Universidade de Aveiro Departamento de Engenharia Mecânica
2020

João Pedro Gonçalves Bastos

Remote Management of Devices

Gestão Remota de Dispositivos

Universidade de Aveiro Departamento de Engenharia Mecânica
2020

João Pedro
Gonçalves Bastos

Remote Management of Devices

Gestão Remota de Dispositivos

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestrado em Engenharia Mecânica, realizada sob orientação científica de José Paulo Oliveira Santos, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro.

Este trabalho teve o apoio financeiro dos projetos UIDB/00481/2020 e UIDP/00481/2020 - FCT - Fundação para Ciência e Tecnologia; e CENTRO-01-0145-FEDER-022083 - Programa Operacional Regional do Centro (Centro2020), no âmbito do Acordo de Parceria Portugal 2020, através do Fundo Europeu de Desenvolvimento Regional.

O júri / The jury

Presidente / President	**Prof. Doutor Marco Paulo Soares dos Santos** Professor Auxiliar Convidado da Universidade de Aveiro
Vogais / Committee	**Prof. Doutor José Paulo Oliveira Santos** Professor Auxiliar da Universidade de Aveiro (orientador)
	Doutor Waldir Aranha Moreira Junior Senior Scientist da Fraunhofer Portugal

Agradecimentos / Acknowledgements

À minha família, em particular aos meus pais e irmã, e à Margarida pelo apoio e motivação dados ao longo do meu percurso académico.
Aos meus amigos e colegas por todos os momentos de trabalho, entreajuda, aprendizagem, convívio e descontração.
Ao Prof. Doutor José Paulo Santos por ter orientado a minha dissertação e pela sua disponibilidade.
Ao Eng. Joaquim Ribeiro e ao Miguel Ferreira pelo forte apoio no desenvolvimento do trabalho, conselhos, sugestões e por toda a sua disponibilidade.

Palavras-chave

Dispositivos; ESP8266; Raspberry Pi; MQTT; *Bosch IoT Hub Bosch IoT Things*; *Bosch IoT Rollouts*; API; HTTP; Interface Gráfica; Aplicação; *Software* e *Firmware*

Resumo

Devido aos avanços tecnológicos ocorridos nos sensores e nos seus controladores, estes têm atualmente mais e melhores características como capacidade de processamento, não serem cablados e o próprio custo. A sua utilização traz benefícios como, por exemplo, o aumento da eficiência da produção ou redução de emissões CO2. Ao assimilar estes benefícios, as empresas utilizam-nos em elevadas quantidades e disperses por uma grande área. Neste projeto de dissertação, foi desenvolvida uma ferramenta que procura direcionar os desafios de *deployment* em massa – configuração inicial, atualização de *firmware* e de *software* e ainda verificação do estado atual do dispositivo ou do agregado de sensores. A ferramenta foi ainda desenvolvida tendo em consideração ser *low-cost* e recorrendo a *software Open-Source* do *working group Eclipse IoT* – neste caso utilizando os serviços *Bosch IoT Hub*, *Bosch IoT Things* e *Bosch IoT Rollouts* (todos serviços da *Bosch IoT Suite*) – e o *Node-RED* para conectar os dispositivos, as APIs e a aplicação desenvolvida. Foram ainda utilizados como *Internet of Thing Devices* o Raspberry Pi e o ESP8266, utilizando protocolos de comunicação MQTT e HTTP.

Keywords Devices; ESP8266; Raspberry Pi; MQTT; Bosch IoT Hub; Bosch IoT Things; Bosch IoT Rollouts; API; HTTP; Graphical Interface; Aplication; Software e Firmware

Abstract Due to technological advances in sensors and their controllers, they currently have more and better features such as processing capacity, not being wired and the cost itself. Its use brings benefits such as, for example, increasing production efficiency or reducing CO2 emissions. By assimilating these benefits, companies use them in large quantities and spread over a large area. In this dissertation project, a tool was developed that seeks to address the challenges of mass deployment - initial configuration, firmware and software update and also verification of the current state of the device or sensor aggregate. The tool was also developed considering low cost and using Open-Source software from the working group Eclipse IoT - in this case using the services Bosch IoT Hub, Bosch IoT Things and Bosch IoT Rollouts (all services from Bosch IoT Suite) - and Node-RED to connect devices, APIs and the developed application. Raspberry Pi and ESP8266 were also used as Internet of Thing Devices, using MQTT and HTTP communication protocols.

Índice

Lista de Figuras

Acrónimos

API Application Programming Interface.

AWS Amazon Web Service.

ERP Enterprise Resource Planning.

FOTA Firmware Over the Air.

HTTP HyperText Transfer Protocol.

I^2C Inter Integrated Circuit.

IoT Internet of Things.

JSON JavaScript Object Notation.

MES Manufacturing Execution System.

MQTT Message Queue Telemetry Transport.

OSS Open Source Software.

SOTA Software Over the Air.

SQL Structured Query Language.

Capítulo 1

Introdução

Ao longo da sua história, a Humanidade já atravessou diversas revoluções, sejam elas de cariz político, socioeconómico ou industrial. Relativamente às industriais, já existiram três revoluções: a primeira decorreu entre 1760 e 1830 e ficou marcada pela mecanização dos sistemas de produção, através da introdução da máquina a vapor que veio substituir a força animal ou humana; a segunda, ocorreu por volta de 1850, trouxe a eletricidade e permitiu a manufatura em massa, produzindo-se de forma mais rápida e com um custo menor; a terceira aconteceu em meados do século XX, com a chegada da eletrónica, da tecnologia da informação e das telecomunicações. Todas elas tiveram o mesmo objetivo, melhorar a eficiência, competitividade e produtividade das indústrias.

O ritmo elevado a que a tecnologia tem evoluído, provoca a diminuição dos custos por um lado, e ao aumento de capacidade de tecnologias como *smartphones, tablets* ou computadores. Atualmente, com a integração da Internet e de novas ou mais modernas tecnologias, a indústria está perante a sua quarta revolução. Na Europa, a quarta revolução industrial é também designada por Indústria 4.0.

A Indústria 4.0 é caracterizada por uma completa digitalização dos sistemas produtivos, interligando máquinas, pessoas e processos. Isso é possível devido à evolução crescente do poder computacional (permitindo maior rapidez e capacidade em processar informação), dos sistemas de comunicação (incluindo Internet e tecnologias *wireless*) e da inovação em *software*. Conceitos como *big data and analytics*, *Internet of Things* (IoT), conectividade, interfaces avançadas homem-máquina, *cloud computing*, entre outros, suportam a Indústria 4.0.

Um dos pontos chave da Indústria 4.0 são os dados. Ainda há bem pouco tempo, a cadeia de informação de uma fábrica estava muito bem estruturada: sensor para controlador, controlador para automação, automação para *Manufacturing Execution System* (MES) e MES para *Entreprise Resource Planner* (ERP) ou similar hierarquia. No entanto, com o aumento da capacidade de processamento dos pequenos dispositivos, com a facilidade de acesso aos dados provocado pela ausência de cabos e com a redução de custos em sensorização, a hierarquia de informação tornou-se parcialmente obsoleta. Dados de processos são trocados entre máquinas, dados do produto acompanham-no ao longo de toda a sua produção, as máquinas apresentam um conhecimento sobre o seu próprio processo e desencadeiam pedidos de manutenção ou avisos ao ser humano quando algo não está correto. Desta forma, os dados que continuam a percorrer a cadeia de informação são já dados selecionados e pré-processados enquanto que muitos outros estão ao nível da máquina, sendo usados para otimização ou reação em tempo real.

O avanço também se fez ao nível dos sensores com o aumento da sua capacidade de processamento, com a interligação de diversas capacidades sensitivas e também com o facto de estes agora serem não cablados, permitindo a sua rápida expansão num chão de uma fábrica.

Além dos sensores, também os seus controladores são agora mais avançados, estando dotados de maior capacidade de processamento para tarefas que anteriormente necessitavam de computadores potentes, como por exemplo a análise de dados. Este tipo de processamento no controlador, passou a ser conhecido como *Edge Computing*. O envio de dados do controlador para uma unidade superior como um MES ou como um *Data Lake* para análise de padrões nos dados ao longo do tempo, passou a ser feito não para unidades de servidores mas sim para um conceito abstrato de implementação designado por FoG ou *Cloud Computing*.

O crescimento destas tecnologias e inovações trouxe também uma quantidade enorme de diferentes modelos com características únicas e grande parte deles benéficos para a eficiência, produtividade ou para redução de CO2, por exemplo. Assumindo que uma empresa procura então o benefício destes equipamentos, estes podem chegar aos milhares e estarem espalhados por uma grande área. Alguns dos desafios enfrentados atualmente passam por realizar uma rápida configuração inicial, atualizar o *software* e *firmware* dos dispositivos já no chão de fábrica, conhecer o seu estado de funcionamento e efetuar o seu diagnóstico.

1.1 Motivação

Percebendo a importância que estes dispositivos têm no processo produtivo de uma empresa, contribuindo para uma produção mais eficiente, aumento de produtividade e excelência operacional, é sem dúvida fundamental encontrar uma solução para os problemas que surgem com o forte aumento do número destes dispositivos no chão de fábrica. Assim, para implementar com sucesso uma solução IoT numa empresa, é crucial considerar a gestão destes dispositivos.

Tendo a Bosch como objetivo passar a ser um *IoT Leading Provider* em todos os seus produtos, a Bosch iniciou uma estratégia de lançamento de ferramentas e infraestruturas que possibilitassem quer o desenvolvimento desses produtos, a produção e mais tarde que esses produtos estivessem em casa das pessoas e fossem inteligentes. Para isso, a Bosch não se pode focar apenas no que vende mas também no que consome. Assim, a Bosch Thermotechnick GmbH, iniciou um conjunto de movimentos, entre os quais a criação de uma equipa de Digitalização na Produção (TT/MFD). A equipa da Termotecnologia *Manufacturing Digitalization* que opera para todas as fábricas do grupo Bosch Termotecnologia, propôs a realização de um projeto conjunto, que deu origem a esta dissertação.

Tal como acontece nos nossos computadores, *smartphones, tablets* e até mesmo automóveis, os seus sistemas operativos necessitam de cuidados frequentes, seja na atualização de *software*, novas configurações para melhorar a segurança ou implementar novas aplicações. Na grande maioria dos casos, todos estes cuidados são realizados remotamente, através de uma ligação à Internet. Por isso, é também fundamental estender esta tecnologia *Over-the-Air* aos dispositivos IoT presentes no chão de fábrica.

1.2 Objetivos

O objetivo principal desta dissertação consiste no desenvolvimento de uma ferramenta que efetue a gestão dos dispositivos com algumas das características referidas anteriormente. Começam a existir alguns softwares que permitem realizar essa gestão, contudo são *softwares* proprietários. No entanto, tendo a Bosch uma parceria estratégica com o *working group Eclipse IoT* para o desenvolvimento de aplicações *Open Source Software (OSS)*, a solução para desenvolver uma ferramenta de baixo custo e que seja capaz de responder a alguns dos desafios já mencionados, passa pela utilização da tecnologia OSS do *Eclipse IoT* ou da *Bosch IoT Suite*.

Através da ferramenta deve ser possível adquirir, processar e monitorizar valores recorrendo aos dispositivos, permitindo ainda ao utilizador a sua visualização. A ferramenta deve também permitir a obtenção do estado de funcionamento de cada dispositivo. Por fim, a ferramenta deve permitir a gestão e atualização do *software* ou *firmware* instalado nos dispositivos, de uma forma fácil e rápida. Potencialmente, esta ferramenta pode ser usada em milhares de dispositivos com diversas funções no chão de fábrica, como por exemplo o controlo do funcionamento de uma máquina, controlo do sistema de AVAC de um determinado local, controlo do sistema de produção de ar comprimido.

De uma maneira geral, a solução passa por integrar as funções dos dispositivos com as tecnologias *Open-Source*. Pretende-se que o dispositivo efetue a sua função principal, ou seja, a aquisição, processamento e monitorização dos valores, adicionando uma vertente de gestão remota, tanto do seu funcionamento como do *software* e *firmware* que tem instalado.

Para atingir o objetivo principal, foram definidas algumas etapas de modo a organizar melhor o trabalho:

- Familiarização com os dispositivos usados;
- Analisar tecnologias *Open-Source*;
- Definir uma arquitetura para a solução do problema;
- Desenvolvimento da ferramenta - adquirir, processar e monitorizar os valores;
- Desenvolvimento da ferramenta - obter remotamente o estado de funcionamento do dispositivo;
- Desenvolvimento da ferramenta - gerir e atualizar remotamente o *software* ou *firmware* instalado no dispositivo

1.3 Estrutura do documento

O documento está organizado em seis capítulos:

Capítulo 1: é feita a contextualização do problema, são expostas as razões que levaram à abordagem do problema, os objetivos que se pretendem alcançar e a estrutura do documento;

Capítulo 2: apresentação da empresa, com a qual foi realizado o trabalho apresentado neste documento;

Capítulo 3: é avaliado o estado da arte e são apresentadas tecnologias relevantes para a dissertação, outros estudos efetuados sobre este tema e por fim são exploradas quais as opções já disponíveis no mercado;

Capítulo 4: caracterização do problema que o trabalho se propõe a resolver, apresentando todas as variáveis e requisitos essenciais para o desenvolvimento da solução. Demonstra-se a arquitetura proposta para o trabalho, explicando cada elemento da mesma e o fluxo de informação entre diferentes nodos;

Capítulo 5: implementação da solução proposta. Apresenta-se a interação dos diferentes nodos na solução proposta, caracterizam-se os diferentes nodos em termos de *hardware* e *software*;

Capítulo 6: é avaliada a solução, verifica-se se os resultados cumprem ou não com os objetivos, são retiradas conclusões sobre o trabalho desenvolvido e são sugeridas melhorias ao mesmo.

Capítulo 2

Bosch Termotecnologia S.A.

2.1 O Grupo Bosch [1]

Robert Bosch inaugurou em Estugarda, no ano de 1886, a Oficina de Mecânica de Precisão e Engenharia Elétrica. A Oficina Mecânica de Precisão e Engenharia Elétrica deu origem ao Grupo Bosch, um dos maiores grupos industriais da Alemanha e um grande grupo multinacional, reconhecido pela qualidade dos produtos, inovação constante e orientação para o cliente (os três pilares que o grupo defende).O Grupo Bosch é líder no fornecimento de tecnologia e serviços e está presente em cerca de 60 países.

As operações do Grupo Bosch dividem-se em quatro áreas de negócio: Soluções de Mobilidade, Tecnologia Industrial, Bens de Consumo e Tecnologia de Energia e Edifícios. Líder em IoT, a Bosch oferece soluções inovadoras para casas e cidades inteligentes, mobilidade e indústria conectada. A empresa utiliza o seu conhecimento em tecnologia de sensores, *software* e serviços, bem como a sua própria *cloud* IoT para oferecer aos seus clientes soluções conectadas e em diversos domínios a partir de uma única fonte. O objetivo estratégico da Bosch é fornecer inovações para uma vida conectada. A Bosch melhora a qualidade de vida em todo o mundo com produtos e serviços inovadores e fascinantes. Desta forma, a empresa oferece mundialmente "Tecnologia para a Vida".

Desde 1964, o Grupo Bosch é detido pela fundação de caridade Robert Bosch Stiftung GmbH, detendo altualmente 92% do Grupo. A fundação mantém atualmente todas iniciativas sociais e de beneficência do fundador da empresa.

2.1.1 O Grupo Bosch em Portugal [2]

Presente em Portugal desde 1911, o Grupo Bosch é um dos Grupos mais reconhecidos no pais. Com uma presença consolidada, exporta mais de 95% da sua produção para mercados internacionais e tem vindo a alargar as atividades de investigação e desenvolvimento em *hardware* e *software* para diferentes áreas de negócio.

Em Portugal, fazem parte do Grupo Bosch três empresas: a Bosch Termotecnologia, em Aveiro, a Bosch *Car Multimedia* Portugal, em Braga e a Bosch *Security Systems* – Sistemas de Segurança, em Ovar. Estas empresas desenvolvem e produzem soluções de água quente, sensores e multimédia automóvel e sistemas de segurança e comunicação. A sede no país está em Lisboa, onde são realizadas atividades de vendas, *marketing*, contabilidade e comunicação, bem como serviços partilhados de recursos humanos e comunicação para o Grupo Bosch.

As exigências do mercado e o objetivo de tornar a Bosch líder em IoT são relevantes também para as atividades em Portugal.

2.2 Bosch Termotecnologia S.A. [3]

A Bosch Termotecnologia S.A. iniciou a sua atividade em Cacia, em 1977, sob a designação Vulcano Termodomésticos S.A., com base num contrato de licenciamento com a Bosch. Ao utilizar a mesma tecnologia que o Grupo Bosch, assim como uma clara estratégia de vendas e assistência pós-vendas, rapidamente a tornou líder do mercado nacional de esquentadores.

Em 1992, torna-se líder europeu e terceiro produtor a nível mundial no mercado de esquentadores e, em 1995, inicia a sua produção de caldeiras.

Até 1998, a empresa era constituída por um capital totalmente nacional, ano no qual o Grupo Bosch adquire a maioria do capital e esta passa a integrar uma divisão de termotecnologia do Grupo, ocorrendo transferência de competências e equipamento, o que leva a um início de especialização desta empresa dentro do grupo. Em 2007, inicia a produção de soluções solares de aquecimento de água.

Figura 2.1: Bosch Termotecnologia S.A., em Aveiro

2.2.1 Organização da Bosch Termotecnologia S.A.

A Bosch Termotecnologia S.A. em termos de infraestruturas está dividida em três grupos: Administração (que coordena todos os projetos, atividades relacionadas com a divisão de termotecnologia e os recursos da empresa), Desenvolvimento (onde se foca a investigação, inovação, criação de novos produtos ou melhoramento dos atuais para responder às expectativas e necessidades do mercado) e Produção (dedicada à fabricação, onde colaboradores diretos e indiretos trabalham com a gestão relacionada ao chão de fábrica, dos recursos disponíveis, e procuram assegurar a produção e melhorar cada vez mais processos associados). Estes três grupos trabalham em sintonia para que tudo funcione da melhor forma.

