



Mestrado em Gestão de Sistemas e Tecnologias de Informação

Dissertação de Mestrado

Alarmística Remota e Operações de Monitorização em Tempo Real

Elaborado por: João Carlos Matos Silva

Nº de Estudante: 201528274

Dissertação orientada por: Professor Doutor Carlos Alberto Galamba Palma Pinto

Barcarena

Setembro, 2018

Alarmística Remota e Operações de Monitorização em Tempo Real – Gestão de Sistemas e Tecnologias de Informação

Alarmística Remota e Operações de Monitorização em Tempo Real – Gestão de Sistemas e Tecnologias de Informação

Universidade Atlântica

Mestrado em Gestão de Sistemas e Tecnologias de Informação

Dissertação de Mestrado

Alarmística Remota e Operações de Monitorização em Tempo Real

Elaborado por: João Carlos Matos Silva

Nº de Estudante: 201528274

Dissertação orientada por: Professor Doutor Carlos Alberto Galamba Palma Pinto

Barcarena

Setembro, 2018

O autor é o único responsável pelas ideias expressas neste relatório.

“O fracasso é a oportunidade de se começar de novo inteligentemente.”

(Henry Ford)

Dedicatória

Dedico este trabalho final de curso de Mestrado em Gestão de Sistemas e Tecnologias de Informação a todos aqueles que me ajudaram, de uma forma ou de outra, a conseguir atingir mais uma etapa na minha vida, em especial à minha família, com especial ênfase à minha esposa e filhas, que abdicaram de muito do seu tempo a auxiliar-me na execução de outras tarefas e com menos possibilidade de usufruir da minha companhia.

Quero também agradecer ao Professor Doutor Carlos Pinto, ao qual agradeço profundamente por toda a paciência e pelo empenho que me incutiu durante todo este processo e ao meu colega e diretor de operações, Nuno Domingues, que me auxiliou nas minhas tarefas profissionais, garantindo que eu tinha tempo para concluir o Mestrado e que essencialmente me fez acreditar que a organização onde nos enquadrámos acredita e aposta nos seus recursos humanos.

Não quero deixar também de agradecer à empresa onde trabalho, Inosat, e a toda a sua hierarquia por acreditar em mim e transmitir-me toda a confiança, transparência e honestidade na relação diária que temos.

A todos em geral, os meus agradecimentos.

Lista de Abreviaturas

Sigla	Descrição
API -	<i>Application Programming Interface</i>
CoAP -	<i>Constrained Application Protocol</i>
DDL -	<i>Data Definition Language</i>
DSR -	<i>Design Science Research</i>
GSM -	<i>Global System for Mobile Communication</i>
GPS -	<i>Global Positioning System</i>
HTTP -	<i>HyperText Transfer Protocol</i>
HTTPS -	<i>Hypertext Transfer Protocol Secure</i>
IoT -	<i>Internet of Things</i>
M2M -	<i>Machine to Machine</i>
MQTT -	<i>Message Queuing Telemetry Transport</i>
NFC	<i>Near Field Communication</i>
QoS -	<i>Quality of Service</i>
ROI -	<i>Retorno do Investimento</i>
SDK -	<i>Software Development Kit</i>
SI -	<i>Sistemas de Informação</i>
SQL -	<i>Structure Query Language</i>
SSL -	<i>Secure Sockets Layer</i>
TCP -	<i>Transmission Control Protocol</i>
TI -	<i>Tecnologias de Informação</i>
TSM -	<i>Time-Structure Merge Tree</i>
TTL -	<i>Time to live</i>
UDP -	<i>User Datagram Protocol</i>

Resumo

Com a crescente divulgação de aspetos bastante atuais em tecnologias de informação, de onde se destacam termos como *Internet of Things (Iot)*, *SmartHome* e *SmartCity's*, muitas das empresas que fornecem soluções nessas áreas têm apresentado soluções aos seus clientes, que são maioritariamente relacionadas com a obtenção remota de informação. Existem, no entanto, algumas necessidades ditas mais “industriais” que ainda não têm soluções muito eficientes e que apresentam problemas em termos de aplicação operacional.

Este trabalho de investigação permitiu definir um modelo de arquitetura para um artefacto que permita dar solução a algumas necessidades encontradas em organizações e que possa ser controlado de forma remota, com base na informação recebida e fluxos de trabalho previamente estabelecidos, e com um elevado grau de automatização na execução de tarefas.

O processo de investigação segue uma abordagem metodológica indutiva, com uma perspetiva de investigação qualitativa, centrada num estudo de caso que serve de contexto para a elaboração de do artefacto. A investigação segue as diretrizes para pesquisas no âmbito dos Sistemas de Informação (SI) do método de investigação *Design Science Research (DSR)*.

A investigação procurou evidenciar o estado da arte, centrando-se no domínio das tecnologias de redes de comunicação e das bases de dados. Através da investigação foi possível definir as características de funcionalidade de um protótipo, sendo as respetivas necessidades validadas através de inquéritos e entrevistas a organizações com necessidades de monitorização remota de instalações e equipamentos.

O modelo de arquitetura proposto nesta investigação pode ser compreendido e utilizado por técnicos e gestores para conceber soluções que tirem partido do artefacto, podendo também proceder a implementações adaptadas às suas necessidades.

Palavras chave: monitorização remota, Internet das Coisas, *datalogger*, redes de comunicação.

Abstract

With the increasing spread of information technology, with new concepts such as Internet of Things (Iot), SmartHome and SmartCity's, many of the companies providing solutions in these areas have presented IT platforms to their customers, which are mostly related to remote retrieval of information. However, these platforms are not suited to some "industrial" in terms of efficiency and present some problems in terms of operational application.

This research work defines a model for an artifact with the aim to solve some needs found in organizations. The artifact can be controlled remotely, based on the information received and on previously established workflows, and a high degree of automation in task execution.

The research process follows an inductive methodological approach, with a qualitative research perspective, focused on a case study that serves as a context for the elaboration of the artifact. The research will follow the guidelines for research in the field of Information Systems (SI) of the method of research Design Science Research (DSR).

The research sought to highlight the state of the art, focusing on the domain of communication networks and databases. Through the investigation it was possible to define the functionality characteristics of a prototype, and the respective needs are validated through surveys and interviews with organizations with remote monitoring needs of facilities and equipment.

The architecture model proposed in this research can be understood and used by technicians and managers to design solutions that take advantage of the artifact, and can also evolve for implementations adapted to their needs.

Keywords: remote monitoring, Internet of Things, datalogger, communication networks.

Índice

Dedicatória	vii
Resumo	ix
Abstract	x
Índice.....	11
Índice de Figuras.....	13
Índice de Tabelas	15
1. Introdução	16
1.1. Problema e motivação.....	17
1.2. Tema	17
1.3. Enquadramento Organizacional.....	18
1.4. Problema e Objetivos de Investigação.....	19
1.5. Métodos de Investigação.....	20
2. Estado da Arte	28
2.1. Redes de Comunicação	28
2.2. Processo de Desenvolvimento de Software	40
2.3. Bases de Dados	41
3. Modelo.....	51
3.1. Descrição do Modelo	51
3.2. Hardware.....	52
3.3. Software	58
3.4. API.....	60
4. Arquitetura Tecnológica do Modelo.....	62
4.1. Plataforma	62
4.2. Máquinas de Estado dos Serviços associados à plataforma.....	70

4.3. Máquina de estados do dispositivo	72
5. Validação dos requisitos do modelo.....	90
5.1. Consulta a interessados	90
5.2. Resultados do questionário	92
6. Conclusões.....	95
6.1. Limitações da Investigação.....	95
6.2. Investigação Futura.....	96
7. Glossário.....	97
8. Bibliografia.....	99
Anexo I - API.....	106

Índice de Figuras

Figura 1- <i>Research Methods</i> for Business Students. (Saunders et al., 2009)	21
Figura 2- Atividades de Desenho Organizacional e de Desenho dos Sistemas de Informação. (Hevner et al., 2004)	23
Figura 3 - Estrutura do Sistemas de Informação do Conhecimento. (Hevner et al., 2004)	25
Figura 4 - As Sete Diretrizes para Design-Science em Pesquisas de SI. (Hevner et al., 2004)...	26
Figura 5- Comparativo entre camadas TCP/IP e IP IoT. (Rotta et al., 2017).....	29
Figura 6 - Publicação de Mensagens (MTQQ).....	36
Figura 7- Broker - Notificação de Desconexão Anormal (MTQQ).....	39
Figura 8 – Protótipo do Equipamento. (Documento Interno da Inosat, 2017)	52
Figura 9- Portas de Entrada Disponíveis no Dispositivo Remoto. (Documento Interno da Inosat, 2017).....	54
Figura 10- Interfaces do Dispositivo Remoto. (Documento Interno da Inosat, 2017)	55
Figura 11- Estrutura do MTQQ.	57
Figura 12 - API Utilização.....	60
Figura 13 - Plataforma - Processo de Configuração	62
Figura 14 - Plataforma - Mockup Autenticação	63
Figura 15 - Plataforma - Mockup Painel de Controlo	64
Figura 16- Plataforma - Mockup Dispositivos	65
Figura 17 - Plataforma - Mockup Dados por Dispositivo	66
Figura 18- Plataforma - Mockup Configurações por Dispositivo	67
Figura 19- Plataforma - Mockup Acesso aos Dados de Conta	68
Figura 20- Plataforma - Mockup Ajuda.....	69
Figura 21 - Máquina de Estados – Envio de Mensagens por SMS.....	70
Figura 22 - Máquina de Estados - Receção de Mensagens.....	71
Figura 23 - Máquina de Estados - Inicialização de Equipamento	72
Figura 24- Máquina de Estados - Ativação de Módulo e Ciclos de Geolocalização.....	73
Figura 25- Máquina de Estados - Ativação do Módulo e Ciclos de GSM	74
Figura 26- Máquina de Estados - Ativação do Módulo e Ciclos da Rede sem Fios	75
Figura 27- Máquina de Estados - Ativação do Módulo e Ciclos do Painel Solar	76
Figura 28- Máquina de Estados - Estado e Ciclos de Alimentação Externa do Dispositivo.....	77

Figura 29- Máquina de Estados – Validação de Carga da Bateria Interna	78
Figura 30- Máquina de Estados - Carregamento da Bateria Interna.....	79
Figura 31- Máquina de Estados - Ligação a Redes sem Fios	80
Figura 32- Máquina de Estados - Estado da Ligação GSM.....	81
Figura 33- Máquina de Estados - Módulo de GPS	82
Figura 34- Máquina de Estados - Estado Painel Solar.....	83
Figura 35- Máquina de Estados - Estado de Porta Digital.....	84
Figura 36- Máquina de Estados - Estado de Porta Analógica	85
Figura 37- Máquina de Estados - Estado da Ligação de Dados	86
Figura 38- Máquina de Estados - Tratamento de Comunicações Recebidas.....	87
Figura 39- Máquina de Estados - Tratamento Dados a Enviar.....	88
Figura 40- Máquina de Estados - Ativação Atuador	89
Figura 41 - API Serviço UserService	106
Figura 42 – API UserService – Método Authenticate	107
Figura 43 - API UserService - Alterar Email do Utilizador	108
Figura 44 - API - UserService - Alterar Número Telemóvel	109
Figura 45 - API UserService - Apagar Mensagem.....	110
Figura 46 - API UserService - Solicita Todas as Mensagens Disponíveis.....	111
Figura 47 - API UserService - Fim de Sessão	112
Figura 48- API UserService - Envia Mensagem	113
Figura 49 - API UserService - Atualizar o Endereço do Serviço PUSH.....	114
Figura 50 - API DeviceService - Lista de Métodos.....	115
Figura 51- API DeviceService - Desativar o Serviço PUSH.....	116
Figura 52- API DeviceService - Listar Todos os Dispositivos.....	117
Figura 53- API DeviceService - Ativar Serviço PUSH.....	118
Figura 54- API DeviceService - Pedir Configuração Atual	119
Figura 55- API DeviceService - Pedir Dados de um Dispositivo entre Datas	120
Figura 56- API DeviceService - Solicita última Informação do Dispositivo	121
Figura 57- API DeviceService - Envio de Configuração para o Dispositivo	122
Figura 58- API DeviceService - Alteração da Configuração de um Dispositivo	123

Índice de Tabelas

Tabela 1 - Formas de Resolução do Problema - Vantagens e Desvantagens	19
Tabela 2 - Comparação entre M2M e IoT.....	31
Tabela 3 - Mensagens de Controlo do Protocolo.....	37
Tabela 4 - Comparação de Protocolos	40
Tabela 5 - Comparativo de Motores de Bases de Dados	50
Tabela 6 - Formas de Resolução do Problema - Vantagens e Desvantagens	92

1. Introdução

Com a crescente divulgação de aspetos bastante atuais em tecnologias de informação, de onde se destacam termos como *Internet of Things (Iot)*, *SmartHome* e *SmartCity's*, muitas das empresas que fornecem soluções nessas áreas têm apresentado soluções aos seus clientes, que são maioritariamente relacionadas com a obtenção remota de informação.

Existem, no entanto, algumas necessidades ditas mais “industriais” que ainda não têm soluções muito eficientes e que apresentam problemas em termos de aplicação operacional. Por outro lado, o aumento da competitividade das empresas, associado à abertura das fronteiras, leva a que as empresas procurem soluções tecnológicas que possam reduzir os custos de aplicação dos seus produtos de forma significativa, mantendo-se assim mais competitivas no mercado em que estão inseridas.

Este trabalho tem como objetivo definir um modelo de arquitetura para um artefacto que permita dar solução a algumas necessidades encontradas em organizações e que possa ser controlado de forma remota, com base na informação recebida e fluxos de trabalho previamente estabelecidos, e com um elevado grau de automatização na execução de tarefas.

O artefacto é composto por duas componentes principais. Uma primeira relativa a um *datalogger*, que permite com a instalação de periféricos recolher informação para uma plataforma central. Uma segunda relativa à plataforma, onde será permitido ao utilizador configurar fluxos de trabalho com base na validação de condições previamente estabelecidas e que permanentemente são aplicadas aos novos dados recebidos e tratados nessa plataforma central.

Esta situação permite criar alertas úteis em diferentes áreas de aplicação industrial e de serviços, com base no tratamento de informação recebida periodicamente, sobre o que se está a monitorizar. Com esta solução possibilita-se o controlo remoto e permanente de situações que normalmente exigiriam a utilização de recursos humanos para o efeito. Por outro lado, com esse controlo, garante a possibilidade de agir proactivamente a eventos de risco, evitando dessa forma prejuízos desnecessários.

1.1. Problema e motivação

Em muitos dos sectores de atividade existe a necessidade de controlar remotamente situações inerentes à atividade da organização. Em muitos destes casos, a forma atual de o fazer é ter recursos humanos que periodicamente se deslocam aos locais a controlar e validam localmente o estado de recursos sensíveis, não permitindo a sua monitorização contínua e com custos significativos que limitam a organização e o respetivo nível de serviço prestado.

Os dispositivos de controlo remoto disponíveis no mercado são maioritariamente dedicados a aplicações de monitorização de veículos e apresentam limitações em termos de funcionalidade, autonomia e custo de implementação, nomeadamente por exemplo em aplicações com muitos recursos a monitorizar, onde é necessária a utilização de sensores de baixo consumo e onde é necessária a comunicação imediata de informação obtidas a partir dos sensores.

Esta comunicação permanente é especialmente crítica para empresas que têm a necessidade de controlar remotamente situações que de uma forma direta ou indiretamente estão associadas à continuidade e desempenho seu negócio e cuja falha de alguma forma poderá colocar em risco o mesmo.

As limitações apresentadas pelos equipamentos existentes podem ser ultrapassadas através da alteração de tecnologia já existente, nomeadamente com desenvolvimento de um *datalogger*, que permita a instalação de periféricos e a recolha de informação para uma plataforma central que permita a monitorização de equipamentos e instalações de forma otimizada.

1.2. Tema

A alarmística é normalmente um conceito associado à monitorização, em tempo real, de locais remotos, onde o principal objetivo é a obtenção de informação periódica que permita executar fluxos de trabalho, com base na informação trabalhada a partir dos dados recebidos.

1.3. Enquadramento Organizacional

Este trabalho incide sobre as necessidades de algumas organizações, que embora de setores de atividade distintos, partilham as mesmas dificuldades e preocupações, ou seja, a monitorização e atuação remota, com base em variados tipos de informação. Empresas, com instalações dispersas geograficamente, mas como necessidades específicas de controlo de informação, derivada da atividade onde se encontram inseridas, pretendem uma solução de baixo custo de operação, que lhes permita ativar fluxos de trabalho com base na informação recebida remotamente. Alguns exemplos de necessidades de controlo remoto e permanente de recursos os negócios seguintes:

- Cadeia de talhos com armazéns de frio que necessita de monitorizar remotamente as temperaturas dos mesmos 24h/7 e atuar de imediato em caso de oscilações de temperatura que ultrapasse o intervalo de temperaturas permitidas;
- Empresa de serviços na comercialização de energia elétrica com postos de transformação de baixa, média e alta tensão, que necessita de controlar acessos não autorizados aos mesmos;
- Empresa de prestação de serviços, na área das telecomunicações, que necessita de controlar remotamente acessos indevidos a locais não autorizados com o intuito de furtar cobre e que pode colocar em causa o serviço prestado pela empresa;
- Empresa na área dos transportes com a necessidade controlar avarias remotamente em equipamentos ferroviários, que podem colocar em causa a correta prestação de serviço, sendo que em alguns casos as avarias não detetadas atempadamente podem colocar em risco a integridade física dos seus colaboradores, parceiros de negócio e clientes;
- A disponibilização de imagens relativos a locais remotos após deteção de movimentos não autorizados.

Neste momento, segundo a pesquisa que foi efetuada ao mercado, não existe uma solução que permita remotamente trabalhar os dados recebidos e ativar fluxos de trabalho, com base em condições aplicadas à análise da informação, análise essa, proveniente do tratamento dos dados enviados remotamente.

1.4. Problema e Objetivos de Investigação

O problema a resolver com o artefacto, centra-se na necessidade de manter sob monitorização um conjunto de recursos da organização, que geralmente são críticos para as suas operações. A criticidade desses elementos leva a que as organizações tenham obrigatoriamente de proceder à sua verificação sistemática, podendo para isso recorrer a uma solução baseada exclusivamente em monitorização humana ou podendo recorrer a uma solução tecnológica como se pretende com o artefacto em estudo. Em reunião de peritos da Inosat foram analisadas as vantagens e desvantagens das várias alternativas atualmente usadas pelas organizações, para várias alternativas de solução, com vista a sustentar a necessidade da solução (tabela 1).

Solução para colmatar a necessidade	Vantagens	Desvantagens
Os colaboradores deslocam-se diariamente aos locais por forma a monitorizar a informação necessária.	- O colaborador está mais familiarizado com o local e deteta mais facilmente situações anómalas. Pode atuar de imediato visto que se encontra junto da instalação.	Não consegue monitorizar 24h/7 dias por semana. No caso das temperaturas, o momento tardio da deteção pode originar elevados prejuízos materiais. Para monitorizar com mais frequência serão necessários mais meios humanos com o maior custo associado.
Subcontratar empresas externas para supervisionar os locais.	- Passam o ónus da responsabilidade para a empresa externa. Será a empresa a mobilizar os elementos necessários para abranger a totalidade do dia. Em caso de falha será a empresa externa a suportar os custos.	A transferência total do risco para terceiros é contratualmente difícil de conseguir. Não consegue monitorizar 24h/7 dias por semana. No caso das temperaturas, o momento tardio da deteção pode originar elevados prejuízos materiais. Não conhecem tão bem o local. Tem custos mais elevados de monitorização.
Não fazer nada em termos de supervisão	Suportar o risco de poder ter um incidente com custos elevados através de outro meio mais simples (ex, seguro). Não têm custos de monitorização.	A transferência total do risco para terceiros é contratualmente difícil. Podem ter custos elevados associado aos eventos (ex. perda de instalações, falha nos níveis de serviço) não monitorizados que podem em última análise levar ao encerramento da empresa.
Instalar equipamento de monitorização remota da atividade 24h/7	Monitorização em tempo real 365 dias por ano. Alertas em tempo real por mensagem de texto, email, API. Aumenta o tempo de reação. Diminui eventuais danos e / ou prejuízos.	- Em caso de falha elétrica, as notificações resultantes de monitorização e atuação poderão ser mais limitadas.

Tabela 1 - Formas de Resolução do Problema - Vantagens e Desvantagens

Verifica-se que as desvantagens das soluções não automatizadas são significativas, representando uma oportunidade para a criação de um artefacto que permita uma solução tecnológica para monitorização contínua 24 horas, 7 dias da semana das instalações

Para resolver o problema proposto define-se como objetivo de investigação:

- A elaboração de um modelo de arquitetura para um artefacto, que permita a monitorização remota de instalações e equipamentos fixos e que permita a adição de funcionalidade, de forma a adaptar-se às suas necessidades específicas;

1.5. Métodos de Investigação

A realização de um estudo requer a definição do respetivo processo de investigação (Saunders, Lewis e Thornhil, 2009). O tipo de investigação a realizar depende igualmente da perceção do investigador sobre o desenvolvimento do conhecimento. O diagrama da figura 1 identifica diferentes opções que caracterizam o processo de investigação, inseridas em camadas sucessivas respetivamente e por esta ordem: filosofia, abordagem, estratégia, tipo de investigação, horizonte temporal e técnica de recolha de dados. O investigador inicia a sua escolha definindo a filosofia e assim sucessivamente até definir elementos do processo de investigação necessários para responder à questão de investigação.

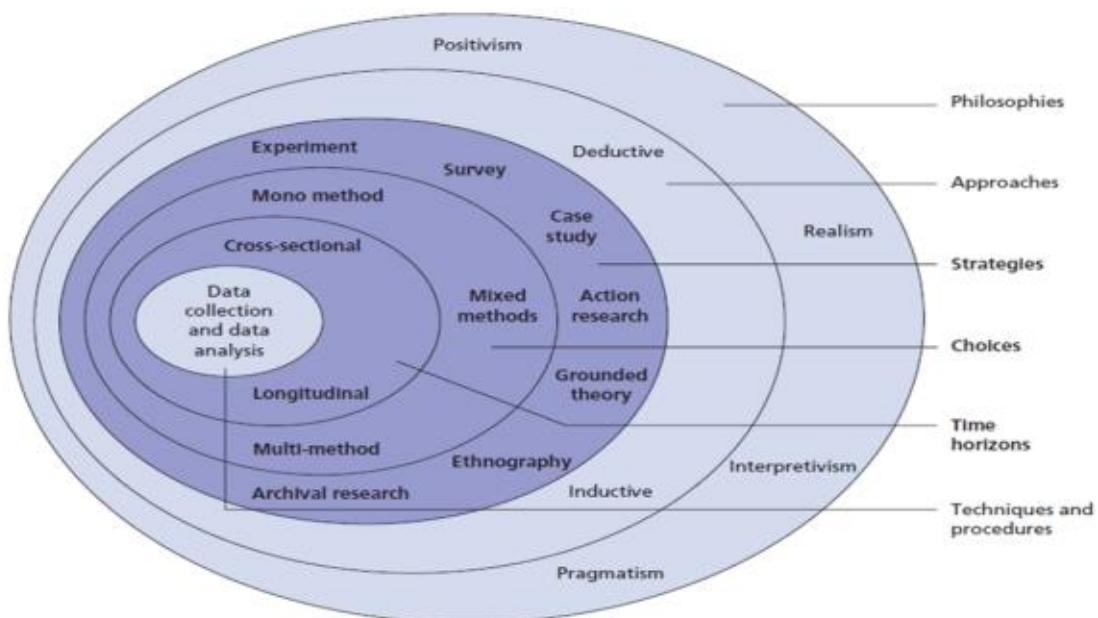


Figura 1- *Research Methods* for Business Students. (Saunders et al., 2009)

1.5.1. Filosofia de Investigação

A filosofia de investigação escolhida é o realismo crítico, segundo a qual um estudo deve ser desenvolvido a múltiplos níveis, podendo alterar a própria perceção que o investigador tem sobre o objeto em estudo. A observação do fenómeno e do artefacto são realizadas simultaneamente, no contexto específico da organização, permitindo a interação dinâmica de várias estruturas, processos e procedimentos (Saunders et al., 2009).

1.5.2. Abordagem Metodológica

A abordagem metodológica é indutiva, consiste na recolha e análise de dados e correspondente formulação de teorias, procurando a compreensão do contexto onde ocorrem os factos em estudo (Saunders et al., 2009), considerando-se que a investigação não requer a generalização das suas conclusões, mas sim aprofundamento o conhecimento sobre um problema num contexto específico.

1.5.3. Perspetiva de investigação

A investigação segue uma qualitativa, direcionada para a observação de fenómenos, a partir da forma como os indivíduos e organizações percecionam a realidade, de forma a compreender as suas necessidades inerentes ao seu contexto de negócio.

1.5.4. Tipo de investigação

O tipo de investigação é o estudo de caso (Yin, 2014). O estudo de caso é um tipo de investigação baseado principalmente no trabalho de campo, estudando uma instituição na sua realidade, através de observações, entrevistas, questionários, documentos e artefactos. Os métodos utilizados neste estudo são essencialmente a análise documental e o questionário.

1.5.5. Horizonte Temporal de Investigação

O horizonte temporal de investigação é *cross-sectional*, na medida em que se examina a realidade num período temporal restrito, relativo às necessidades inerentes ao problema de investigação no momento atual.

1.5.6. Técnica de Análise de Dados

Tesch (2013) propõem três formas distintas de análise dos dados: interpretativa – organizando e classificando todos os dados recolhidos segundo categorias que permitam explorar e explicar o fenómeno a investigar; estrutural – procurando encontrar padrões nos dados, que possam ajudar a explicar ou clarificar um determinado fenómeno, ou situação a investigar; reflexiva - avaliando e interpretando o fenómeno a ser estudado, através da perceção, julgamento ou intuição do investigador. Neste trabalho é utilizada uma análise de dados essencialmente interpretativa e reflexiva, com vista a agrupar as necessidades em categorias/funções necessárias às organizações, obtidas através de ferramentas de investigação como questionários e reuniões de peritos.

De acordo com o Simon (1996), a ciência do design suporta um paradigma de pesquisa que por si só exige a conceptualização de artefactos inovadores para resolver problemas do mundo real. Segundo Hevner, Ram, March e Park (2004) “O bom design é uma atitude renascentista que combina tecnologia, ciência cognitiva, necessidade humana e beleza para produzir algo que o mundo não sabia que estava faltando”.

