokP>, Acesso: 21 Out 2019.
- Holdings, A., 2020. **Arm Cortex-M0+ Processor: The Most Energy-Efficient Arm Processor Available for Constrained Embedded Applications**. Disponível em: <https://www.arm.com/products/silicon-ip-cpu>, Acesso: 04 Abr 2020.
- IEEE, 2017. **IEEE Standards Activities in the Internet of Things**. IEEE Standards Association Disponível em: <https://standards.ieee.org/content/dam/ieee-standards/standards/web/documents/other/iot.pdf>, Acesso: 21 Out 2019.
- IEEE, 2018. **Internet of Things for Telecom Engineers: A Report on Current State and Future Techonologies**. Disponível em: <http://forms1.ieee.org/rs/682-UPB-550/images/IEEE-IOT-White-Paper.pdf>, Acesso: 21 Out 2019.
- Manyika, J., Woetzel, J., Dobbs, R., 2015. **Unlocking the potential of the Internet of Things**. McKinsey Global Institute Reports Disponível em: <https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/the-internet-of-things-the-value-of-digitizing-the-physical-world>, Acesso: 21 Set 2019.
- Microchip, 2020. **SAM D21/DA1 Family, Low-Power, 32-bit Cortex M0+ MCU with Advanced Analog and PWM**. Disponível em: <https://www.microchip.com/wwwproducts/en/ATSAMD21G18>, Acesso: 04 Abr 2020.

- Proakis, J., Salehi, M., 2008. **Digital Communications**. McGraw-Hill.
- Rama, G., 2017. **AWS Market Share is Triple Azure's**. Disponível em: <https://awsinsider.net/articles/2017/08/01/aws-market-share-3x-azure.aspx>, Acesso: 04 Abr 2020.
- Rappaport, T., 2009. *Wireless Communications: Principles and Practice*. Prentice Hall communications engineering and emerging technologies series, Dorling Kindersley. URL: <https://books.google.com.br/books?id=11qEWkNFFwQC>.
- Ruckebusch, P., Giannoulis, S., Moerman, I., Hoebeke, J., Poorter, E.D., 2018. **Modelling the energy consumption for over-the-air software updates in LPWAN networks: SigFox, LoRa and IEEE 802.15.4g**. *Journal of Internet of Things* 3, 104–119. Disponível em: <https://biblio.ugent.be/publication/8579474/file/8579475.pdf>, Acesso: 23 Out 2019.
- Santos, C., 2018. **O Desafio de Conectar "Coisas" a Baixo Custo no Brasil**. Disponível em: <https://bit.ly/2BDj5RQ>, Acesso: 24 Out 2019.
- Sigfox, 2017. **Sigfox Geolocation - The Simplest and Cheapest IoT Location Service**. Disponível em: <https://bit.ly/3c3r5vR>, Acesso: 04 Abr 2020.
- Sigfox, 2018. **Radio Technology Keypoints**. Disponível em: <https://www.sigfox.com/en/sigfox-iot-radio-technology>, Acesso: 14 Nov 2019.
- uDev Soluções Tecnológicas Ltda., 2018. **EDK Maker's Kit**. Disponível em: <http://sigfox.udev.com.br/edk-makers-kit/>, Acesso: 10 Nov 2019.
- Souad Amghar, Safae Cherdal, S.M., 2018. **Which NoSQL database for IoT applications?**. 2018 International Workshop on Architectures for Future Mobile Computing and Internet of Things Disponível em: <https://bit.ly/2ytxNX>, Acesso: 24 Mai 2019.
- Synchrologix, 2018. **LPWAN Technologies**. Disponível em: <https://bit.ly/2MNJ3st>, Acesso: 24 Out 2019.
- Tabbane, S., 2017. **Internet of Things: A technical overview of the ecosystem**. Disponível em: <https://bit.ly/31ItSF4>, Acesso: 23 Out 2019.
- Texas Instruments, 2017. **Wireless Connectivity for the Internet of Things: One Size Does Not Fit All**. Disponível em: <http://www.ti.com/lit/wp/swry010a/swry010a.pdf>, Acesso: 21 Set 2019.
- Van Brummelen, G., 2012. **Heavenly mathematics: The forgotten art of spherical trigonometry**. Princeton University Press.
- Walsh, J., 2018. **Low Power Wide Area Technologies for IoT Use Cases**. SCTE-ISBE Expo 2018. .
- Wisol, 2017. Wisol / wssfm11r2dat data sheet rev.01. Disponível em: <https://bit.ly/33nnXbn>, Acesso: 13 Set 2020.