2.2.2 A equipa

A equipa da Termotecnologia *Manufacturing Digitalization* (TT/MFD), que opera para todas as fábricas do grupo Bosch Termotecnologia, trabalha para produzir inovações na indústria.

Um dos seus focos é a aplicação do *Nexeed Manufacturing Execution System* [4] no grupo Bosch Termotecnologia. O *Nexeed MES* fornece exatamente as informações de que as empresas precisam para um sistema de controlo ativo, documentação contínua, rastreabilidade, garantia de qualidade e controlo da sua produção ao longo de toda a cadeia de valor. O objetivo é otimizar continuadamente os processos com base nesses dados, para responder às mudanças rapidamente. Consequentemente, isso gera um aumento na eficiência da produção. O *Nexeed Manufacturing Execution System* garante integração vertical, em particular, desde o sensor até à *cloud*.

Outro objetivo da equipa é preparar as fábricas para o futuro, através da análise dos dados que são recolhidos e também recorrendo a soluções de visão artificial.

Como um dos grandes objetivos do Grupo Bosch é atingir a neutralidade carbónica ainda em 2020, a equipa também procura desenvolver ferramentas que permitam reduzir a utilização de papel, optando pela digitalização e a integração de novas tecnologias como *Robotic Process Automation* (RPA) ou *Intelligent Process Automation* (IPA).

Sendo uma equipa de inovação na indústria, a integração de novas soluções *low cost* para monitorizar dados como dispositivos IoT, a utilização de outras infraestruturas que não relacionadas com o *Manufacturing Execution System* como a *Bosch IoT Suite* e implementação do *Use Case Energy Management*, são exemplos de projetos.

Capítulo 3

Revisão Bibliográfica

3.1 Gestão de dispositivos

A gestão de dispositivos refere-se a todas a ferramentas, recursos e processos necessários para oferecer suporte às soluções de IoT de forma eficaz.

Depois dos dispositivos serem instalados é necessário efetuar a correção de erros e atualizações de *software*, ou por vezes os dispositivos falham sendo preciso repará-los ou substituí-los. Em ambos os casos, pretende-se que o tempo de inatividade dos dispositivos seja o menor possível.

Deste modo, além da instalação da infraestrutura IoT, soluções robustas para gestão de dispositivos são essenciais para o sucesso de qualquer solução geral. É assim essencial que estas infraestruturas estejam preparadas para o futuro. [5]

Existem quatro requisitos fundamentais para a gestão de dispositivos [6]:

- provisionamento e autenticação;
- configuração e controlo;
- monitorização e diagnóstico;
- manutenção e atualização de *software*.

Provisionamento e Autenticação

O provisionamento é o processo de inscrição de um dispositivo no sistema. A autenticação faz parte desse processo, onde apenas os dispositivos que apresentam as credenciais apropriadas são registados.

A autenticação do dispositivo é o ato de estabelecer com segurança a identidade do dispositivo para garantir que ele seja de confiança. Um serviço hospedado na *cloud*, ao qual o dispositivo se conecta, precisa de saber que o dispositivo é realmente um dispositivo genuíno, está a executar um *software* confiável e está a trabalhar em nome de um utilizador de confiança.

Uma solução robusta para a gestão de dispositivos e que possua métodos de autenticação fortes e medidas de segurança, reduz significativamente o risco de conexão de dispositivos desconhecidos.

Configuração e Controlo

Depois da implementação dos dispositivos, podem haver configurações que têm de ser ajustadas ao longo do tempo.

A capacidade de configurar e controlar dispositivos, mesmo após a implantação, é fundamental para garantir a funcionalidade, melhorar o desempenho e proteger contra ameaças à segurança.

Monitorização e diagnóstico

Ao longo do funcionamento do dispositivo, podem surgir problemas operacionais imprevistos ou erros de *software* que têm de ser resolvidos. Contudo, antes de resolvê-los, é preciso identificá-los.

Portanto, a capacidade de monitorizar e detetar quando algo está errado, é essencial para identificar e diagnosticar possíveis problemas.

Existe a necessidade de incluir a monitorização e o diagnóstico de forma ininterrupta para minimizar o tempo de inatividade e otimizar as operações.

Manutenção e atualização de *software*

Ao identificar erros ou falhas de segurança nos dispositivos, é fundamental corrigi-los através da atualização do *software*.

No entanto, com o elevado número de dispositivos que a infraestrutura IoT pode conter, obter o acesso físico a cada dispositivo para efetuar a sua atualização manualmente não é de todo uma opção viável. A longo prazo, esta opção comprometeria a infraestrutura IoT.

Por isso, a capacidade de realizar a manutenção e a atualização do *software* remotamente é fundamental para a gestão de dispositivos.

3.2 Indústria 4.0

A indústria é uma parte da economia que produz bens materiais, através de processos mecanizados e automatizados. Desde o início da industrialização, diversos saltos tecnológicos levaram a mudanças de paradigma, criando assim as chamadas Revoluções Industriais.

Com base na digitalização avançada dentro das fábricas, a combinação de tecnologias da Internet e tecnologias orientadas para o futuro no campo de objetos inteligentes, como máquinas e produtos, surge uma nova mudança de paradigma fundamental para a produção industrial. O termo Indústria 4.0 foi estabelecido na Europa, como sendo a quarta revolução industrial.

O termo Indústria 4.0 surgiu na Alemanha em 2011, como parte da estratégia para promover a digitalização da indústria. A Indústria 4.0 é um conceito abrangente, bem como uma tendência na manufatura, baseado na integração de um conjunto de tecnologias que permitem ecossistemas de fábricas inteligentes, autónomas, descentralizadas e integração de produtos e serviços.

A Indústria 4.0 visa a recolha e aplicação de dados e informações em tempo real, por meio de uma conexão em rede de todos os os elementos individuais, como o objetivo de

diminuir a complexidade das operações, aumentando a eficiência e a eficácia, reduzindo os custos a longo prazo. A Índustria 4.0 procura operacionalizar os resultados promissores na Internet of Things (IoT), Controlo Distribuído e Descentralizado, *Embedded Systems, Cyber-Physical Systems (CPS) e Big Data*. A grande variedade de tecnologias e a sua integração, são características da Indústria 4.0. A implementação destas tecnologias visa [7]:

- **Aumentar a mecanização e automação:** consiste na criação de mais ajudas ao trabalho físico, bem como soluções automáticas que executam operações versáteis;
- **Digitalização e rede:** a crescente digitalização da fabricação resulta numa grande quantidade de dados recolhidos por sensores e atuadores, os quais podem suportar funções de controlo e análise. A evolução dos processos digitais e o aumento da digitalização dos bens e serviços produzidos, levam a ambientes completamente digitalizados.
- **Minimização:** atualmente, dispositivos com desempenho semelhante ao dos computadores, necessitam de muito menos espaço para serem instalados. Isto permite novos campos de aplicação especialmente no contexto da produção e da logística.

A discussão e a aplicação desta quarta revolução industrial, envolve vários conceitos, tais como [8]:

- ***Cyber-physical systems (CPS):*** estes sistemas integram computação, rede e processos físicos. Caracterizam-se pela utilização de *embedded systems* para monitorizar e agir sobre processos físicos. Algoritmos inteligentes estão em execução reagindo aos dados e controlando o processo. Estes sistemas permitem que as linhas de produção sejam flexíveis e altamente automatizadas;
- ***Cloud manufacturing*:** transforma recursos de fabricação em serviços. Diversas tecnologias permitem digitalizar esses recursos com a intenção de usar essas informações para fornecer serviços partilhados ao longo de toda a cadeia de valor. Os dados armazenados na *cloud* podem estender-se a todo o ciclo de vida de uma produto, incluindo design, fabricação e manutenção;
- ***Smart factory*:** uma fábrica de utiliza os seus sensores, atuadores e sistemas autónomos para ser inteligente e flexível. Os processos podem ser melhorados através da auto-otimização e tomada de decisão autónoma. As bases de dados contém registos detalhados que podem ser usados para *machine learning*. Uma *smart factory* implementa os conceitos *Cyber-physical systems*, fabricação habilitada para IoT e *cloud manufacturing*.

3.3 Automação Industrial

O aumento da competitividade no setor industrial traduz-se em produtos de maior qualidade e consistência a preços mais competitivos. Para enfrentar este desafio as indústrias adotam a criação de novos produtos e técnicas de produção integradas, assim como o uso de dispositivos automatizados.

Um dos movimentos notáveis e influentes para obter as soluções do desafio mencionado, é a automação industrial. A automação industrial facilita o aumento da qualidade, confiabilidade e taxa de produção, reduzindo os custos de produção e *design*, adotando tecnologias e serviços novos, inovadores e integrados.

A automação industrial pode ser definida como o uso de tecnologias e dispositivos de controlo que resultam na operação e controlo automático de processos industriais sem intervenção humana significativa, alcançando assim um desempenho superior. As tecnologias incluem diversos sistemas de comunicação industrial, enquanto que os dispositivos incluem computadores, PLC's *(Programmable Logic Controller)*, microcontroladores, entre outros [9].

As principais vantagens de um sistema automatizado são:

- aumento da produtividade;
- redução dos custos de produção;
- produtos de melhor qualidade;
- redução de verificações de rotina;
- aumento dos níveis de segurança dos trabalhadores.

Com o objetivo de definir um modelo *standard* para o processo de automação industrial, foi criado o modelo hierárquico vertical apresentado na Figura 3.1. Este modelo chama-se ISA-95 tendo sido definido pela *International Society of Automation* (ISA). O modelo divide os sistemas de operações de produção em cinco níveis, os quais podem ser associados a tipos de informações, sistemas e prazos [10].

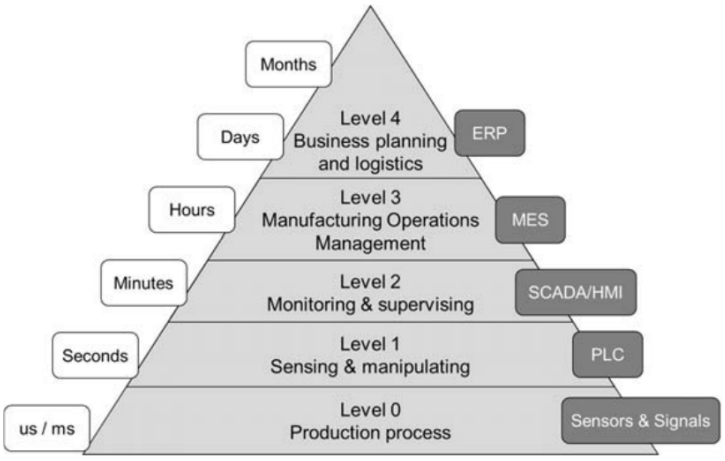

Figura 3.1: Hierarquia de um sistema de automação industrial [10]

O nível superior está relacionado com o planeamento dos negócios e com a logística, associado-se geralmente o ERP *(Enterprise Resource Planning)*. O nível seguinte controla as operações de fabricação, como por exemplo quais os processos que devem ser efetuados e em que ordem. Essa gestão é feita através do MES *(Manufacturing Execution System)*. No nível abaixo encontra-se a monitorização e a supervisão, responsável pela monitorização de um equipamento através de uma HMI *(Human Machine Interface)*

ou de um sistema de controlo e aquisição de dados (SCADA). O equipamento é controlado pelo nível de deteção e manipulação, usando-se geralmente PLC's. O último nível é composto por sensores e atuadores.

3.4 *Internet of Things* (IoT) [11]

A *Internet of Things* compreende todos os aparelhos e objetos que se encontram habilitados a estarem permanentemente ligados à Internet, sendo capazes de se identificar na rede e de comunicar entre si.

Relativamente à Indústria 4.0, o conceito vai muito para além da mera implantação de sistemas eletrónicos e de TICs em geral nos processo de produção nas fábricas, que caracterizou a Indústria 3.0. Esta nova vaga tecnológica, viabiliza uma grande interação entre os diversos dispositivos instalados ao longo da cadeia de produção no chão de fábrica, incluindo a cadeia logística, proporcionando que os processos de fabrico resultem de uma comunhão entre o mundo físico e o virtual.

Quer os equipamentos nas linhas de produção, quer os produtos que estão a ser fabricados, quer os centros logísticos são capazes de interagir autonomamente, mais uma vez com o objetivo de melhorar o processo produtivo e assim fabricar produtos de maior qualidade, mais alinhados com os requisitos do cliente e com uma melhorada eficiência em toda a cadeia de valor. Por outras palavras, a tecnologia digital em que se baseia a Indústria 4.0, quer na componente de produção quer na componente logística, contempla a simbiose da informação digital proveniente de várias fontes e locais, tendo em vista o comando e controlo do ato físico de produzir e distribuir um bem ou conjunto de bens.

Como se pode ver na Figura 3.1, a arquitetura da IoT pode ser dividida em cinco níveis.

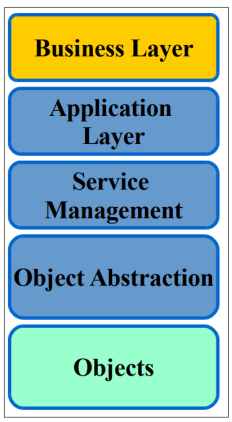

Figura 3.2: Os cinco *layers* da *Internet of Things* [11]

Objects Layer

O primeiro nível, os objetos (dispositivos) ou nível de percepção, representa os sensores físicos da IoT que obtêm e processam informação. Este nível inclui sensores e

atuadores para executar diferentes funcionalidades, tais como obter a temperatura, humidade, vibrações, movimento. Este nível digitaliza e transfere os dados para o nível seguinte de forma segura. A *Big Data* criada pela IoT é iniciada neste nível.

Object Abstraction Layer

Este nível tem como função transferir os dados produzidos pelo nível dos objetos para o nível que corresponde à Gestão do Serviço, também através de meios seguros. Os dados podem ser transferidos através de várias tecnologias tais como RFID, *WiFi*, *Bluetooth*, Infravermelho, entre outras. Funções como *cloud computing* e gestão de dados são também tratados neste nível.

Service Management Layer

Este nível emparelha um serviço solicitado com o seu solicitante com base em endereços e nomes. Este nível permite que os programadores de aplicações IoT trabalhem com objetos heterogéneos sem considerar um plataforma de *hardware* específica. Além disso, este nível processa os dados recebidos, toma decisões e entrega os serviços necessários através dos protocolos de rede.

Application Layer

Este nível fornece os serviços solicitados pelo cliente. Pode fornecer por exemplo medições de temperatura ou humidade do ar ao cliente que solicita esses dados. Este nível é capaz de fornecer serviços de alta qualidade para atender às necessidades dos clientes. É caracterizado por abranger vários mercados, tais como residências inteligentes, automação industrial, assistência médica e transportes.

Business Layer

O nível de negócios gere as atividades e serviços gerais do sistema IoT. As responsabilidades deste nível são criar modelos de negócio, gráficos, fluxogramas com base nos dados recebidos do nível anterior. Também deve projetar, analisar implementar, avaliar monitorizar e desenvolver elementos relacionados com o sistema IoT. Este nível possibilita o suporte a processos de tomada de decisão com base na análise da *Big Data*. A monitorização e a gestão dos outros quatro níveis são executados neste nível.

3.5 Condition Monitoring [12]

Condition Monitoring é definido como a medição de parâmetros específicos de um equipamento, como vibrações de uma máquina, a temperatura ou a condição do seu óleo, observando quaisquer alterações significativas que possam indicar uma falha eminente. A monitorização contínua das condições de um equipamento e a observação de quaisquer irregularidades que normalmente reduzem a vida útil de um equipamento, permitem que a manutenção ou outras ações preventivas sejam agendadas para resolver os problemas antes que eles se transformem em falhas mais graves.

Condition Monitoring é um grande componente da manutenção preditiva. Os dados recolhidos ao longo do tempo fornecem informações importantes sobre o estado atual e o

histórico de um equipamento. A evolução de um equipamento pode ser usada para antecipar o seu desempenho ao longo do tempo e prever como se pode degradar, permitindo o agendamento da manutenção com base nessas previsões. A manutenção preditiva funciona com base nestas previsões, procurando antecipar quais as falhas que podem ocorrer e qual a manutenção que deve ser realizada para evitar que as falhas ocorram.

As técnicas de *Condition Monitoring* são aplicadas nos mais diversos equipamentos, tais como equipamentos rotativos, compressores, motores elétricos, prensas. Essas técnicas passam pela análise de vibrações, análise da qualidade do óleo, termografia por infravermelhos, ultra-sons e emissões acústicas.

Os cinco benefícios mais importantes do *Condition Monitoring* são:

- Redução dos custos de manutenção;
- Redução do tempo de paragem dos equipamentos, logo maximização da produção;
- Aumento da vida útil dos equipamentos;
- Priorização das tarefas de manutenção;
- *KPIs (Key Performance Indicators)* e análises de desempenho precisas.

***Condition Monitoring* e *Internet of Things* [12]**

Os sistemas de *Condition Monitoring* associados à *Internet of Things*, permitem a integração de vários tipos de *software* de monitorização num sistema em tempo real, em qualquer lugar do mundo e em vários dispositivos.

Esta integração traz enormes vantagens às empresas, facilitando a obtenção do desempenho de cada equipamento, deteção de problemas e até mesmo o agendamento automatico das manutenções a efetuar. Algumas das vantagens da integração incluem:

- Armazenamento na *cloud* dos dados recolhidos;
- Análise sofisticada - algoritmos inteligentes permitem realizar um diagnóstico mais preciso;
- Capacidade de usar dados de vários equipamentos criando algoritmos de *machine learning* mais rapidamente e mais precisos.
- Menos necessidade da atividade humana, devido à capacidade dos sistemas realizarem a monitorização remotamente. É um benefício substancial para equipamentos que estejam em lugares remotos, tais como turbinas eólicas marítimas, plataformas petrolíferas, entre outros.

3.6 *Updates Over-The-Air* [13]

Atualmente, qualquer utilizador está bastante habituado às atualizações de *software* do seu *smartphone* seja ao nível do seu sistema operativo, ou ao nível das suas aplicações. Além disso este processo de atualização deve ser tão simples como clicar num botão.