Estes autores propõe um método de investigação designado *Design Science Research* (DSR), que apresenta um conjunto de diretrizes para pesquisas no âmbito dos Sistemas de Informação (SI), considerando-se que pode ser utilizado nesta investigação, a qual requer a criação de um artefacto inovador para o problema detetado (Hevner et al., 2004).

Segundo Hevner (Hevner et al., 2004), a estratégia das Tecnologias de Informação (TI) de qualquer empresa deve estar alinhada com a estratégia do negócio da empresa. Da mesma forma,

que a infraestrutura dos sistemas de informação deve estar alinhada com a infraestrutura da organização, em que todos as suas atividades. A figura seguinte ilustra os alinhamentos essenciais entre as estratégias de negócios e de tecnologia da informação e entre as infraestruturas organizacionais e de sistemas de informação, tais como referenciados anteriormente.

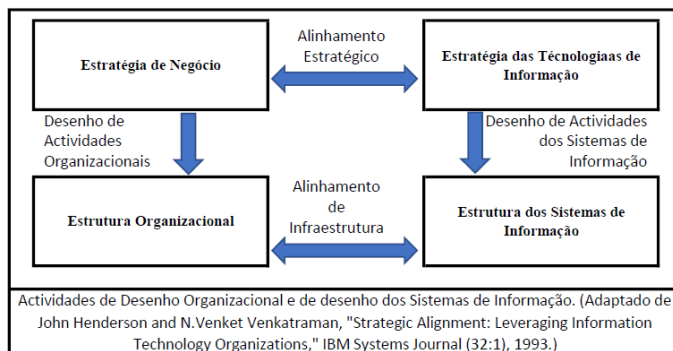


Figura 2- Atividades de Desenho Organizacional e de Desenho dos Sistemas de Informação. (Hevner et al., 2004)

Neste modelo, o desenho das atividades da estrutura organizacional nasce da estratégia de negócio, e o desenho das atividades da estrutura dos sistemas de informação deriva da estratégia de tecnologias de informação. Essas são atividades de design interdependentes que são centrais para a disciplina de SI. Portanto, a pesquisa de SI deve abordar a interação entre estratégia de negócios, estratégia de TI, infraestrutura organizacional e infraestrutura de SI.

Essa interação é tanto mais crucial quando mais importantes forem as tecnologias da informação como facilitadoras da estratégia de negócios e da infraestrutura organizacional (Kalakota e Robinson, 2001; Orlikowski e Barley, 2001). Recursos de Tecnologias de Informação (TI) disponíveis e emergentes são assim um fator significativo na determinação das estratégias que guiam uma organização.

O processo de design é uma sequência de atividades especializadas que produz um produto inovador (ou seja, o artefacto de design). A avaliação do artefacto fornece informações de feedback e um melhor entendimento do problema, a fim de melhorar a qualidade do produto e o processo de design. As atividades de design são endêmicas em muitas profissões. Em particular,

a profissão de engenharia produziu uma considerável literatura sobre design (Dym, 1994; Pahl e Beitz, 1996; Petroski, 1996).

No âmbito dos SI, muitas atividades de design foram extensivamente estudadas, formalizadas tornando-se rotineiras. A pesquisa em ciência do design em SI aborda os que são considerados problemas iníquos (Brooks, 1987, 1996; Rittel e Webber, 1984). Ou seja, aqueles problemas caracterizados por:

- Requisitos instáveis e restrições baseadas em contextos ambientais mal definidos;
- Interações complexas entre subcomponentes do problema e sua solução;
- Flexibilidade inerente para alterar processos de design, bem como artefactos de design (ou seja, processos maleáveis e artefactos);
- Uma dependência crítica das habilidades cognitivas humanas (por exemplo, criatividade) para produzir soluções eficazes;
- Uma dependência crítica das habilidades sociais humanas (por exemplo, trabalho em equipe) para produzir soluções eficazes.

De acordo com Simon (1996), uma teoria do design em sistemas de informação está, por necessidade, em constante estado de revolução científica (Kuhn, 1996). Os avanços tecnológicos são o resultado de processos inovadores e criativos da ciência do design. Se não forem caprichosos, são pelo menos arbitrários (Brooks, 1987) relativamente às necessidades de negócio e ao conhecimento existente.

Inovações, como sistemas de gestão de bases de dados, linguagens de alto nível, computadores pessoais, componentes de software, agentes inteligentes, tecnologia de objetos, *Internet* e *World Wide Web*, tiveram impactos dramáticos e, por vezes, não intencionais, na forma como os sistemas de informação são projetados, concebidos, implementados e geridos.

A pesquisa em SI, abrange pessoas, organizações e respetivas tecnologias existentes ou planeadas (Silver et al., 1995), incluindo os objetivos, tarefas, problemas e oportunidades que definem as necessidades percebidas de negócio. Tais perceções são moldadas pelos papéis, capacidades e características das pessoas dentro da organização.

As necessidades de negócios são avaliadas dentro do contexto de estratégias organizacionais, estrutura, cultura e processos de negócios existentes, tendo em conta as pessoas e respetivas regras, competências e características, assim como a tecnologia disponível, conduzindo ao desenvolvimento e avaliação de teorias e artefactos, com base no conhecimento aplicável e respetivas fundamentações e metodologias (Figura 3).

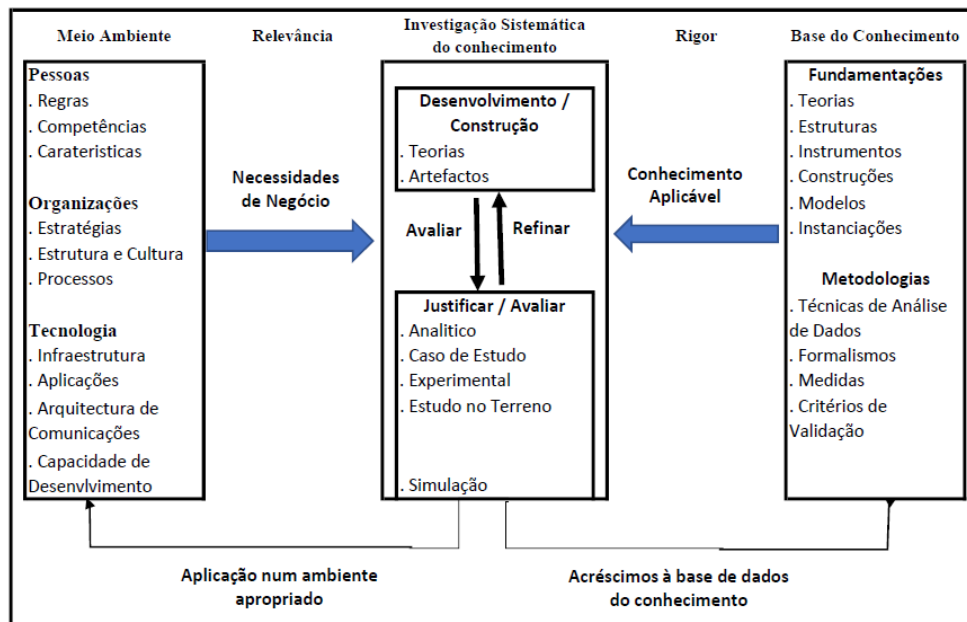


Figura 3 - Estrutura do Sistema de Informação do Conhecimento. (Hevner et al., 2004)

O artefacto deverá ter uma avaliação, no sentido de garantir a utilidade do mesmo. Por forma a colaborar positivamente, em termos de pesquisa e contribuição, o artefacto deve resolver um problema nunca antes resolvido ou a fornecer uma solução que seja mais efetiva que as soluções até ao momento apresentadas. Em termos de construção e avaliação, o artefacto deve ser realizado de forma rigorosa e os resultados da pesquisa devem ser apresentados de forma eficaz, tanto para os destinatários orientados à gestão, como para os destinatários orientados às tecnologias de informação.

Os modelos usam construções para representar uma situação do mundo real, o problema de design e seu espaço de solução (Simon, 1996). Os modelos auxiliam no entendimento do

problema e da solução e frequentemente representam a relação entre o problema e os componentes da solução, permitindo a exploração dos efeitos das decisões de projeto e das mudanças no mundo real. Segundo Hevner et al. (2004), existem sete diretrizes para uma pesquisa científica de design (Figura 4).

Diretrizes de Pesquisa em Design-Science	
Diretriz	Descrição
Diretriz 1: Desenhar / projetar como um artefacto	A pesquisa em <i>Design-Science</i> deve produzir um artefato viável na forma de um construto, um modelo, um método ou uma instanciação.
Diretriz 2: Relevância do problema	O objetivo da pesquisa em <i>Design-Science</i> é desenvolver soluções baseadas em tecnologia para problemas de negócios importantes e relevantes.
Diretriz 3: Avaliação do projecto / desenho	A utilidade, a qualidade e a eficácia de um artefato de design devem ser rigorosamente demonstradas por meio de métodos de avaliação bem executados.
Diretriz 4: Contribuições da pesquisa	A pesquisa eficaz em <i>Design-Science</i> deve fornecer contribuições claras e verificáveis nas áreas do artefato de design, fundamentos de design e / ou metodologias de projeto.
Diretriz 5: Rigor da pesquisa	A pesquisa em <i>Design-Science</i> depende da aplicação de métodos rigorosos tanto na construção quanto na avaliação do artefato de design.
Diretriz 6: Desenho como projecto de pesquisa	A busca por um artefato efetivo requer a utilização dos meios disponíveis para alcançar os fins desejados enquanto satisfaz as leis do ambiente problemático.
Diretriz 7: Comunicação da Pesquisa	A pesquisa em <i>Design-Science</i> deve ser apresentada de forma eficaz tanto para o público orientado para a tecnologia quanto para o público orientado para a gestão.

Figura 4 - As Sete Diretrizes para *Design-Science* em Pesquisas de SI. (Hevner et al., 2004)

Diretriz 1: Desenhar/projetar como um artefacto - o resultado da pesquisa em ciência do design em SI é, por definição, um artefacto de TI viável criado para resolver um problema organizacional importante. Deve ser descrito de forma eficaz, permitindo a sua implementação e aplicação em um domínio apropriado.

Diretriz 2: Relevância do Problema - o objetivo da pesquisa em sistemas de informação é adquirir conhecimento e compreensão que permitam o desenvolvimento e a implementação de soluções baseadas em tecnologia para problemas empresariais ainda não resolvidos e importantes.

Diretriz 3: Avaliação de Projeto - A avaliação é um componente crucial do processo de pesquisa. O estudo do artefacto recorre a uma análise observacional, estudando a utilidade do artefacto no ambiente de negócios (caso de estudo) sendo também utilizado o método analítico examinando o artefacto relativamente à qualidades estáticas e arquitetura técnica de SI.

Diretriz 4: Contribuições de pesquisa - a pesquisa eficaz em *design science* deve fornecer contribuições clara, nomeadamente conhecimento relativo à construção de projeto (ou seja, fundamentos) e/ou conhecimento de avaliação de projeto (isto é, metodologias). Procura-se neste estudo contribuir através do próprio desenho do artefacto, apresentando uma solução, assim como o seu fundamento com o modelo de funcionamento proposto.

Diretriz 5: Validação do modelo – Neste estudo parte-se de um artefacto já testado em ambiente empresarial e são propostas alterações de funcionalidade cuja utilidade é verificada através de inquéritos a um conjunto de organizações com características operacionais que os tornam possíveis interessados.

Diretriz 6: Design como um processo de pesquisa - o design é essencialmente um processo de investigação para validar uma solução eficaz para um problema. A resolução de problemas pode ser vista como a utilização dos meios disponíveis para alcançar os fins desejados enquanto satisfaz as leis existentes no ambiente (Simon, 1996).

Diretriz 7: Comunicação da investigação- A investigação em ciência do design deve ser apresentada tanto para o público orientado para a tecnologia quanto para o público orientado para a gestão. Os públicos orientados para a tecnologia precisam de detalhes suficientes para permitir que o artefacto descrito seja construído (implementado) e usado dentro de um contexto organizacional apropriado. Isso permite que os profissionais aproveitem os benefícios oferecidos pelo artefacto e permite que os pesquisadores construam uma base de conhecimento cumulativa para maior extensão e avaliação.

2. Estado da Arte

2.1. Redes de Comunicação

A Internet das Coisas, mais conhecida por Internet of Things (IoT), é um paradigma tecnológico que tem como objetivo principal a interoperabilidade de equipamentos eletrónicos, promovendo a comunicação entre si, equipamentos esses que podem ser smartphones, sensores, atuadores entre outros. O objetivo principal é a interação entre equipamentos com objetivos comuns (Atzori et al., 2010) ou seja, onde a internet esteja presente existirá a possibilidade de ter equipamentos passíveis de serem controlados remotamente e executarem funcionalidades cujo objetivo será comum.

Pretende-se que o IoT seja uma das bases para mudar a forma como vivemos e trabalhamos, seja num âmbito pessoal como profissional. Na área doméstica, este conceito/tecnologia poderá estar na génese das aplicações em casas automatizadas, *e-health* e Enhanced Learning. (Gai e Qiu, 2018). Em ambiente de trabalho, os benefícios surgem na área de automação de tarefas industriais, transporte de pessoas e bens, assim como na vigilância, entre outras.

O Termo “Internet das coisas” é usado para se referir (Nayyar, 2017):

- (i) Rede global de objetos inteligentes interligados entre si por meio de tecnologias de Internet;
- (ii) O Conjunto de Suporte necessárias para isso, isto é, RFID, Sensores, entre dispositivos de comunicação de máquinas e;
- (iii) O Conjunto de Aplicativos e serviços que aproveitam essas tecnologias para abrir novas oportunidades de negócios e marketing.

2.1.1. Protocolos do IoT

Como modelo de referência será utilizado o TCP/IP, usado na maior parte das aplicações associadas ao IoT. De referenciar que não é necessário utilizar exclusivamente este protocolo, poderão ser utilizados outros tais como o UDP. A única razão porque foi eleito o TCP/IP é

devido ao facto de não ser necessário a validação da entrega dos pacotes de informação nos destinatários, sendo essa função parte integrante do protocolo.

Como é explicado na tabela seguinte, o modelo é dividido em 4 camadas (Rede – Internet – Transporte e Aplicação) (Tanenbaum, A, 2010).

Camadas	Internet TCP/IP	Ip IoT
Aplicação	HTTP/HTTPS/FTP/SSH/etc..	CoAP, MQTT
Transporte	TCP/UDP	UDP
Internet	IPv4/IPv6	6LoWPAN
Rede	IEEE 803.2 Ethernet / 802.11 Wifi	IEEE 802.11 / 802.15

Figura 5- Comparativo entre camadas TCP/IP e IP IoT. (Rotta et al., 2017)

2.1.2. Camadas do IoT

As camadas do IoT consistem:

- **Camada de Rede** - a camada de rede, que nós designamos por física, corresponde às tecnologias que usamos para a troca de pacotes de informação. No caso do IoT, a camada de rede que por norma utilizamos é baseada em comunicações em redes sem fios (Wi-Fi). Foram especificados para esta camada, diversos padrões tais como IEEE 802.11, 802.15 (Sutaria e Govindachari, 2010);
- **Camada Internet** - a camada internet tem como tarefa a entrega de pacotes onde foram necessários (Tanenbaum, 2010). A ideia desta camada é a gestão dos endereçamentos de rota por forma a entregar os mesmos nos destinatários sem que cause congestionamentos na rede;
- **Camada Transporte** - No modelo TCP/IP, a camada transporte está acima da camada Inter-Redes que tem como finalidade permitir que as entidades possam comunicar desde os *hosts* de origem aos *hosts* de destino sem interrupções (Tanenbaum, 2010);

- **Camada Aplicação** - são protocolos de serviços específicos de aplicações para promover comunicação de dados ao nível processo-a-processo. O protocolo de comunicação para serviços Web é o HTTP / HTTPS, no entanto, o mesmo corresponde a um nível energético muito elevado para os dispositivos IoT.

2.1.3. Seleção do Protocolo de Comunicação

O protocolo de comunicação é a forma como o equipamento irá comunicar com o servidor / serviço e vice-versa. Este protocolo deverá incluir a possibilidade de ser iniciado tanto pelo equipamento como pela plataforma e terá como missão transmitir os dados recolhidos pelo dispositivo e garantir a sua entrega no servidor ou guarda-los em memória caso se encontre *off-line* e posterior transmissão ao servidor assim que o canal de comunicação esteja disponível.

A seleção do protocolo de comunicação tem como principal objetivo a seleção do protocolo que mais vantagens tragam ao artefacto, tendo em conta vários pontos fundamentais tais como:

- Segurança;
- Fiabilidade;
- Taxa de transferência de dados;
- Latência;
- Custo da implementação na conceção física do dispositivo.

Os protocolos de comunicação disponíveis para implementar nos sistemas de IoT são bastante diversificados. Os responsáveis pelo desenvolvimento de novos dispositivos eletrónicos vocacionados para o IoT tem uma gama vasta de opções disponíveis no mercado que podem incorporar nos seus projetos. Alguns dos protocolos mais clássicos tais como o Wi-Fi e Bluetooth são bastante conhecidos na comunidade de fabricantes e utilizadores.

No entanto, tem-se vindo a verificar algumas alternativas emergentes tais como Thread e Z-Wave, ambas focadas na automação residencial, mas menos difundidas que as mais conhecidas. Com a expansão do conceito de IoT, começam a emergir uma serie de protocolos de

comunicação que embora não sejam tão conhecidos e utilizados, são opções disponíveis e de certa forma mais vocacionados para os conceitos IoT e *Machine to Machine* (M2M).

Com o aparecimento do IoT verificou-se a necessidade de adaptar os protocolos existentes ou mesmo criar novos protocolos que venham ao encontro das necessidades de interoperabilidade e comunicação. A necessidade de criação de novos protocolos, surgiu como solução para a grande expansão da IoT e dos respetivos *middlewares*, que gerou diversos desafios de interoperabilidade e comunicação (Rotta et al., 2017). A figura seguinte compara os conceitos IOT e o M2M, focando-se nas diferenças (tabela 2).

M2M	IoT
Comunicação ponto a ponto habitualmente integrada no hardware do lado do cliente.	Os dispositivos comunicam utilizando redes sobre IP incorporando vários protocolos de comunicação.
Muitos dispositivos usam redes celulares ou sem fios.	Os dados são transmitidos através de uma camada intermédia hospedada na nuvem.
Os dispositivos não têm uma real dependência de uma ligação à internet.	Na maior parte dos casos, os dispositivos requerem uma ligação à internet ativa
Opções de integração limitadas devido ao facto de os dispositivos terem de respeitar os padrões de comunicação estabelecidos.	Número ilimitado de possibilidades de integração, mas que requer soluções de implementação de sistemas de comunicação.

Tabela 2 - Comparação entre M2M e IoT

n.d., Sítio: <https://www.incognito.com/iot-and-m2m-whats-the-difference/> Ultima visita:27/09/2018

O mais importante a reter, é que tanto o M2M como o IoT referem-se a dispositivos que comunicam entre si, sendo que, M2M refere-se a instâncias de comunicações isoladas de dispositivo a dispositivo e o IoT refere-se a uma escala maior, onde existe uma sinergia entre programas verticais que permitem a gestão e automatização das comunicações e os vários dispositivos.

Os operadores que queiram fornecer suporte à IoT devem ter um posicionamento o mais independente possível dos fornecedores, tendo a possibilidade de gerir várias ligações a partir de

vários protocolos de comunicação, para que os utilizadores/subscritores do serviço possam aproveitar todos os novos dispositivos que se enquadram no conceito de *smart-home*. Nestes casos, existem algumas dificuldades na gestão de comunicações tão variadas, mas desde que haja uma comunicação IP com um software que permita essa mesma gestão permitindo que os dispositivos comuniquem entre si.

O protocolo utilizado neste estudo irá transmitir pacotes de dados cuja dimensão é muito reduzida e onde a rapidez na transmissão da informação terá grande relevância. Quanto ao protocolo poderemos optar por seleccionar um protocolo já existente e bastante utilizado, garantindo dessa forma que o mesmo se encontra testado ou criar um protocolo de raiz tendo em conta todos os inconvenientes que possam daí advir.

Nos pontos abaixo são apresentadas algumas das alternativas disponíveis de forma a avaliar qual delas será mais vantajosa (Vidal, 2017):

- **LoRaWAN** - mais vocacionado para aplicações em redes WAN, o LoRa foi especificado para promover comunicações de baixo consumo energético (como qualquer protocolo para IoT) e oferecer funcionalidades específicas na área de segurança em comunicações M2M. Desenhado e mantido pelo Lora Alliance, este protocolo suporta redes de grandes dimensões, com milhões de dispositivos em simultâneo e possui velocidades de transmissão entre 0.3 Kbps e 50 Kbps. É um dos protocolos mais populares;
- **Neul** - o Protocolo Neul baseia-se num conceito similar ao do protocolo SigFox. É baseado num chip específico “Iceni”, desenvolvido pela empresa que criou o Neul. A tecnologia base está desenhada com base nas redes WAN e foi criada para competir com as redes já existentes GPRS, 3G, CDMA e LTE WAN. A taxa de transferência de dados vai desde o bit por segundo até ao 100Kbps. Tem um alcance de 10Km e consumos na ordem dos 20 a 30 mA;

- **SigFox** - a Sigfox é uma alternativa entre as redes WIFI e as redes de longo alcance, tais como as redes celulares (3G, 4G, etc). A banda utilizada é a ISM (*Industrial, scientific and medical band*). A ideia que originou o aparecimento desta rede foi a necessidade que muitas das aplicações M2M a funcionar em equipamentos com bateria onde os níveis energéticos são baixos ou pouco potentes e que existe a necessidade de transmissão de dados, onde o alcance do WIFI é muito curto e em paralelo o custo de transmissão de dados em redes celulares são muito caras e com elevados consumos energéticos. A tecnologia base do Sigfox é a UNB (*UltraNarrow Band*) e foi idealizada apenas para níveis de transferência de dados muito baixa, entre os 10 bit/ e 1 Kbit/s. O Alcance desta rede é de 30-50Km. Por curiosidade, a Sigfox também é o nome da empresa responsável pelo desenvolvimento do protocolo em 2009;
- **NFC** - a NFC (*Near Field Communication*) é também um protocolo bem conhecido atualmente pela sua exponenciação de penetração no mercado de novos dispositivos IoT. É uma tecnologia para troca de informações entre equipamentos. Basicamente é uma extensão da tecnologia RFID que permite aos equipamentos trocarem informações entre si desde que se enquadrem numa distância máxima de poucos centímetros. É uma tecnologia que tem vindo a equipar essencialmente *Smartphones* pois tem a capacidade de troca de informação entre os dispositivos sem a necessidade de intervenção. O alcance geral do protocolo é de 10cm e com taxas de transferências de 100 a 420Kbps. O padrão NFC foi estabelecido com base nas normas ISO/IEC 18000-3;
- **Wi-Fi** – o Wi-Fi é dos protocolos mais conhecidos dada a sua taxa de penetração no mercado, tanto ao nível pessoal como empresarial. Devido ao seu sucesso é uma opção óbvia para muitos dos fabricantes de soluções IoT. É um protocolo LAN e tem capacidade de gerir grandes quantidades de dados com taxas de transmissão bastante altas. Dentro dos padrões mais utilizados estão o 802.11ac e 802.11n, que oferecem taxas de velocidade de transmissão na ordem das centenas de megabit por segundo chegando a atingir os 1300 Mbps com o 802.11ac. Em contrapartida, ambos têm consumos energéticos muito altos o que poderá ser um fator impeditivo para algumas aplicações

IoT. O Alcance das redes Wi-Fi é de sensivelmente 50m, com a última versão 802.11-ac a oferecer taxas entre os 500Mbps e 1Gbps;

- **Thread** - desenvolvido essencialmente para o nicho de automação residencial, o Thread é baseado noutro protocolo, o 6LowPan (explicado abaixo), como tal é bem diferente dos tradicionais protocolos Bluetooth e Zigbee. Do ponto de vista aplicacional, é um protocolo complementar do WI-FI para sistemas residenciais;
- **6LowPAN** - o 6LowPAN é um protocolo IP (Internet Protocol). O nome é a abreviatura de IPv6 Low-power wireless *Personal Area Network*. Em vez de uma tecnologia Bluetooth ou ZigBee, o 6LowPAN é um protocolo que define o encapsulamento, *headers* e mecanismos de compressão. O Atributo chave é o *IPV6 Stack*, que foi um passo muito importante para viabilizar o IoT. Com o IPV6, é possível atribuir cada objeto ou equipamento do mundo, o seu IP único e liga-lo à Internet;
- **Z-Wave** - o Z-Wave também foi desenhado para automação residencial e produtos domésticos. Possui taxas de transmissão de até 100Kbps. Permite a implementação de redes mesh com um limite máximo de 232 nós. Uma das suas vantagens principais é que é um protocolo de fácil implementação, fator que se revela importante no processo de integração;
- **ZigBee** - o ZigBee é um dos protocolos mais usados em ambientes empresariais do que propriamente em ambientes residenciais. É baseado no padrão IEEE802.15.4, que é essencialmente um padrão a seguir nas redes wireless industriais na faixa dos 2.4Ghz. As aplicações que utilizam este protocolo, por norma, não requerem mudanças constantes na taxa de transmissão de dados. Todos os principais fabricantes de semicondutores têm módulos ZigBee no seu portefólio de produtos. O alcance deste protocolo é de 10 a 100 metros. A taxa de transmissão alcança no máximo 250Kbps.
- **Bluetooth** - o Bluetooth, outro protocolo também bastante conhecido e utilizado, já teve várias versões lançadas, mas foi na versão 4.0 (Bluetooth *low energy*) e na mais recente

versão 5 que o foco se concentrou no IoT. O Protocolo é mantido para Bluetooth SIG e possui bastante documentação, exemplos de aplicações, o que facilita quem vai implementar novas soluções. O alcance varia de acordo com a classe do módulo. Chips de classe 1 tem alcance de 100m e potência de 100mW. Módulos da classe 2 tem alcance até 10 metros e potência de 2,5mW. A classe 3 tem alcance de apenas 1 metro e dissipa no máximo 1 mw. O Bluetooth 5.0 tem um alcance de até 240 metros e taxa de transmissão de 50 Mbit/s (“Bluetooth Technology Website,” n.d.).