Zuñiga, J.C., 2017. **Technology Overview for W3C WoT Group**. Disponível em: <https://bit.ly/2CuwGLE>, Acesso: 10 Nov 2019.

Apêndices

A Visão Geral do Amazon Identity and Access Management (IAM)

O Amazon Identity and Access Management (IAM) é um serviço AWS que permite controlar acessos e permissões aos serviços AWS para cada recurso e usuário.

Diferentemente de outros serviços, a Amazon não cobra pela utilização deste sistema de gestão de acessos, sendo por si só gratuito. A cobrança é feita apenas sob as aplicações e serviços que venham utilizar o IAM (?).

O IAM atua como um serviço auxiliar para verificação dos tokens de segurança e definição de papéis ("roles") e permissões para cada aplicação.

A Figura 53 ilustra o papel do IAM na aplicação desenvolvida.



Figura 53: Diagrama Ilustrativo do Papel do IAM na Aplicação Desenvolvida

Fonte: Autor

O IAM depende de alguns termos e nomenclaturas padrões do ambiente AWS para gerenciar os níveis de acesso:

O acesso no AWS é gerenciado através da criação de políticas e sua consequente designação para identificadores IAM que podem se referir a usuários específicos, grupos de usuários ou papéis ("roles"). Uma política é um objeto AWS que, quando associado a um identificador ou recurso, define suas permissões. O AWS avalia essas políticas quando um outro identificador IAM ou token de segurança realiza uma requisição. Permissões das políticas determinam se a requisição será ou não atendida. Como muitos outros objetos AWS, as políticas são armazenadas em formato JSON (Amazon, 2020).

A Figura 54 ilustra o papel do IAM dentro de uma aplicação com mais de um usuário e mais de um serviço AWS em uso (neste caso, EC2 e S3).