O termo SOTA (*Software Over-The-Air*) significa que as atualizações de *software* são efetuadas remotamente através da Internet. FOTA (*Firmware Over-The-Air*) é outro

termo associado aos *updates over-the-air*. Enquanto o SOTA se refere à atualização do *software*, o FOTA relaciona-se com a atualização de *firmware*.

Estes dois termos vêm facilitar o processo de atualização na perspetiva do cliente, uma vez que deixa de existir a necessidade de ele se deslocar ao fabricante ou a um centro de apoio técnico. No entanto, do lado do fabricante torna-se mais complexo, uma vez que as atualizações precisam de ser fornecidas como um *download* e devem ser atribuídas aos dispositivos elegíveis.

Sendo a IoT um ambiente muito dinâmico, as melhorias constantes são essenciais. Assim, as atualizações de *software* e *firmware* têm um papel crucial. Duas das grandes vantagens destas atualizações são a resolução de problemas de segurança e a implementação de novas funcionalidades.

Com o elevado número de dispositivos distribuídos, é impensável instalar as atualizações manualmente. No entanto, recorrendo ao SOTA e ao FOTA, as empresas conseguem ultrapassar esse desafio. Esta abordagem tem vários benefícios, tais como:

- Processo de atualização conveniente para o cliente;
- Problemas de segurança resolvidos rapidamente;
- Atender a novas necessidades de um cliente;
- Estender o ciclo de vida de um produto;
- Gerar novos fluxos de receita;
- Gestão eficiente de lançamentos de *software*;
- Agendamento de lançamentos de *software*.

3.6.1 Principais funcionalidades de um sistema de atualização de *software* da IoT [14]

A primeira funcionalidade é o repositório de dispositivos e *software*. Ou seja, deve conter as informações do dispositivo relacionadas com as atualizações de *software* incluindo, o histórico de atualizações, informações sobre qual a versão instalada no momento e como é que o serviço consegue alcançar o dispositivo.

A segunda funcionalidade é a entrega do *artifact* que contém a atualização ao dispositivo.

A gestão de atualizações e lançamentos de *software* são a terceira funcionalidade. Em vez de atualizar os dispositivos individualmente, cria uma abordagem orquestrada e gerida para atualizações de *software* em larga escala. Desta forma, os problemas são minimizados caso existam erros no processo de atualização.

Por fim, o sistema deve conter uma função de relatório e monitorização que acompanhe o lançamento da atualização, garantindo a sua segurança.

3.6.2 Alguns casos de aplicação [13]

Indústria automóvel

Os automóveis de hoje em dia têm cada vez mais *softwares* incorporados, desde assistentes de condução a ofertas de entretenimento. Em muitos casos, estas atualizações de *software* ou a correção de falhas de segurança são realizadas manualmente, o que é ineficiente para o fabricante e aborrecido para o clienet. No entanto, graças ao SOTA este processo por ser realizado remotamente o que economiza bastante tempo e não afeta a satisfação do cliente.

Medição e aquecimento inteligentes

Os produtores de soluções de energia estão a investir cada vez mais em ferramentas e serviços de atualização de *software*. Devido ao elevado número de utilizadores dos sistemas inteligentes de aquecimento, a sua atualização é um grande desafio. Posto isto, o SOTA é uma solução viável para este desafio.

Atualização de *software* como serviço

Um outro caso de aplicação do SOTA, ainda que não tão óbvio, são as atualizações de *software* como serviço. Para fornecedores de *software* ou serviços, estes podem desenvolver atualizações como solução de serviço para os seus clientes. A utilização de um serviço reutilizável a partir da *cloud* pode ajudar as empresas a criar valor, fornecendo aos seus clientes componentes personalizados, mais adequados às suas necessidades.

3.7 Plataformas IoT

A gestão de alguns dispositivos IoT seria bastante simples, no entanto as empresas começam a ter um elevado número de dispositivos IoT a realizarem as mais diversas tarefas. Estes dispositivos podem ser dos mais diversos tipos, não existindo padrões IoT industriais comuns ou métodos de conexão. Posto isto, gerir uma rede industrial de IoT significa ter a capacidade de responder ao desafio extremamente complexo, que é monitorizar e controlar uma variedade enorme de dispositivos IoT.

Felizmente, existem plataformas IoT que auxiliam as empresas a gerir com toda a segurança as pessoas, sistemas e dispositivos conectados a um ecossistema IoT [15].

As plataformas IoT fornecem um avanço na criação de sistemas de IoT, fornecendo ferramentas e recursos internos para tornar a IoT mais fácil e barata para as empresas, responsáveis pelo desenvolvimento e utilizadores. Uma plataforma IoT facilita a comunicação, fluxo de dados, gestão de dispositivos e a funcionalidade das aplicações.

As grandes características das plataformas IoT são [16]:

- conectar *hardware* como sensores e dispositivos;
- lidar com diferentes protocolos de comunicação de *hardware* e *software*;
- fornecer segurança e autenticação para dispositivos e usuários;
- obter, visualizar e analisar dados que os sensores e dispositivos adquirem;

- integrar tudo isto com sistemas comerciais existentes ou com outros serviços da *web*.

Uma plataforma IoT a nível industrial torna-se a base de uma implementação IoT bem sucedida. Sem uma plataforma eficaz, qualquer implementação de IoT industrial em larga escala irá falhar. Deste modo, a utilização de plataformas IoT a nível industrial traz os seguintes benefícios [15]:

- redução de custos - a plataforma IoT centraliza todo o processo de gestão, reduzindo grande parte da carga e dos custos;
- melhorar as operações - as soluções de IoT industrial fornecem informação em tempo real sobre a *performance* de equipamentos e dos funcionários, o que permite otimizar e melhorar os processos de negócios e os fluxos de trabalho;
- melhorar a produção - a plataforma IoT fornece a base para implantar novas aplicações de IoT industrial, as quais podem impulsionar a inovação e a eficiência;
- converter os dados recolhidos em lucro - a plataforma IoT permite recolher e analisar os dados de cada estágio da produção e do uso do produto, o que permite a criação de novos serviços e produtos;
- melhorar a segurança IoT - uma vez que os dispositivos possuem pouca segurança, a plataforma IoT garante que não surjam ataques a partir dos dispositivos.

De seguida apresentam-se três Plataformas IoT: *Bosch IoT Suite, Amazon Web Services IoT e Microsoft Azure IoT Suite.*

3.7.1 *Bosch IoT Suite*

A *Bosch IoT Suite* é a plataforma IoT com *software* Bosch, baseada em ferramentas *open source*. O núcleo da *Bosch IoT Suite* é construído sobre os projetos *open-source* do *Eclipse IoT*, que fornecem a tecnologia necessária para criar uma plataforma IoT de nível comercial. Esses projetos são [17]:

- *Eclipse Ditto* - local onde os dispositivos IoT e os seus *digital-twins* se reúnem. Este projeto é a base para a *Bosch IoT Things*;
- *Eclipse hawkBit* - permite criar uma solução para implantar atualizações de *software*. É a base do *Bosch IoT Rollouts*;
- *Eclipse Hono* - fornece interfaces de serviço remoto para conectar um grade número de dispositivos. É a base do *Bosch IoT Hub*;
- *Eclipse Vorto* - permite descrever os recursos abstratos e a funcionalidade dos dispositivos como *Informations Models*. A *Bosch IoT Suite* usa esse modelos para normalizar a *payload* dos dispositivos na utilização de APIs.

A partir desta plataforma a Bosch, os seus clientes e parceiros podem criar uma ampla variedade de soluções, serviços e projetos IoT. Esta plataforma IoT incorpora o *know-how* do Grupo Bosch, e está disponível numa grande diversidade de indústrias.

A *Bosch IoT Suite* oferece todas as funcionalidade necessárias para criar aplicações IoT, nomeadamente [18]:

- conexão e gestão confiáveis de dispositivos, sensores e *gateways*;
- gestão de acesso seguro;
- criação das representações digitais dos dispositivos físicos;
- visualização de todos os dados de diversas fontes num *dashboard*;
- execução do processo de atualização de *firmware* e lançamento de *software*;
- gestão e análise de dados;

Na Figura 3.3 apresentam-se os diferentes *layers* da *Bosch IoT Suite*.

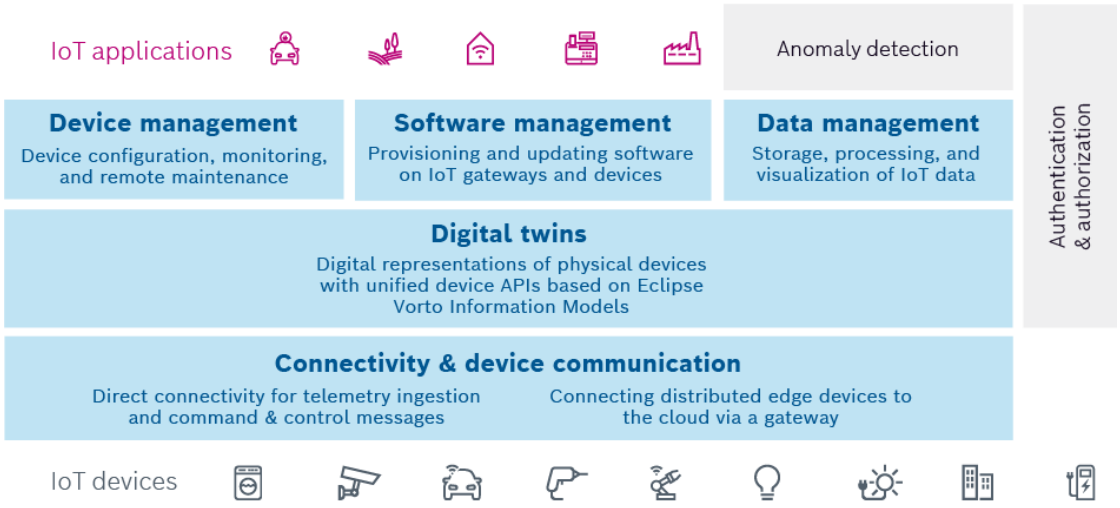

Figura 3.3: *Layers* da *Bosch IoT Suite* [19]

A plataforma divide-se em três níveis: conectividade, *digital twins* e gestão [19].

O nível da conectividade na *Bosch IoT Suite* permite conectar os dispositivos físicos diretamente à *cloud* ou indiretamente através de *gateways*. Faz parte deste nível o *Bosch IoT Hub*, o qual permite conectar de maneira fácil e segura os dispositivos a aplicações IoT. Suporta protocolos de comunicação IoT, tais como HTTP e MQTT [20].

Com o nível *digital twins* é possível criar na *cloud* uma representação dos dispositivos físicos. Através das suas capacidades, os *digital twins* permitem que os mundos real e digital estejam sincronizados. O serviço *Bosch IoT Things* faz parte deste nível. Este serviço permite que os dispositivos comuniquem com os *digital twins* com recurso ao *Bosch IoT Hub*, e por outro lado, que aplicações IoT interajam com os *digital twins* através de APIs [21].

No nível da gestão, esta pode ser feita ao nível do dispositivo, do *software* e dos dados.

Quanto ao dispositivo, as funcionalidades de gestão que a *Bosch IoT Suite* fornece permitem gerir o ciclo de vida de um dispositivo ou *gateway*, efetuar remotamente a configuração e o controlo de um dispositivo, bem como efetuar a monitorização e o diagnóstico remotamente.

Relativamente à gestão do *software*, a *Bosch IoT Suite* oferece todas as funcionalidades necessárias para efetuar de uma forma segura a atualização do *software* dos dispositivos e das *gateways*. O serviço responsável pela gestão do *software* dos dispositivos é o *Bosch IoT Rollouts*. Este serviço é um meio seguro e confiável de lidar com processos de implantação de *software* que envolvem um grande número de dispositivos [22].

Já as funcionalidades de gestão de dados permitem obter, processar e armazenar os dados. Além disso, a *Bosch IoT Suite* permite efetuar a análise dos dados recorrendo a algoritmos inteligentes.

3.7.2 *Amazon Web Service IoT*

A *Amazon Web Service (AWS)* tem serviços IoT abrangentes e especializados, desde a *edge* até à *cloud*. Reúne gestão e análise sofisticadas de dados em serviços fáceis de usar, projetados especificamente para dados IoT.

A AWS une a inteligência artificial e a IoT para criar dispositivos mais inteligentes. Os modelos podem ser criados na *cloud*, sendo depois implantados nos dispositivos.

A AWS IoT oferece serviços para todas as camadas de segurança, incluindo mecanismos de segurança preventivos.

Esta plataforma foi desenvolvida com base numa infraestrutura na *cloud* segura e comprovada, além de ser dimensionada para um elevado número de dispositivos e mensagens. A AWS IoT é integrada com outros serviços da AWS para criar soluções mais completas [23].

Os serviços da AWS IoT podem ser divididos em três grupos:

- *software* de dispositivo - permite conectar os dispositivos estando estes em execução na *edge*;
- conectividade e serviços de controlo - permitem proteger, controlar e gerir os dispositivos na *cloud*;
- serviços de análises - permitem trabalhar com dados de IoT mais rapidamente e assim extrair valor desses dados.

Alguns casos de uso desta plataforma são ao nível doméstico, comercial e industrial. Ao nível doméstico, a AWS IoT permite a criação de aplicações para automação residencial, redes domésticas e segurança. Ao nível comercial, através do desenvolvimento de aplicações para monitorização de tráfego, segurança pública e cuidados de saúde.

Ao nível industrial a AWS IoT pode ser usada para criar modelos de qualidade preditiva que ajudam a criar produtos melhores. Pode também ser usada na monitorização das condições de um equipamento. Através recolha de diferentes dados relativos a um equipamento é possível verificar se a sua *performance* é a ideal. Por fim, numa perspetiva de manutenção preditiva, com a AWS IoT é possível monitorizar continuadamente o *status*, a integridade e a *performance* dos equipamentos, permitindo detetar os problemas em tempo real [24].

A AWS IoT é uma plataforma indicada para aplicações industriais uma vez que:

- possui serviços que facilitam a implementação e a gestão de uma solução IoT industrial;

- fornece serviços projetados especificamente para facilitar a aquisição, o armazenamento e a análise de dados dos dispositivos;
- permite dimensionar aplicações industriais de IoT para um elevado número de dispositivos;
- garante a segurança dos dados e dos dispositivos.

Na Figura 3.4, apresenta-se a arquitetura e os serviços da AWS IoT para implementar aplicações industriais.

Figura 3.4: Arquitetura e os serviços da AWS IoT para implementar aplicações industriais [24]

3.7.3 *Microsoft Azure IoT Suite* [25]

Com os serviços e soluções da *Microsoft Azure IoT Suite*, as empresas conseguem atingir rapidamente grandes resultados comerciais usando dispositivos IoT.

A *Azure* desempenha um papel muito importante nas soluções de IoT dos seus clientes, fornecendo:

- serviços globalmente disponíveis, infinitamente escaláveis e económicos para dispositivos IoT;
- conectividade segura, usando protocolos padrão do setor e abordagens de segurança;
- um local central para recolher dados, enviar comandos e gerir dispositivos distribuídos geograficamente;
- análises avançadas que ajudam os seus clientes a extrair as principais informações dos dados recolhidos;

- um vasto leque de serviços que permitem extrair o valor na IoT e que simplificam a interação com a *Azure IoT Suite*;

- soluções pré-configuradas que ajudam os clientes a provisionar e obter rapidamente valor comercial da IoT.

Uma das soluções pré-configuradas entregues pela *Azure IoT Suite* é a monitorização remota, a qual simplifica a monitorização dos dados obtidos pelos dispositivos, a procura por condições específicas nos dados obtidos e a execução de ações. Com base nesta solução pré-configurada, são criadas instâncias dos seguintes serviços da *Azure*:

- *IoT Hub* - é a peça central de uma solução IoT da Azure, servindo como *cloud gateway* para estabelecer a conexão com os dispositivos. Permite a conexão de uma elevada quantidade de dispositivos, bem como processar uma grande quantidade de dados. Suporta protocolos de comunicação como o HTTP e o MQTT;

- *Stream Analytics* - é um serviço de análise de dados em tempo real que permite detetar anomalias e arquivar dados dos dispositivos IoT. Os resultados desse processamento podem ser enviados para outros serviços, tais como *Event Hubs, Power BI* ou serviços de armazenamento;

- Blob Storage - oferece um local para armazenar os dados que os dispositivos enviam para a *cloud*;

- *DocumentDB* - utilizado pela *Azure IoT Suite* para gerir os dados dos dispositivos provisionados, tais como configurações, estado e propriedades de segurança;

- *Web App* - é implantado para hospedar uma aplicação *web* usada para inspecionar o painel de dados do dispositivo, configurar e enviar comandos para os dispositivos, criar e atualizar a lógica de negócios e gerir ações orientadas a eventos, como enviar alertas quando determinados limites forem atingidos;

- *Logic App* - é implantado para a ajudar a integrar a solução de IoT à infraestrutura já existente e automatizar o processo de fluxo de trabalho. Este serviço permite que os responsáveis pelo desenvolvimento projetem fluxos de trabalho que se iniciam com o *trigger* e executam uma série de etapas.

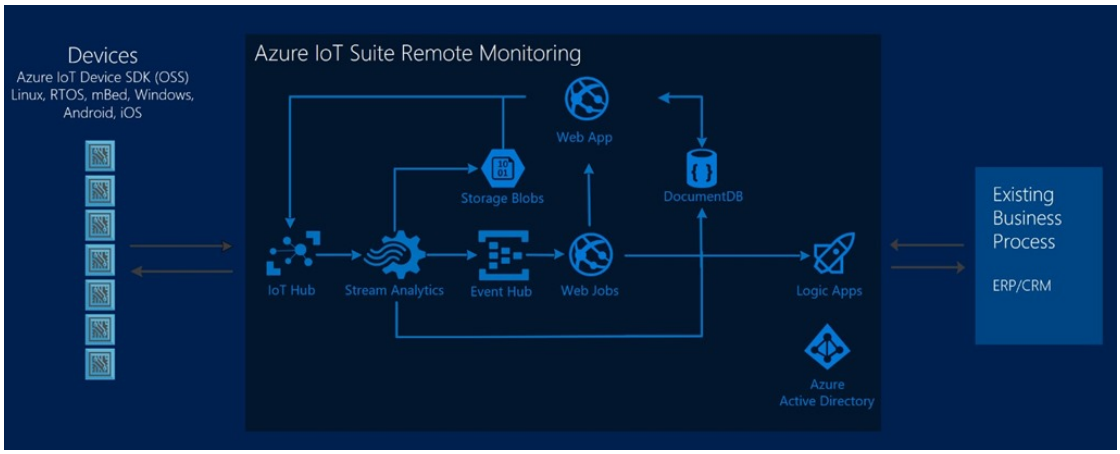

Figura 3.5: Arquitetura da solução pré-configurada *Azure IoT Suite Remote Monitoring*

3.8 Alguns protocolos de comunicação usados na IoT

3.8.1 MQTT [26]

O protocolo de comunicação MQTT *(Message Queue Telemetry Transport)* foi inventado e desenvolvido inicialmente pela IBM, tendo por base outro protocolo de comunicação o TCP/IP.