- **MQTT** (*Message Queue Telemetry Transport*) - inventado pela IBM para comunicação entre satélites e equipamentos de extração, possui uma confiabilidade e baixo consumo energético como requisitos para a sua implementação, o que o torna um grande candidato a ser aplicado nas redes IoT. Utiliza um modelo “*Publish/subscribe*” e necessita de um broker MQTT para gerir e redirecionar as mensagens dentro da rede. É descrito como um protocolo de comunicação *many-to-many* Uma das grandes vantagens do MQTT é a eficiência energética do modelo “pub/sub”, modelo esse que se comporta muito bem no que diz respeito à escalabilidade. Devido ao facto de utilizar o TCP/IP, o MQTT já se encontra munido da segurança de rede pelo que a implementação da mesma já não é necessária. A dependência do broker e a utilização do TCP/IP pode ser uma desvantagem, pois ambos necessitam de mais processamento para desempenhar bem as suas funções, requerendo desta forma mais/melhor hardware. Por outro lado, a necessidade de um broker, associada ao alojamento na *Cloud* pode ser uma vantagem competitiva dado que o próprio servidor *Cloud* pode fazer de Broker interagindo assim entre os dispositivos IoT e os outros serviços;

Neste protocolo de troca de mensagens, desenhado para utilização com baixa largura de banda, existem 4 conceitos básicos:

- ***Publish / Subscribe*** - permite publicar mensagens para determinado tópico ou dispositivo, mas também pode ser subscrito um tópico e receber todas as mensagens desse tópico

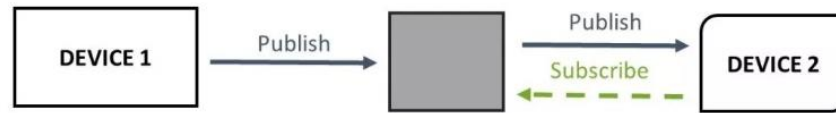


Figura 6 - Publicação de Mensagens (MQTT)

Sítio: <https://randomnerdtutorials.com/what-is-mqtt-and-how-it-works/> última visita em: 28/09/2018

Por exemplo o dispositivo publica num determinado tópico e o device 2 recebe a notificação e posteriores mensagens desse mesmo tópico. Na arquitetura do *publish subscribe* as mensagens MQTT são entregues de forma assíncrona ("push") por meio da arquitetura de assinatura de publicação. O protocolo MQTT funciona trocando uma série de pacotes de controle MQTT de forma definida. Cada pacote de controle tem uma finalidade específica e cada bit no pacote é cuidadosamente criado para reduzir os dados transmitidos pela rede. Uma topologia MQTT possui um servidor MQTT e um cliente MQTT. O cliente e o servidor MQTT comunicam entre si através de diferentes pacotes de controle. A tabela abaixo descreve resumidamente cada um desses pacotes de controle.

Control packet	Direction of flow	Description
CONNECT	Client to Server	Client request to connect to Server
CONNACK	Server to Client	Connect acknowledgment
PUBLISH	Client to Server or Server to Client	Publish message
PUBACK	Client to Server or Server to Client	Publish acknowledgment
PUBREC	Client to Server or Server to Client	Publish received (assured delivery part 1)
PUBREL	Client to Server or Server to Client	Publish release (assured delivery part 2)
PUBCOMP	Client to Server or Server to Client	Publish complete (assured delivery part 3)
SUBSCRIBE	Client to Server	Client subscribe request
SUBACK	Server to Client	Subscribe acknowledgment
UNSUBSCRIBE	Client to Server	Unsubscribe request
UNSUBACK	Server to Client	Unsubscribe acknowledgment
PINGREQ	Client to Server	PING request
PINGRESP	Server to Client	PING response
DISCONNECT	Client to Server	Client is disconnecting

Tabela 3- Mensagens de Controlo do Protocolo.

n.d., Sítio:

https://www.ibm.com/developerworks/community/blogs/5things/entry/5_things_to_know_about_mqtt_the_protocol_for_internet_of_things?lang=en ultimo acesso: 25/09/2018

O protocolo permite a comunicação em redes com restrições (ex. baixa largura de banda, alta latência, limites de dados e conexões frágeis). Os cabeçalhos de pacotes de controle MQTT são mantidos com o menor tamanho possível. Cada pacote de controle MQTT consiste em três partes, um cabeçalho fixo, cabeçalho variável e carga útil. Cada pacote de controle MQTT possui um cabeçalho fixo de 2 bytes. Nem todo o pacote de controle tem os cabeçalhos variáveis e carga útil. Um cabeçalho variável contém o identificador de pacote, se usado pelo pacote de controle. Um *payload* de até 256 MB pode ser anexado nos pacotes. Ter uma pequena sobrecarga de cabeçalho torna esse protocolo apropriado para IoT, diminuindo a quantidade de dados transmitidos em redes com restrições;

- **Messages** - mensagens são a informação que queremos partilhar entre dispositivos. Poderá ser comandos ou dados. Os níveis de qualidade de serviço (QoS – Quality of Service) determinam como cada mensagem MQTT é entregue e devem ser especificados para cada mensagem enviada por meio do MQTT. É importante escolher o valor de QoS adequado para cada mensagem, porque esse valor determina como o cliente e o servidor comunicam entre si para entregar a mensagem. Três QoS para entrega de mensagens podem ser obtidos usando o MQTT:
 - QoS 0 (no máximo uma vez) - onde as mensagens são entregues de acordo com os melhores esforços do ambiente operacional. A Perda de mensagem pode ocorrer.
 - QoS 1 (pelo menos uma vez) - onde as mensagens são garantidas que chegam ao destinatário, mas as duplicatas podem ocorrer.
 - QoS 2 (Exatamente uma vez) - quando a mensagem é garantida que chega exatamente uma vez.

Há uma regra simples ao considerar o impacto no desempenho da QoS. É “Quanto mais alta a QoS, menor o desempenho.” O MQTT fornece flexibilidade aos dispositivos IoT, para escolher a QoS apropriada que eles precisam para seus requisitos funcionais e ambientais;

- **Topics** - tópicos são a forma como queremos registar interesse pela determinada receção de mensagens, ou seja, quais as mensagens de determinados assuntos que teremos interesse em receber. Os tópicos são representados por palavras e divididos por barras: Exemplo: “Armazem1/temperaturas” ou “Posto Transformador/Portas”;
- **Broker** - o *Broker* é uma espécie de mediador entre dispositivos em que irá ser responsável pela receção das mensagens, filtra as mesmas e decidir, quais as que têm interesse para que dispositivos e de seguida publicá-las nos subscritores registados.

Quando um cliente MQTT se conecta ao servidor MQTT, ele pode definir um tópico e uma mensagem que precisa ser publicada automaticamente nesse tópico quando

desconectar inesperadamente (figura 11). Isso também é chamado de “*Last will and testament*” (LWT). Quando o cliente desconecta inesperadamente, o *timer keep alive* no lado do servidor deteta que o cliente não enviou nenhuma mensagem ou o *keep alive PINGREQ*. Portanto, o servidor publica imediatamente a mensagem *Will* no tópico *Will* especificado pelo cliente. O recurso LWT pode ser útil em alguns cenários. Por exemplo, para um cliente MQTT remoto, esse recurso pode ser usado para detetar quando os dispositivos IoT saem da rede. O recurso LWT pode ser usado para criar notificações para uma aplicação que está a monitorizar a atividade do cliente.



Figura 7- Broker - Notificação de Desconexão Anormal (MQTT)

Sítio: <https://randomnerdtutorials.com/what-is-mqtt-and-how-it-works/> última visita em: 28/09/2018

O MQTT é um protocolo aberto e padronizado pelo Comitê Técnico OASIS. Isso torna esse protocolo fácil de adotar para a grande variedade de dispositivos, plataformas e sistemas operacionais de IoT. Muitas aplicações do MQTT podem ser desenvolvidas apenas implementando os pacotes de controle *CONNECT*, *PUBLISH*, *SUBSCRIBE* e *DISCONNECT*. Diversas bibliotecas do cliente MQTT são disponibilizadas por meio do projeto Eclipse Paho.

- **CoAP** - protocolo que utiliza um modelo “Cliente/Servidor”, o qual disponibiliza uma interação “*request/response*” um para um, podendo também suportar *multicast*. Diferente do MQTT, o CoAP emergiu para superar a necessidade de protocolos IoT, desenvolvido para interoperar em HTTP e com arquiteturas *RestFul*, através de simples *proxies*, tornando-se dessa forma compatíveis com a Internet. Por utilizar o protocolo UDP, o CoAP apresenta menos consumos energéticos e de processamento. A utilização deste protocolo permite um menor tempo de resposta quando acionado, pois, mantém uma ligação ativa entre nodos. Este protocolo é mais indicado para envio de comandos em nodos locais, por se tratar de uma arquitetura mais semelhante ao HTTP. Também é

mais utilizado em dispositivos com menos poder de processamento / recursos mais reduzidos;

A tabela seguinte resume os protocolos em termos comparativos.

	<i>LoRaWAN</i>	<i>Neul</i>	<i>SigFox</i>	<i>NFC</i>	<i>WI-FI</i>	<i>Thread</i>	<i>6LowPAN</i>	<i>Z-Wave</i>	<i>ZigBee</i>	<i>Bluetooth</i>	<i>MQTT</i>	<i>CoAP</i>
Distância comunicação	Sem limite	10 Km	30 a 50Km	10 cm	50mts	N/A	N/A	N/A	Até 100 Mts	Até 240 Mts	Sem Limite	Sem Limite
Confiabilidade	sim	sim	Sim	Não	Sim	Sim	Sim	N/A	Não	Sim	Sim	Sim
Gerir grandes quantidades de dados	sim	Não	Não	Não	Sim	Não	Não	N/A	Não	Não	Sim	Não
Protocolos utilizados	TCPIP	TCPIP	TCPIP	N/A	TCPIP	IP	IP	N/A	TCPIP	TCPIP	TCPIP	HTTP / UDP
Consumo energético	baixo	20 a 30 mA	Baixo	Baixo	Alto	Alto	Alto	N/A	Baixo	Baixo	Baixo	Baixo
Suporta redes de grandes dimensões	sim	Não	Não	Não	Sim	Não	Não	Não	Não	Não	Sim	Não
Velocidades de transmissão	entre 0.3 Kbps e 50 Kbps	Até 100Kps	Até 1Kbps	Até 420 Kbps	1Gbps	N/A	N/A	100 Kbps	Até 250Kbps	Até 50Mbps	Sem Limite	Sem Limite

Tabela 4 - Comparação de Protocolos

2.2. Processo de Desenvolvimento de Software

O desenvolvimento ágil de Software ou método ágil é um processo de desenvolvimento de software que promove um desenvolvimento iterativo e incremental, consistindo num conjunto de metodologias guiadas, por quatro valores e doze princípios definidos no Manifesto Ágil (Teles, 2008) nomeadamente:

- Indivíduos e a interação entre eles. mais que processos e ferramentas;
- Software em funcionamento mais que documentação abrangente;
- Colaboração com o cliente mais que negociação de contratos;
- Responder a mudanças mais que seguir um plano.

Os 12 princípios expandem os valores de forma mais detalhada dando mais ênfase aos itens à esquerda dos valores do que os da direita. Devido à simplicidade de seus valores e ênfase na organização e gestão das equipas, tem-se a impressão de que documentar os requisitos, definir um projeto de arquitetura, fazer análise e design não são obrigatórios ou não precisam de nenhum tipo de rigor.

Porém, também tem como princípio a atenção contínua à excelência técnica e a um bom *design* salientando a importância de se ter um *design* satisfatório, o que só é possível com certo rigor e formalismo no projeto de arquitetura que por sua vez é baseado nos requisitos levantados. Desde

as décadas de 80 e 90, são aplicadas metodologias designadas como “ágeis”, destacando-se entre elas o Scrum desde 1986 (Takeuchi e Nonaka, 1986) e a Programação Scrum (Sutherland, 2013).

2.3. Bases de Dados

Entre os melhores motores de bases de dados direcionadas para IoT Apps e M2M, destacam-se:

- **InfluxDB** - é um motor de base de dados de séries temporárias distribuídas, de código aberto, desenvolvido pela InfluxData. Esta base de dados foi escrita na linguagem de programação *Go* e é baseado no LevelDB, uma base de dados de Valor-chave. Além de um *Front-end*, uma interface HTTP e de bibliotecas de que são fornecidas aos utilizadores para interação com a base de dados, a principal vantagem do InfluxDb é a sua capacidade de agregar valores em intervalos de tempo, em tempo real, sem intervenção manual. O InfluxDB está disponível por acesso via plataforma ou através de outras ferramentas externas de consultas de dados, tais como, Grafana, que é uma poderosa ferramenta de *front-end* que fornece recursos de visualização de dados de series temporais.

Este motor de base de dados não possui dependências externas e são usadas consultas sql para consultar os dados tais como medições, series ou pontos. Cada ponto consiste em vários pares de valores-chave chamados *fieldset* e *timestamp*. Os valores podem ser inteiros de 64 Bits, pontos flutuantes de 64 bits, strings e booleanos. Os pontos são sempre indexados pelo seu tempo e conjunto de *tags*. O InfluxDB armazena dados via HTTP, TCP e UDP. Tem como principais características:

- Ser escrito exclusivamente na linguagem de programação GO, facilitando a compilação num único binário sem dependências externas;
- Armazena dados com alto desempenho no armazenamento dos mesmos, especialmente para dados de séries temporais. O mecanismo de TSM do InFluxDB permite armazenamentos e compactação de dados eficientes e de alta velocidade;

- Existem plugins que suportam outros protocolos de armazenamento de dados como o Graphite, Collectd ou mesmo OpenTSDB;
- Está equipada com ferramenta de *front-end* Web que incorpora administração da base de dados e de utilizadores;
- Tem bom desempenho na união de várias séries de dados.

Endereço oficial Web: <https://www.influxdata.com/>

- **CrateDB** - o CrateDb é um gestor de base de dados distribuído, de código fonte aberto, desenvolvido pela Crate.io Inc, que integra totalmente um sistema de armazenamento de dados orientado a documentos pesquisável. O CEO da Crate.io, Christian Lutz, disse que no momento da fundação da Crate.io, decidiram reinventar o SQL para a era dos dados provenientes de máquinas. Segundo ele, hoje 75% dos clientes usam CrateDB para gerir dispositivos IoT, devido a facilidade de utilização, desempenho e versatilidade. O CrateDB torna possível o acesso a dados provenientes de máquinas, não estruturados, aos utilizadores de SQL. O CrateDB combina SQL com versatilidade de pesquisa e facilidade de escalabilidade de “*containers*”.

Este gestor de base de dados fornece uma boa alternativa para ferramentas de armazenamento de dados analíticos como o *Splunk*. Esta plataforma, inclui um mecanismo de consulta de dados distribuída onde permite a utilização de ligações entre entidades, agregações e consultas de ad-doc mais rápidas, SQL com pesquisa integrada por versatilidade de dados e de consulta. Tem arquitetura de “*container*” e *sharding* de dados automático para dimensionamento simples. A principal linguagem de programação utilizada é o SQL mas também permite a utilização de linguem noSQL, ou seja, orientada a documentos ou informação não relacional. Este motor de base de dados permite a inclusão do Facebook Presto como ferramenta analítica do motor de base de dados assim como ferramenta de análise de previsão. Tem uma interface de administração embutido. O Crate Shell CLI permite aos utilizadores fazer consultas SQL interativas e tem como principais características:

- Facilidade em escalabilidade - as atualizações à base de dados / motor de base de dados é feita de forma simples bastando para isso adicionar novas máquinas e

consequente atualização do cluster, não existindo a necessidade de redistribuição de dados, dado que essa tarefa será feita automaticamente pelo CreateDB;

- Alta disponibilidade - o CrateDb permite valores muito elevados de disponibilidade mesmo em situações mais anormais já que fornece um sistema de replicação automatizado de dados em todo o cluster, até mesmo as atualizações de hardware e software não interrompem as operações de dados normais. O CrateDB tem a capacidade de autorrecuperação quando deteta nós inoperacionais;
- O CrateDB tem uma capacidade de tratamento e armazenamento de dados em tempo real muito acima do normal mesmo em situações extremas em que se esteja a escrever grandes quantidades de dados e ao mesmo a consultar dados, o CrateDb tem a capacidade de se auto desbloquear em “*overheads*”;
- Suporta vários tipos de dados relacionais como também suporta documentos em JSON. Também permite “*blob Storage*” para guardar documentos de vídeo, áudio ou texto de grandes dimensões e não estruturados;
- Suporta consultas geoespaciais e esquemas dinâmicos, tornando totalmente flexível.

Endereço oficial Web: <https://crate.io>

- **Riak time series database** - motor de base de dados vocacionado para séries temporais (TS). É um motor de base de dados otimizado para armazenamento de dados “valor-chave” noSQL de código aberto num âmbito de aplicações IoT. Com o RTSB existe a possibilidade de se associar uma grande quantidade de pontos a um ponto específico de tempo. Este motor é baseado numa arquitetura sem mestre, na qual cada nó do cluster é capaz de atender a pedidos de leitura e armazenamento de dados. Este motor permite a replicação e distribuição de dados pelo cluster de forma automática mantendo o alto desempenho e disponibilidade.

A base de dados em Riak TS é altamente otimizado para aplicações, cujo requisito a dados seja a predominante. Suporta integração com Apache Spark, o que possibilita o suporte à integração para *streaming* do spark, “*dataframes*” e Spark SQL. O Riak TS pode ser instalado diretamente no data center ou numa cloud. A AWS Amazon Machine

Images (AMI) também tem disponibilidade para esse tipo de base de dados, de forma a facilitar os utilizadores a experimentar esta plataforma em ambiente de trabalho. O Riak RS tem como principais características:

- Suportar a adição de novos nós à arquitetura de Cluster já existente sem “*sharding*”; os dados são automaticamente e uniformemente distribuídos pela Cluster;
- Suportar DDL para tabelas e campos, assim como também suporta armazenamento de dados não estruturados e semiestruturados;
- Suportar replicação para mais que um cluster com facilidade na administração do sistema de replicação de dados, tanto em *data-center* interno como em *data-center* na *cloud*, em qualquer parte do mundo;
- Suportar consulta de dados do tipo SQL pelos utilizadores ou aplicações, com acesso fácil e flexível a base de dados globais;
- Suportar integração de aplicações com API's e bibliotecas de clientes em vários idiomas tais como Java, Ruby, Python, Erlang, Go, Node.js e .NET;
- A “frameWork” Riak Meso permite a gestão eficiente de recursos de cluster e um dimensionamento ajustável, tanto na redução como na incrementação de nós;
- Suportar integração integral com o Apache Spark para análise operacional de série de dados por tempo.

Endereço oficial Web: <http://basho.com/products/riak-ts/>

- **MongoDB** - o MongoDB é um motor de base de dados muito poderoso, flexível, livre e de código aberto, orientado a documentos, escalável e para uma utilização generalizada. Tem a capacidade de expansão de recursos tais como índices secundários, consultas em intervalos de tempo, classificações, agregações e índices sobre geolocalizações. É caracterizável por ser um motor de bases de dados noSQL, pois utiliza JSON como documentos e esquemas.

O MongoDB adiciona preenchimento dinâmico de documentos e pré-aloca arquivos de dados para negociar o uso de espaço extra para um desempenho consistente. Faz uma utilização da RAM de forma eficiente, nomeadamente de armazenamentos em cache e corrige query's para utilização efetiva dos índices. Suporta uma linguagem de consulta

avançada para poder dar suporte a operações de leitura e gravação (CRUD), bem como agregação de dados, pesquisa de texto e consultas “geospacial”. O MongoDB tem como principais características

- Suportar índices secundários genéricos para uma variedade de consultas rápidas e fornece recursos de indexação exclusivos, compostos, geoespaciais e índices de “*full text*” para os utilizadores;
- Suportar agregação por “pipelines” para construção complexa de agregações com base em peças simples para otimização da base de dados;
- Suportar TTL para dados que devem expirar depois de determinado período de tempo;
- Suportar protocolo “*easy-to-use*” para armazenamento de ficheiros de grandes dimensões e ficheiros de meta-dados;
- Suportar JSON para arquivo e transmissão de dados. O JSON sendo um protocolo padrão, é uma grande vantagem, tanto para Web como para base de dados;
- Suportar Map-Reduce do lado do servidor para processamento de informação utilizando funções JavaScript;
- Dispõe da ferramenta “MongoDB Management Service (MMS) para permitir que os utilizadores possam procurar a informação na base de dados assim como fazer a cópia integral dos dados;
- Suportar automatização de configuração de “*load balancing*” devido à colocação de grandes volumes de dados em particionamentos (*data shards*).

Endereço oficial Web: <https://www.mongodb.com/>

- **RethinkDB** – o RethinkDB é uma base de dados distribuída e de código fonte aberto, utilizado principalmente para armazenar documentos JSON com capacidade de escalabilidade. Este motor de base de dados é considerado como primeira e principal escolha por parte de quem faz desenvolvimento de aplicações baseadas em IoT, com alimentação de dados em tempo real. Foi responsável pelo revolucionamento completo da arquitetura de base de dados tradicional, invocando novos modelos de acesso para atualização de resultados das consultas para aplicações em tempo real.

O RethinkDB oferece uma linguagem de consulta flexível para monitorização de APIs e é altamente fácil de configurar, aprender e aplicar. O RethinkDB oferece várias vantagens comparando com o MongoDB, nomeadamente:

- Linguagem de consulta avançada, que suporta ligações de dados entre tabelas, com “*sub-queries*”;
- Com uma operação e monitorização poderosa com a inclusão de API com uma integração perfeita na linguagem utilizada na consulta de dados o que torna o RethinkDb uma ferramenta de fácil integração e utilização;
- Uma interface de administração simples e bonita que permite fragmentar e replicar de forma fácil e oferece documentação on-line assim como sugestões de idioma de consulta.

Tem como principais características:

- Tolerância a falhas. Ele suporta a mudança automática de servidor se detetar falhas no servidor principal;
- Fácil adição de nós no cluster. *Plug-and-play* de nós em tempo real, sem qualquer tempo de inatividade, nem mesmo por um milissegundo;
- Inclui interface de programação de aplicativos assíncronos suportando desta forma consultas assíncronas via *EventMachine* em Ruby e Tornado;
- Suportar acesso SSL para ter acesso seguro ao RethinkDB via Internet pública;
- Suporta funções de cálculo como o floor, Ceil, round entre outras.

Endereço oficial Web: <https://rethinkdb.com/>

- **SQLite** – o SQLite é uma base de dados relacional cujo código fonte é aberto e incorporado, projetado para fornecer uma forma fácil de as aplicações gerirem dados sem sobrecarga. É altamente portátil, fácil de usar, compacto, eficiente e confiável. O SQLite é compatível com ACID. Implementa a maioria dos padrões SQL e usa sintaxe SQL dinâmica. O mecanismo SQLite não é um processo independente como noutros motores de base de dados. Poderá ser associado a aplicações estáticas ou dinâmicas. Tem como principais características:

- Não requer um servidor independente ou processos separados para poder operar. Pode funcionar num ambiente sem servidor;
- Não requer qualquer administração do motor de base de dados nem de um servidor dedicado dado que os requisitos deste motor de base de dados é muito pouco exigente em termos de hardware;
- Sem dependências externas (Independente);
- Linguagem de programação em ANSI-C e com fornecimento de API simples e fácil de utilizar;
- Compatível com qualquer plataforma, seja ela, Windows, unix mac-OS, etc;
- As transações são totalmente compatíveis com ACID, permitindo acesso seguro a partir de vários processos;
- Suporta todas as consultas SQL encontradas no SQL92;
- Código totalmente testado e verificado no SQLite, livre erros e atualizado.

Endereço oficial Web: <https://www.sqlite.org>

- **Apache Cassandra** – o Apache Cassandra é considerado uma base de dados cujo código fonte é aberto, altamente escalonável e conceptualizado para gerir grandes quantidades de dados estruturados em muitos servidores. Em comparação com outros motores de base de dados de código fonte aberto, este oferece vários recursos adicionais de alta performance em termos de disponibilidade e desempenho de escalabilidade. Este motor de base de dados também tem a característica de, com facilidade, ter as suas bases de dados em cluster, dividindo assim os dados pelos vários servidores.

O Cassandra foi desenvolvido pelo Facebook com o objetivo principal de facilitar a pesquisa na caixa de entrada e acabou por transformar-se num *open source* em 2008. Este motor de base de dados foi implementado com um modelo de replicação “*Dynamo-style*”, o qual é conhecido por não ter nenhum ponto de falha e associado a este facto tem também a particularidade de ter um modelo de dados mais poderoso, denominado “*column family*”. Tem como principais características:

- Ser massivamente escalonável, sem uma hierarquia de nós, onde todos os nós estão ao mesmo nível o que proporciona uma simplicidade operacional e de fácil dimensionamento. Arquitetura “*masterless*”;
- Com a Arquitetura “*masterless*” os dados podem ser escritos em qualquer um dos nós;
- Desempenho exponencial linear em que à medida que o número de nós aumenta, o desempenho do Cassandra também aumenta;
- A deteção e recuperação dos nós com falha é efetuada com sucesso à medida que as situações vão sendo detetadas, são facilmente restaurados e recuperados;
- Modelo de dados flexível e dinâmico onde existe suporte a tipos de dados com rápida leitura e escrita;
- Forte proteção de dados com confirmação do sucesso das operações de escrita assim como com mecanismos implementados de base para cópias de segurança e recuperação de dados;
- Suporta ajuste de consistência de dados em toda a arquitetura distribuída;
- Suporte Replicação para múltiplos *data-centers*;
- Tem a capacidade de comprimir dados até 80% sem sobrecarga;
- Linguagem de construção de consultas idêntica ao SQL;
- Muito fácil para os arquitetos de desenvolvimento migrar uma base de dados de qualquer estrutura para o Cassandra.

Endereço oficial Web: <http://cassandra.apache.org>

- **Firestore** - o Firestore Realtime é um motor de base de dados noSQL alojado na *Cloud*. Com esta ferramenta pode-se armazenar dados e sincroniza-los pelos vários utilizadores. Tem como objetivo armazenar dados em JSON e foi especialmente criado para trabalhar com dispositivos moveis e aplicações web. Esta ferramenta tem facilidade de escalabilidade sem baixar o nível de performance. O sistema robusto de sincronização de dados entre dispositivos em tempo real é uma das principais vantagens. Os SDKs existentes para Android, iOS e Javascript sincronizam os dados das aplicações de forma quase instantânea. Esta também será uma das vantagens na criação de aplicações reativas com sincronização automática de dados entre

dispositivos assim como a sinergia entre recursos colaborantes. Caso não seja requisito a sincronização em tempo real, o Firebase tem um poderoso sistema de leitura “*one-time*” que fará todo o controlo.