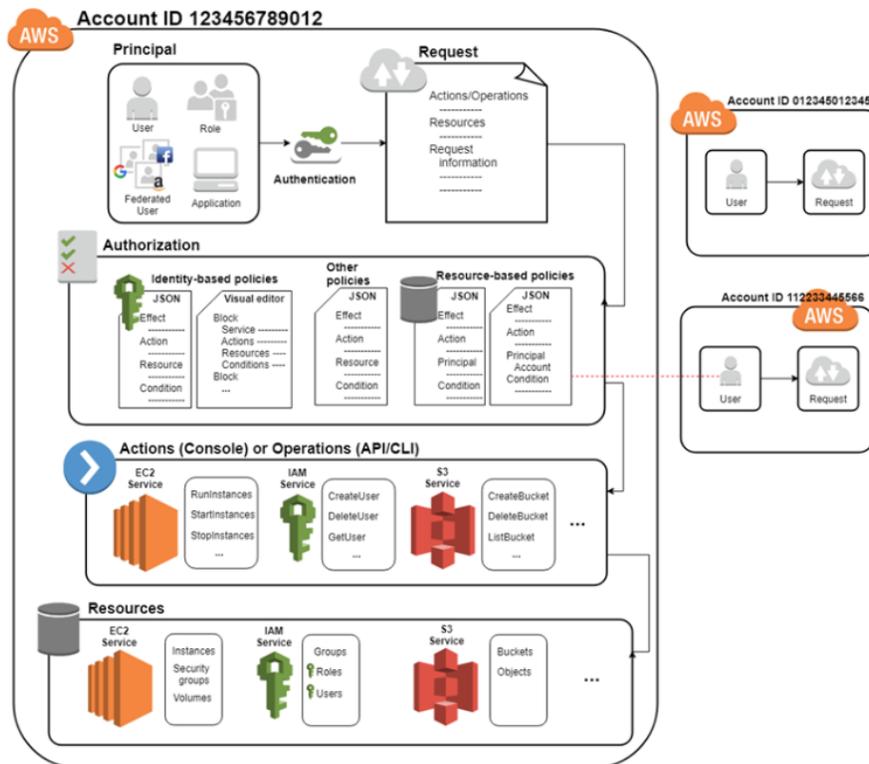


Figura 54: Exemplo do Papel do IAM numa Aplicação Multi-Usuário e com Mais de Um Serviço

Fonte: AWS

B *Template* para Criação do Stack no AWS CloudFormation

```
{
  "AWSTemplateFormatVersion": "2020-05-16",

  "Description": "This template is used to create a role with cross account permissions
  that can connect and push data to AWS IoT.",

  "Parameters": {
    "ExternalId": {
      "Type": "String",
      "MinLength": "2",
      "MaxLength": "1224",
      "AllowedPattern": "[\\w+=,\\.@:\\\\/-]*",
      "Default": "a1b2c3d4e5f6g7h8i9",
    }
  }
}
```

```
    "Description": "Please copy/paste the External Id assigned to you in the Sigfox console."
  },
  "Region": {
    "Type": "String",
    "AllowedPattern": "([a-z]{2}-[a-z]+-\\d{1})",
    "Default": "us-east-1",
    "ConstraintDescription": "The value must match an AWS region name.",
    "Description": "Input the AWS Region you wish to use."
  },
  "TopicName": {
    "Type": "String",
    "MinLength": "1",
    "Default": "sigfox",
    "ConstraintDescription": "The value must match the name of an MQTT topic.",
    "Description": "Input the name of the topic Sigfox will use to publish data."
  },
  "AWSAccountId": {
    "Type": "String",
    "AllowedPattern": "(\\d{12})",
    "Default": "012345678901",
    "ConstraintDescription": "The value must be an AWS Account ID.",
    "Description": "Input your AWS Account ID."
  }
},
"Resources": {
  "crossAccountRoleForAWSIoT": {
    "Type": "AWS::IAM::Role",
    "Properties": {
      "AssumeRolePolicyDocument": {
        "Statement": [{
          "Effect": "Allow",
          "Principal": {
            "AWS": "arn:aws:iam::225344712516:root"
          },
          "Condition": {"StringEquals": {"sts:ExternalId": {"Ref": "ExternalId"}}},
          "Action": [
            "sts:AssumeRole"
          ]
        }]
      }
    }
  }
},
"CrossAccountPolicy": {
  "Type": "AWS::IAM::Policy",
  "Properties": {
    "PolicyName": "IAMpolicyIoT",
    "PolicyDocument": {
      "Statement": [{
        "Effect": "Allow",
```

```

    "Action": ["iot:Connect","iot:DescribeEndpoint"],
    "Resource": "*"
  }, {
    "Effect": "Allow",
    "Action": ["iot:Publish"],
    "Resource": {
      "Fn::Join" : [ "", [ "arn:aws:iot:",{ "Ref" : "Region" },":",
        { "Ref" : "AWSAccountId" },
        ":topic/",{ "Ref" : "TopicName" } ] ]
    }
  ]
},
"Roles": [{
  "Ref": "crossAccountRoleForAWSIoT"
}]
}
}
},
"Outputs": {
  "ARNRole": {
    "Description": "Please copy/paste the ARN of the cross account role
    created into the Sigfox console",
    "Value": { "Fn::GetAtt" : [ "crossAccountRoleForAWSIoT", "Arn" ] }
  },
  "Region": {
    "Description": "Please copy/paste the name of the AWS Region into the Sigfox console",
    "Value": { "Ref": "Region" }
  },
  "Topic": {
    "Description": "Please copy/paste the name of IoT topic into the Sigfox console",
    "Value": { "Ref": "TopicName" }
  }
}
}
}

```