O protocolo MQTT suporta a comunicação assíncrona entre as partes, o que permite desacoplar o emissor e o recetor da mensagem, tanto no espaço como no tempo. Deste modo, o protocolo é escalável em ambientes de rede não confiáveis.

O MQTT é um protocolo leve e flexível, o que permite a sua implementação em dispositivos restritos, e em redes com uma baixa largura de banda e alta latência. A sua flexibilidade permite o suporte a diversos cenários de aplicação dos dispositivos e serviços IoT.

Um importante recurso deste protocolo é o modelo *publish and subscribe*. O protocolo MQTT define dois tipos de entidades na rede: o *Broker* e os clientes. O *Broker* tem a função de receber as mensagens dos clientes, encaminhado-as para os clientes de destino. A comunicação entre os clientes e o *Broker* é estabelecida da seguinte maneira:

- um cliente conecta-se ao *Broker* e subscreve um ou vários tópicos;
- por outro lado, um cliente conecta-se e envia mensagens para o *Broker* através da sua publicação em tópicos;
- por fim, o *Broker* encaminha a mensagem para todos os clientes que subscrevem esses tópicos.

Uma vez que as mensagens do protocolo MQTT são organizadas por tópicos, é fácil definir quais os tópicos com que cada cliente interage, seja publicando ou subscrevendo.

O modelo *publish and subscribe* permite que os clientes comuniquem um-para-um, um-para-muitos e muitos-para-muitos.

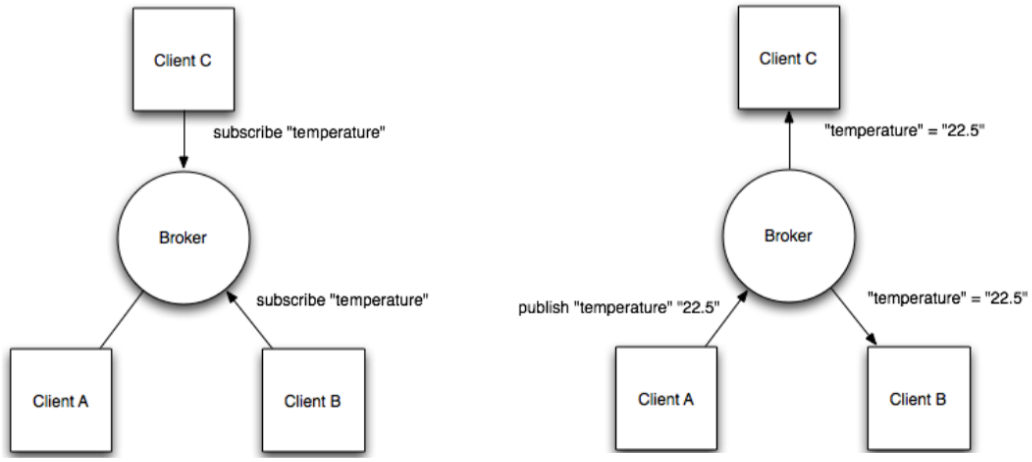

Figura 3.6: Esquema do modelo *publish and subscribe* com três clientes e o *Broker* [27]

Na Figura 3.6 apresenta-se um esquema do modelo *publish and subscribe* com três clientes e o *Broker* [27].

Os três clientes estão conectados ao *Broker* e os clientes B e C subscrevem o tópico *temperature*.

Por sua vez, o cliente A publica o valor 22,5 no tópico *temperature*. De seguida, o *Broker* encaminha a mensagem para os clientes que subscrevem este tópico.

Um recurso importante do protocolo MQTT é que ele permite *QoS (Quality of Service)*. Este recurso indica com que consistência as mensagens precisam de ser entregues aos clientes. Existem três níveis de QoS [28]:

- Nível 0 - no máximo uma vez: o destinatário não confirma a recessão da mensagem e o remetente não a armazena nem reenvia;
- Nível 1 - pelo menos uma vez: garante que uma mensagem seja entregue pelo menos uma vez ao destinatário;
- Nível 2 - exatamente uma vez: garante que cada mensagem seja recebida apenas uma vez pelo destinatário.

Ao nível da segurança, o *Broker* pode requerer a autenticação dos clientes para que estes se conectem. Esta autenticação é efetuada através de um *username* e de uma *password*. Além disso, para garantir a privacidade, a ligação TCP pode ser criptografada através dos certificados digitais SSL/TLS.

3.8.2 HTTP [29]

O HTTP (Hypertext Transfer Protocol) é a base para a comunicação de dados, sendo usado desde 1990 na World-Wide Web.

Este protocolo funciona com base no modelo de solicitação e resposta. Um cliente envia uma solicitação ao servidor, composta pelo tipo de solicitação, URI (Uniform Resource Identifier) e a versão do protocolo. A solicitação pode conter ainda informações do

cliente e um corpo com conteúdo. Por sua vez, o servidor responde enviando uma *status line* composta pela versão do protocolo e o código de sucesso ou de erro. A resposta pode ainda ser constituída por informações do servidor e pelo conteúdo solicitado pelo cliente. Depois de enviada a resposta por parte do servidor, a ligação é terminada.

As mensagens HTTP são formatadas de acordo com a Figura 3.7.

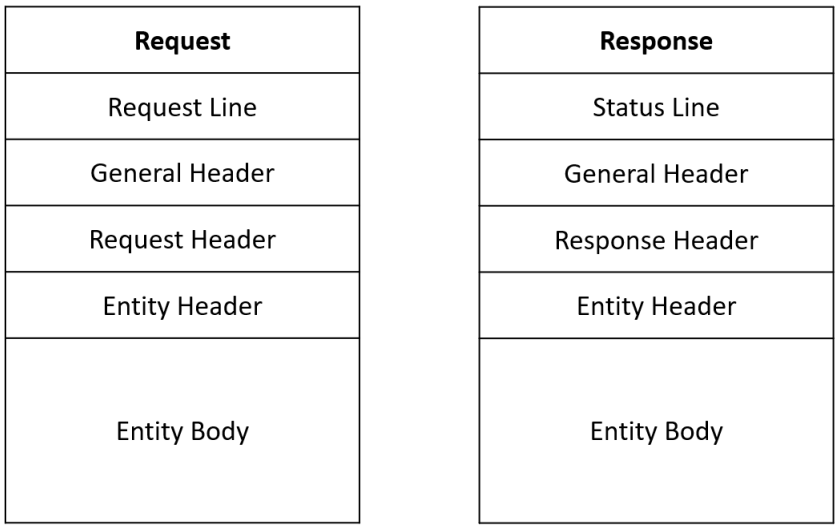

Figura 3.7: Formato das mensagens enviadas pelo cliente (à esquerda) e pelo servidor (à direita), com base em [29]

De maneira geral, a comunicação HTTP é iniciada pelo cliente e consiste numa solicitação a ser aplicada num recurso do servidor.

A comunicação HTTP geralmente ocorre através de conexões TCP/IP. A porta padrão é a TCP 80, ainda que outras portas possam ser usadas.

Os recursos solicitados pelo cliente são identificados através do URI. O URI é uma *string* que inclui o nome, o local e qualquer outra característica de um determinado recurso. No protocolo HTTP, um recurso pode ser solicitado através da seguinte *string*:

"http://"servidor [":"porta] [localização_do_recurso ["?"parâmetros]]

Depois de estabelecida a ligação, o cliente pode efetuar diversos tipos de pedidos ao servidor, por exemplo:

- GET - através deste método, o servidor responde com informações sobre o recurso pedido, incluindo o seu conteúdo;
- HEAD - semelhante ao método anterior, mas o servidor não envia o conteúdo do recurso solicitado;
- POST - este método é utilizado para submeter uma entidade ao recurso solicitado;
- DELETE - com este método é possível eliminar um determinado recurso.

Consoante o tipo de pedido efetuado pelo cliente e dependendo da forma como o servidor foi capaz de reconhecer e executar esse pedido, o servidor envia uma mensagem de resposta. Um dos constituintes dessa mensagem é o *status code*, o qual é constituído por três dígitos. A classe da resposta é definida através do primeiro dígito do *status code*, sendo que existem cinco classes:

- 1xx: Informativo - solicitação recebida, processo contínuo;
- 2xx: Sucesso - a ação foi recebida, compreendida e aceite com êxito;
- 3xx: Redirecionamento - devem ser tomadas mais ações para concluir a solicitação;
- 4xx: Erro do cliente - a solicitação contém sintaxe incorreta ou não pode ser atendida;
- 5xx: Erro do servidor - o servidor falhou ao atender uma solicitação aparentemente válida.

HTTPS [30]

Com o aumento do uso do protocolo HTTP em aplicações confidenciais, surgiu a necessidade de implementar medidas de segurança. Essas medidas são implementadas com recurso a dois certificados o SSL *(Secure Sockets Layer)* e o TLS *(Transport Layer Security)*. Deste modo, o HTTPS tem por base o protocolo HTTP adicionando-se os certificados de segurança referidos.

Relativamente ao protocolo HTTP, surgem duas diferenças significativas: o número da porta utilizada e o formato do URI. Quanto ao número da porta, este passa a ser o 443. Esta alteração é efetuada para que seja possível destinguir qual o protocolo usado. Quanto ao formato do URI, passa a iniciar-se por "https://".

Capítulo 4

Solução Proposta

A ferramenta de gestão remota de dispositivos foi dividida em duas partes. Uma parte relacionada com a aquisição, processamento e monitorização de valores por parte dos dispositivos e outra parte relacionada com a gestão do *firmware* e do *software* que os dispositivos têm instalado.

4.1 Requisitos da ferramenta

A ferramenta deve cumprir os seguintes requisito relativamente à parte relacionada com a aquisição, processamento e monitorização de valores:

- Aquisição de dados provenientes de sensores e realizar o seu processamento;
- Permitir o registo de dispositivos na Plataforma IoT (Monitorização);
- Conexão com a Plataforma IoT e realizar o envio dos dados obtidos;
- Armazenar os dados numa Base de Dados;
- Monitorização dos valores obtidos;
- Obter o estado de funcionamento do dispositivo.

Quanto à parte relacionada com a gestão do *firmware* ou *software*, a ferramenta deve cumprir os seguintes requisitos:

- Permitir o registo de dispositivos na Plataforma IoT (Atualização);
- Criar pacotes de atualizações;
- Atualizar os dispositivos através de tecnologias *Over-the-Air* (SOTA e FOTA);
- Obter informações sobre a atualização atualmente instalada;
- Permitir a gestão das atualizações criadas;

4.2 *Use Cases*

De modo a encontrar uma solução para a gestão remota de dispositivos, são desenvolvidos dois *Use Cases*: o *Use Case 1* em que o dispositivo que adquire os dados é um IoT *Device*, isto é, tem ligação à Internet, e o *Use Case 2* em que não, sendo necessário o uso de uma *gateway* de suporte.

Assim, para o *Use Case 1* é proposta a implementação da Figura 4.1 e para o *Use Case 2* a implementação da Figura 4.2.

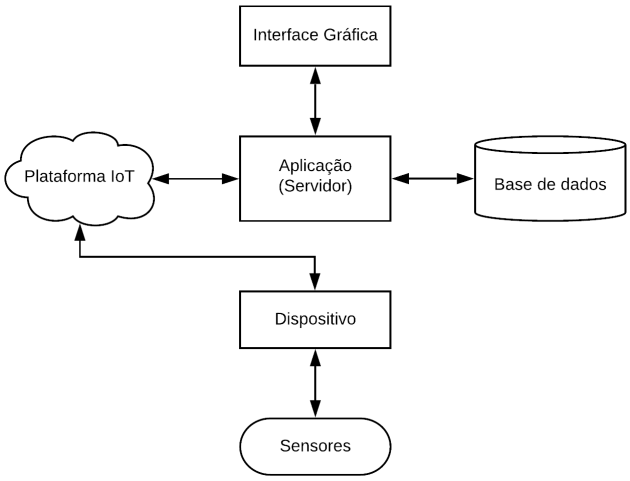

Figura 4.1: Proposta de implementação do *Use Case 1*

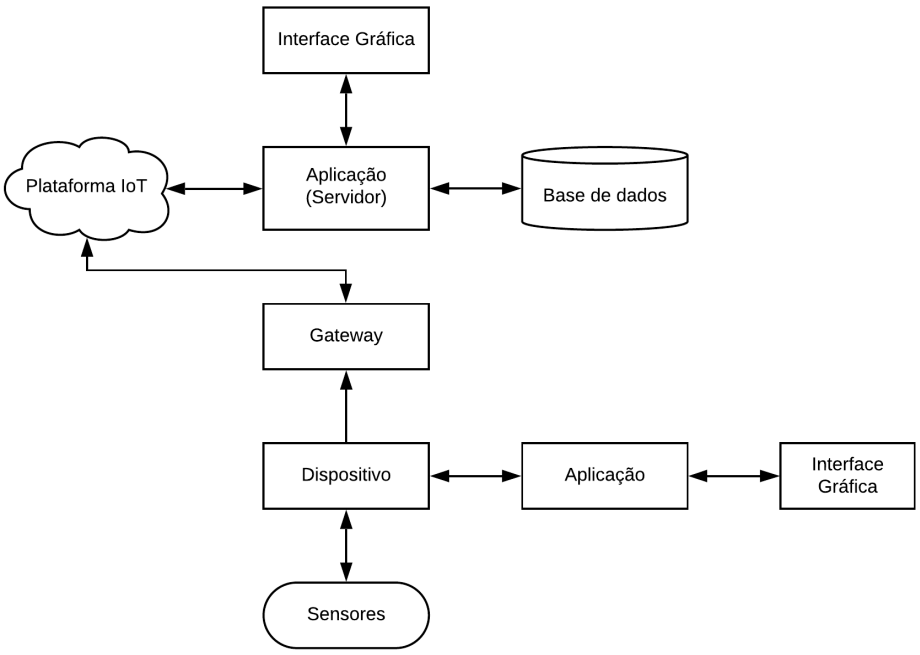

Figura 4.2: Proposta de implementação do *Use Case 2*

De uma forma geral, os *Use Cases* são muito semelhantes. Ambos contêm os sensores, o dispositivo, a Plataforma IoT, a Aplicação (Servidor) e a Interface Gráfica associada a ela, e a Base de Dados. Também em termos de fluxo de informação, a arquitetura é a mesma.

Tal como referido, o que diferencia os dois *Use Cases* é o facto do dispositivo utilizado na Figura 4.2 não ter ligação à Internet. Assim, apesar do fluxo de informação ser o mesmo, foi adicionada uma *gateway*. Além disso, a implementação da Figura 4.2 conta ainda com uma outra Aplicação e a Interface Gráfica associada a ela.

Uma vez que as implementações propostas apresentadas são baseadas na arquitetura IoT, é relevante realizar a associação dos elementos das implementações com os *layers* de uma arquitetura IoT. Essa associação é apresentada de seguida:

- Sensores, Dispositivo e *Gateway* - *Objects Layer*;
- Comunicação entre o Dispositivo ou a *Gateway* e a Plataforma IoT - *Object Abstraction Layer*;
- Plataforma IoT e Base de Dados - *Service Management Layer*;
- Aplicação e Aplicação (Servidor) - *Application Layer*;
- Interfaces Gráficas - *Business Layer*.

4.3 Descrição dos elementos das implementações

Apresentadas as implementações de cada *Use Case*, importa fazer uma explicação das funções de cada elemento das implementações.

4.3.1 Sensores

Começando pelos sensores, estes são os responsáveis pela recolha de valores ao longo do tempo, sendo posteriormente enviados para o dispositivo periodicamente. Estes valores podem ser dos mais variados tipos, tais como condições ambientais de um local de trabalho, condições de funcionamento de um equipamento, cadência de produção de uma linha de montagem, entre outros. Estes valores poderão ser depois tratados e analisados de forma a otimizar um processo produtivo ou consumos energéticos, por exemplo.

4.3.2 Dispositivo

Passando agora para o dispositivo, uma das suas funções comum aos dois *Use Cases* é a comunicação com os sensores. Como já referido, os valores adquiridos pelos sensores são enviados para o dispositivo, e aqui são sujeitos a alguns tipos de tratamento, nomeadamente a sua conversão para valores com os quais o ser humano tem maior sensibilidade.

A outra função do dispositivo é diferente de acordo com o *Use Case*. No caso do *Use Case 1*, como o dispositivo consegue estabelecer a ligação diretamente com a Plataforma IoT, este tem também a função de enviar os valores para a Plataforma IoT.

Quando se trata do *Use Case 2*, como é a *gateway* que estabelece a ligação com a Plataforma IoT, o dispositivo tem a função de enviar os valores para a *gateway* e para a Aplicação.

Quanto à atualização do dispositivo, esta só se aplica ao dispositivo do *Use Case 1*, dada a sua capacidade de comunicação com a Plataforma IoT. Para se atualizar, o dispositivo verifica através da Plataforma IoT se tem alguma atualização para fazer. Se tiver, transfere o *firmware* ou *software* correspondente à atualização e efetua a sua instalação.

Relativamente às comunicações estabelecidas pelo dispositivo, no *Use Case 1* estas são estabelecidas com os sensores e com a Plataforma IoT, enquanto que no *Use Case 2*, estas são estabelecidas com os sensores, com a *gateway* e com a Aplicação.