Usa coleções e documentos para estruturar e consultar dados. Este modelo de dados é muito familiar e intuitivo para muitos dos programadores nesta área. Também permite consultas de dados expressivas em que basicamente permite ter a mesma performance, sendo o resultado da “*Query*” um registo ou 10000000 de registos. Permite acesso a dados em *Off-line* devido a uma “base de dados” no dispositivo. Esta característica permite ter uma aplicação móvel on-line que também trabalha em offline. Esta funcionalidade está disponível em iOS, Android e Web.

Permite o desenvolvimento de aplicativos sem que seja necessário o acesso direto aos servidores de base de dados. Os SDKs do “lado do cliente” controlam o método complexo de autenticação e do código referente ao processo de autenticação que normalmente é necessário que seja implementado. Do lado do *Back-End*, é fornecido um conjunto de regras de segurança que é passível de ser controlado pela aplicação por forma a garantir o controlo de acesso aos dados. As ditas regras de segurança são passíveis de ser aplicadas por utilizador e documentos assim como permite a aplicação de lógica de validação complexa. As ditas funcionalidades permitem ter um elevado nível de segurança aos dados.

Permite ativar o *Firebase Cloud Functions* para execução de código personalizado sempre que são armazenados dados na Cloud ou no momento em que é feita a autenticação Firebase a partir dos SDKs. Pode ser redimensionado sem qualquer problema e de forma automática, permitindo a replicação de dados em várias regiões, para que sejam recuperáveis ou estejam disponíveis em curto espaço de tempo. É um motor de base de dados munido de escalabilidade, consistência e distribuído.

O FireBase também permite aos programadores poderem interagir do lado do servidor (*back-end*) através de SDKs que neste momento permitem interação em linguagens como Java, Go, Python e Node.js. Tem como principais características:

- Documentos e coleções de documentos podem ser listados com altos níveis de performance;
- Compatível com IOS e Android assim como Web, existindo SDKs disponíveis manuseamento e armazenamento de dados quando se encontra em off-line;
- Sincronização em tempo real com taxas de *delays* muito reduzidas;
- Sistema automático de replicação de dados multi-região com elevados níveis de consistência;
- Elevado nível de integração com variadas linguagens de programação via SDK tais como Node, Python, Go e Java;
- São também objetivos a facilidade e simplicidade de integrar sistemas com este motor de base de dados;
- Permite desenvolvimento sem acesso direto ao servidor;

Endereço oficial Web: <https://firebase.google.com>

A tabela 5 faz uma análise comparativa dos diferentes motores de base de dados relativamente a aspetos essenciais de desempenho. Para cada uma das mais relevantes características foi atribuído um valor entre 0 e 10, sendo que 0 corresponde a não ter a característica e 10 ter a característica muito bem implementada.

	Escalabilidade	Arquitetura distribuida	Segurança	Desempenho	Desenvolvimento Rápido	Replicação	Pontuação total
InfluxDB	2	6	4	5	2	6	25
CrateDB	7	8	5	6	3	8	37
Riak time serie	7	7	6	7	2	4	33
MongoDB	6	6	3	4	4	5	28
RethinkDB	7	7	6	7	6	7	40
SQLite	0	0	2	8	6	0	16
Apache Cassandra	8	7	7	7	6	7	42
Firebase	9	10	10	9	8	8	54

Tabela 5- Comparativo de Motores de Bases de Dados

3. Modelo

Com base em necessidades de negócio das organizações, foi formulada uma solução, que será validada através de um artefacto, verificando se colmata as referidas necessidades e mantendo ou reduzindo custos, com aumento da disponibilidade de dados.

3.1. Descrição do Modelo

A solução baseia-se num equipamento eletrónico que tem a capacidade de receber sinais digitais e/ou analógicos e, após processamento, poder transmiti-los via internet, através de rede móvel ou fixa para um servidor central, que tem a capacidade de processar os dados enviados, com base num fluxo de trabalho definido pelo utilizador, promovendo ações de alerta. Esta solução também terá a possibilidade de incorporar um atuador que poderá ser configurado para atuar após deteção de parâmetros previamente configurados.

A solução em causa terá componentes base e que irão colmatar as necessidades da maior parte das empresas, oferecendo ainda algumas funcionalidades adicionais, não contempladas de série (ex. painéis solares em casos de ausência de fonte de alimentação externa) e outras que irão permitir abranger clientes com necessidades especiais tais como os do setor de alimentação (ex. temperaturas, humidades). A possibilidade de utilizar canais de acesso internet, alternativos à rede GSM, permite utilizar recursos já existentes localmente, reduzindo os custos inerentes às comunicações.

A necessidade de localização através de coordenadas GPS, também será uma funcionalidade adicional (considerando-se eventualmente desnecessária em locais fixos), não estando presente na solução base, de forma a minimizar o seu custo. Esta funcionalidade só terá sentido se existir a necessidade de relocalizar o equipamento com frequência e em situações em que seja de difícil localização. O equipamento base pode ser acoplado aos módulos adicionais, que irão colmatar algumas das necessidades específicas do cliente.

3.2. Hardware

Nesta fase do trabalho foi estudada a arquitetura hardware do artefacto de forma a poder encontrar a solução que mais opções pudesse dar face às necessidades e constrangimentos apresentados pelos interessados.

3.2.1. Descrição

Tendo em conta a variedade de ambientes de aplicação, definiu-se que para a solução base se deveria dar preferência à utilização de uma caixa estanque para o equipamento remoto (figura 8), integrando no próprio corpo todos os elementos necessários ao processamento, localização georreferenciada, comunicações (na imagem a opção de GSM), alimentação interna e sensor de movimento.

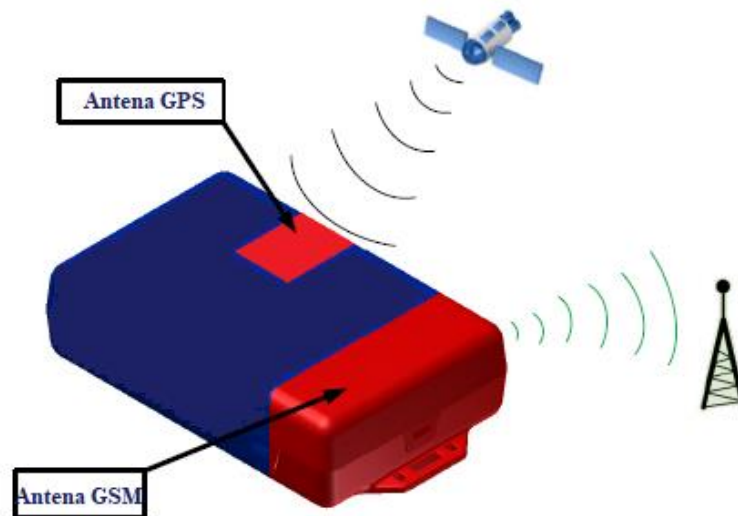


Figura 8 – Protótipo do Equipamento. (Documento Interno da Inosat, 2017)

3.2.2. Objetivos

O equipamento terá como principal finalidade a receção de sinais (digitais e/ou analógico, processá-los e transmiti-los para um servidor central. A transmissão dos dados poderá ser executada por um dos canais disponíveis. Este equipamento também terá um atuador que terá como missão emitir sinal (massa ou tensão) com a finalidade de remotamente poder fazer atuar um outro dispositivo, via relé por exemplo. Poderá ter como função a ativação de um alarme, de uma câmara, de um qualquer outro sistema.

Este equipamento também irá suportar sensores de temperatura que permitirão ter a informação periódica das temperaturas locais. O dispositivo estará também preparado para integrar informação dos sensores de porta, sensores estes que permitem detetar o estado da porta onde o sensor se encontra instalado, permitindo remotamente ter por exemplo a informação de quando uma determinada porta/janela abre ou fecha. Sendo um equipamento estanque, enviará informação ao servidor sempre que seja aberto, sempre que a sua alimentação externa seja interrompida ou sempre que detete movimento do próprio equipamento (sensor interno de movimento).

Este equipamento teve como base uma solução já existente no mercado, produzida pela Inosat e vocacionada para a gestão de frotas (veículos). Esta base é alterada essencialmente em termos de sinais admissíveis nas suas portas digitais, tendo sido feito igualmente o desenvolvimento de novas funcionalidades ao nível do firmware que permitem a sua gestão, assim como a gestão da forma de comunicação do dispositivo, permitindo o envio permanente de informação. Estas alterações permitem expandir significativamente o âmbito de utilização do equipamento nomeadamente em aplicações fixas.

3.2.3. Componentes base do equipamento remoto

Este equipamento é composto por um processador, responsável por todo o tratamento de dados e das funcionalidades existentes. Tem uma memória volátil que permite fazer a gestão das variáveis locais e uma memória não volátil que guarda os dados a transmitir ao servidor central.

Esta memória está dividida em duas listas com apontadores que identificam os endereços onde escrever os próximos dados e outro apontador que indica quais os dados por enviar.

O equipamento da solução base inclui:

- **Portas digitais e analógicas** - O equipamento tem 4 portas digitais série que permitem receber sinais de tensão entre 3v e 30v e os converte aos sinais digitais. e com quatro portas série analógicas. Tem também duas portas analógicas, para sensores de temperatura, tendo a possibilidade de receber sinais analógicos através do protocolo I2C, rececionando os dados, com tratamento local e envio para o servidor central. A figura 9 ilustra o posicionamento das portas de entrada disponíveis no equipamento.

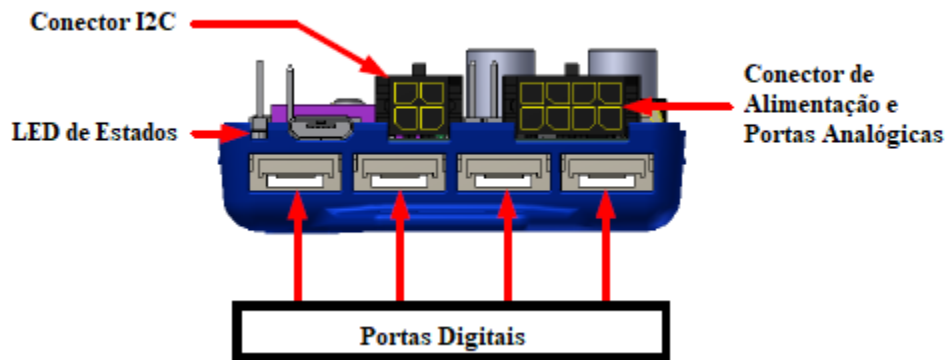


Figura 9- Portas de Entrada Disponíveis no Dispositivo Remoto. (Documento Interno da Inosat, 2017)

Estas portas são registadas via I2C, protocolo que permite registar vários dispositivos no mesmo canal, mas com Ids diferentes. Tem também uma porta, denominada de atuador, que permite emitir sinais de tensão e consequentemente ativar por esta mesma via outros dispositivos. A figura 10 ilustra as ligações cabladas a interfaces do dispositivo remoto;

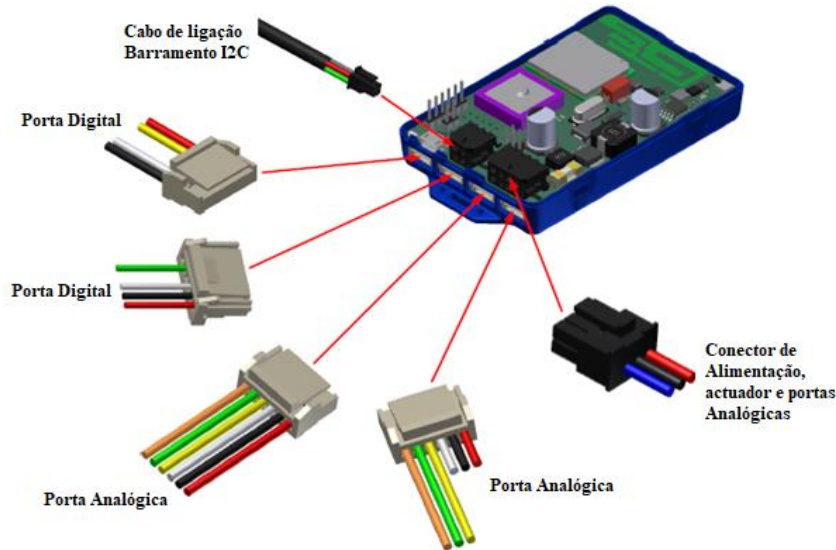


Figura 10- Interfaces do Dispositivo Remoto. (Documento Interno da Inosat, 2017)

- **Processador** - o processador selecionado foi um PIC32MX795F (MicroChip) dado que se trata de um processador de baixo consumo energético, com um custo baixo, com todas as funcionalidades necessárias para implementação deste projeto;
- **Modem 3G** - foi escolhido o Modem Quectel M95 dado que é um modem 3G, com um custo de aquisição e energético muito baixo, tendo todas as funcionalidades necessárias para o projeto a implementar;
- **Memória** - Este equipamento vem por defeito com Memória 16M-BIT *SERIAL FLASH MEMORY* no *chip* Winbond 25Q16BVSIG que é o suficiente para o tipo e quantidade de informação que recolhe e trata. Com a implementação de novas funcionalidades, prevê-se a necessidade de adicionar mais memória;
- **Atuador** - esta funcionalidade permite enviar sinais, através de porta específica, que pode originar a execução de funcionalidades externas, como ativar alarme, ativar camaras, bloquear acesso, entre outras;

- **Transformador de corrente elétrica** – o equipamento vem de série com adaptador de corrente eléctrica, de 12v - 2A. Para além do adaptador de corrente, o equipamento encontra-se também equipado com bateria interna de segurança;
- **Barramento I2C para expansão de portas** - este equipamento vem de origem integrado com um barramento I2C. Este barramento permite a comunicação entre componentes/ periféricos padronizados e a troca de informação de uma forma eficiente;
- **Modem de rede sem fios** (componente adicional) - modem wireless que permite a ligação a uma rede Wi-Fi local para envio dos dados para ao servidor deixando assim de ser necessário a integração de um sim de GSM para transmissão de dados;
- **Placas Solares** (componente adicional) - alternativa, em termos de fonte de energia, permitindo a reposição dos níveis energéticos da bateria. Com esta solução, o equipamento poderá ser instalado em locais sem ligação à rede elétrica;
- **Módulo GPS** (componente adicional) – para equipamentos que possam ser movimentados com frequência de um local a outro ou em casos em que a frequência com que se instalem e desinstalem equipamentos em locais distintos seja muito elevada, permitindo a sua localização absoluta em tempo real;
- **Expansor de Portas** (componente adicional) - para casos em que as portas fornecidas por defeito no equipamento não são suficientes. Com este expansor, pode-se adicionar mais portas digitais/analógicas ao equipamento para ligação a mais sensores, evitando a aquisição de outro equipamento remoto.

3.2.4. Protocolo de comunicação de dados

O protocolo seleccionado para comunicação entre os equipamentos remotos e o servidor, a implementar no artefacto, foi o MQTT (figura 11) devido a fatores que se descrevem seguidamente.

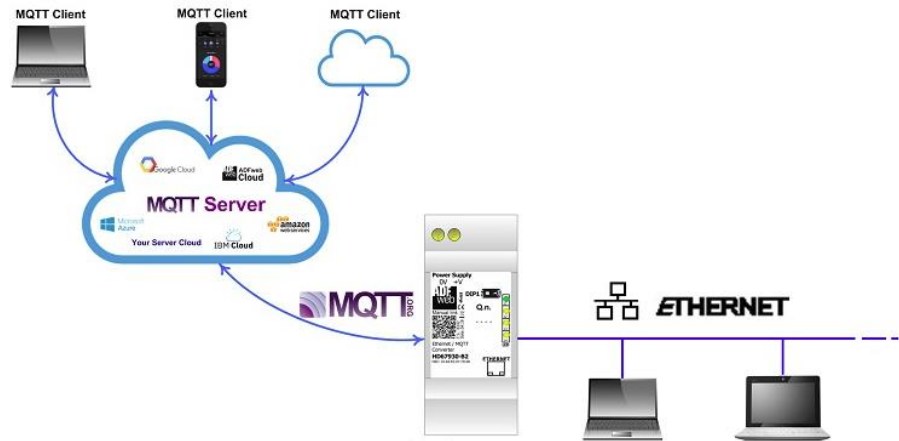


Figura 11- Estrutura do MTQQ.

n.d., Sítio:

http://www.adfweb.com/Home/products/Ethernet MQTT.asp?frompg=nav28_1&loc_phy=20874&k001=b&dg-k2=%2Bmqtt%20protocol%20architecture&d=c&pos=1o1&gclid=Cj0KCCQiAr93gBRDSARIsADvHiOr_tJ-usuPjKA24JXGkQGpuTsgvq0WCVYqCProF-Cam48Xzz9nWEhMaAuc2EALw_wcB último acesso: 20/09/2018

É um protocolo de fácil implementação, onde é utilizado um modelo “*Publish/subscribe*” em que existe um *broker* pelo meio que fará a gestão de entrega de informação, basicamente será o responsável por fazer chegar aos destinatários os dados enviados pelos remetentes. Permite interagir com qualquer dispositivo ligado à *Cloud*, devido ao já referenciado *Broker*. Com este sistema podem-se estabelecer comunicações com múltiplos dispositivos, onde existe a possibilidade de receber e enviar mensagens (leia-se pacotes de dados) dentro da rede.

3.3. Software

Nos próximos pontos irão ser abordados todos os componentes que fazem parte da plataforma de software e dos serviços inerentes. O controle e automatismo de procedimentos são objetivos inerentes ao IoT. Com estes objetivos, as aplicações poderão gerar grande quantidade de dados, dependendo da informação a recolher e da periodicidade com que é recolhida. O envio contínuo de dados heterogéneos pode criar problemas de eficiência no armazenamento, transferência e gestão dos dados.

Os sistemas de base de dados tradicionais usam uma linguagem estruturada (SQL - *Structured Query Language*) que suporta todos os requisitos do utilizador em conjunto com simplicidade, robustez, flexibilidade, escalabilidade e desempenho eficiente. A utilização deste tipo de base de dados enfrenta uma limitação determinante que é o esquema estático da mesma, que torna o RDBMS não adequado para aplicações de IoT. Por outro lado, estão a emergir rapidamente no mercado bases de dados com um desempenho muito superior ao sistema de base de dados convencionais (SQL). As bases de dados ditas “não relacionais” (NoSQL), são base de dados sem relações entre entidades e sem esquema de estrutura são base de dados, sendo de fácil configuração, replicação e possuindo escalabilidade. Foi verificado também que as bases de dados não estruturadas têm um desempenho muito superior para aplicações direcionadas para o IoT.

3.3.1. Motor de base de dados

Dado que o tipo de aplicação que será desenvolvida está relacionado com IoT e tendo em conta as necessidades relativas à base de dados que são exigidas neste tipo de aplicações, foram feitas pesquisas e posterior comparação dos principais motores de bases de dados vocacionadas para o IoT (a maior parte delas gratuitas). Para conseguir manipular os dados provenientes da aplicação, de forma eficiente, é muito importante o motor de base de dados se enquadre de forma eficiente com a plataforma IoT que está a ser projetada. Existem inúmeros fatores que devem ser levados em conta, tais como a escalabilidade, a capacidade de lidar com grandes volumes de dados, que tenha um esquema de dados flexível, com altos níveis de segurança, disponibilidade, tolerância a

falhas e seja passível de portabilidade no que respeita a ferramentas analíticas. Os custos, nomeadamente de manutenção também serão tidos em conta.

A possibilidade de implementar bases de dados distribuídas, com réplicas de dados em vários servidores é também uma mais valia, dado que mitiga o risco de perdas de dados e de indisponibilidade. Se algum dos servidores falhar, existe um servidor que irá assumir o papel e responder aos pedidos de dados e tarefas de armazenamento até que o servidor principal volte a estar disponível. Todas as plataformas relacionadas com IoT devem ter motores de base de dados com alto nível de disponibilidade tendo em conta os elevados volumes de dados. Se um servidor de base de dados ficar inativo ou se tiver uma quantidade muito alta de dados para armazenar, os dados poderão ser armazenados em sistemas de mensagens até que seja processado o *backlog* de dados atrasados ou que sejam adicionados mais servidores ao cluster de base de dados principal.

Com base na análise comparativa, entre os diferentes motores de base de dados, efetuada no capítulo do Estado da Arte, relativamente a aspetos considerados relevantes para o IoT, conclui-se que o Firebase é o que melhores condições reúne para a implementação do artefacto.

Com este motor de base de dados reúne-se a maior parte das funcionalidades que queremos implementar com baixo custo de desenvolvimento e recursos. O Firebase está preparado para trabalhar com os mais conceituados sistemas operativos tais como Windows, Linux, Android, IOS entre outros. Tem um sistema otimizado de autenticação que permite o desenvolvimento de aplicações com processos de autenticação seguros e variados em termos de registo de utilizadores. Tem uma interface de fácil utilização, como sistema de desenvolvimento *drag and drop*, compatível com *drop-in*, personalizável e de código aberto, que permite processar os fluxos da interface do utilizador para fazer o processo de autenticação. O Processo de autenticação FirebaseUI implementa práticas recomendadas para autenticação em dispositivos móveis e sites.

Ao nível da segurança, o Firebase tem um sistema de segurança que foi desenvolvido pela mesma equipa que desenvolveu os sistemas de segurança do google, nomeadamente o *smartlock*, gerador de senhas do Chrome e que contém uma vasta experiência em gestão de dados de uma

das maiores empresas do mundo em termos de números de utilizadores. Ao nível da implementação, a configuração do sistema de autenticação, nesta plataforma é bastante rápida e requer menos de 10 linhas de código. Não necessita de ter uma equipa dedicada à implementação, monitorização do sistema dado que com esta plataforma tudo foi previsto e desenvolvido para despende o mínimo tempo e recursos possível.

3.4. API

O objetivo da API é facilitar a comunicação entre plataformas/programas tendo a possibilidade de ter acesso a dados através dos serviços disponíveis e respetivos métodos. A API está dividida em serviços de acordo com o tipo de informação ou operação que se pretende fazer e em cada um dos serviços estão representados um conjunto de métodos com determinadas funcionalidades.

Para utilizar a API, é obrigatório iniciar o processo pela autenticação por forma a ter um ID (GUID), ID esse que será sempre necessário em todos os restantes pedidos. No final, devemos terminar a sessão de utilização da API com o fim de sessão (*Logout*).

Máquina de Estados Utilização API

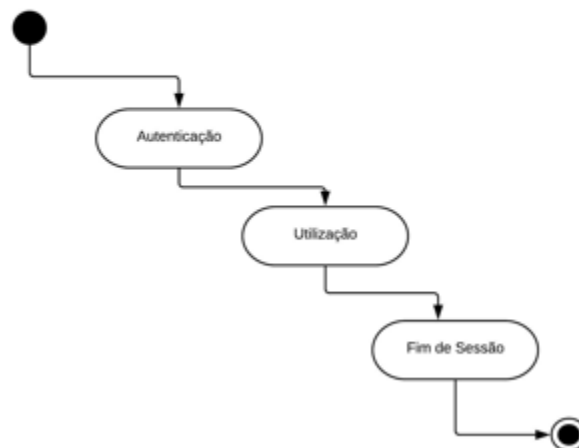


Figura 12 - API Utilização

A API está dividida em vários Serviços, encontrando-se descrita no Anexo I.

- UserService - Nesta secção será representado todos os métodos relacionados com o utilizador.
- DeviceService - Nesta secção será representado todos os métodos relacionados com os dispositivos.

4. Arquitetura Tecnológica do Modelo

A arquitetura tecnológica do modelo será descrita sob a forma de *mockups*, ilustrando assim as funcionalidades da plataforma.

4.1. Plataforma

Relativamente à plataforma, esta ficará alojada na *Cloud* e é composta por alguns procedimentos básicos que devem ser executados antes de se poder utilizar em pleno a plataforma.

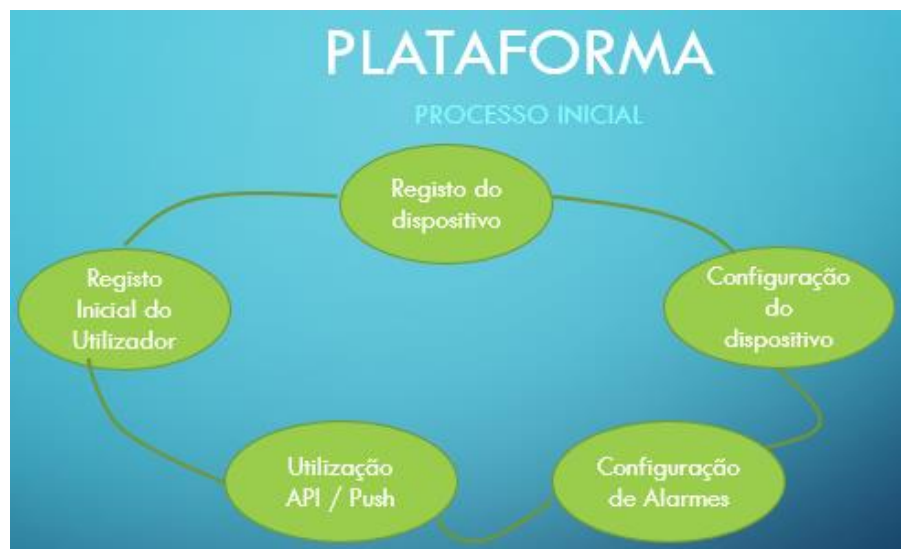


Figura 13 - Plataforma - Processo de Configuração

4.1.1. Processo de autenticação

Quando é subscrito o serviço, ser-lhe-á dado as credenciais de acesso à aplicação. As mesmas devem ser introduzidas para se autenticar na plataforma e poder usufruir dos respetivos serviços.