A comunicação dispositivo - sensores é do tipo *master* - *slave*, uma vez que o dispositivo pede aos sensores os valores que estes depois enviam.

No *Use Case 1* a comunicação dispositivo - Plataforma IoT é bidirecional, já que o dispositivo é capaz de enviar para a Plataforma os valores que está a monitorizar, mas também é capaz de receber comandos enviados através da Plataforma. Também a atualização do dispositivo é efetuada através desta comunicação bidirecional. No entanto, para que esta comunicação possa ser estabelecida, o dispositivo tem de estar registado na Plataforma IoT e configurado para efetuar as comunicações.

Já no *Use Case 2* a comunicação dispositivo - *gateway* ocorre apenas no sentido do envio dos valores do dispositivo para a *gateway*. Por outro lado, a comunicação dispositivo - Aplicação é bidirecional, para que o dispositivo possa enviar os valores, mas também para que possa receber as mensagens enviadas pela Aplicação.

4.3.3 Aplicação

O próximo elemento está apenas relacionado com o *Use Case 2*, e trata-se da Aplicação. Esta Aplicação tem como objetivos receber os valores que são enviados pelo dispositivo, obter informações acerca do dispositivo e dos sensores ao nível do seu funcionamento, e obter informações acerca das comunicações estabelecidas entre eles. Para a sua obtenção, a Aplicação e o dispositivo têm de ter uma ligação estabelecida. Como referido na Secção anterior, essa ligação é bidirecional.

Interface Gráfica

Associada à Aplicação existe uma Interface Gráfica, na qual é possível visualizar os valores recebidos e as informações obtidas. É também a a partir da Interface Gráfica que se enviam ordens para que a Aplicação envie mensagens para o dispositivo.

4.3.4 *Gateway*

Outro elemento relacionado apenas com o *Use Case 2* é a *gateway*. Uma das funções deste elemento é a recessão dos valores enviados pelo dispositivo. A outra função é estabelecer a ligação com a Plataforma IoT, uma vez que o dispositivo utilizado não o consegue fazer.

A ligação estabelecida entre a *gateway* e a Plataforma IoT é bidirecional, para que a *gateway* envie para a Plataforma IoT os valores recebidos, mas também para que receba comandos enviados através da Plataforma IoT. É também através desta ligação bidirecional que a *gateway* efetua as suas atualizações.

4.3.5 Plataforma IoT

A Plataforma IoT é um dos elementos comuns a ambos os *Use Cases*. Esta Plataforma reside na Internet e possui serviços fundamentais para desenvolver uma solução para a gestão remota de dispositivos. No entanto, para que se possam utilizar os serviços que disponibiliza, é necessário subscrevê-los e também efetuar o registo dos dispositivos nesses serviços.

É aqui que ficam armazenados temporariamente os valores que estão a ser monitorizados e que são enviados pelo dispositivo ou pela *gateway*.

Através da Plataforma IoT é possível enviar comandos para o dispositivo no *Use Case 1* ou para a *gateway* no *Use Case 2*. Este envio de comandos pode servir, por exemplo, para receber uma mensagem do dispositivo ou da *gateway* e desta forma perceber se estão em funcionamento ou não.

É também através da Plataforma IoT que se efetua a criação das atualizações de *firmware* ou *software*, ficando aqui guardadas e prontas a serem transferidas e instaladas pelo dispositivo no *Use Case 1* ou pela *gateway* no *Use Case 2*.

4.3.6 Aplicação (Servidor)

A Aplicação (Servidor) é também comum a ambos os *Use Cases* e é através dela que se estabelece a integração com os serviços que a Plataforma IoT disponibiliza. A Aplicação (Servidor) terá uma parte relacionada com a monitorização dos valores e outra parte relacionada com a gestão do *firmware* e *software*.

Quanto à parte da monitorização, a Aplicação (Servidor) tem a função de extrair os valores da Plataforma IoT, armazená-los numa Base de Dados e permitir a sua visualização. Ainda nesta parte, tem também a função de obter alguns dados sobre o funcionamento dos dispositivos.

No que diz respeito à parte relacionada com a gestão do *software* ou *firmware*, através da integração da Plataforma IoT com a Aplicação (Servidor), é possível obter informações sobre a atualização que cada dispositivo ou *gateway* tem instalada e sobre todas as atualizações criadas, criar atualizações e atualizar o dispositivo ou a *gateway*.

Quanto às comunicações estabelecidas, a Aplicação (Servidor) estabelece comunicações com a Plataforma IoT e com a Base de Dados.

A comunicação Aplicação (Servidor) - Plataforma IoT é bidirecional, uma vez que no caso da monitorização de valores a Aplicação (Servidor) obtém os valores armazenados na Plataforma IoT, e é também através desta comunicação que a Aplicação (Servidor) consegue enviar comandos para o dispositivo no *Use Case 1* ou para a *gateway* no *Use Case 2*. No caso da gestão do *firmware* ou *software* também se utiliza essa comunicação bidirecional, uma vez que a Aplicação (Servidor) pode obter os dados relacionados com todas as atualizações criadas, mas quando se cria uma atualização, a Aplicação (Servidor) procede ao envio de dados para a Plataforma IoT.

Relativamente à comunicação Aplicação (Servidor) - Base de Dados, esta também é bidirecional, já que em primeiro lugar a Aplicação (Servidor) efetua o armazenamento dos dados que extrai da Plataforma IoT, mas depois seleciona os valores para permitir a sua visualização na Interface Gráfica.

Interface Gráfica

Associada à Aplicação (Servidor) existe uma Interface Gráfica. Nesta Interface Gráfica é possível visualizar os valores que vão sendo extraídos da Plataforma IoT, os dados relacionados com o *firmware* ou *software* instalado no dispositivo e na *gateway* e ainda os dados relacionados com todas as atualizações criadas. É também a partir da Interface Gráfica que se enviam comandos para a Aplicação (Servidor), para que esta efetue diversas ações através da comunicação com a Plataforma IoT.

4.3.7 Base de Dados

Através da Aplicação (Servidor), os valores obtidos vão sendo armazenados numa Base de Dados. Estes valores serão depois visualizados na Interface Gráfica associada à Aplicação (Servidor), podendo ser também objeto de análise no âmbito da *Condition Monitoring*. Como referido, a comunicação com a Base de Dados é feita exclusivamente pela Aplicação (servidor).

Capítulo 5

Implementação

Neste capítulo apresentam-se as implementações de cada *Use Case* , assim como componentes, serviços e ferramentas usados e as comunicações estabelecidas. Em cada uma das implementações existe uma parte de aquisição, processamento e monitorização de valores e outra parte relacionada com a gestão do *software* ou *firmware* de cada dispositivo. A implementação do *Use Case 1* é apresentada na Figura 5.1, e a implementação do *Use Case 2* é apresentada na Figura 5.2.

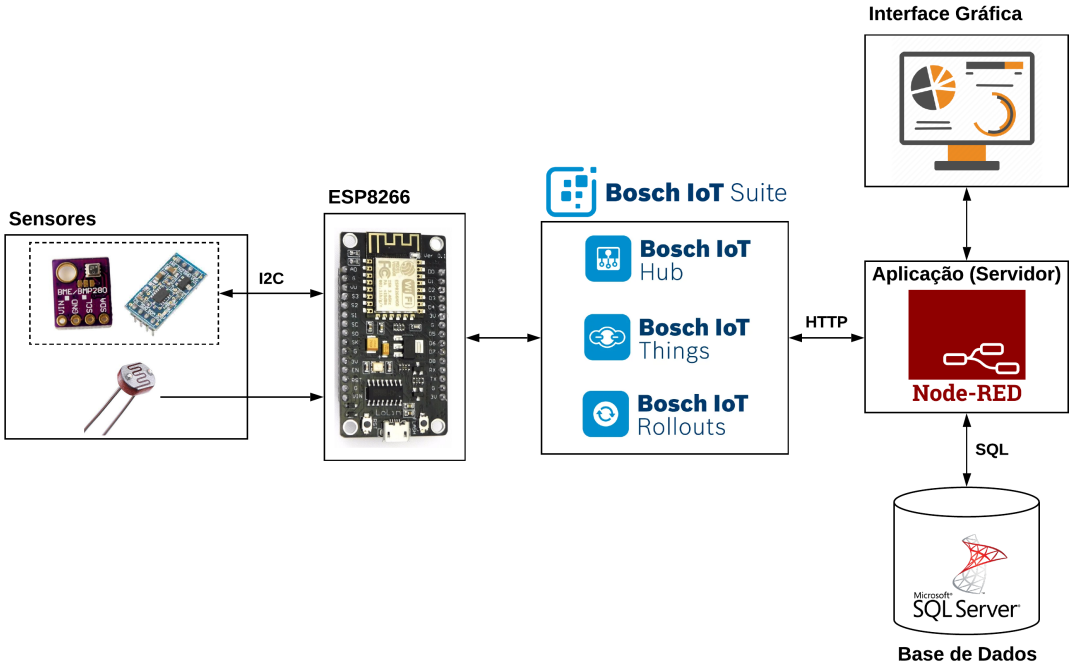

Figura 5.1: Implementação do *Use Case 1*

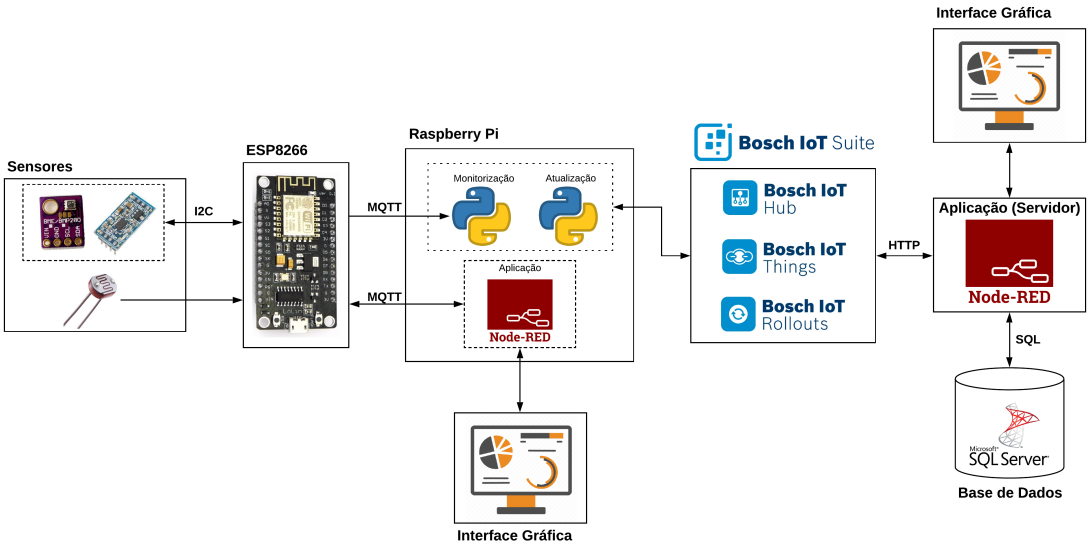

Figura 5.2: Implementação do *Use Case 2*

Fazendo uma comparação entre as Figuras 5.1 e 5.2 e a Figura 3.3, que representa os *layers* da *Bosch IoT Suite*, verifica-se que o ESP8266 no *Use Case 1* e o *Raspberry Pi* no *Use Case 2* correspondem ao *layer* dos *IoT Devices*. Os serviços *Bosch IoT Hub, Bosch IoT Things* e *Bosch IoT Rollouts* correspondem aos *layers Connectivity & device communication, Digital twins* e *Software Management*, respetivamente. Por fim, a Aplicação (Servidor) corresponde ao *layer IoT applications*, sendo este componente desenvolvido integralmente pelo autor. A par da Aplicação (Servidor) também a programação e configuração dos dispositivos foi efetuada pelo autor.

5.1 Descrição geral das implementações

5.1.1 Aquisição, processamento e monitorização

Relativamente à parte da aquisição, processamento e monitorização de valores, estes correspondem a valores de temperatura, humidade, acelerações e luminosidade, os quais são adquiridos pelos sensores BME280, MMA7455 e por uma fotorresistência.

Depois da sua obtenção são enviados para o ESP8266, onde são convertidos para valores mais usuais. No *Use Case 2*, os valores são ainda enviados pelo ESP8266 para o Raspberry Pi e para a Aplicação. As comunicações ESP8266 – Raspberry Pi (*script Python* Monitorização) e ESP8266 – Aplicação são estabelecidas através do protocolo de comunicação MQTT.

A Aplicação presente no *Use Case 2* é desenvolvida com recurso ao *Node-RED* e permite obter os valores enviados pelo ESP8266, informações acerca do dispositivo e dos sensores e das comunicações estabelecidas. Associada à Aplicação existe uma Interface Gráfica que permite visualizar essas informações e ainda executar algumas ações.

De seguida, os valores adquiridos são enviados para a *Bosch IoT Things* pelo ESP8266 no *Use Case 1*, ou pelo Raspberry Pi no *Use Case 2*. A *Bosch IoT Things* é um serviço da *Bosch IoT Suite*, sendo a comunicação com este serviço estabelecida através de um outro serviço da *Bosch IoT Suite*, o *Bosch IoT Hub*.

Na Aplicação (Servidor), também desenvolvida em *Node-RED*, os valores são extraídos da *Bosch IoT Things* e são guardados numa Base de Dados SQL Express. Além de extrair os valores e efetuar o seu armazenamento, a Aplicação (Servidor) também permite obter informações sobre o estado de funcionamento do dispositivo e enviar alertas. A Aplicação (Servidor) permite ainda o registo de dispositivos na *Bosch IoT Suite*.

Associada à Aplicação (Servidor) existe uma Interface Gráfica composta por seis menus, sendo que dois deles estão relacionados com a parte da monitorização de valores. Num é possível visualizar os valores que estão a ser obtidos e armazenados, visualizar as informações sobre o estado de funcionamento do dispositivo e executar algumas ações. O outro menu, juntamente com as funcionalidades da Aplicação (Servidor), permite registar um dispositivo na *Bosch IoT Suite*.

5.1.2 Gestão do *firmware* ou *software*

No que diz respeito à parte relacionada com a gestão do *firmware* ou *software*, essa gestão afeta o ESP8266 no *Use Case 1* e o Raspberry Pi no *Use Case 2*.

Para efetuar a gestão do *firmware* ou *software* dos dispositivos, a Aplicação (Servidor) estabelece a comunicação com o *Bosch IoT Rollouts*, outro serviço da *Bosch IoT Suite*. Com a Aplicação (Servidor) é possível criar atualizações, obter os dados relacionados com todas as atualizações criadas, atribuir atualizações aos dispositivos e obter informações sobre o *firmware* ou *software* que o dispositivo tem instalado.

Como referido na Secção anterior, associada à Aplicação (Servidor) existe uma Interface Gráfica. Dos seis menus que a compõem, os restantes quatro estão relacionados com a gestão do *firmware* ou *software*. Deste modo, um dos menus serve para criar uma atualização e atribuí-la a um dispositivo, outro menu serve para visualizar os dados da atualização que cada dispositivo tem atualmente instalada, e por fim, os outros dois menus servem para visualizar as informações relacionadas com todas as atualizações criadas.

Do lado do dispositivo, seja o ESP8266 no *Use Case 1* ou o Raspberry Pi no *Use Case 2*, este também estabelece comunicação com o *Bosch IoT Rollouts*. É através dessa ligação que o dispositivo verifica se tem alguma atualização atribuída. Caso tenha, deve fazer a sua transferência e posterior instalação. No fim da instalação, seja com sucesso ou não, o dispositivo deve enviar o *feedback* para o *Bosch IoT Rollouts*. No caso do ESP8266, o mesmo *firmware* tem as funções de monitorização e de atualização. Em relação ao Raspberry Pi, existe o *script Python* Atualização, responsável pelas atualizações.

5.2 Considerações sobre alguns dos componentes

Tendo em conta os dispositivos e *softwares* usados, importa referir as razões que os levaram a ser escolhidos.

ESP8266

Começando pelo Dispositivo, a escolha recaiu sobre o ESP8266 dado o seu poder de processamento face ao custo e dada a capacidade de se poder comunicar por *Wi-Fi*. Além disso, é um dispositivo usado na Bosch e também já existia alguma familiarização com o dispositivo de outras Unidades Curriculares.

Raspberry Pi

Quanto ao Raspberry Pi, foi escolhido para desempenhar a função de *gateway*, uma vez que possui elevadas capacidades de processamento, possui um sistema operativo e tem um preço reduzido. Tal como o ESP8266, é um dispositivo usado na Bosch.

Node-RED

O *Node-RED* é uma ferramenta desenvolvida pela IBM para programação rápida de dispositivos no *Edge*. Esta ferramenta permite a interligação de dispositivos, APIs e serviços utilizando fluxos. Tanto a Aplicação como a Aplicação (Servidor) são desenvolvidas com recurso a esta plataforma *low code*. Tal como no caso do ESP8266, uma das razões para a sua escolha é o facto da sua utilização noutras Unidades Curriculares.

Bosch IoT Suite

A Plataforma IoT escolhida foi a *Bosch IoT Suite*. Depois de analisadas outras plataformas IoT, efetuou-se essa escolha uma vez que a *Bosch IoT Suite* contém serviços e tecnologias que permitem atingir os objetivos propostos, bem como cumprir os requisitos definidos para a ferramenta que se pretende desenvolver. A compatibilidade dos serviços da *Bosch IoT Suite* com os outros componentes das implementações foi um aspeto positivo que contribuiu para a sua escolha. Por exemplo, o *Bosch IoT Hub* possibilita a utilização de múltiplos protocolos de comunicação, e a *Bosch IoT Things* disponibiliza o acesso aos dados via API.

Outra vantagem encontrada relaciona-se com o facto de ser *pay as you grow*, sendo possível no caso em questão a sua utilização sem qualquer custo ou limitação no tempo de utilização. Naturalmente, existem limitações como o número máximo de dispositivos associados a cada serviço e de ainda estar em desenvolvimento. No entanto, o facto de pertencer à Bosch, também facilita no acesso à documentação e aos próprios *developers*.

Por fim, esta plataforma estava também a ser utilizada pela equipa da Bosch o que permitiu troca de conhecimento e aprendizagem.

5.3 *Hardware*

5.3.1 ESP8266 NodeMCU [31]

O ESP8266 NodeMCU é uma placa com WiFi integrado, desenvolvido pela *Espressif Systems* para aplicar em soluções de IoT. Possui um módulo ESP-12E que contém o *chip* ESP8266 com um microprocessador Tensilica Xtensa LX106 RISC de 32 bits que opera entre 80 e 160MHz. Possui ainda 128 kB de memória RAM e 4 MB de memória Flash. O transeptor WiFi que integra com 802.11 b/g/n, permite que se possa não apenas conectar a uma rede WiFi, mas também criar a sua própria rede, permitindo que outros dispositivos se conectem a ele. [32]

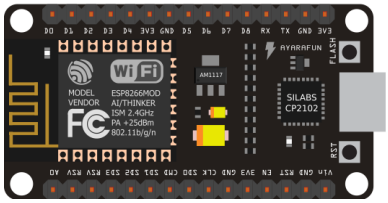

Figura 5.3: ESP8266 NodeMCU

Relativamente à tensão de operação do ESP8266, esta varia entre 3V e 3,6V. No entanto, a existência de um regulador de tensão permite que esta tenha um valor constante de 3,3V. Consegue ainda fornecer até 600 mA de intensidade de corrente. Contém 3 pinos de 3,3V que permitem alimentar dispositivos ligados a ele. A alimentação da placa é efetuada através de um cabo Micro USB, ou em alternativa, por uma fonte regulada de 5V diretamente no pino VIN. [32]

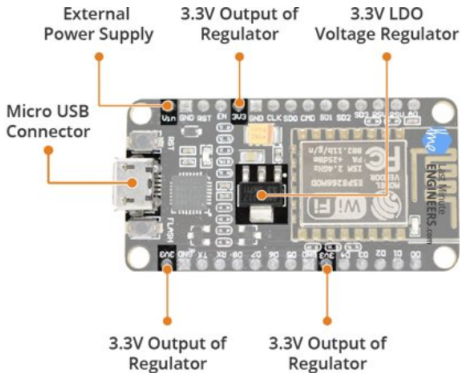

Figura 5.4: Alimentação e tensões na placa [32]

O ESP8266 NodeMCU possui 2 botões: o botão de RESET usado para reiniciar o chip e o botão FLASH usado durante a atualização de *firmware*. Para além dos botões, a placa possui um LED programável pelo utilizador e ligado ao pino D0 da placa. Para a comunicação em série, inclui um CP2102 USB-to-UART Bridge Controller, que converte os sinais USB para série, permitindo que o computador comunique com o chip e o programe.

Em relação aos periféricos e I/O, o ESP8266 NodeMCU possui 30 pinos, dos quais 17 GPIOS, divididos pelos dois lados da placa. [32]

- 4 Pinos de alimentação: 1 pino VIN para alimentar a placa com uma tensão até 5V e 3 pinos para alimentar equipamentos com uma tensão de 3,3V;
- 4 Pinos terra (GND);
- 2 Pinos I2C;
- 17 Pinos GPI0 – podem ser configurados para outras; funcionalidades, tais como I2C, I2S, UART, PWM, SPI, entre outras;

- 1 Pino ADC;
- 5 Pinos UART;
- 4 Pinos SPI + 4 Pinos HSPI;
- 6 Pinos SDIO (SD Card);
- 4 Pinos PWM;
- 4 Pinos Controlo.

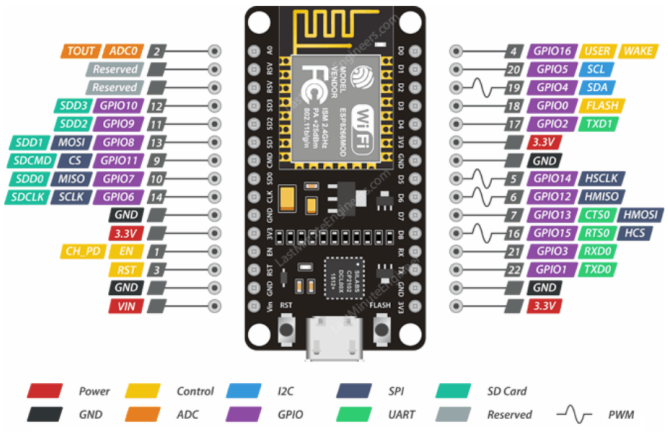

Figura 5.5: Esquema dos pinos do ESP8266 NodeMCU [32]

5.3.2 Raspberry Pi 3 Model B+

O Raspberry Pi é um computador de baixo custo e de pequenas dimensões desenvolvido no Reino Unido pela Fundação Raspberry Pi. Para a sua utilização, apenas é necessário ligar um rato, um teclado e um monitor. Caso esteja ligado à Internet, é possível aceder remotamente ao Raspberry Pi, através dos protocolos SSH ou VNC.

A principal função do Raspberry Pi é oferecer uma alternativa barata, prática e acessível para que se possa explorar todas as capacidades da computação. Além disso, também permite a familiarização com linguagens de programação tais como Scratch e Python.

Apesar das pequenas dimensões e do seu aspeto pouco convencional, o Raspberry Pi é um computador como outro qualquer. Com o Raspberry Pi é possível efetuar muitas tarefas que se realizam noutros computadores, como por exemplo navegar na Internet, criar textos, reproduzir conteúdo multimédia ou até mesmo jogar [33].

Neste trabalho utiliza-se o Raspberry Pi 3 Model B+ (Figura 5.6).

Este modelo possui um processador Broadcom BCM2837B0 de 4 núcleos que opera a 1,4 GHz. Conta ainda com um dissipador de calor, para que a sua refrigeração seja mais eficiente.

Relativamente aos modelos anteriores, a maioria das mudanças são relacionadas com as comunicações. No Raspberry Pi 3 Model B+ foi introduzido WiFi de banda-dupla, permitindo que este comunique com redes que operam nas frequências de 2,4 GHz ou 5 GHz. Este modelo também suporta o Bluetooth 4.2.

Figura 5.6: Raspberry Pi 3 Model B+ [34]

Outra grande novidade deste modelo é a incorporação do Gigabit Ethernet. Contudo, a comunicação é limitada pela velocidade máxima do USB 2.0. Assim, a velocidade máxima de recessão de dados é 300 Mbps em vez de 1 Gbps. É também possível alimentar o Raspberry Pi através do PoE (Power over Ethernet), no entanto é necessário *hardware* adicional (PoE HAT) [35].

O Raspberry Pi Model 3 B+ apresenta as seguintes especificações [34]:

Processador: Broadcom BCM2837B0 SoC, Cortex-A53 (ARMv8) 64-bit Quadcore @ 1.4GHz;

Memória RAM: 1GB LPDDR2 SDRAM;

Memória flash: cartão de memória microSD,

Conectividade sem fio: Wireless 2.4GHz e 5GHz, Bluetooth 4.2/BLE;

Conexão com fio: 40 pinos (26 GPIO), Gigabit Ethernet (até 300 Mbps);

Multimédia: H.264, MPEG-4 decode (1080p30), H.264 encode (1080p30), OpenGL ES 1.1;

Conectores: slot cartão microSD, 4 × USB 2.0, Ethernet, HDMI, MIPI DSI (display), MIPI CSI (câmara), áudio estéreo de 4 pólos;

Fonte de alimentação: 5V / 2.5A DC (via microUSB ou GPIO), Power over Ethernet (requer HAT).

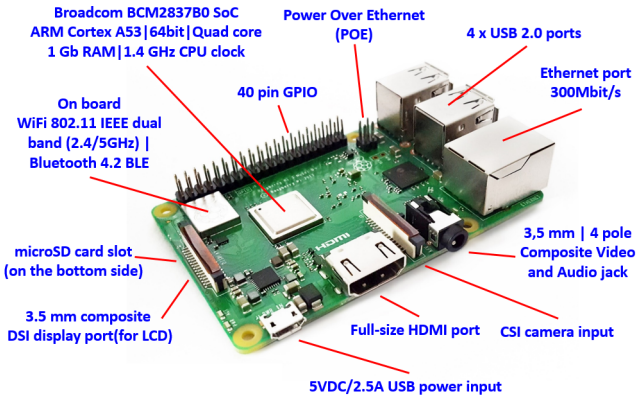

Figura 5.7: Componentes e especificações do Raspberry Pi 3 Model B+ [36]

5.3.3 BME280 [37]

Para adquirir os valores de temperatura e humidade, recorre-se ao sensor BME280.

O BME280 (Figura 5.8) é um sensor digital de temperatura, humidade relativa e pressão de última geração, fabricado pela Bosch. É um sucessor de sensores como o BMP180, BMP085 ou BMP183.

Figura 5.8: Sensor BME280

Este sensor permite medir a temperatura entre -40ºC e 84ºC com precisão de ± 1ºC, a humidade relativa entre 0 e 100% com precisão de ± 3% e a pressão barométrica entre 300Pa e 1100hPa com precisão de ± 1hPa. Uma vez que as medidas de pressão são tão precisas, é também possível obter a altitude com precisão de ± 1m.

Além do sensor BME280, o módulo possui um regulador de tensão de 3,3V e um conversor do nível de tensão para a comunicação I2C. Deste modo, pode ser usado com um microcontrolador de 3,3V ou 5V, como por exemplo o ESP8266.

O BME280 tem um baixo consumo de energia, o que permite a sua implementação em dispositivos que funcionam com bateria.

O módulo possui uma interface I2C simples de dois fios para comunicação com um microcontrolador qualquer, por exemplo o ESP8266. O módulo BME280 terá o endereço I2C padrão 0x76 [38].

O módulo BME280 possui 4 pinos (Figura 5.9):

- o pino VIN, a partir do qual se efetua a alimentação do módulo, entre 3,3V e 5V;
- o pino GND, que se conecta ao GND do microcontrolador;
- o pino SCL, pino de relógio da interface I2C;
- o pino SDA, pino de dados da interface I2C.

Figura 5.9: Módulo BME280

5.3.4 MMA7455 [39]

Para adquirir os valores das acelerações, utiliza-se um acelerómetro. Um acelerómetro é um sensor que deteta as forças que a aceleração e a gravidade exercem sobre uma pequena massa existente no seu interior.

O acelerómetro escolhido foi o MMA7455. Este é capaz de medir a aceleração ao longo dos eixos X, Y e Z.

O módulo MMA7455 tem vários recursos integrados, entre eles um conversor analógico-digital, filtro passa baixo e três faixas de sensibilidade (±2g, ±4g ou ±8g). O acelómetro pode ser configurado para detetar movimentos rápidos ou situações de queda livre em qualquer dos eixos.

O módulo é caracterizado por uma elevada flexibilidade e compatibilidade. Possui um regulador de tensão e comutadores de nível de tensão dos I/O, o que permite ligá-lo facilmente a qualquer microcontrolador, por exemplo o ESP8266. A sua tensão de alimentação varia entre 2,5V e 5,5V e estabelece comunicações através dos protocolos SPI e I2C [40].

As suas características principais são:

- funciona com dispositivos de 3,3V e 5V;
- interfaces I2C e SPI para comunicar com qualquer microcontrolador;
- baixo consumo de energia;
- compacto e fácil de usar.

Figura 5.10: Módulo MMA7455

Como se pode ver na Figura 5.10, o módulo MMA7455 possui 8 pinos [39]:

- o pino VCC, a partir do qual se efetua a alimentação do módulo, entre 3,3V e 5V;
- o pino GND, que se conecta ao GND do microcontrolador;
- o pino SCL, pino de relógio das interfaces I2C e SPI;
- o pino SDA, pino de troca de dados da interface I2C e de entrada de dados na interface SPI;
- o pino SDO, pino de saída de dados na interface SPI;
- o pino CS, conecta-se para a interface I2C e desconecta-se para a interface SPI;
- os pinos IN1 e IN2 são utilizados para efetuar interrupções.

5.3.5 Fotorresistência (LDR – *Light Dependent Resistor*)

Uma fotorresistência é uma resistência de valor variável, isto é, a sua resistência depende da intensidade de luz que incide sobre ela. O material que constitui estas resistências é semicondutor e tem uma elevada resistência térmica. Assim, quando a luz incide sobre a fotorresistência, são libertados eletrões que melhoram a sua condutividade e diminuem a resistência.

Deste modo, se com a incidência de luz a resistência diminui, a queda de tensão nos terminais da resistência é menor. Ou seja, quanto maior a intensidade de luz que incide na fotorresistência, menor a diferença de potencial nos terminais da resistência.

Posto isto, o valor da luminosidade é calculado a partir da tensão que chega ao microcontrolador.

Figura 5.11: Fotorresistência (LDR – *Light Dependent Resistor*)

5.4 Aquisição, processamento e monitorização de valores

Como referido no início deste Capítulo, uma das partes das implementações apresentadas relaciona-se com a aquisição, processamento e monitorização de valores. Desde a sua obtenção até à sua visualização, passando pelo seu armazenamento, os valores passam por diversos locais. O caminho que os valores percorrem será explicado detalhadamente nas próximas Secções.

No *Use Case 1*, os valores são adquiridos pelos sensores e enviados para o ESP8266, através das comunicações *master-slave*. De seguida, são enviados para a *Bosch IoT Things*, através do *Bosch IoT Hub*. Por fim, a Aplicação (Servidor) efetua a sua extração, realizando de seguida o seu armazenamento e permitindo a sua visualização graficamente. Estas interações estão ilustradas na Figura 5.12.

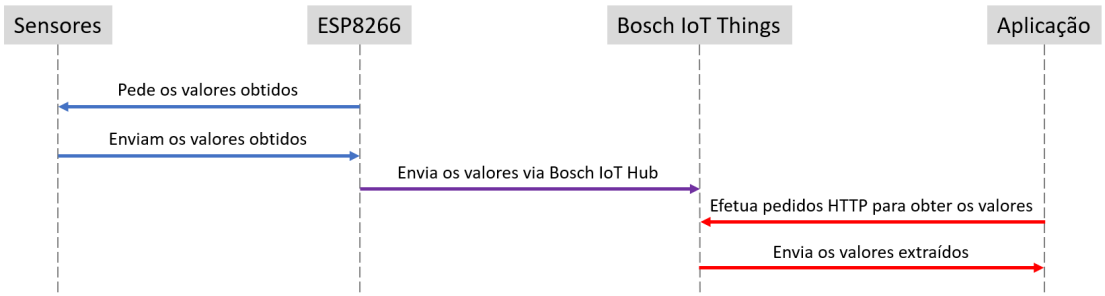

Figura 5.12: Diagrama de interações para aquisição, processamento e monitorização dos valores no *Use Case 1*

Relativamente ao *Use Case 2*, os valores são também adquiridos pelos sensores e enviados para o ESP8266, através das comunicações *master-slave*. De seguida, são enviados para o Raspeberry Pi com recurso ao protocolo de comunicação MQTT. Por sua vez, o Raspberry Pi envia os valores para a *Bosch IoT Things* através do *Bosch IoT Hub*. Por fim, e à semelhança do que acontece no *Use Case* anterior, a Aplicação (Servidor) efetua a sua extração, realizando posteriormente o seu armazenamento e permitindo a sua visualização graficamente. Estas interações estão ilustradas na Figura 5.13.

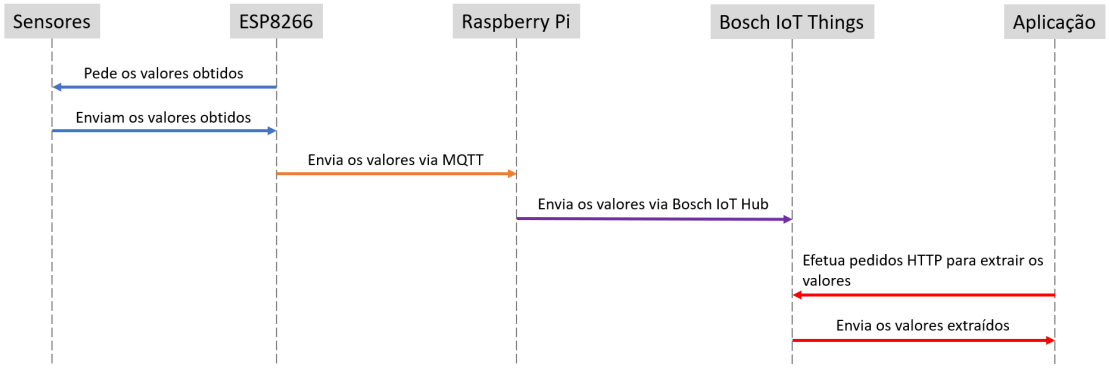

Figura 5.13: Diagrama de interações para aquisição, processamento e monitorização dos valores no *Use Case 2*

5.4.1 Aquisição e processamento no ESP8266

A primeira etapa da monitorização dos valores é a sua obtenção. Para obter os valores da temperatura e humidade utiliza-se o sensor BME280, utiliza-se o sensor MMA7455 para obter o valor das acelerações ao longo dos eixos X, Y e Z e utiliza-se uma fotorresistência para obter o valor da luminosidade.

Uma vez que os sensores BME280 e MMA7455 possuem interface I2C, a comunicação com o ESP8266 é estabelecida com recurso a esse protocolo de comunicação. É uma comunicação *master - slave*, onde o ESP8266 é o *master* e os sensores os *slaves*. Deste modo, o ESP8266 envia para os sensores um pedido para que estes lhe enviem os valores que recolheram.

Para que a comunicação seja estabelecida, efetuam-se as seguintes ligações:

- o pino VIN ou VCC do sensor liga-se a uma saída de 3,3V do ESP8266;
- o pino GND do sensor liga-se a um pino GND do ESP8266;
- o pino SCL do sensor liga-se ao pino D1 do ESP8266 (pino de relógio I2C);
- o pino SDA do sensor liga-se ao pino D2 do ESP8266 (pino de dados I2C).

Nas Figuras 5.14 e 5.15 é possível visualizar as ligações a efetuar entre o ESP8266 e o BME280 e entre o ESP8266 e o MMA7455, respetivamente.