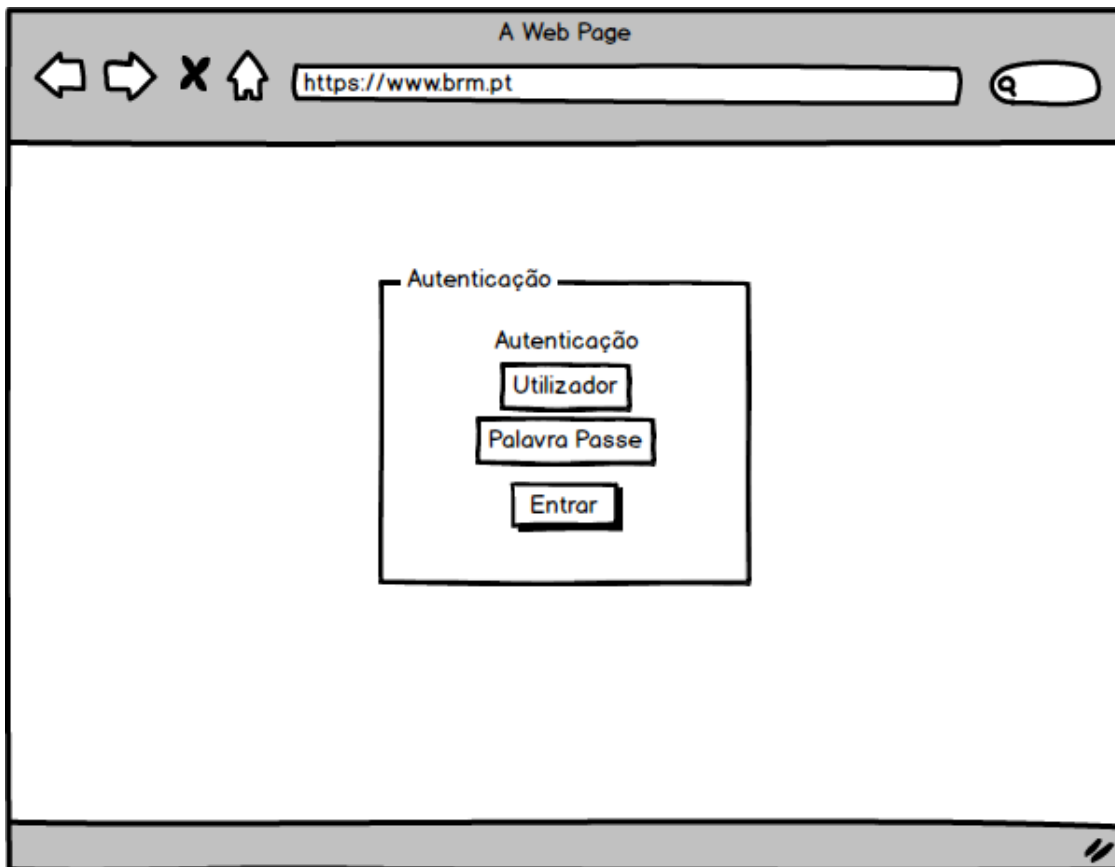


Figura 14 - Plataforma - Mockup Autenticação

4.1.2. Painel de controlo

Nesta página, é permitido visualizar a informação sobre os dispositivos e o serviço dos mesmos em geral. Pode-se ver todos os dispositivos que estão configurados para esta conta na plataforma assim como a última comunicação, ação e estado. Podem ser analisados os alarmes reportados por cada um dos dispositivos assim como os que ainda não foram visualizados.

Temos também a informação dos alarmes, em acumulado, de cada um deles desde o início do mês e do ano. Da mesma forma existe a possibilidade de ter acesso às mensagens despoletadas por cada um dos dispositivos, tanto ao nível de acumulados por mês e ano, assim como das mensagens novos, pendentes de leitura. Pode-se ter acesso à utilização da API e PUSH *Service* através de gráfico de utilização.

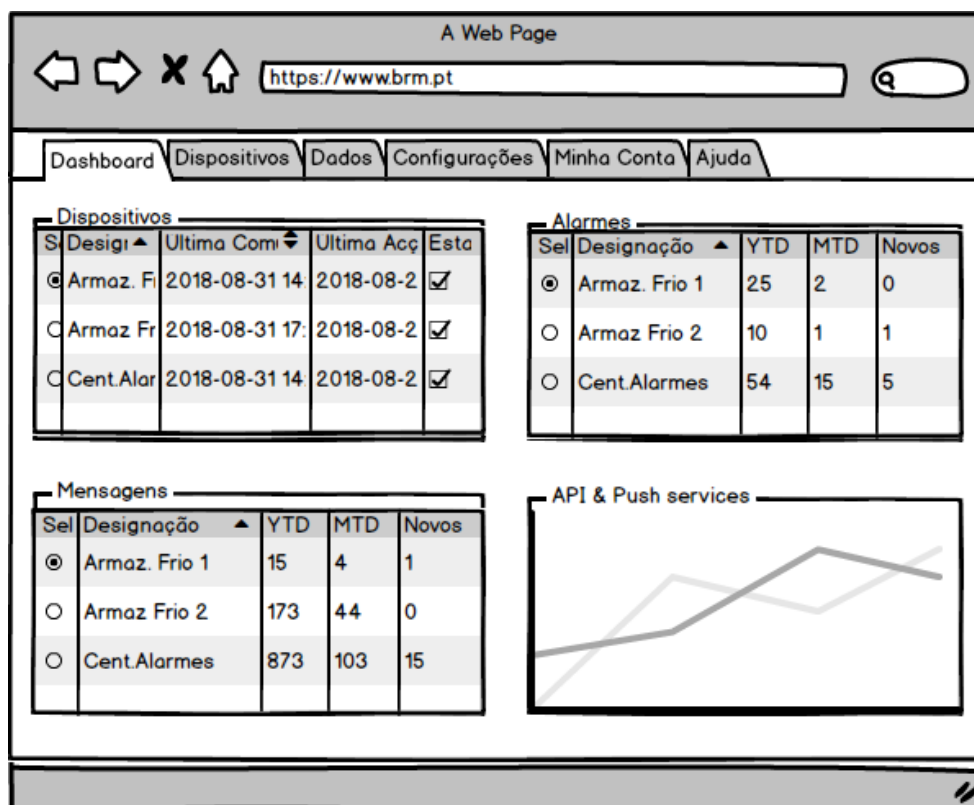


Figura 15 - Plataforma - Mockup Painel de Controlo

4.1.3. Dispositivos

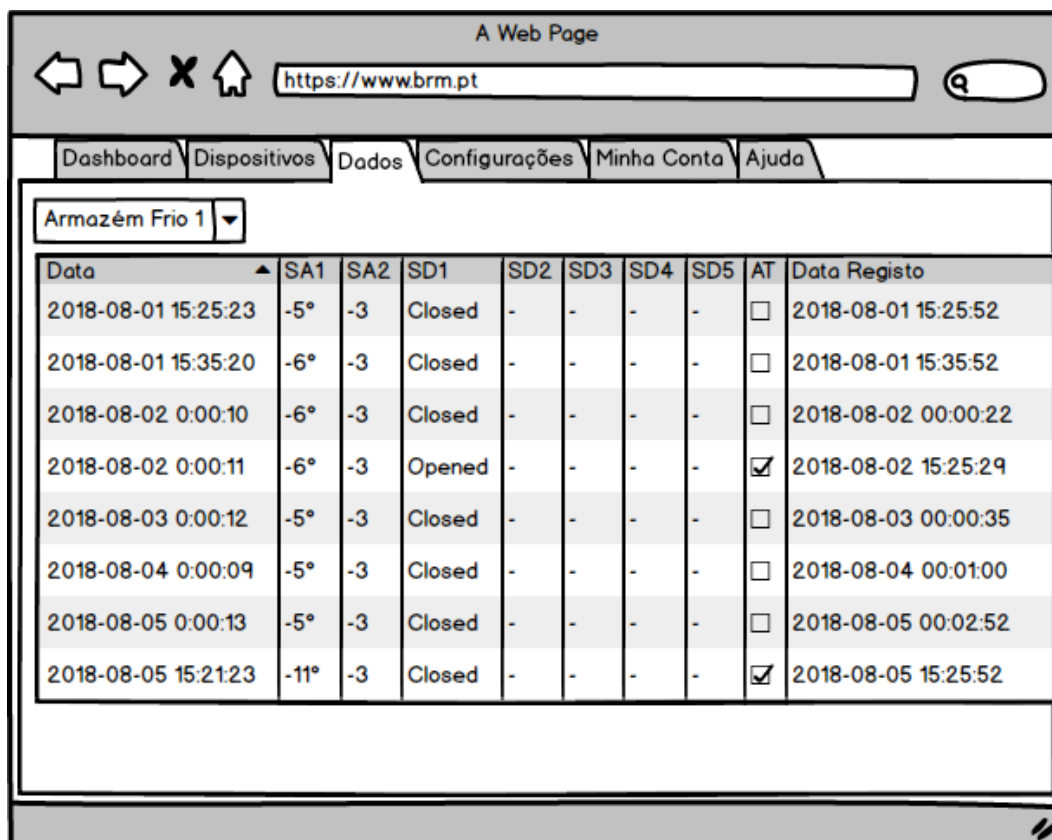
Nesta secção pode-se aceder aos dispositivos já registados por nós na plataforma. Neste tabulador temos acesso aos dados dos dispositivos, tais como, data da última comunicação, data da última configuração, enviar a configuração da mesma assim como ativar ou desativar o dispositivo.

Selecção	Designação	Última Comunicação	Última Acção	Última Configuração	Conf.	Estado
<input checked="" type="radio"/>	Armaz. Frio 1	2018-08-31 14:24:55	2018-08-24 11:21	2018-06-24 07:55	CFG	<input checked="" type="checkbox"/>
<input type="radio"/>	Armaz. Frio 2	2018-08-31 17:24:55	2018-08-24 11:21	2018-06-24 07:55	CFG	<input checked="" type="checkbox"/>
<input type="radio"/>	Cent.Alarmes	2018-08-31 14:24:55	2018-08-24 11:21	2018-06-24 07:55	CFG	<input checked="" type="checkbox"/>
<input type="radio"/>	Posto Transf 1	2018-05-31 14:24:55	2018-08-24 11:21	2018-06-24 07:55	CFG	<input type="checkbox"/>

Figura 16- Plataforma - Mockup Dispositivos

4.1.4. Dados

Nesta secção pode-se aceder aos dados de cada um dos dispositivos, sendo que para isso, será necessário seleccionar de qual dos dispositivos se pretende visualizar os dados.



The screenshot shows a web browser window titled 'A Web Page' with the address bar containing 'https://www.brm.pt'. The navigation menu includes 'Dashboard', 'Dispositivos', 'Dados', 'Configurações', 'Minha Conta', and 'Ajuda'. A dropdown menu is open, showing 'Armazém Frio 1'. Below this is a table with the following data:

Data	SA1	SA2	SD1	SD2	SD3	SD4	SD5	AT	Data Registo
2018-08-01 15:25:23	-5°	-3	Closed	-	-	-	-	<input type="checkbox"/>	2018-08-01 15:25:52
2018-08-01 15:35:20	-6°	-3	Closed	-	-	-	-	<input type="checkbox"/>	2018-08-01 15:35:52
2018-08-02 0:00:10	-6°	-3	Closed	-	-	-	-	<input type="checkbox"/>	2018-08-02 00:00:22
2018-08-02 0:00:11	-6°	-3	Opened	-	-	-	-	<input checked="" type="checkbox"/>	2018-08-02 15:25:29
2018-08-03 0:00:12	-5°	-3	Closed	-	-	-	-	<input type="checkbox"/>	2018-08-03 00:00:35
2018-08-04 0:00:09	-5°	-3	Closed	-	-	-	-	<input type="checkbox"/>	2018-08-04 00:01:00
2018-08-05 0:00:13	-5°	-3	Closed	-	-	-	-	<input type="checkbox"/>	2018-08-05 00:02:52
2018-08-05 15:21:23	-11°	-3	Closed	-	-	-	-	<input checked="" type="checkbox"/>	2018-08-05 15:25:52

Figura 17 - Plataforma - Mockup Dados por Dispositivo

4.1.5. Configuração

Nesta secção temos a possibilidade de registar os nossos novos dispositivos assim como de configurar cada um deles de acordo com as suas funcionalidades. Cada um dos dispositivos pode acumular mais do que uma funcionalidade. Para cada porta, terá a respetiva configuração, que poderá ser conjugada com filtros/condições previamente criadas.

A Web Page
https://www.brm.pt

Dashboard Dispositivos Dados Configurações Minha Conta Ajuda

Registrar Novo Dispositivo

Código de Registo Novo Dispositivo Designação a Atribuir Validar

Armazém Frio 1

SA1	Sensor Temperatura	Sensor Temperatura Frig1	min	Max	<input type="checkbox"/> Activa AT	Sensor Temp. c1	<input checked="" type="checkbox"/> Send Data on temp.Chg.
SA2	Sensor Temperatura	Sensor Temperatura Frig2	min	Max	<input checked="" type="checkbox"/> Activa AT	Condição	<input checked="" type="checkbox"/> Send Data on temp.Chg.
SD1	Sensor Porta	Porta Armazém	Opened	Closed	Condição	<input checked="" type="checkbox"/> Activa AT	
SD2	Botão de Estado	Designação	Designação ON	Designação OFF	Condição	<input type="checkbox"/> Activa AT	
SD3	Botão de Pressão	Designação	Designação ON	Designação OFF	Condição	<input type="checkbox"/> Activa AT	
SD4	Selecionar Tipo	Designação	Designação ON	Designação OFF	Condição	<input type="checkbox"/> Activa AT	
SD5	Selecionar Tipo	Designação	Designação ON	Designação OFF	Condição	<input type="checkbox"/> Activa AT	

Armazém Frio 1

Tipo Sensor Designação

Sensor Temperatura Sensor Temp. c1

Tipo	Condição	Valor	Unidades		
Temperatura	Maior que	min	°c)	Ou <input checked="" type="checkbox"/> Activa AT

Apagar Condições Apagar último Gravar Condição

Enviar configuração dispositivo Data Última Configuração Estado Última Configuraç

Condição

Temperatura > 5, °c, Sim, ou
Temperatura < -10, °c, Não, ou
Tempo > , 30, min, Não,

Figura 18- Plataforma - Mockup Configurações por Dispositivo

4.1.6. A Minha Conta

No separador “Minha Conta”, existem 4 temas diferentes: utilizador, que será o utilizador que se encontra a utilizar a plataforma e a configurar os dispositivos, alarmes e condições; empresa, com os dados da empresa cliente; gestão de utilizadores, com gestão de outros utilizadores da conta; definições de conta, com configurações de utilização de smsgateway externo e/ou Push API.

The screenshot shows a web browser window with the URL <https://www.brm.pt>. The page has a navigation menu with the following items: Dashboard, Dispositivos, Dados, Configurações, Minha Conta (selected), and Ajuda. The main content area is titled 'Minha Conta' and is divided into four sections:

- Utilizador:** A form with fields for 'Nome', 'Morada', 'Localidade', 'Codigo Postal', 'Cidade', 'Telefone', 'Telemovel', and 'Email'.
- Empresa:** A form with fields for 'Nome', 'Morada', 'Localidade', 'Codigo Postal', 'Cidade', 'Telefone', and 'Email'. It also includes a 'Dispositivos Contractados' dropdown menu set to '5' and a 'Certificado API' field containing the value 'l3dwD67YK*4c6\$adkbTv/jv-VT'BL'.
- Definições de Conta:** A section with a sub-header 'Definições de Conta' and two checked checkboxes: 'Gateway SMS Pessoal' and 'Push API'. Below these are input fields for 'Gateway URL' and 'Push URL'.
- Gestão Utilizadores:** A table listing users with columns for 'Designação', 'Utilizador', 'Iniciais', 'Data Expiração', and 'Activo'.

Designação	Utilizador	Iniciais	Data Expiração	Activo
Administrador	admin	ADM	2018-12-31	<input checked="" type="checkbox"/>
João Silva	jsilva	JSL	2018-12-31	<input checked="" type="checkbox"/>
Ana Figueiredo	afigueira	AFL	2018-12-31	<input checked="" type="checkbox"/>
Sandra Gonçalves	sgoncas	COB	2018-12-31	<input checked="" type="checkbox"/>
Tekas Isabel	tsantos	Tekas	2018-12-31	<input checked="" type="checkbox"/>
Anabela Silva	asilva	AS	2018-12-31	<input checked="" type="checkbox"/>
Joana Silva	jnsilva	JFS	2018-12-31	<input type="checkbox"/>
João Santos	jsantos	BBLS	2018-12-31	<input checked="" type="checkbox"/>

Figura 19- Plataforma - Mockup Acesso aos Dados de Conta

4.1.7. Ajuda

Nesta secção temos toda a informação sobre o equipamento, a plataforma, as configurações e interações entre os aparelhos.

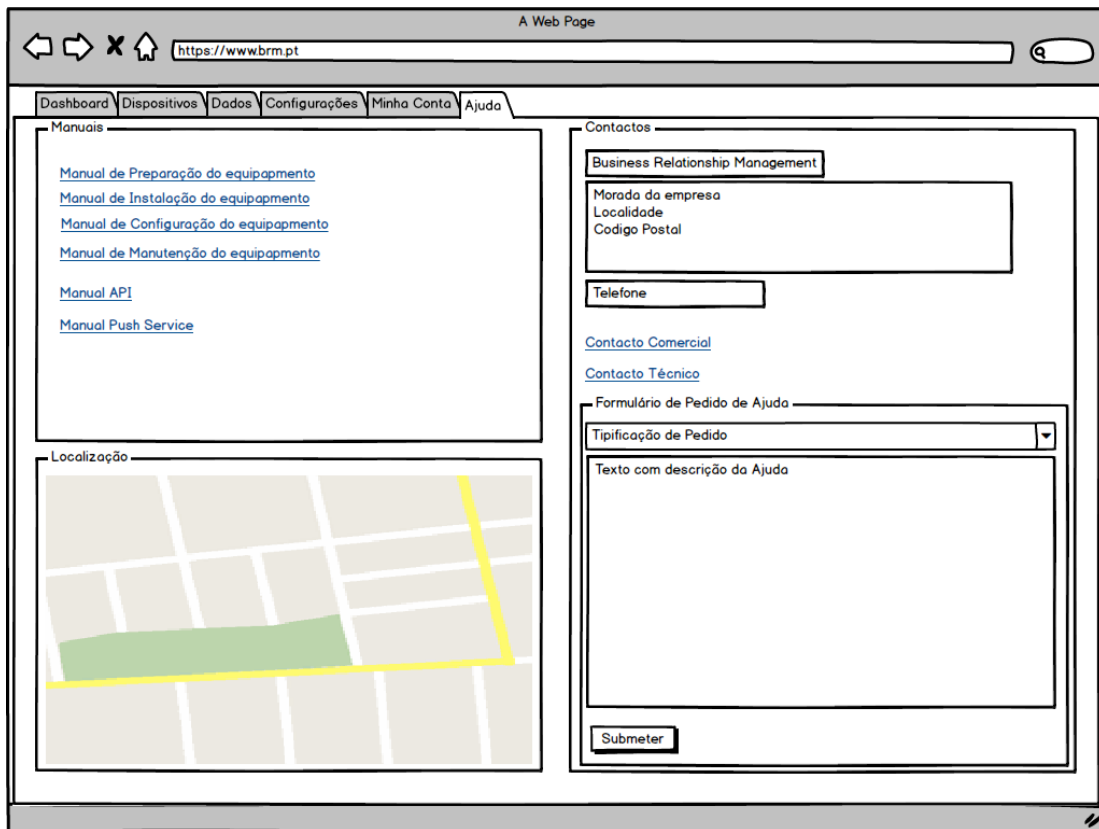


Figura 20- Plataforma - Mockup Ajuda

4.2. Máquinas de Estado dos Serviços associados à plataforma

Neste capítulo serão descritos os serviços associados à plataforma e que serão utilizados de acordo com o tipo de serviço contratado pelos utilizadores.

4.2.1. Serviço de envio de mensagens por SMS

Este serviço vai verificar na plataforma quais as mensagens de texto (SMS) por enviar e vai tratar cada uma delas.

Máquina de Estados Envio de SMS

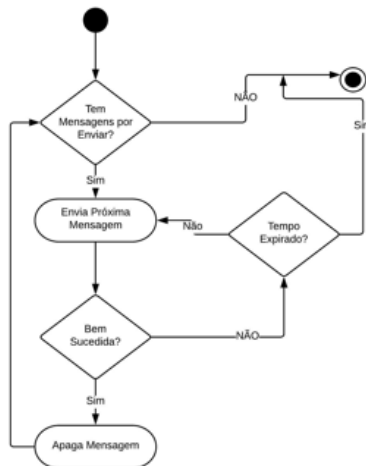


Figura 21 - Máquina de Estados – Envio de Mensagens por SMS

4.2.2. Serviço SMS Recebidos no Gateway Interno

Neste serviço irão ser tratadas todas as mensagens recebidas pela Gateway, registando as mesmas na base de dados e executando as que forem enviadas pelos dispositivos e que estejam abrangidas pelo protocolo de comunicação.

Máquina de Estados Mensagens Recebidas no Gateway Interno

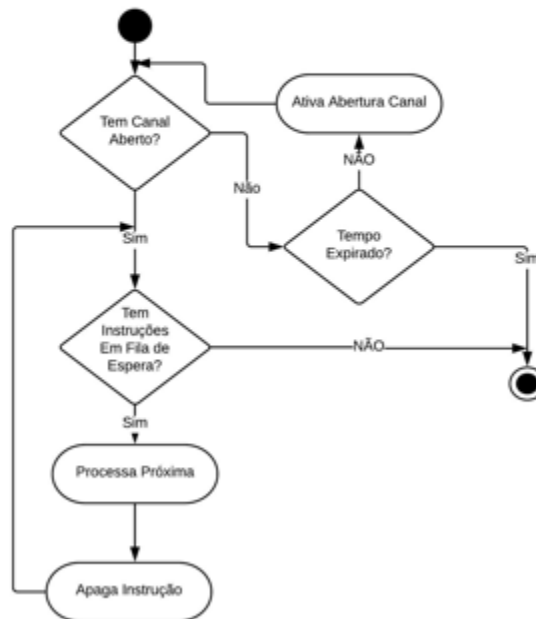


Figura 22 - Máquina de Estados - Receção de Mensagens

4.3. Máquina de estados do dispositivo

Nesta secção estão identificados todos os diagramas de estados do dispositivo. Será identificado em cada um deles quais as variáveis a validar, os dados a recolher e a respetiva transmissão dos mesmos.

4.3.1. Inicialização de Equipamento

O equipamento quando é ligado/inicializado terá de fazer uma série de validações, leituras de configuração, inicialização de módulos e ciclos de forma a garantir que o equipamento está em condições de operar. Essas informações devem ficar registadas na memória não volátil do equipamento assim como deve ser visível para o operador que o instala, através de combinações de estados e cores do led do equipamento. O dispositivo, de acordo com o hardware instalado, nomeadamente módulo GPS, módulo de Wireless, Módulo de Painéis Solares e Módulo de GSM, deve ativar ou desativar os módulos, assim como os ciclos de verificação e execução.

Máquina de Estados Inicialização do Dispositivo

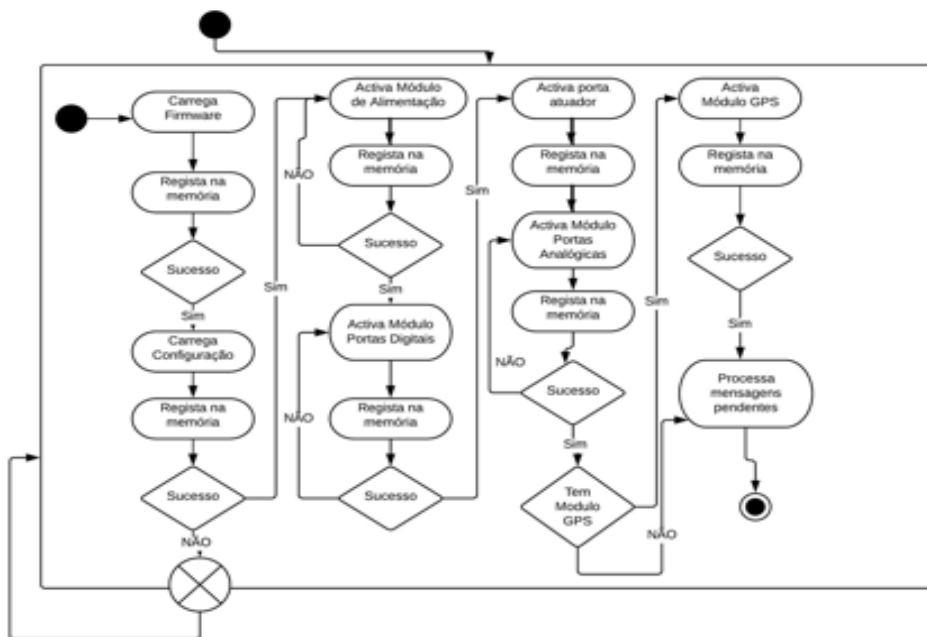


Figura 23 - Máquina de Estados - Inicialização de Equipamento

4.3.2. Ativação Módulo e Ciclos de Geolocalização

Neste processo será validado se o equipamento tem o módulo de GPS incorporado e caso tenha, será inicializado o módulo assim como os ciclos de verificação e registo de dados de geolocalização. Caso o equipamento não tenha o módulo instalado, dado que se trata de um módulo opcional, o equipamento permite ter um valor estático de geolocalização (latitude e longitude).

Máquina de Estados

Ativação Módulo e Ciclos de Geolocalização

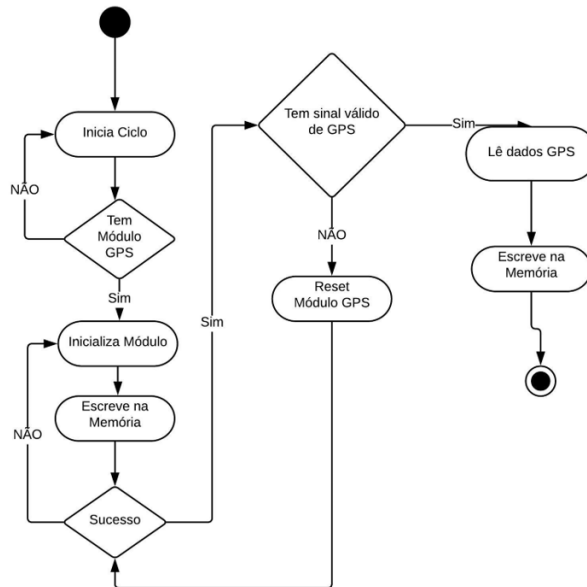


Figura 24- Máquina de Estados - Ativação de Módulo e Ciclos de Geolocalização

4.3.3. Ativação Módulo e Ciclos de GSM

Neste processo será validado se o equipamento tem o módulo de GSM incorporado e caso tenha, será inicializado o módulo assim como os ciclos de verificação, ligação à rede e transmissão de dados. Caso o equipamento não tenha o módulo instalado, dado que se trata de um módulo opcional, o equipamento permite utilizar outro canal de comunicação, tais como Wireless ou comunicação por satélite.