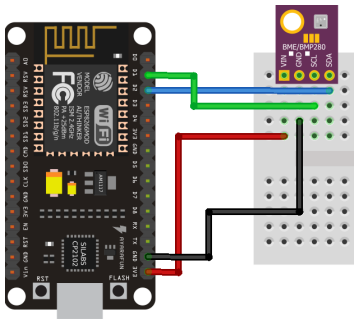

Figura 5.14: Esquema das ligações do ESP8266 com o BME280

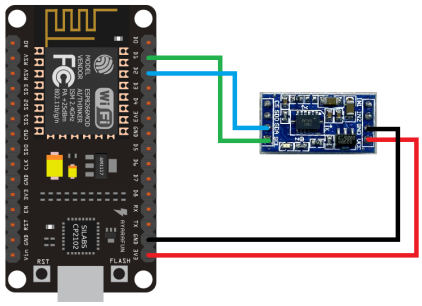

Figura 5.15: Esquema das ligações do ESP8266 com o MMA7455

Relativamente ao valor da luminosidade, este é obtido através da leitura da tensão que chega à entrada analógica do ESP8266 (pino ADC ou A0). Este pino possui uma resolução de 10 bits, o que significa que os valores obtidos variam entre 0 e 1023.

Na Figura 5.16 é possível visualizar as ligações a efetuar entre a fotorresistência e o ESP8266. A resistência a usar é de 10kΩ.

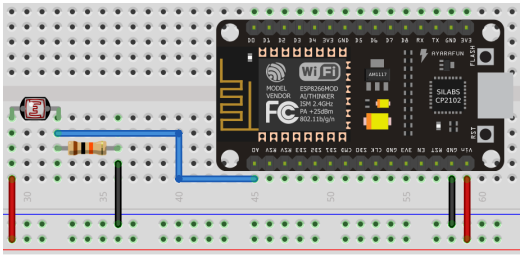

Figura 5.16: Esquema das ligações do ESP8266 com a fotorresistência

A segunda etapa é tratar os valores enviados para o ESP8266. O tratamento passa por converter os valores do formato em que chegam para um formato mais usual e com o qual existe maior sensibilidade. No caso da luminosidade, esse tratamento é a conversão do valor lido para um valor em percentagem. Por exemplo, se o valor lido da entrada analógica for 1023, corresponde a uma luminosidade de 100%. Posto isto, a conversão da tensão para luminosidade é feita através da divisão do valor lido por 1023, multiplicando-se depois por 100 para que o valor seja em percentagem.

Por fim, no caso do *Use Case 1* os valores são já enviados para a *Bosch IoT Things*. No caso do *Use Case 2* são ainda enviados para o Raspberry Pi e para a Aplicação.

5.4.2 Comunicações MQTT

Como referido, no *Use Case 2*, os valores são enviados para o Raspberry Pi e para a Aplicação. As comunicações ESP8266 – Raspberry Pi (*script Python* Monitorização) e ESP8266 – Aplicação efetuam-se através do protocolo de comunicação MQTT. Os três conceitos mais importantes desde protocolo são o Broker MQTT, os clientes MQTT e os tópicos.

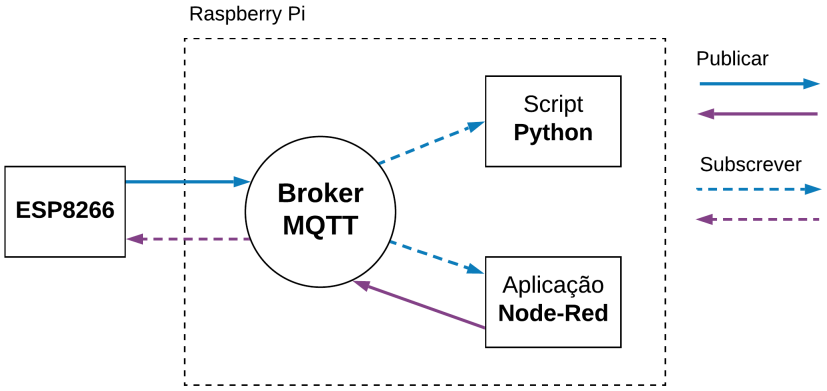

Figura 5.17: Esquema das comunicações MQTT estabelecidas

Como está ilustrado na Figura 5.17, nas comunicações MQTT que se estabelecem, o Broker MQTT está ativo no Raspberry Pi e os clientes são o ESP8266, o *script Python* e a Aplicação (os dois últimos em execução no Raspberry Pi).

Os tópicos para os quais o ESP8266 publica são subscritos pelo *script Python* e pela Aplicação. Por outro lado, a Aplicação publica para tópicos que apenas são subscritos pelo ESP8266.

Os valores de temperatura, humidade, luminosidade e das acelerações nos três eixos (X, Y e Z) são enviados pelo ESP8266 através da sua publicação nos tópicos *temp*, *humid*, *lumin*, *xAcc*, *yAcc* e *zAcc*, respetivamente. O *script Python* e a Aplicação recebem estes valores através da subscrição dos tópicos acima mencionados. O ESP8266 publica nos tópicos de cinco em cinco minutos.

Por sua vez, a Aplicação envia mensagens para o ESP8266 através dos tópicos *resend_ temp*, *resend_ humid*, *resend_ lumin* e *resend_ accelerations*.

5.4.3 Processamento no Raspberry Pi (*Gateway*)

No *Use Case 2*, uma vez que o dispositivo não tem capacidade para estabelecer a comunicação com a *Bosch IoT Suite*, é necessário um componente adicional que execute essa função. Tal como descrito no inicio do Capítulo e ilustrado na Figura 5.2, esse componente é o Raspberry Pi.

Uma vez que o Raspberry Pi é um dispositivo com sistema operativo, é possível efetuar a instalação de diversos programas e ferramentas. Assim, dentro do Raspberry Pi existem pelo menos três processos em execução.

Como já foi referido na Secção 5.4.2, o Boker MQTT está ativo no Raspberry Pi. Assim, um desses processos é o Broker MQTT, o qual permite que as ligações MQTT sejam estabelecidas. Se o Broker MQTT não estiver ativo, não é possível estabelecer estas comunicações. Outro processo é o scrip Python responsável por receber os valores vindos do ESP8266 e depois proceder ao seu envio para a *Bosch IoT Things*. Por fim, está também em execução o *Node-RED*. O *Node-RED* permite desenvolver a Aplicação e que esta esteja em funcionamento.

A Aplicação desenvolvida permite obter os valores que estão a ser enviados pelo ESP8266, obter dados sobre as comunicações estabelecidas entre o ESP8266 e os sensores e obter dados sobre o funcionamento do dispositivo e dos sensores.

Interface Gráfica

Como se pode ver na Figura 5.2, associada à Aplicação existe uma Interface Gráfica. Nela é possível observar os valores que chegam do ESP8266, a data (entenda-se data e hora) das comunicações entre a Aplicação e o ESP8266, nomeadamente a data da receção do último valor enviado pelo ESP8266 - *Reception Date* - e a data da última mensagem enviada pela Aplicação - *Resend Date*. Devido à rapidez das comunicações e do processamento, estas datas serão muito próximas das datas das comunicações do ESP8266 com os sensores. É também possível visualizar o estado de funcionamento de cada sensor e efetuar algumas ações clicando nos botões existentes. Na Figura 5.18 apresenta-se a Interface Gráfica.

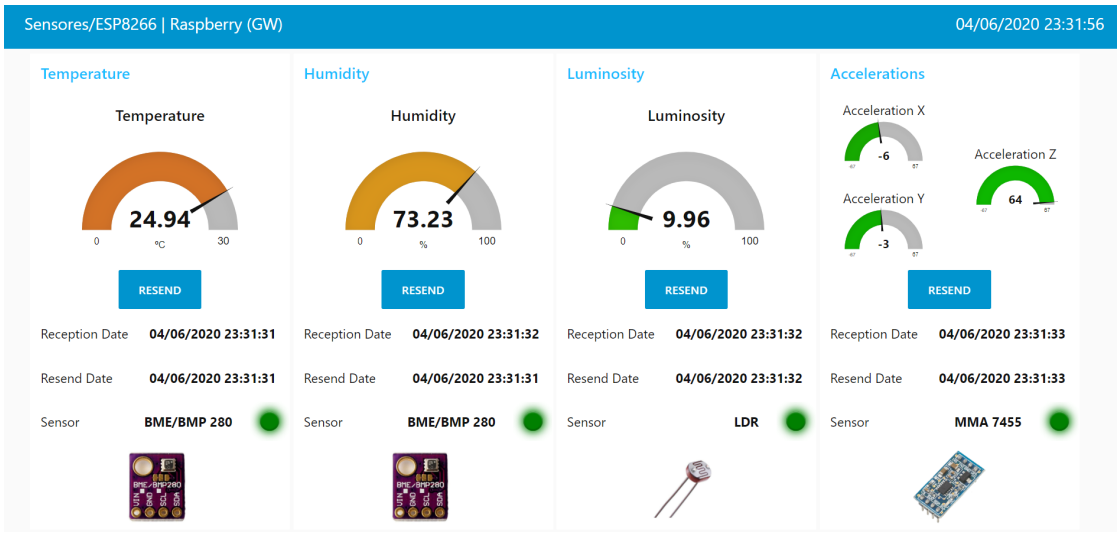

Figura 5.18: Interface Gráfica associada à Aplicação em execução no Raspberry Pi

Uma vez que o ESP8266 publica mensagens para os tópicos de cinco em cinco minutos, a data de receção do último valor também se deve alterar com essa cadência. Uma forma de perceber se tudo está a funcionar corretamente, é controlar precisamente essa data. No entanto, uma vez que seria um processo demorado, encontrou-se um método mais eficiente para fazer essa verificação. Em vez de aguardar que cheguem novos valores, força-se o ESP8266 a enviá-los num instante qualquer. Desta forma, clicando nos botões correspondentes, a Aplicação publica nos tópicos *resend_ temp*, *resend_ humid*, *resend_ lumin* ou *resend_ accelerations*, enviando deste modo mensagens para o ESP8266.

Estes tópicos são subscritos pelo ESP8266, deste modo, quando num deles chega uma mensagem, o ESP8266 obtém o novo valor e envia-o através da sua publicação no tópico correspondente. Por exemplo, se no lado da Interface Gráfica se clicar no botão *RESEND* na coluna *Temperature*, a Aplicação envia uma mensagem através do tópico *resend_temp*, o ESP8266 recebe a mensagem, adquire o novo valor para a temperatura e envia-o através da sua publicação no tópico *temp*.

Na Figura 5.19 apresenta-se o diagrama de interações para a Aplicação.

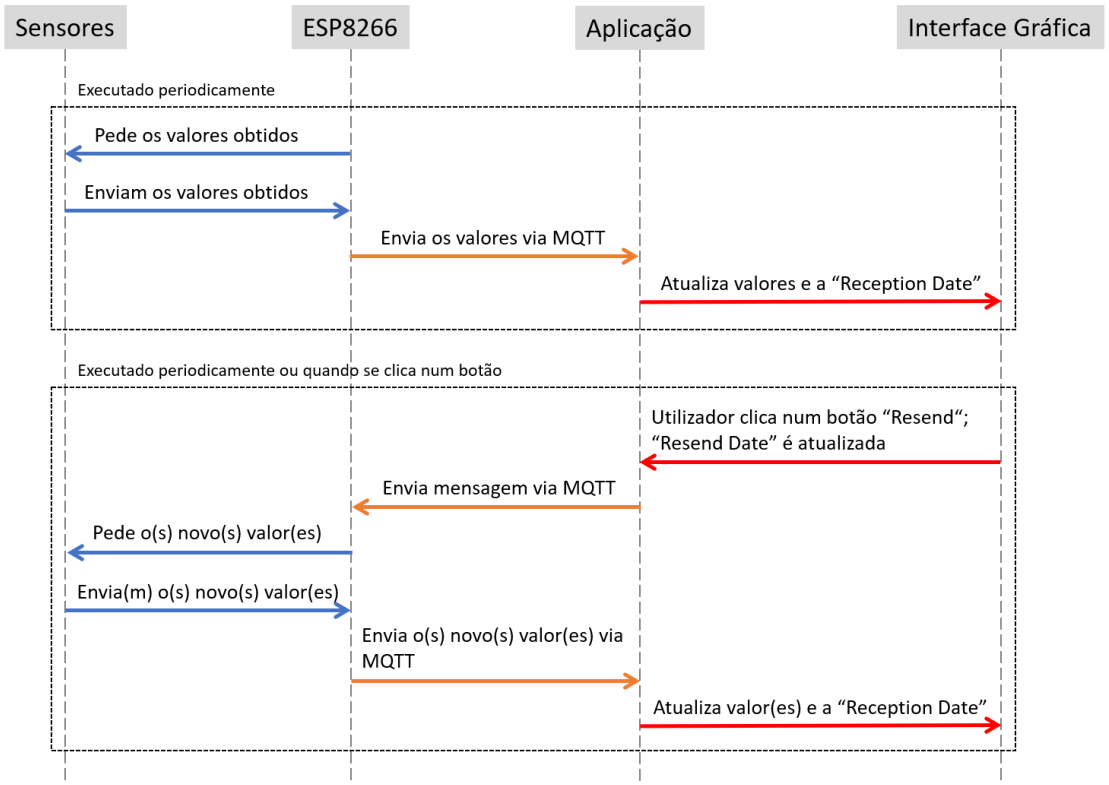

Figura 5.19: Diagrama de interações para a Aplicação e Interface Gráfica

De seguida a Aplicação analisa a data e hora em que enviou a mensagem para o ESP8266 e analisa também a data e hora da chegada do último valor enviado pelo ESP8266. Geralmente, caso a diferença de tempo entre o clicar no botão a pedir o novo valor e a recessão desse valor seja superior a dez segundos, significa que o ESP8266 ou algum sensor não estão a funcionar corretamente.

5.4.4 Comunicação com a *Bosch IoT Suite*

Como se pode ver na Figura 5.1 e 5.2, enquanto que no *Use Case 1* a comunicação com a *Bosch IoT Suite* é efetuada pelo ESP8266, no *Use Case 2* essa comunicação é efetuada pelo Raspberry Pi, mais concretamente pelo *script Python* Monitorização, que também estabelece a comunicação com o ESP8266 através do protocolo MQTT. Além do ESP8266 e do Raspberry Pi, também a Aplicação (Servidor) comunica com a *Bosch IoT Suite*.

Para se poder usufruir dos serviços da *Bosch IoT Suite*, nomeadamente do *Bosch IoT*

Hub e da *Bosch IoT Things*, é necessário ter uma conta na *Bosch IoT Suite*, subscrever o plano *Bosch IoT Suite for Asset Communication*, efetuar algumas configurações no plano subscrito e por fim registar o dispositivo nos dois serviços. Todo este procedimento está explicado detalhadamente no Apêndice A.

O serviço *Bosch IoT Hub* contém, entre outros, um *MQTT Adapter*, permitindo que os dispositivos físicos comuniquem com serviços e aplicações remotas de forma fácil, eficaz e segura. Recorrendo a esta funcionalidade do *Bosch IoT Hub*, as comunicações podem ser estabelecidas de uma forma bastante semelhante ao protocolo MQTT comum.

A *Bosch IoT Things* permite que as aplicações efetuem a gestão dos *digital twins* dos dispositivos ativos. Com base no *digital twin* as aplicações podem armazenar e atualizar os dados e propriedades dos dispositivos ativos, e podem ser notificadas de todas as alterações importantes. Os dispositivos físicos podem interagir com o *digital twin* através do *Bosch IoT Hub*. Por outro lado, as aplicações podem interagir com o *digital twin* usando APIs, por exemplo.

Para que os dispositivos comuniquem com a *Bosch IoT Things* através do *Bosch IoT Hub*, o *firmware* ou *software* instalado tem de incluir algumas configurações relacionadas com o plano subscrito e com as credenciais de registo do dispositivo.

As comunicações entre o dispositivo físico, o seu *digital twin* e a aplicação remota, podem ser estabelecidas nos dois sentidos. Na Figura 5.20 apresenta-se uma visão esquemática das comunicações estabelecidas.

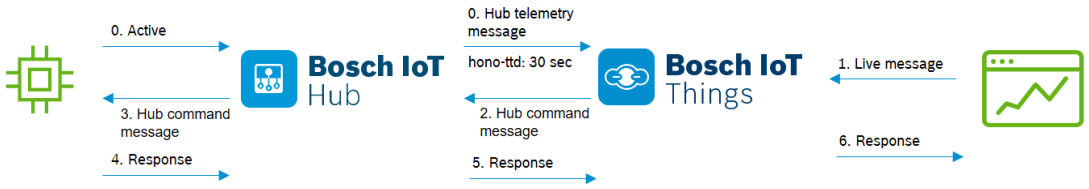

Figura 5.20: Comunicações estabelecidas entre o dispositivo físico, o *digital twin* e uma aplicação remota [41]

Pode-se considerar uma primeira comunicação que apenas envolve o dispositivo físico e o seu *digital twin* na *Bosch IoT Things*. Esta comunicação é explicada pelos seguintes tópicos:

- o dispositivo tem de estar preparado para enviar e receber mensagens através do *Bosch IoT Hub*;
- através do *Bosch IoT Hub* o dispositivo envia para a *Bosch IoT Things* os valores que está a monitorizar;

Outra comunicação existente é a que envolve o dispositivo físico, o seu *digital twin* e a aplicação remota. Esta aplicação é do tipo *command and control* e é explicada da seguinte forma:

- a aplicação remota envia uma mensagem para a *Bosch IoT Things* através de um pedido HTTP à API;
- esta mensagem não afeta o *digital twin* e é reencaminhada para o *Bosch IoT Hub*, ficando a *Bosch IoT Things* à espera de uma resposta;

- por sua vez, o *Bosch IoT Hub* envia a mensagem para o dispositivo, ficando também a aguardar uma resposta;
- o dispositivo processa a mensagem, executa uma operação e envia uma resposta para o *Bosch IoT Hub*;
- o *Bosch IoT Hub* reencaminha a resposta para a *Bosch IoT Things*;
- por fim, a *Bosch IoT Things* envia a resposta para a aplicação remota.