Máquina de Estados Ativação Módulo e Ciclos de GSM

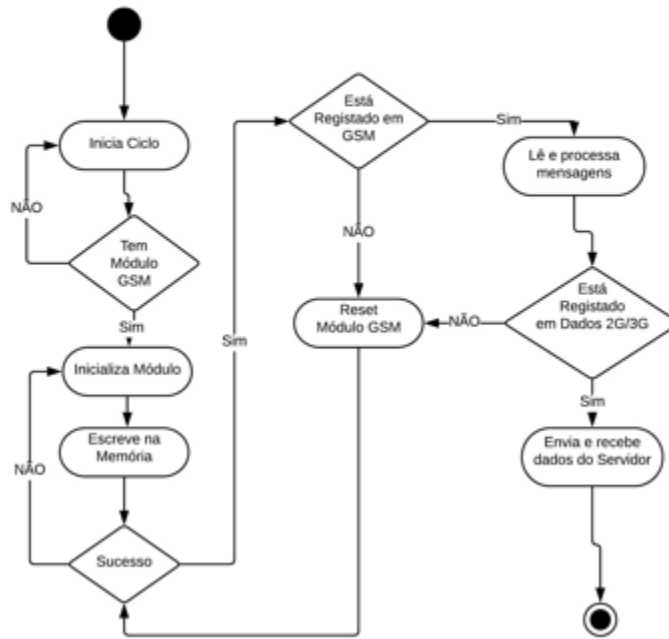


Figura 25- Máquina de Estados - Ativação do Módulo e Ciclos de GSM

4.3.4. Ativação do Módulo e Ciclos da Rede sem Fios

Neste processo irá ser validado se o equipamento tem o modulo de Wireless incorporado e caso tenha, será inicializado o módulo assim como os ciclos de verificação, ligação à rede e transmissão de dados. Caso o equipamento não tenha o módulo instalado, dado que se trata de um módulo opcional, o equipamento permite utilizar outro canal de comunicação, tais como GSM ou comunicação por satélite.

Máquina de Estados

Ativação Módulo da Rede sem Fios

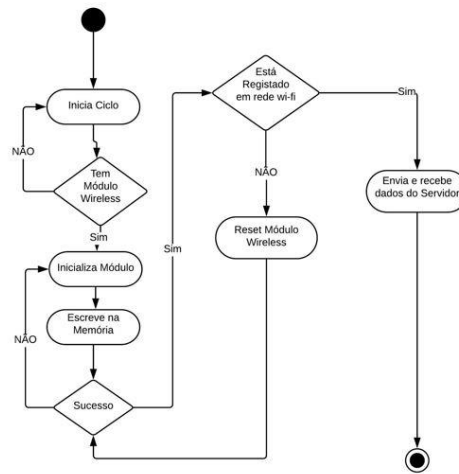


Figura 26- Máquina de Estados - Ativação do Módulo e Ciclos da Rede sem Fios

4.3.5. Ativação do Módulo e Ciclos do Painel Solar

Neste processo irá ser validado se o equipamento tem incorporado o modulo de Painel Solar e caso tenha, será inicializado o módulo assim como os ciclos de verificação, transformação e armazenamento de energia. Caso o equipamento não tenha o módulo instalado, dado que se trata de um módulo opcional, o equipamento não ativará os processos relacionados com o mesmo, mas continua a validar o primeiro ponto (está instalado).

Máquina de Estados

Ativação Módulo e Ciclos do Painel Solar

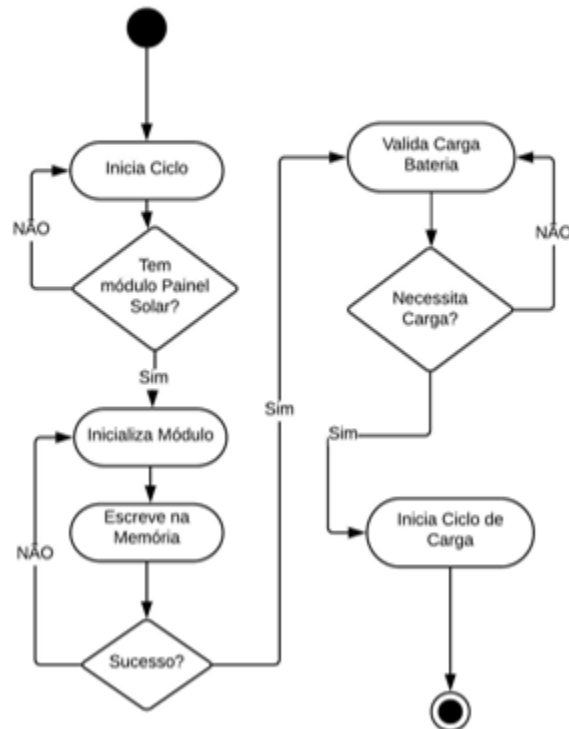


Figura 27- Máquina de Estados - Ativação do Módulo e Ciclos do Painel Solar

4.3.6. Alimentação externa do dispositivo

Logo que se dê início ao processo, a primeira variável a validar é se o dispositivo está equipado com entrada de alimentação externa. Caso esteja equipado, valida se a mesma tem tensão e qual o valor da tensão, regista em memória para posteriormente ser enviada ao servidor. No final aguarda o tempo limite de espera para iniciar novamente o processo.

Máquina de Estados Alimentação do Dispositivo



Figura 28- Máquina de Estados - Estado e Ciclos de Alimentação Externa do Dispositivo

4.3.7. Validação de Carga de Bateria Interna

Iniciando-se o processo de validação de carga da bateria e tendo validado que se consegue estabelecer ligação entre o dispositivo e a bateria, verificamos qual os valores de carga da mesma, registando os mesmos em memória não volátil. Se os valores de carga foram inferiores a 80% e os valores de temperatura previamente recolhidos foram entre os 10°C positivos e os 55°C positivos, dar-se-á a ativação do processo de carregamento. Caso não se consiga comunicar entre o dispositivo e a bateria, será regista a informação e o ciclo ficará encerrado.

Máquina de Estados Carga da Bateria Interna

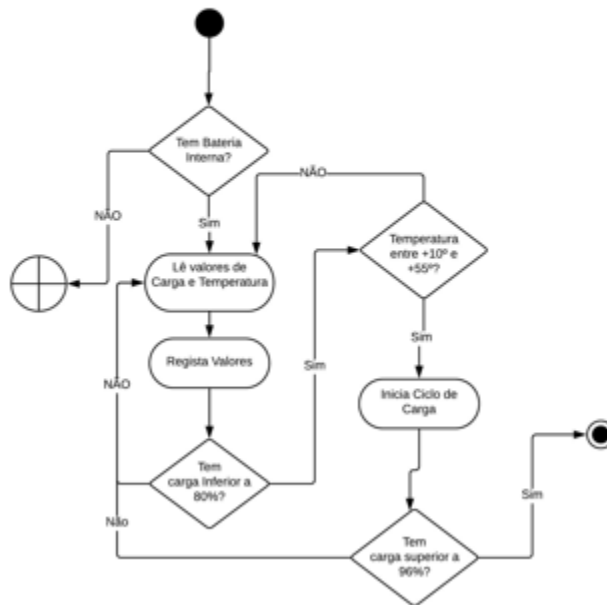


Figura 29- Máquina de Estados – Validação de Carga da Bateria Interna

4.3.8. Processo de Carregamento da Bateria Interna

Neste ciclo será explicado o processo de carregamento da bateria. Por norma, as baterias têm algumas particularidades que devemos ter em conta no momento de implementar processos de carregamento, nomeadamente níveis de humidade, temperatura e valores de carga existentes. Neste diagrama pode-se concluir que existe um estado do processo (Ativo e Inativo) que é controlado pela máquina de estados dos níveis de bateria e que permite ativar e desativar este processo de acordo com a conectividade e as leituras registadas nesse processo. Por questões de segurança e de prolongamento do tempo de vida útil da bateria, os carregamentos devem ser efetuados em condições normais de temperatura ($>10^{\circ}\text{C}$ e $<55^{\circ}\text{C}$) em que a carga do equipamento se encontra abaixo dos 90%.

Máquina de Estados Processo de carregamento da Bateria Interna

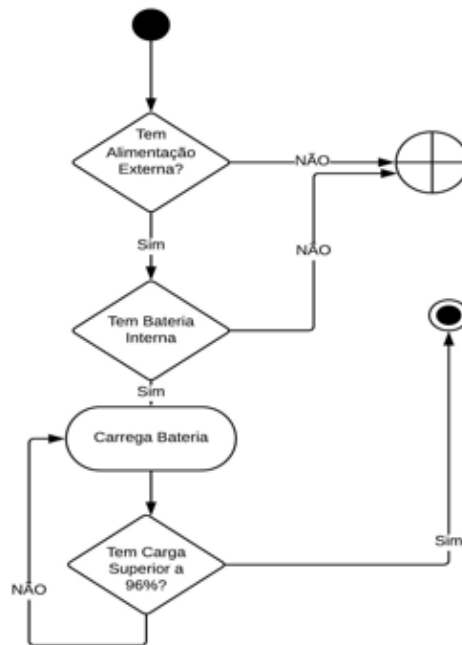


Figura 30- Máquina de Estados - Carregamento da Bateria Interna

4.3.9. Estado de Ligação Rede sem Fios

Quando o módulo está instalado, periodicamente o dispositivo vai validar qual o estado do módulo. Basicamente irá validar se está com canal disponível para comunicação. Caso não esteja registado, tentará e dará essa informação ao processador do equipamento.

Máquina de Estados Estado da Ligação Rede Sem Fio

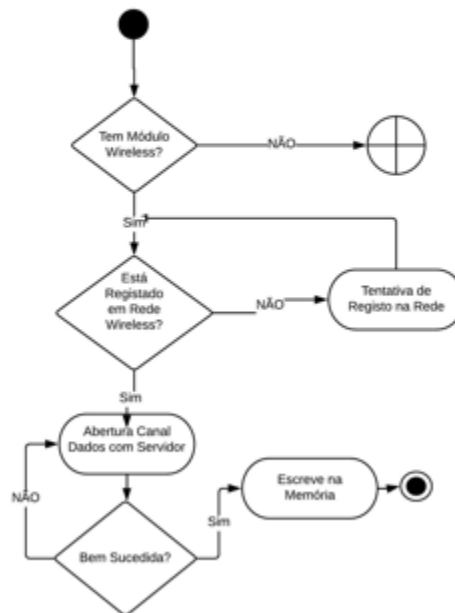


Figura 31- Máquina de Estados - Ligação a Redes sem Fios

4.3.10. Estado de Ligação GSM

Quando o módulo está instalado, periodicamente o dispositivo vai validar qual o estado do módulo. Basicamente irá validar se está com canal disponível para comunicação. Caso não esteja regista e dá essa informação ao processador do equipamento.

Máquina de Estados Estado da Ligação GSM

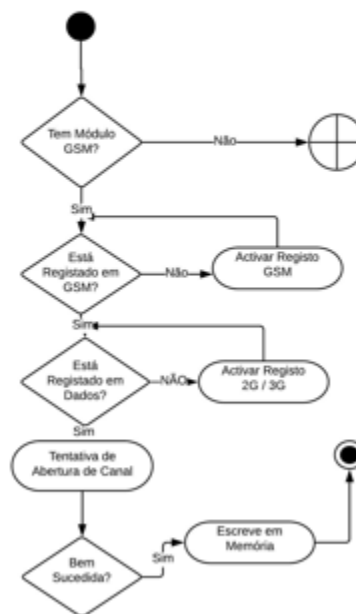


Figura 32- Máquina de Estados - Estado da Ligação GSM

4.3.11. Estado de Sinal GPS

Quando o módulo está instalado, periodicamente o dispositivo vai validar qual o estado do módulo, nomeadamente da qualidade de sinal, triangulação entre satélites, tempo limite de reset entre outras. Caso o sinal seja de fraca qualidade, sem dados de GPS ou com coordenadas inválidas o mesmo terá de efetuar um “reset” após findado o tempo limite. A informação sobre o estado do mesmo ficará registada ao nível do dispositivo e comunicada ao servidor.

Máquina de Estados Estado do Sinal GPS

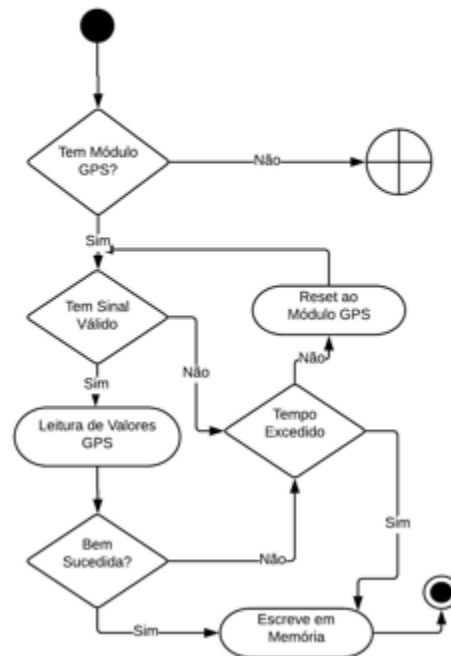


Figura 33- Máquina de Estados - Módulo de GPS

4.3.12. Estado de Painel Solar

Uma vez iniciado o processo e verificado que o dispositivo se encontra equipado com painel solar, irá dar-se início às verificações periódicas nomeadamente comunicação do mesmo com o dispositivo, leituras de transformação de energia, leituras de carga e tensão.

Máquina de Estados Estado do Painel Solar

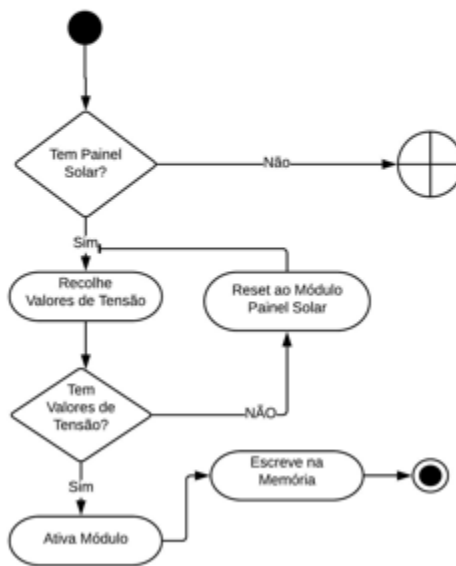


Figura 34- Máquina de Estados - Estado Painel Solar

4.3.13. Estado de Porta Digital

Neste processo, será analisado se o equipamento tem alguma porta digital configurada e qual o estado dela. Este ciclo será efetuado periodicamente sempre que o equipamento esteja ligado e a porta configurada. O processo demonstrado será o mesmo para qualquer uma das portas digitais (de 1 a 4).

Máquina de Estados Estado da Porta Digital

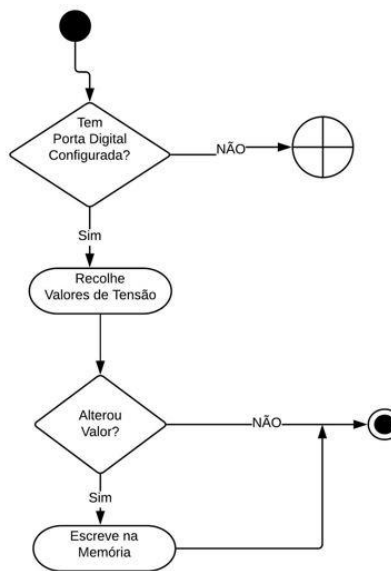


Figura 35- Máquina de Estados - Estado de Porta Digital

4.3.14. Estado de Porta Analógica

Neste processo, é validada a existência no equipamento de alguma porta analógica configurada e qual o estado dela. Este ciclo será efetuado periodicamente sempre que o equipamento esteja ligado e a porta configurada. O processo demonstrado será o mesmo para qualquer uma das portas analógicas (1 ou 2).

Máquina de Estados Estado da Porta Analógica

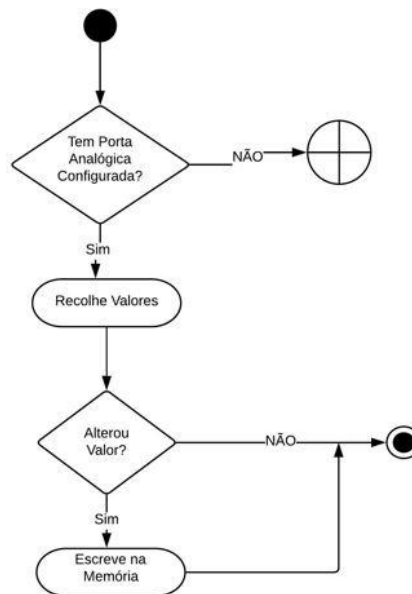


Figura 36- Máquina de Estados - Estado de Porta Analógica

4.3.15. Estado da Ligação de Dados

Neste processo será validada a comunicação entre o equipamento e a plataforma, leia-se servidores, através de qualquer um dos componentes disponíveis para o efeito. De salientar que existe um nível de prioridades na utilização do canal de comunicação. Por exemplo, se existir dois canais de comunicação disponíveis, será dada prioridade ao canal com maior largura de banda.

Máquina de Estados Estado do Ligação de Dados

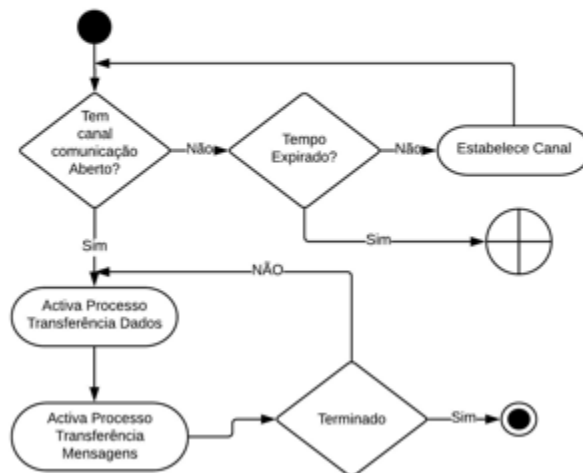


Figura 37- Máquina de Estados - Estado da Ligação de Dados

4.3.16. Tratamento de Comunicações Recebidas

O Equipamento estará sempre a tentar manter uma comunicação estabelecida com o servidor, contudo estará sempre dependente de fatores externos tais como canal de comunicações. Sempre que exista comunicação estabelecida, o equipamento periodicamente fará sincronização com o servidor, envia dados e recebe dados. Esses dados serão depositados numa fila e processados por ordem cronológica de registo no dispositivo.

Máquina de Estados Tratamento de Comunicações Recebidas

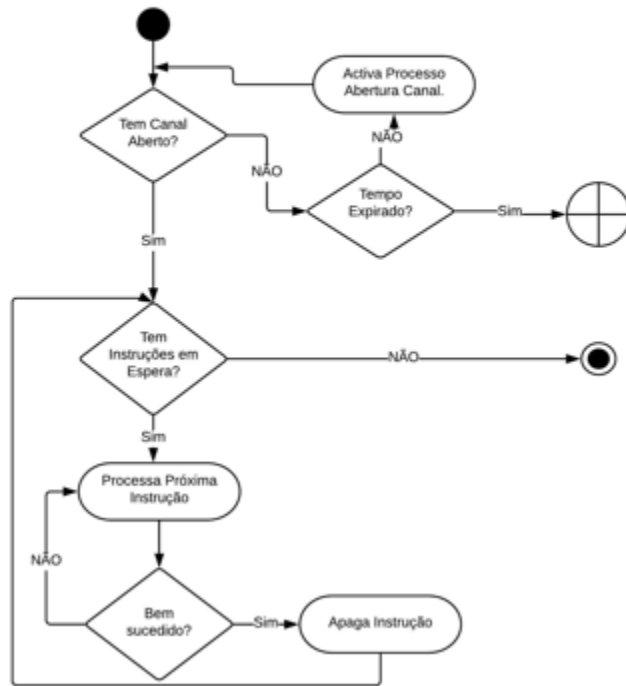


Figura 38- Máquina de Estados - Tratamento de Comunicações Recebidas

4.3.17. Tratamento de Dados a Enviar

O Equipamento estará sempre a tentar manter uma comunicação estabelecida com o servidor, contudo estará sempre dependente de fatores externos tais como canal de comunicações, tendo em conta que o mesmo se encontra estabelecido, mas validando sempre antes de iniciar o processo de envio de dados. Sempre que exista comunicação estabelecida, o equipamento periodicamente fará sincronização com o servidor, envia dados e recebe dados. Esses dados serão depositados numa fila e processados por ordem cronológica de registo no dispositivo.

Máquina de Estados Tratamento de Comunicações a Enviar

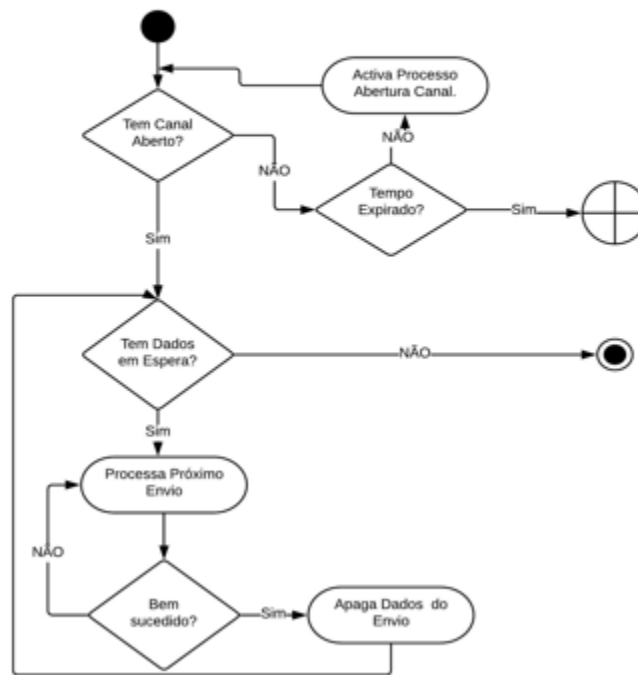


Figura 39- Máquina de Estados - Tratamento Dados a Enviar

4.3.18. Ativar Atuador

Esta funcionalidade tem por base receber uma instrução da central para Ativar o Atuador. Esta função, quando ativada emite um sinal que permite ativar remotamente outras funcionalidades como ligar ou desligar camaras, acionar alarmes, entre outras.

Máquina de Estados Ativar Atuador

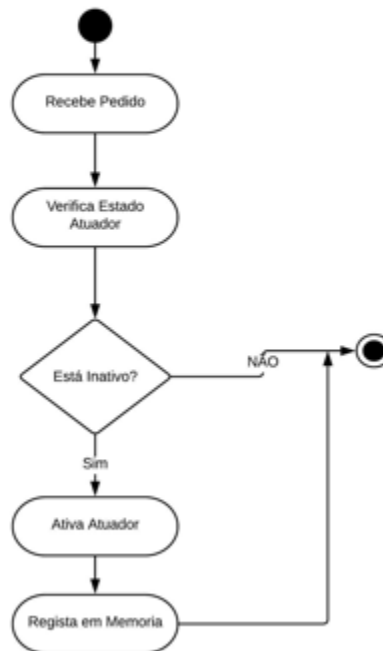


Figura 40- Máquina de Estados - Ativação Atuador

5. Validação dos requisitos do modelo

O objetivo essencial de um artefacto é garantir que o mesmo vem colmatar uma necessidade não abrangida por outras soluções atualmente disponíveis. Para o efeito foram questionadas algumas organizações de setores, cuja atividade de negócios abrange atividades e recursos que possam tirar partido da aplicação deste artefacto.

5.1. Consulta a interessados

A base para recolha da amostra foram as empresas atualmente em carteira de clientes na empresa Inosat, atualmente com 5678 contactos, dos quais foram selecionados 50 clientes, com base nos seguintes critérios, definidos em reunião com a hierarquia:

1. Pertencer a uma área de negócio com atividade operacional relevante para a aplicação deste artefacto em instalações fixas;
2. Já ser cliente do produto que está na base tecnológica deste artefacto, tendo por isso o conhecimento prévio da sua utilização em ambiente móvel;
3. Ter uma dimensão operacional relevante em termos do produto base já instalado (mínimo 5 a 6 equipamentos instalados em veículos);

Relativamente ao primeiro critério, as áreas de negócio foram selecionadas em reunião com dois peritos das áreas comercial e dois peritos da área técnica da empresa, sendo propostos os setores alimentar, soluções energéticas, telecomunicações, segurança e transportes, após discussão sobre o tipo de operações e instalações fixas existentes. Dentro de cada setor foram identificadas empresas que cumpriam os dois critérios seguintes (dois e três) e na proporção em que o setor está representado na base de dados, tendo sido selecionados para cada setor:

- 17 empresas no setor alimentar;
- 3 empresas no setor da energia;
- 6 empresas no setor das telecomunicações;
- 14 empresas no setor da segurança;
- 4 empresas no setor dos transportes;

- 6 empresas de outros setores.

Em reunião com dois elementos da área comercial e três elementos da área técnica foram também definidas as questões a colocar às empresas, no sentido de recolher informação sobre a eventual necessidade de controlo periódica de informação em locais remotos, assim como tipificar o sistema necessário para obter essa informação.

As questões 1 a 4 foram definidas com o objetivo de tipificar a organização.

1. Nº de Colaboradores da Organização
2. Empresa de Âmbito Nacional ou Internacional
3. Tipificação da Organização. Serviços / Industrial /Sector Público
4. A Organização tem responsável dos Ativos

A questão 5 foi definida com o objetivo de saber se a organização está interessada no artefacto.

5. A empresa tem necessidade de monitorizar algo em locais remotos?

As questões 6 a 13 foram definidas com o objetivo de tipificar a utilização do produto, tendo em conta a experiência dos peritos.

6. Em caso afirmativo, que tipo de monitorização necessita:
 - a. Acessos
 - b. Temperaturas
 - c. Movimentos
 - d. Ativações / desativações (elétricas ou mecânicas)
 - e. Outras
 - i. Se sim quais?
7. Quem realiza a monitorização: Interno ou Externo (subcontratado)?
8. Os locais a monitorizar são visitados periodicamente por superiores?
9. Qual o método atualizado atualmente?
10. Deseja ter um método automatizado que aumente a frequência de monitorização e reduza os custos inerentes? Numa escala de 1 a 10, sendo que 1 é muito pouco e 10 é muito, qual o valor que atribuíu a essa necessidade?
11. Qual a frequência de monitorização?
 - a. Minuto a minuto
 - b. 15 em 15 minutos
 - c. 30 em 30 minutos
 - d. 1 em 1 hora
 - e. 6 em 6 horas

- f. 12 em 12 horas
 - g. 24 em 24 horas
12. No local remoto a monitorizar, tem energia elétrica?
13. No local remoto a monitorizar tem rede Wi-Fi/ acesso internet?

5.2. Resultados do questionário

Dos 50 questionários enviados, foram recebidas 50 respostas, tendo sido feito o seguimento do envio junto dessas empresas para as sensibilizar para a necessidade de resposta. A tabela seguinte resume, para cada um dos setores, o resultado das respostas obtidas, podendo-se constatar quais as necessidades das empresas por cada um dos setores de atividade.

Necessidades	Setor de Atividade					
	Alimentar	Energético	Telecomunicações	Segurança	Transportes	Outros
Monitorização remota	17	3	6	14	4	5
Métodos alternativos	17	3	5	14	4	5
Monitorização de Temperatura	17	0	4	12	0	1
Controlo de acessos	1	3	4	2	0	0
Despoletar ações após validação	1	3	4	2		1
Equipado com						
Energia eléctrica no Local	17	3	4	2	4	3
Acesso rede wi-fi no local	5		4	5	2	0

Tabela 6 - Formas de Resolução do Problema - Vantagens e Desvantagens

Pode-se constatar, relativamente a cada necessidade:

- Monitorização remota - a maioria das empresas inquiridas, nos setores de atividade selecionados, têm a necessidade de realizar monitorização remota. Só uma das 50 empresas, nomeadamente associada ao setor agrícola, é que não tem a necessidade de monitorização remota. Todas as restantes empresas, devido à especificidade do seu negócio, têm essa necessidade, sendo que neste momento, todas elas têm métodos alternativos de monitorização, nomeadamente com a visitas regulares a esses locais

remotos para validar o estado. Pode-se concluir que 98% das empresas questionadas tem necessidades de monitorização remota.

- Métodos alternativos - do total de empresas inquiridas, só duas das empresas, uma do setor de telecomunicações e outra de outro sector, é que não verificaram a necessidade de métodos alternativos aos utilizados atualmente. Pode verificar que 96% das empresas têm a necessidade de métodos alternativos que representem maior eficácia aos atualmente em curso.
- Monitorização de temperatura - 34 das 50 empresas têm a necessidade de periodicamente monitorizar temperaturas em locais remotos. Esta constatação de fatos representa 96% das empresas que foram questionadas. Esta necessidade está mais relacionada com o setor alimentar, setor esse, que tem regras apertadas no acondicionamento e preservação dos alimentos armazenadas.
- Controlo de acessos - 10 das 50 empresas inquiridas tem a necessidade de controlar remotamente o acesso a locais ou zonas de acesso restrito. A maioria das empresas com estas necessidades encontram diretamente relacionadas com empresas do setor dos serviços (telecomunicações), energético (derivado ao fenómeno de roubo de cobre) e segurança. Este resultado representa cerca de 20% do total dos inquiridos.
- Despoletar ações após verificação das condições instituídas - verificou-se que 11 das 50 empresas que responderam ao questionário necessitam de despoletar ações posteriores à verificação de determinadas condições antecipadamente determinadas, o que corresponde a 21,8% do total de empresas inquiridas.
- Energia elétrica no local - dos 50 questionários recolhidos, 33 orgânicos têm fonte de energia elétrica nos locais que pretende monitorizar remotamente pelo que os restantes (33%) necessitam de fonte de alimentação alternativa.

- Acesso rede Wi-Fi no local - verificou-se que do total de inquiridos, 16 empresas têm rede Wi-Fi no local que necessitam de monitorizar remotamente. Nestes casos (32%), não seria necessário colocar no equipamento remoto um modem GSM mas sim um módulo Wi-Fi,

A maioria dos inquiridos tem a necessidade de monitorização remota e desejam ter métodos alternativos aos utilizados atualmente. Pode-se concluir também que dentro da monitorização, a maior parte dos inquiridos tem a necessidade de monitorizar acessos remotos (78%) e em segundo lugar vem a necessidade de controlar temperaturas nesses mesmos locais (15%). De realçar que, dos inquiridos com necessidades de monitorização remota, cerca de 7% tem necessidade de despoletar remotamente ações tais como ativar camaras, acionar alarme (Sirene), entre outras e cerca de 66% dos casos tem energia elétrica no local a monitorizar e 32% tem disponíveis acessos à internet via Wi-Fi, aspetos que condicionam o equipamento do artefacto.

Desta recolha de necessidades pode-se concluir que existem funcionalidades que não necessitam de fazer parte do produtor base, podendo ser apresentados como opcionais, aspeto essencial para o desenho de uma solução escalável de custo controlado.

6. Conclusões

Este trabalho de investigação permitiu definir um modelo para um artefacto que permita dar solução a algumas necessidades encontradas em organizações e que possa ser controlado de forma remota, com base na informação recebida e fluxos de trabalho previamente estabelecidos, e com um elevado grau de automatização na execução de tarefas.

A investigação procurou evidenciar o estado da arte, centrando-se no domínio das tecnologias de redes de comunicação e das bases de dados. Através da investigação foi possível definir as características de funcionalidade de um protótipo, sendo as respetivas necessidades validadas através de inquéritos e entrevistas a organizações com necessidades de monitorização remota de instalações e equipamentos.

O modelo de arquitetura proposto nesta investigação pode ser compreendido e utilizado por técnicos e gestores para conceber soluções que tirem partido do artefacto, podendo também proceder a implementações adaptadas às suas necessidades.

6.1. Limitações da Investigação

Embora não pareçam existir soluções disponíveis no mercado nacional, que tenham sido desenvolvidas especificamente para satisfazer as necessidades de monitorização de instalações e equipamentos fixos, considera-se que poderia ter constituído uma mais valia para a definição da arquitetura, a realização de uma análise comparativa de diferentes soluções aplicáveis em ambientes móveis.

A consulta aos interessados poderia igualmente ter sido realizada com mais profundidade, abrangendo um conjunto mais alargado de organizações e eventualmente selecionando alguns dos interessados para a realização de entrevistas prévias, complementando assim as ferramentas de investigação e recebendo contributos para o conteúdo do questionário.

6.2. Investigação Futura

Considera-se que em investigação futura será possível implementar o artefacto e testar o modelo em ambiente de produção, envolvendo um conjunto de organizações dispostas a partilhar a informação gerada, no sentido de avaliar desempenho do artefacto e melhorar o modelo de arquitetura. Esta avaliação permitira igualmente expandir o âmbito do modelo através da inclusão de sensores e equipamentos controlados e da análise das condições ambientais de aplicação.

7. Glossário

Conceito	Descrição
<i>Add-on</i>	Processo de adição.
<i>Back-End</i>	É um sistema que funciona na retaguarda e que não é visível ao utilizador final, mas que permite a execução de funcionalidades que implementam lógica nos processos.
<i>Blob storage</i>	Funcionalidade, nomeadamente em Cloud, que permite armazenar informação não estruturada, em grandes quantidades, tais como vídeos, áudios ou texto.
Bluetooth	Protocolo de comunicação
<i>Cheap</i>	Pequeno dispositivo eletrónico que nos permite aceder a determinadas funcionalidades.
<i>Cloud</i>	Na nuvem.
<i>Column family</i>	Famílias de colunas
<i>Data Shards (sharding)</i>	Tipo de base de dados que particiona grandes bases de dados em mais pequenas e rápidas bases de dados. Esses particionamentos são chamados de “data Shards”
<i>Delay</i>	Atrasos
Dynamo-style	Processo que implementa particionamento de informação, replicação de informação e manipulação de falhas.
<i>E-health</i>	Eletrónico <i>Health</i> é um termo utilizado para definir o tratamento ou práticas relacionadas com a saúde através de meios eletrónicos.
<i>Enhanced Learning</i>	Intensidade no gosto pela aprendizagem com especial foco para a absorção de conceitos de forma eletrónica.
<i>Hosts</i>	Computador ou dispositivo numa rede informática que necessite de endereçamento para funcionar.
Inter-dispositivos	Ligação / comunicação entre dispositivos em que existe uma comunicação entre todos os membros da rede
IoT	Internet das coisas, mas conhecido por IoT, é um paradigma tecnológico que tem por finalidade ligar entre si aparelhos eletrónicos e utilizá-los remotamente via internet.
IPV6	Novo protocolo de comunicação que permite expandir o número de dispositivos em simultâneo na mesma rede do que o seu antecessor.
M2M	Nome atribuído ao processo que permite o estabelecimento de comunicação entre duas máquinas (ponto a ponto).
<i>middlewares</i>	São programas que estão em execução de forma privada e que permitem executar determinadas funções que não queiramos que sejam publicas.
Multicast	Transmissão de informação para múltiplos destinatários ao mesmo tempo.
<i>Proxies</i>	É um servidor que serve como intermediário.
<i>RealTime</i>	Em tempo real
<i>RestFul</i>	Em um <i>web service RESTful</i> , requisições feitas a um URI de recurso extrairá uma resposta que pode estar em XML, HTML, JSON ou algum outro formato.
Smart-home	É o conceito de automatizar ações ou funcionalidades em nossa casa.

<i>SmartCity</i>	São cidades Inteligentes onde existe um uso intensivo de tecnologias sejam elas na vertente de comunicação ou informação.
<i>Software</i>	Programa eletrónico que executa determinadas funcionalidades.
<i>SQL92</i>	Revisão da linguagem SQL após a revisão SQL-89
<i>subqueries</i>	É uma consulta de dados sobre o resulta de uma outra consulta de dados, restringindo dessa forma o volume dados apresentado.
<i>Thread</i>	Espaço de memória reservado a execução de determinado conjunto de instruções que normalmente designamos de código.
<i>Wi-Fi</i>	Nome da tecnologia <i>Wireless Networking Technology</i> que usa ondas de rádio para fornecer um rede e acesso à Internet de alta velocidade

8. Bibliografia

Atzori, L., Iera, A. e Morabito, G., (2010). The Internet of Things: A survey. *Computer. Netw.* 54, 2787–2805. <https://doi.org/10.1016/j.comnet.2010.05.010>, último Acesso 2018-07-17.

Atzori, L., Iera, A. e Morabito, G. (2010). The Internet of Things: A survey, Sítio: <http://www.sciencedirect.com/science/article/pii/S1389128610001568>, Último Acesso: 2018-07-17 10:36:39

Agency, E. (2006). Urban sprawl - Europe's ignored environmental challenge., European Environment Agency disponível em: <https://www.eea.europa.eu/media/newsreleases/urban-sprawl-europes-ignored-environmental-challenge> último Acesso: 2018-09-18

Asay, M. (25 de maio de 2015). Postgres pushes past MySQL in developer hearts. Obtido de TechRepublic: <http://www.techrepublic.com/article/postgres-pushes-past-mysql-in-developer-hearts/> Último Acesso: 25-09-2018

Bardin, L. (1977). *L'Analyse de Contenu.* , France: Presses Universitaires de France

Benbasat, I., Goldstein, D. 3 Mead, M. (1987). The case research strategy in studies of information systems. *MIS quarterly*, 370.

Bluetooth Technology Website, n.d. Sítio: <https://www.bluetooth.com>, Ultimo Acesso: 2018-09-25

Boehm, B. (1981). *Software Engineering Economics* . England: Prentice Hall

Borouh, M. (1998). *Understanding RISK Analysis.* American Chemical Society (ACS) Resources for the future disponível em: http://www.rff.org/mise_does/risk_book.pdf Ultimo acesso: 2018-01-15

Brooks, D. (1996) Brooks, F. P., Jr. (1996) “The Computer Scientist as Toolsmith II,” *Communications of the ACM* , 39 (3), pp. 61-68.

Butler, T. (1997). A case study of user participation in the information systems development process. Association for Information Systems Atlanta, GA, USA ©1997.

Collis, J. e Hussey, R. (2003). Business Research. Red Globe Press

Computer-Networks---A-Tanenbaum---5th-edition.pdf, n.d., Sítio:

<http://iips.icci.edu.iq/images/exam/Computer-Networks---A-Tanenbaum---5th-edition.pdf>

Último Acesso: 20-09-2018

Daft, R. (1985). System influence on Organization Decision-Making: The case of Resource Allocation. Academy of Management Journal

Darke, P., Shanks, G. e Broadbent, M. (2002). Successfully completing case study research: combining rigour, relevance and pragmatism.

Davis, A. (2003). The Art of Requirements Triage. ACM Digital Library, Volume 36 (March 2003).

DB-Engines. (junho de 2015). DB-Engines. Obtido de DB-Engines: <http://db-engines.com/en/ranking> Último Acesso: 25-09-2018

Dooley, J. (2011). Software Development and Professional Practice. Apress. USA

Dooley, L. (2013). Case Study Research and Theory Building. Disponível em :

https://www.researchgate.net/publication/255051693_Case_Study_Research_and_Theory_Building

Dubé, L. e Paré, G. (2003). Rigor in Information Systems Positivist Case Research: Current Practices, Trends, and Recommendations.

Dym, C. (1994). Engineering Design: A Synthesis of Views, England: Cambridge University Press

Fairley, R. (1997). Software Engineering Concepts. McGraw-Hill Companies; 1st edition (July 1, 2017)

Freeman, R. (2010). Strategic Management: A Stakeholder Approach.

Friedman, V. (31 de janeiro de 2008). 10 Principles Of Effective Web Design. Obtido em 09 de setembro de 2015, de Smashing Magazine: <http://www.smashingmagazine.com/2008/01/10-principles-of-effective-web-design/> Último Acesso: 25-09-2018

Gai, K. e Qiu, M. (2018). Optimal resource allocation using reinforcement learning for IoT content-centric services. Appl. Soft Computer. 70, 12–21. Disponível em: <https://doi.org/10.1016/j.asoc.2018.03.056>

Grady, R. (1992). Practical Software Metrics for Project Management and Process Improvement. Prentice Hall, 1992

Gupta, R. (2014), 5things, 2014. 5 Things to Know About MQTT – The Protocol for Internet of Things (5 Things To Know IBM Redbooks Blog), Sítio: https://www.ibm.com/developerworks/community/blogs/5things/entry/5_things_to_know_about_mqtt_the_protocol_for_internet_of_things, ultimo Acesso: 2018-07-18

Hakim, C. (2000). Research Design: Successful Designs for Social and Economic Research Routledge, Abingdon.

Hall, R. (1984). Organizações: estrutura e processos. Prentice Hall Brasil, 2004

Hevner et al. (2004). Ciência do design (metodologia). Obtido de [https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-PT&prev=search&rurl=translate.google.pt&sl=en&sp=nmt4&u=https://en.wikipedia.org/wiki/Design_science_\(methodology\)&xid=17259,15700002,15700023,15700105,15700124,15700149,15700168,15700173,1](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-PT&prev=search&rurl=translate.google.pt&sl=en&sp=nmt4&u=https://en.wikipedia.org/wiki/Design_science_(methodology)&xid=17259,15700002,15700023,15700105,15700124,15700149,15700168,15700173,1) ultimo acesso: 20/08/2018

Hevner, A., Ram, S., March, S. e Park. J. (2004). Design Science Research in Information Systems. MIS Quarterly. Vol. 28, nº 1, pp. 75-105.

Home - NFC Forum | NFC Forum, n.d. IoT and M2M, What's the Difference? 2017. . n.d.

Kalakota, Ravi; Robinson, Marcia (2001): e-Business 2.0. Roadmap for Success, Boston: Addison-Wesley.

Nayyar, D.A., 2017. Best open source databases for IoT applications. Open Source You. Disponível em: <https://opensourceforu.com/2017/05/best-open-source-databases-iot-applications/>

News – NB-IoT standard – Neul [WWW Document], n.d. Sítio <http://neul.com/neul-news/news-nb-iot-standard/> (Último Acesso 17-07-2018).

Kendall, K. e Kendall, J. (2005). Systems Analysis and Design. Prentice Hall

Krug, S. (2000). Don't make me think: a common sense approach to web usability. New Riders

Lai, V. e Mahapatra, R. (1997). Exploring the research in information technology implementation. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0378720697000220?via%3Dihub>

Latorre, A. (2003). La Investigación - Acción. Editorial Grao

Lee, A. (1989). A scientific methodology for MIS case studies. ABI/INFORM.

Leffingwell, D. e Widrig, D. (2003). Managing Software Requirements: A Use Case Approach. Addison-Wesley

Lewis, J. P. (2005). Project, planning, scheduling & control. MCGraw-Hill

Malhotra, Y. (1996). Competitive Intelligence Programs: An Overview. Disponível em: <http://www.brint.com/papers/ciover.htm>

Marcus, A. (4 de July de 2000). Crosscurrents: cultural dimensions and global Web user-interface design. Obtido de ACM Digital Library: <http://dl.acm.org/citation.cfm?id=345238>

Meyer, M. e Zack, M. (1996). The design and implementation of knowledge products.

Miles, M., Huberman, A. e Saldana, J. (2014). *Qualitative Data Analysis: A Methods Sourcebook*. Disponível em: https://www.researchgate.net/publication/272566756_Qualitative_Data_Analysis_A_Methods_Sourcebook

MQTT, Sítio: <http://mqtt.org/>, Último acesso: 2018-07-17 18:10:11

Mintzberg, H. (1973). *The Nature of Managerial Work*. Disponível em: <https://thinkers50.com/blog/the-nature-of-managerial-work/>

n.d. About LoRaWAN™ | LoRa Alliance™, sítio: <https://www.lora-alliance.org/about-lorawan>, Último Acesso: 2018-07-17 12:16:10

n.d. 10 protocolos de IoT que você deveria conhecer, 2017. . Profissionais TI. Sítio <https://www.profissionaisiti.com.br/2017/11/10-protocolos-de-iot-que-voce-deveria-conhecer/> (último Acesso 17-07-2018).

Nonaka, I. e Takeuchi, H. (1997). *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press

Oliveira, I. (s.d.). *A Interação Homem-Computador Através dos Tempos*. Obtido de http://www.google.pt/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwjAkt_X0oXbAhXLwBQKHdRsAScQFggnMAA&url=http%3A%2F%2Fwww.olharcientifico.kinghost.net%2Findex.php%2Ffolhar%2Farticle%2Fdownload%2F28%2F29&usg=AOvVaw3EkY9bZYsKjLGaIAATxS5V Último Acesso: 25-05-2018

Oliveira, M., Maçada, A. e Goldoni, V. (2006). *Análise da Aplicação do Método Estudo de Caso na Área de Sistemas de Informação*. EnANPAD 2006.

Orlikowski, J., Barley, R.. 2001. "Technology and Institutions: What Can Research on Information Technology and Research on Organizations Learn from Each Other?,". USA, University of Minnesota: *MIS Quarterly*, (25: 2)

Pahl, G., Beitz, W. 1996. "Engineering Design: A Systematic Approach", 2d ed. London: Springer

- Petroski, H. (1996). "Invention by Design: How Engineers Get from Thought to Thing". England, Harvard University Press
- Ponchirolli, O. e Fialho, F. (2005). Gestão Estratégica do conhecimento como parte da estratégia empresarial. Disponível em: <http://teses.eps.ufsc.br>
- Pressman, R. S. (2005). Software Engineering: A Practitioner's Approach. MCGraaw-Hill Higher Education
- Rittel, H., Webber M. (1984). "Planning problems are wicked problems", In N. Cross (Ed.), Developments in Design Methodology, Wiley, pp. 135-144.
- Robson, C. (2011). Real World Research. Wiley
- Rotta et al. (2017), Um estudo sobre protocolos de comunicação para amb.pdf, n.d. sitio: <http://www.lbd.dcc.ufmg.br/colecoes/erad/2017/091.pdf> Ultimo Acesso: 2018-07-15 20:18
- Rotta, G., Charao, A. e Dantas, M., n.d. Um estudo sobre protocolos de comunicação para ambientes de Internet das Coisas 4. Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/erad/2017/091.pdf>
- Rummler, G., & Brache, A. (1990). Improving Processes.
- Saunders, M., Lewis, P. e Thornhill, A. (2009). Research methods for business students; 5th Ed.: Pearson Education., New York
- Sekaran, U. e Bougie, R. (2000). Research Methods for Businnes: A Skill Building Approach.
- Sigfox - The Global Communications Service Provider for the Internet of Things (IoT), n.d. Sítio <https://www.sigfox.com/en> Ultimo Acesso: 17-072018
- Simon, H. (1996). The Sciences of the Artificial. 3rd ed. Cambridge: MIT Press
- Singer, R. (2004). 37signals: An Introduction to Using Patterns in Web Design.
- Stake, R. (1995). The art of case study research.

Stewart, T. (1998). Capital intelectual: a nova vantagem competitiva das empresas.

Sutaria, R., n.d. Making Sense of Interoperability - Protocols and Standardization Initiatives in IOT 4. Disponível em:

http://www.cymbet.com/pdfs/Low_power_IoT_ComNet_2013_Mindtree.pdf último acesso: 27/08/2018

Teles, V. (2008), Manifesto Ágil,

Tesch, R. (2013). Qualitative research: Analysis types and software tools. New York: Routledge.

Thomas H. Davenport, L. P. (1998). Working Knowledge.

Vidal, V. (2017), 10 protocolos de IoT que você deveria conhecer), sítio:

<https://www.profissionaisti.com.br/2017/11/10-protocolos-de-iot-que-voce-deveria-conhecer/>,
Último Acesso: 2018-07-17 11:46:42

Walsham, G. (2006). Doing interpretive research.

Weinberg, G. (1975). Introduction to General Systems Thinking.

Wi-Fi Definition is Not Wireless Fidelity | Webopedia Reference , n.d. Sitio

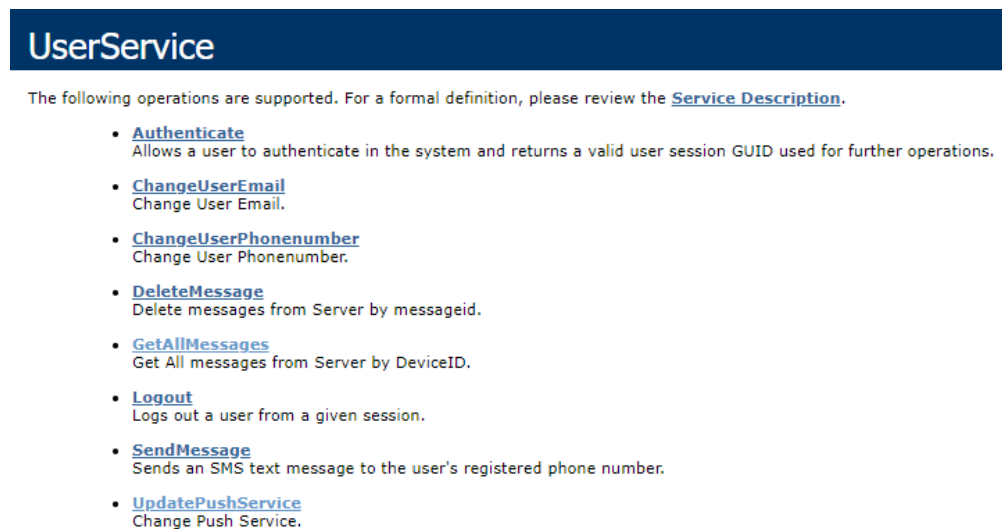
https://www.webopedia.com/DidYouKnow/Computer_Science/wifi_explained.asp Ultimo Acesso: 15-07-2018.

Yin, K. (2014). Case study research: Design and methods (Fifth Edition). USA: Sage.

Anexo I - API

1. UserService

Nesta secção estará representado todos os métodos relacionados com o utilizador. Para poder utilizar a API devemos ter antecipadamente um certificado fornecido pela empresa, que em conjunto com a definição de um utilizador (com direitos de acesso à API) permite proceder ao processo de autenticação no sistema.



UserService

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [Authenticate](#)
Allows a user to authenticate in the system and returns a valid user session GUID used for further operations.
- [ChangeUserEmail](#)
Change User Email.
- [ChangeUserPhonenumber](#)
Change User Phonenumber.
- [DeleteMessage](#)
Delete messages from Server by messageid.
- [GetAllMessages](#)
Get All messages from Server by DeviceID.
- [Logout](#)
Logs out a user from a given session.
- [SendMessage](#)
Sends an SMS text message to the user's registered phone number.
- [UpdatePushService](#)
Change Push Service.

Figura 41 - API Serviço UserService

2. Authenticate

O método *Authenticate*, associado ao serviço *UserService*, é o ponto de início da utilização da API. Com as credenciais e o certificado podemos executá-lo e obter o **aspNetSessionId** necessário para todas as operações futuras.

UserService

Clique [aqui](#) para obter uma lista completa de operações.

Authenticate

Allows a user to authenticate in the system and returns a valid user session GUID used for further operations.

Testar

Para testar a operação utilizando o protocolo HTTP POST, clique no botão 'Invocar'.

Parâmetro	Valor
Certificate:	<input type="text"/>
user:	<input type="text"/>
Password:	<input type="text"/>

Figura 42 – API UserService – Método Authenticate

Certificate: Certificado de Utilização.

User: Utilizador.

Password: Palavra Passe.

Após execução será devolvida a `aspNetSessionId` necessária para invocação de todos os restantes métodos.

3. ChangeUserEmail

UserService

Click [here](#) for a complete list of operations.

ChangeUserEmail

Change User Email.

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
email:	<input type="text"/>
aspNetSessionId:	<input type="text"/>

Figura 43 - API UserService - Alterar Email do Utilizador

aspNetSessionId: Código de identificação da sessão.

newEmailAddress: Novo endereço de email do utilizador.

Devolve o estado verdadeiro se a execução do método foi bem-sucedida ou falso se a execução do método não foi bem-sucedida.

4. ChangeUserPhonenumber

UserService

Click [here](#) for a complete list of operations.

ChangeUserPhonenumber

Change User Phonenumber.

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
newPhonenumber:	<input type="text"/>
aspNetSessionId:	<input type="text"/>

Figura 44 - API - UserService - Alterar Número Telemóvel

aspNetSessionId: Código de identificação da sessão.

newPhonenumber: Novo número de telefone do utilizador.

Devolve o estado verdadeiro se a execução do método foi bem-sucedida ou falso se a execução do método não foi bem-sucedida.

5. DeleteMessage

UserService

Click [here](#) for a complete list of operations.

DeleteMessage

Delete messages from Server by messageid.

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
deviceID:	<input type="text"/>
aspNetSessionId:	<input type="text"/>
messageID:	<input type="text"/>

Figura 45 - API UserService - Apagar Mensagem

aspNetSessionId: Código de identificação da sessão.

messageID: Número identificativo da mensagem.

Devolve o estado verdadeiro se a execução do método foi bem-sucedida ou falso se a execução do método não foi bem-sucedida.

6. GetAllMessages

UserService

Click [here](#) for a complete list of operations.

GetAllMessages

Get All messages from Server by DeviceID.

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
Deviceid:	<input type="text"/>
aspNetSessionId:	<input type="text"/>

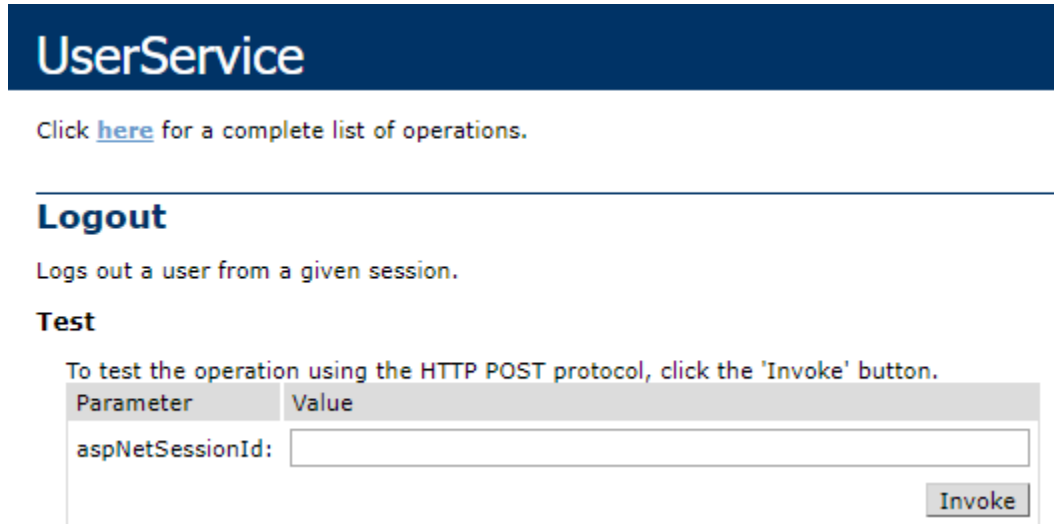
Figura 46 - API UserService - Solicita Todas as Mensagens Disponíveis

aspNetSessionId: Código de identificação da sessão.

deviceID: Número identificativo do dispositivo.

Devolve uma lista de mensagens associadas ao dispositivo indicado.

7. Logout



The screenshot shows the documentation for the 'Logout' operation in the 'UserService' API. At the top, there is a dark blue header with the text 'UserService' in white. Below the header, a link is provided: 'Click [here](#) for a complete list of operations.' The 'Logout' section is titled in bold blue text. Below the title, a description reads: 'Logs out a user from a given session.' A 'Test' section follows, with the instruction: 'To test the operation using the HTTP POST protocol, click the 'Invoke' button.' Below this instruction is a table with two columns: 'Parameter' and 'Value'. The table contains one row with the parameter 'aspNetSessionId' and an empty text input field for its value. To the right of the input field is a button labeled 'Invoke'.

UserService

Click [here](#) for a complete list of operations.

Logout

Logs out a user from a given session.

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
aspNetSessionId:	<input type="text"/>

Invoke

Figura 47 - API UserService - Fim de Sessão

aspNetSessionId: Código de identificação da sessão.

Devolve o estado verdadeiro se a execução do método foi bem-sucedida ou falso se a execução do método não foi bem-sucedida.

8. SendMessage

UserService

Click [here](#) for a complete list of operations.

SendMessage

Sends an SMS text message to the user's registered phone number.

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
deviceID:	<input type="text"/>
aspNetSessionId:	<input type="text"/>
message:	<input type="text"/>

Figura 48- API UserService - Envia Mensagem

aspNetSessionId: Código de identificação da sessão.

deviceID: Número identificativo do dispositivo.

Message: Mensagem que queremos enviar

Devolve o estado verdadeiro se a execução do método foi bem-sucedida ou falso se a execução do método não foi bem-sucedida.

9. UpdatePushService

UserService

Click [here](#) for a complete list of operations.

UpdatePushService

Change Push Service.

No definition found. [Search the web for "urladdress" »](#) ...ocol, click the 'Invoke' button.

urlAddress:

aspNetSessionId:

Invoke

Figura 49 - API UserService - Atualizar o Endereço do Serviço *PUSH*

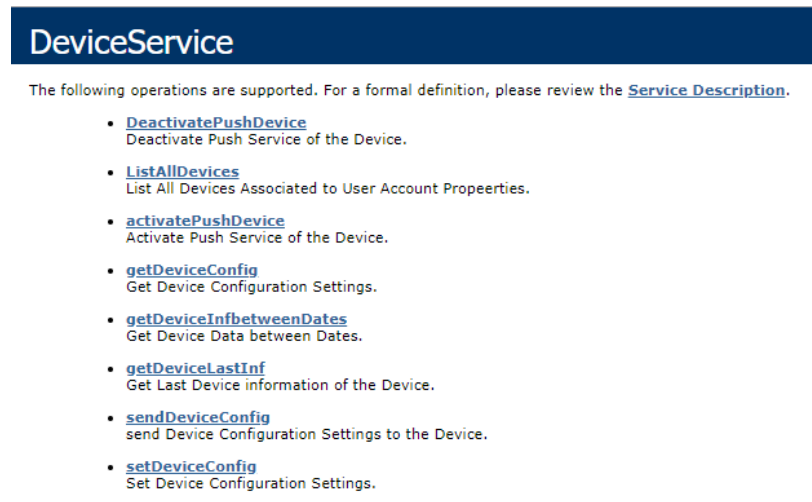
aspNetSessionId: Código de identificação da sessão.

urlAddress: Endereço para o qual queremos que as chamadas sejam efetuadas.

Devolve o estado verdadeiro se a execução do método foi bem-sucedida ou falso se a execução do método não foi bem-sucedida.

10. DeviceService

Neste serviço estão todos os métodos relacionados com os dispositivos. Pode-se listar todos os serviços que estão associados à conta do utilizador, ter dados de acesso e de configuração dos mesmos assim como ativar/desativar serviços relacionados com os respetivos dispositivos.



DeviceService

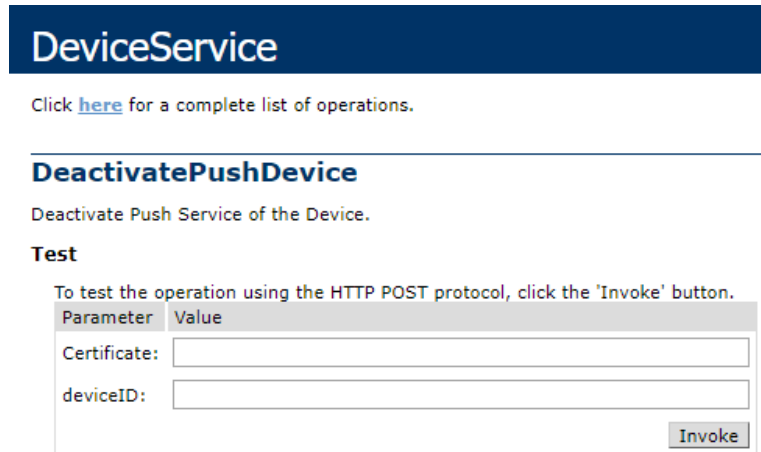
The following operations are supported. For a formal definition, please review the [Service Description](#).

- [DeactivatePushDevice](#)
Deactivate Push Service of the Device.
- [ListAllDevices](#)
List All Devices Associated to User Account Properties.
- [activatePushDevice](#)
Activate Push Service of the Device.
- [getDeviceConfig](#)
Get Device Configuration Settings.
- [getDeviceInfbetweenDates](#)
Get Device Data between Dates.
- [getDeviceLastInf](#)
Get Last Device information of the Device.
- [sendDeviceConfig](#)
send Device Configuration Settings to the Device.
- [setDeviceConfig](#)
Set Device Configuration Settings.

Figura 50 - API DeviceService - Lista de Métodos

11. DeactivatePushDevice

Desativa o serviço de *PushService*.



The screenshot shows the API interface for DeviceService. At the top, there is a dark blue header with the text "DeviceService". Below the header, there is a link: "Click [here](#) for a complete list of operations." The main section is titled "DeactivatePushDevice" and contains the description "Deactivate Push Service of the Device." Below this, there is a "Test" section with the instruction "To test the operation using the HTTP POST protocol, click the 'Invoke' button." A table with two columns, "Parameter" and "Value", is present. The first row has "Certificate:" and an empty text input field. The second row has "deviceID:" and another empty text input field. To the right of the input fields is a button labeled "Invoke".

Parameter	Value
Certificate:	<input type="text"/>
deviceID:	<input type="text"/>

Invoke

Figura 51- API DeviceService - Desativar o Serviço *PUSH*

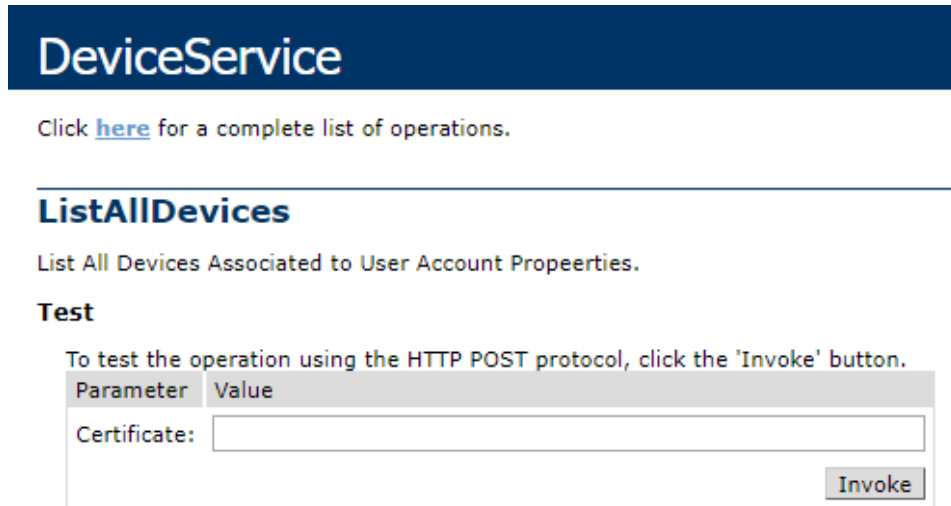
aspNetSessionId: Código de identificação da sessão.

deviceID: Número identificativo do dispositivo.

Devolve o estado verdadeiro se a execução do método foi bem-sucedida ou falso se a execução do método não foi bem-sucedida.

12. ListAllDevices

Lista todos os dispositivos que foram associados à conta do utilizador.



The screenshot shows the DeviceService API interface. At the top, there is a dark blue header with the text "DeviceService". Below the header, there is a link: "Click [here](#) for a complete list of operations." A horizontal line separates this from the "ListAllDevices" section. The section title is "ListAllDevices" in bold blue text. Below the title is the description: "List All Devices Associated to User Account Properties." Underneath is a "Test" section. It contains the instruction: "To test the operation using the HTTP POST protocol, click the 'Invoke' button." Below this instruction is a table with two columns: "Parameter" and "Value". The first row has "Certificate:" in the "Parameter" column and an empty text input field in the "Value" column. To the right of the input field is a button labeled "Invoke".

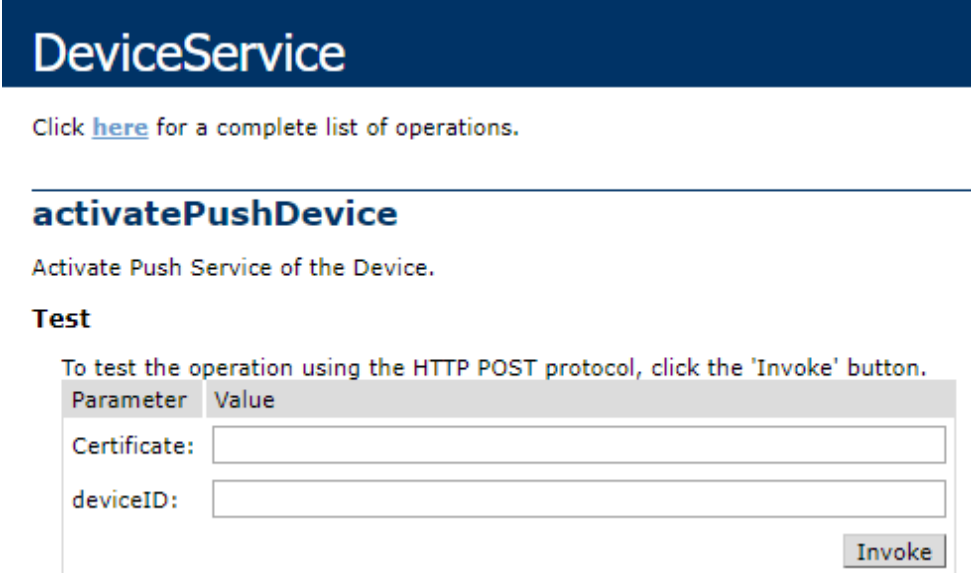
Figura 52- API DeviceService - Listar Todos os Dispositivos

aspNetSessionId: Código de identificação da sessão.

Devolve uma tabela com os dados de todos os dispositivos que estão associados à conta (ID, descrição)

13. ActivatePushDevice

Ativa o serviço de envio automático de dados para sistema de terceiros via chamada http/https.



The screenshot shows a web interface for the DeviceService API. At the top, there is a dark blue header with the text "DeviceService" in white. Below the header, there is a link "Click [here](#) for a complete list of operations." followed by a horizontal line. The main heading is "activatePushDevice" in bold. Below this, it says "Activate Push Service of the Device." and "Test". A note states: "To test the operation using the HTTP POST protocol, click the 'Invoke' button." Below the note is a form with two input fields: "Certificate:" and "deviceID:". To the right of the "deviceID:" field is a button labeled "Invoke".

Parameter	Value
Certificate:	<input type="text"/>
deviceID:	<input type="text"/>

Invoke

Figura 53- API DeviceService - Ativar Serviço *PUSH*

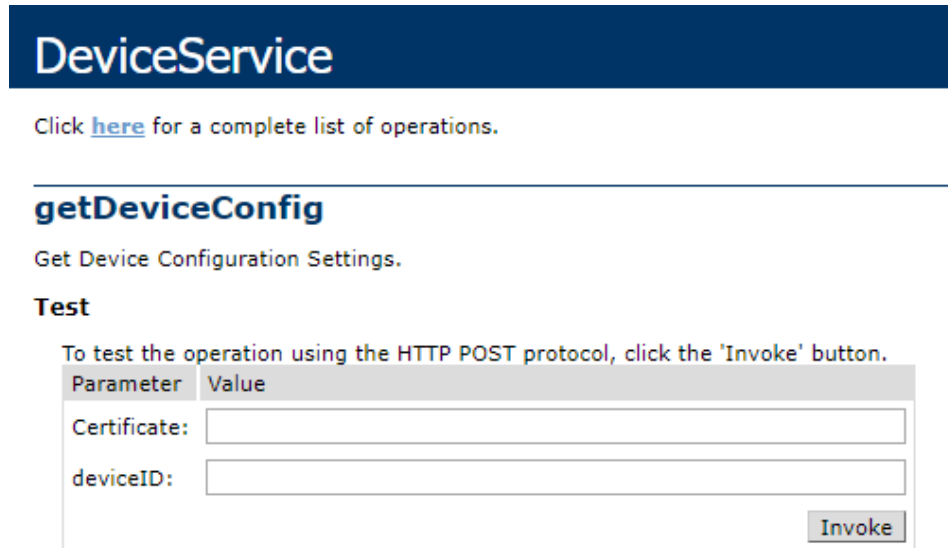
aspNetSessionId: Código de identificação da sessão.

deviceID: Número identificativo do dispositivo.

Devolve o estado verdadeiro se a execução do método foi bem-sucedida ou falso se a execução do método não foi bem-sucedida.

14. GetDeviceConfig

Este método permite recolher os últimos dados de configuração do dispositivo.



The screenshot shows the DeviceService API interface. At the top, there is a dark blue header with the text "DeviceService". Below the header, there is a link: "Click [here](#) for a complete list of operations." A horizontal line separates this from the "getDeviceConfig" section. The "getDeviceConfig" section has the title "getDeviceConfig" in bold, followed by the description "Get Device Configuration Settings." Below this is a "Test" section with the instruction "To test the operation using the HTTP POST protocol, click the 'Invoke' button." There is a table with two columns: "Parameter" and "Value". The table contains two rows: "Certificate:" and "deviceID:". Each row has an empty text input field to its right. At the bottom right of the table is a button labeled "Invoke".

Parameter	Value
Certificate:	<input type="text"/>
deviceID:	<input type="text"/>

Invoke

Figura 54- API DeviceService - Pedir Configuração Atual

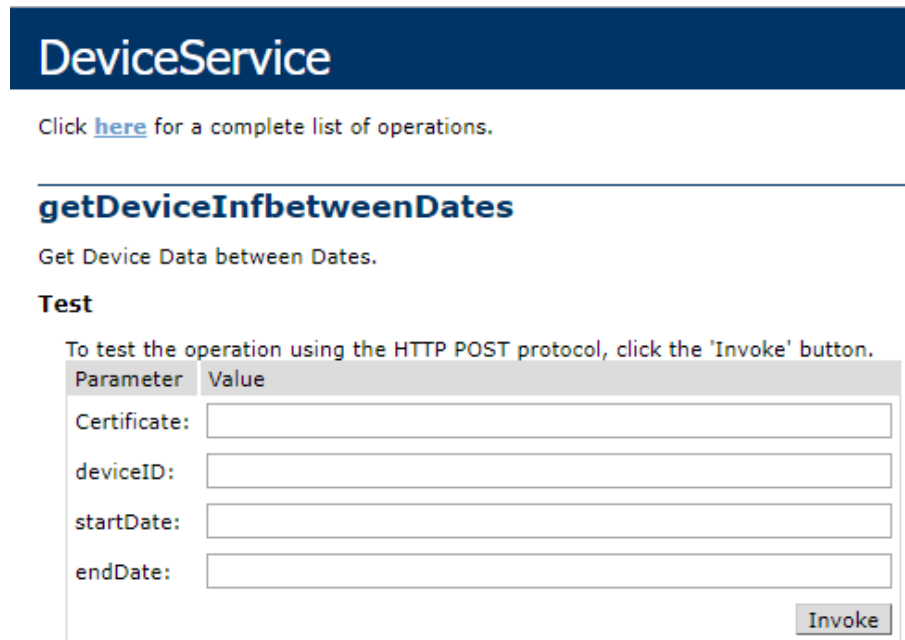
aspNetSessionId: Código de identificação da sessão.

deviceID: Número identificativo do dispositivo.

Devolve uma lista com todos os valores das opções de configuração do dispositivo cujo deviceID foi o que passámos na invocação do método.

15. GetDeviceInfoBetweenDates

Devolve os dados que o dispositivo enviou num determinado período cujas datas inicial e final são invocadas no método.



The screenshot shows the DeviceService API interface. At the top, there is a dark blue header with the text "DeviceService". Below the header, there is a link: "Click [here](#) for a complete list of operations." The main section is titled "getDeviceInfoBetweenDates" in bold blue text, followed by the description "Get Device Data between Dates." Below this, there is a "Test" section with the instruction "To test the operation using the HTTP POST protocol, click the 'Invoke' button." The test form consists of a table with two columns: "Parameter" and "Value". The parameters are "Certificate:", "deviceID:", "startDate:", and "endDate:", each with an adjacent text input field. An "Invoke" button is located at the bottom right of the form.

Parameter	Value
Certificate:	<input type="text"/>
deviceID:	<input type="text"/>
startDate:	<input type="text"/>
endDate:	<input type="text"/>

Invoke

Figura 55- API DeviceService - Pedir Dados de um Dispositivo entre Datas

aspNetSessionId: Código de identificação da sessão.

deviceID: Número identificativo do dispositivo.

StartDate: Data a partir da qual queremos recolher dados.

endDate: Data até à qual queremos recolher dados.

Devolve uma lista de dados do dispositivo cujas datas se enquadrem dentro do período invocado.

16. GetDeviceLastInf

Este método permite recolher os dados mais atualizados do dispositivo.



The screenshot shows the DeviceService API interface. At the top, there is a dark blue header with the text "DeviceService" in white. Below the header, there is a link "Click [here](#) for a complete list of operations." followed by a horizontal line. The main section is titled "getDeviceLastInf" in bold blue text. Below the title, it says "Get Last Device information of the Device." and "Test". There is a note: "To test the operation using the HTTP POST protocol, click the 'Invoke' button." Below this note is a form with two input fields: "Certificate:" and "deviceID:". The "deviceID:" field has a small "deviceID:" label to its left. At the bottom right of the form is a button labeled "Invoke".

Parameter	Value
Certificate:	<input type="text"/>
deviceID:	<input type="text"/>

Invoke

Figura 56- API DeviceService - Solicita última Informação do Dispositivo

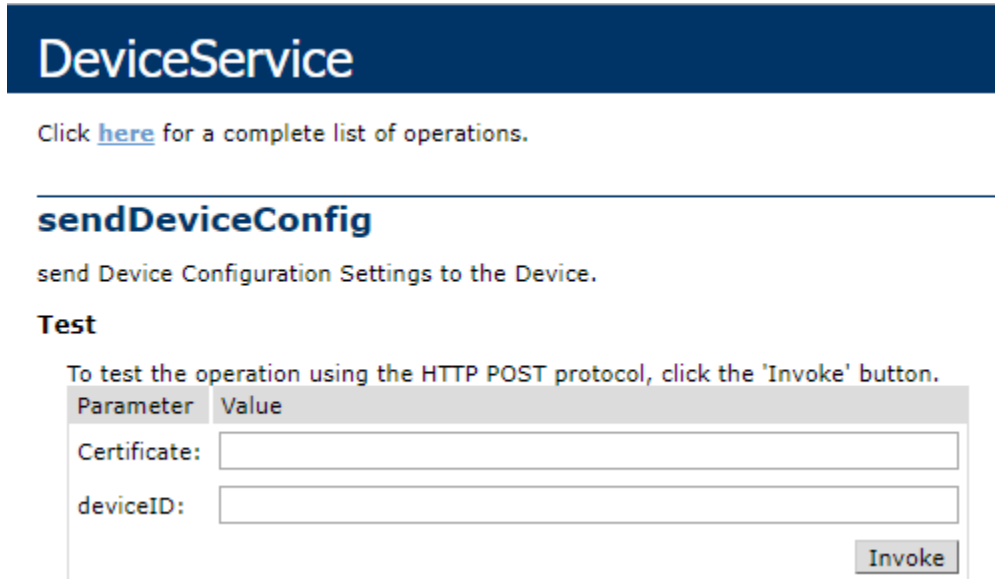
aspNetSessionId: Código de identificação da sessão.

deviceID: Número identificativo do dispositivo.

Devolve uma lista dos últimos dados referidos ao dispositivo invocado.

17. SendDeviceConfig

Envia a configuração para o dispositivo.



The screenshot shows the DeviceService API interface. At the top, there is a dark blue header with the text "DeviceService" in white. Below the header, there is a link: "Click [here](#) for a complete list of operations." A horizontal line separates this from the "sendDeviceConfig" section. The section title "sendDeviceConfig" is in bold blue text, followed by the description "send Device Configuration Settings to the Device." Below this is a "Test" section with the instruction "To test the operation using the HTTP POST protocol, click the 'Invoke' button." The test form has a table with two columns: "Parameter" and "Value". There are two rows: "Certificate:" and "deviceID:". Each row has a text input field. At the bottom right of the form is a button labeled "Invoke".

Parameter	Value
Certificate:	<input type="text"/>
deviceID:	<input type="text"/>

Invoke

Figura 57- API DeviceService - Envio de Configuração para o Dispositivo

aspNetSessionId: Código de identificação da sessão.

deviceID: Número identificativo do dispositivo.

Devolve o estado verdadeiro se a execução do método foi bem-sucedida ou falso se a execução do método não foi bem-sucedida.

18. SetDeviceConfig

Este método permite alterar os dados de configuração do dispositivo. Após invocar este método deverá invocar o método de SendDeviceConfig.

setDeviceConfig

Set Device Configuration Settings.

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
Certificate:	<input type="text"/>
deviceID:	<input type="text"/>
Phonenumber:	<input type="text"/>
username:	<input type="text"/>
Pwd:	<input type="text"/>
APN:	<input type="text"/>
wirelessusername:	<input type="text"/>
wirelesspwd:	<input type="text"/>
wirelessnetworkname:	<input type="text"/>
srlport1:	<input type="text"/>
srlport2:	<input type="text"/>
srlport3:	<input type="text"/>
srlport4:	<input type="text"/>
srlport5:	<input type="text"/>
AnalgPort1:	<input type="text"/>
AnalgPort2:	<input type="text"/>
actwireless:	<input type="text"/>
actGSM:	<input type="text"/>
ACTSolarPanel:	<input type="text"/>

Figura 58- API DeviceService - Alteração da Configuração de um Dispositivo

i. Serviço de Push de Equipamentos

Este serviço não é mais que uma configuração que é feita na página da minha conta e que permite que seja chamada um endereço de internet externo e que lhe sejam passados os novos dados recebidos de cada dispositivo. Esta opção permite que um sistema de terceiros tenha toda a informação atualizada relativamente a cada um dos dispositivos.

Ex:

```
POST  
/M2M/ManagedConnectivity/v1/notification?url=http%3a%2f%2fjssl.brm.com%2fdevi  
cereps%2fHTTP/1.1  
Connection: Keep-Alive  
Content-Length: 254  
Content-Type: application/json  
Accept: application/json  
Accept-Encoding: gzip, deflate  
Authorization: REMOVED  
Host: services.bk.sapo.pt
```

```
{"ALARM_TYPE\":"Temperature Change\","DATETIME\":"25/10/2016  
03:07:40\","MSISDN\":"927960706\","ALARM_CONDITION\":"id_8675436\","ICCID\":"  
Dispositivo Armazem1\","ENDPOINT\":" \"}"
```