As comunicações do dispositivo com a *Bosch IoT Things* efetuam-se através de três tópicos específicos do *Bosch IoT Hub*: um tópico para o dispositivo enviar os dados que está a monitorizar, outro tópico a partir do qual a *Bosch IoT Things* envia os comandos e, por fim, outro tópico através do qual o dispositivo envia a resposta aos comandos.

Recorrendo ao *Bosch IoT Hub*, tanto o ESP8266 no *Use Case 1*, como o Raspberry Pi (*script Python* Monitorização) no *Use Case 2*, podem estabelecer a ligação com a *Bosch IoT Things*. Através da publicação de mensagens para o tópico destinado ao envio de dados, os dispositivos referidos anteriormente conseguem enviar para a *Bosch IoT Things* os valores de temperatura, humidade, luminosidade e acelerações que estão a ser monitorizados. Estes dados passam a ficar armazenados no *digital twin* associado a cada dispositivo físico.

Por outro lado, a Aplicação (Servidor) - equivalente à aplicação remota referida na explicação das comunicações - estabelece a comunicação com a *Bosch IoT Things* através de pedidos HTTP à *Bosch IoT Things - API v2*. Através desses pedidos, a Aplicação (Servidor) consegue extrair dos *digital twins* os valores que eles armazenam e também enviar-lhe mensagens com determinados comandos, os quais vão sendo reencaminhados até ao dispositivo físico.

5.4.5 Registo de um dispositivo através da Aplicação (Servidor)

O plano *Bosch IoT Suite for Asset Communication* contém os serviços *Bosch IoT Things* e *Bosch IoT Hub*. No entanto, para que um dispositivo possa usufruir dos serviços referidos, tem de estar registado na *Bosch IoT Suite*.

O atual plano *Bosch IoT Suite for Asset Communication* subscrito, destina-se a dispositivos que estejam a monitorizar a temperatura, a humidade, a luminosidade e as acelerações ao longo dos três eixos (X, Y e Z). Tanto o ESP8266 no *Use Case 1* como o Raspberry Pi no *Use Case 2* foram registados na *Bosch IoT Suite*, ficando associados a este plano.

Para efetuar o registo de um dispositivo existem duas possibilidades: através da API *Bosch IoT Suite – Device Provisioning* ou através do *Vorto Repository* e do *Postman*. Estas duas possibilidades são explicadas detalhadamente no Apêndice A.

Para efetuar o registo de um dispositivo são necessários os seguintes dados:

- *Test Token* para aceder à API;
- *Service Instance ID* do plano;
- *namespace* criado durante a subscrição do plano;

- *id*, no formato <namespace:dispositivo_id>
- *password*.
- *JSON* com o corpo do pedido à API (o *id* e a *password* integram este *JSON*).

O processo de registo de um dispositivo através das duas possibilidades anteriores é complexo e demorado. Obter o *Test Token*, o *Service Instance ID* e o *namespace* e criar o *JSON* manualmente leva o seu tempo. Contudo, através do *Vorto Repository* e do *Postman* a criação do *JSON* já se torna mais rápida, ainda assim com um grande número de passos a realizar, e a necessidade de obter o *Test Token*, o *Service Instance ID* e o *namespace* continua presente.

No entanto, quando se registam dispositivos no mesmo plano, as diferenças entre eles apenas residem no *Test Token*, se este tiver expirado entretanto, e no *JSON*, uma vez que o *id* e a *password* variam. Visto que o *Service Instance ID* e o *namespace* são dados relacionados com o plano, o seu valor é constante.

Dadas estas pequenas diferenças, desenvolveu-se uma forma mais simples e rápida de registar um novo dispositivo. Através desta nova forma, o *JSON* que de alguma maneira terá de ser criado para registar o primeiro dispositivo no plano, será guardado e reutilizado para o registo de novos dispositivos.

Posto isto, através da Aplicação (Servidor) implementou-se uma nova maneira de registar um dispositivo no plano subscrito. Como referido no início do Capítulo, esta Aplicação (Servidor) foi desenvolvida através do *Node-RED*.

Deste modo, visto que o valor do *Service Instance ID* não varia, passa a ser uma constante do desenvolvimento da Aplicação (Servidor). Através das funcionalidades do *Node-RED*, é possível editar o ficheiro *JSON* e alterar os campos que correspondem às credenciais *id* e *password*. Resta agora obter o *Test Token* para que a Aplicação (Servidor) consiga aceder à API *Bosch IoT Suite – Device Provisioning* e efetue o pedido HTTP para registar o dispositivo.

O *Test Token* é também obtido através de um pedido HTTP realizado pela Aplicação (Servidor). O pedido HTTP é realizado periodicamente antes do *Test Token* expirar, e tem a seguinte estrutura:

```
POST /token HTTP/1.1
Host:  https://access.bosch-iot-suite.com
Accept:  application/json
Content-Type:  application/x-www-form-urlencoded
```

Payload:

```
"grant_type":  "client_credentials"
"client_id":  "e4e96af4-e856-470b-b71d-a393ac20daff"
"client_secret":  "XOkr4fxõe0c"
"service:iot-hub-prod":  "t5f7ed5ae8357493c914ff61b8287a3e9_hub/full-ac
cess service:iot-things-eu-1:5f7ed5ae-8357-493c-914f-f61b8287a3e9_things/f
ull-access"
```

O *Test Token* está contido na resposta a este pedido. Com a sua obtenção, passa a ser uma variável da Aplicação (Servidor), uma vez que também é usado nos outros pedidos

HTTP que a Aplicação (Servidor) efetua. Esses pedidos serão descritos nas próximas Secções.

Assim, para criar um dispositivo dentro do mesmo plano e que neste caso se destina à monitorização dos valores da temperatura, da humidade, da luminosidade e das acelerações ao longo dos três eixos, apenas é necessário inserir na Aplicação (Servidor) três parâmetros:

- o caminho completo com a localização do ficheiro *JSON*;
- o *id* que o dispositivo terá, no formato <namespace:dispositivo_id>;
- a *password* para o dispositivo aceder ao *Bosch IoT Hub*.

O pedido que a Aplicação (Servidor) efetua à API *Bosch IoT Suite – Device Provisioning* para registar o dispositivo, tem a seguinte estrutura:

```
POST /api/1/<Service Instance ID>/devices HTTP/1.1
Host:  https://deviceprovisioning.eu-1.bosch-iot-suite.com
accept:  application/json
Authorization:  Bearer <Test Token>
Content-Type:  application/json
```

Payload:

```
<Conteúdo do ficheiro JSON>
```

Em caso de sucesso, a resposta contém o código *HTTP 201 Created*, e o seu conteúdo é referente ao dispositivo que se acaba de registar.

As interações estabelecidas entre a Aplicação (Servidor) e a API da *Bosch IoT Suite* para efetuar o registo de um dispositivo, estão representadas na Figura 5.21.

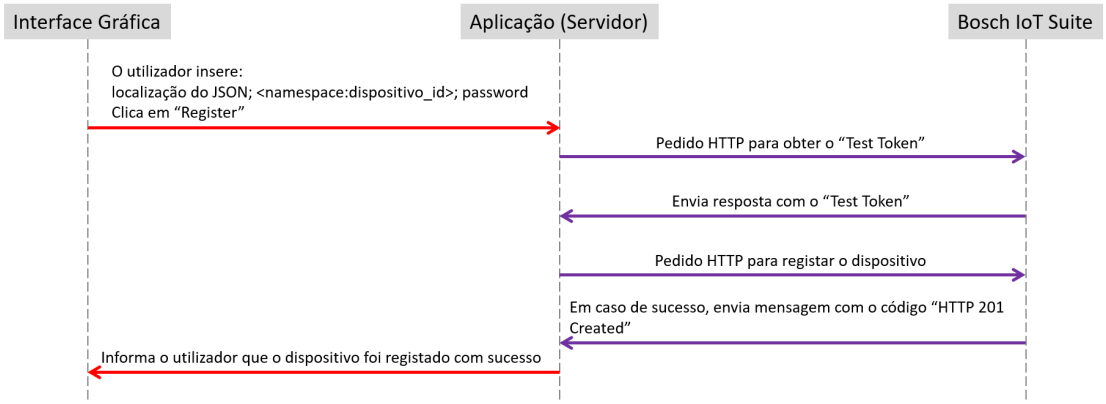

Figura 5.21: Diagrama de interações para registar um dispositivo através da Aplicação (Servidor)

Menu da Interface Gráfica

O menu que se apresenta de seguida, juntamente com as funcionalidades da Aplicação (Servidor), permite registar um dispositivo no plano *Bosch IoT Suite for Asset Communication* subscrito. O menu pode ser visualizado na Figura 5.22.

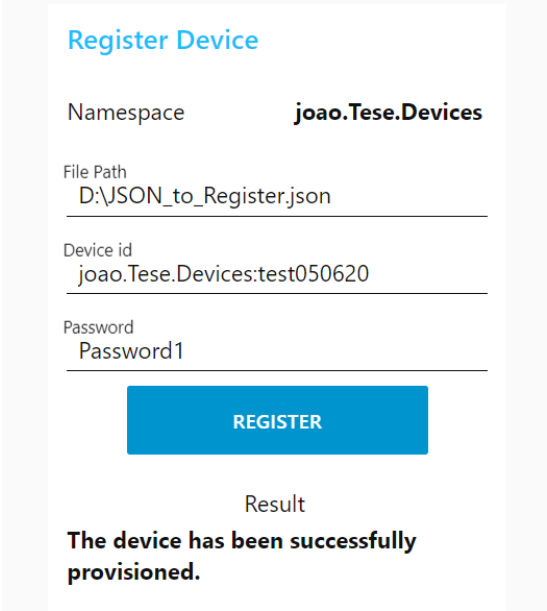

Figura 5.22: Menu para efetuar o registo de um dispositivo no plano subscrito

Assim, com a utilização deste menu apenas existe a necessidade de inserir os três parâmetros acima mencionados. Depois de inseridos, clica-se no botão *Register* para efetuar o pedido HTTP e, consequentemente, o registo do dispositivo. O utilizador é ainda informado sobre o sucesso do registo do dispositivo.

Estas interações também se encontram ilustradas na Figura 5.21.

5.4.6 Processamento na Aplicação (Servidor)

Um componente comum aos dois *Use Cases* e que tem uma importância muito significativa para que seja possível efetuar a gestão remota dos dispositivos, é a Aplicação (Servidor). A Aplicação (Servidor) foi desenvolvida com o *Node-RED*, e tem uma parte relativa ao processamento e monitorização de valores e outra parte relativa à gestão do *firmware* ou *software* dos dispositivos. A parte relativa ao processamento e monitorização de valores será abordada de seguida, enquanto que a parte relativa à gestão do *firmware* ou *software* será abordada mais à frente na Secção 5.5.

Relativamente à monitorização, o primeiro passo é obter os valores que foram previamente enviados para a *Bosch IoT Things*.

Só quando estão em funcionamento é que o ESP8266 no *Use Case 1* e o Raspberry Pi (*script Python* Monitorização) no *Use Case 2* enviam valores novos, caso contrário, deixam de enviar valores e o último valor enviado é o que fica guardado na *Bosch IoT Things*.

Uma vez que a Aplicação (Servidor) extrai os valores da *Bosch IoT Things*, se os dispositivos não enviarem valores novos, os valores extraídos serão repetidos. Para evitar que isto aconteça, antes de se efetuar o pedido HTTP à API a pedir esses valores, efetua-se um outro pedido, o qual permite enviar uma mensagem (comando) para o dispositivo e eventualmente receber uma resposta.

Periodicamente, a Aplicação (Servidor) efetua ambos os pedidos HTTP à *Bosch IoT Things – API v2*. Também para aceder a esta API é necessário uma autenticação, a qual é efetuada através do *Test Token*. Como já referido na Secção 5.4.5, este *Test Token* é obtido periodicamente pela Aplicação (Servidor) antes de expirar, podendo ser usado sempre que necessário.

Começando pelo pedido HTTP que envia a mensagem para o dispositivo, tem a seguinte estrutura:

```
POST /api/2/things/<namespace:dispositivo_id>/inbox/messages/<subject>
HTTP/1.1
Host: https://things.eu-1.bosch-iot-suite.com
Authorization: Bearer <Test Token>
Content-Type: application/json
```

Payload:

```
{}
```

Através do pedido HTTP descrito acima, a Aplicação (Servidor) envia uma mensagem para o *digital twin* na *Bosch IoT Things*. De seguida, tal como referido na Secção 5.4.4, esta mensagem é enviada para o dispositivo por intermédio do *Bosch IoT Hub*, utilizando o tópico destinado a esse fim. Por sua vez, o dispositivo tem de subscrever esse tópico, para conseguir receber a mensagem, efetuar um determinado procedimento e enviar uma resposta. Assim, quando a mensagem chega ao dispositivo, este executa uma determinada ação e envia uma resposta para a *Bosch IoT Things*. A resposta é enviada através do *Bosch IoT Hub*, utilizando o tópico destinado ao envio das respostas. Por fim, essa resposta é enviada pela *Bosch IoT Things* para a Aplicação (Servidor). Posto isto, quando está em funcionamento, o dispositivo recebe a mensagem (comando) enviado pela Aplicação (Servidor) e envia uma resposta. Caso esteja desligado, não recebe a mensagem enviada, pelo que também não envia uma resposta.

Relativamente ao pedido HTTP para obter os valores enviados previamente pelos dispositivos para a *Bosch IoT Things*, tem a seguinte estrutura:

```
GET /api/2/things/<namespace:dispositivo_id> HTTP/1.1
Host: https://things.eu-1.bosch-iot-suite.com
accept: application/json
Authorization: Bearer <Test Token>
```

A resposta obtida é em formato *JSON*, no entanto, através das funcionalidades do *Node-RED*, é possível descodificar a resposta e extrair os valores que estão a ser monitorizados.

Na Figura 5.23 apresenta-se as interações descritas anteriormente, realizadas entre a Aplicação (Servidor) e a *Bosch IoT Things*, as quais permitem extrair os valores armazenados no *digital twin*.

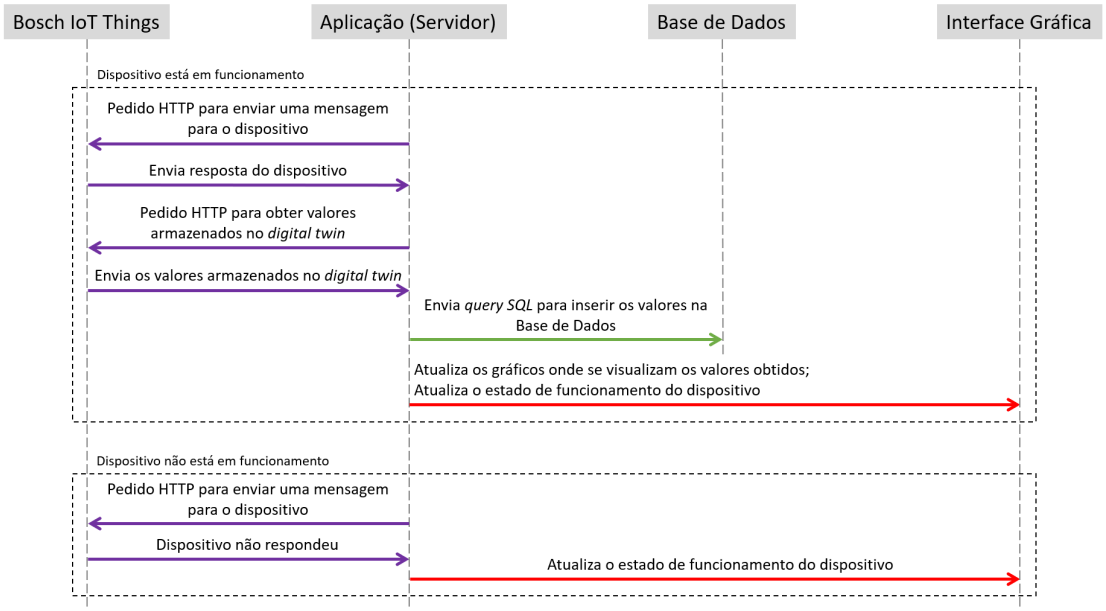

Figura 5.23: Diagrama das interações efetuadas durante a comunicação Aplicação (Servidor) - *Bosch IoT Things* para aquisição dos valores

Depois de extraídos os valores da *Bosch IoT Things*, estes são enviados para uma Base de Dados. A Base de Dados é desenvolvida com recurso ao SQL Express, chama-se *bd_node-red* e contém a tabela *dbo.monitoring*. A tabela *dbo.monitoring* pode ser visualizada na Figura 5.24.

	ID	Device_ID	Temperature	Humidity	Luminosity	Acceleration_X	Acceleration_Y	Acceleration_Z	DateTime
1	129	ESP8266_as_Gateway	23.97	67.17	50.49	3	6	64	2020-06-13 01:37:32.577
2	128	Raspberry_as_Gateway	23.96	67.21	3.027	0	-1	64	2020-06-13 01:32:37.560
3	127	ESP8266_as_Gateway	23.97	67.22	50.49	5	-6	67	2020-06-13 01:32:32.557
4	126	Raspberry_as_Gateway	23.99	67.12	3.027	-1	-1	64	2020-06-13 01:27:37.470
5	125	ESP8266_as_Gateway	23.99	67.15	50.39	-6	-1	67	2020-06-13 01:27:32.680
6	124	Raspberry_as_Gateway	23.98	67.2	3.027	1	0	66	2020-06-13 01:22:37.643
7	123	ESP8266_as_Gateway	23.95	67.2	50.39	-3	4	67	2020-06-13 01:22:32.560
8	122	Raspberry_as_Gateway	23.98	67.15	3.027	1	0	65	2020-06-13 01:17:37.567
9	121	ESP8266_as_Gateway	24.01	67.11	50.59	-3	-4	65	2020-06-13 01:17:32.550
10	120	Raspberry_as_Gateway	23.98	67.12	3.027	1	1	64	2020-06-13 01:12:37.460
11	119	ESP8266_as_Gateway	23.99	67.09	50.39	1	6	63	2020-06-13 01:12:32.570
12	118	Raspberry_as_Gateway	24.01	67.04	2.929	4	6	66	2020-06-13 01:07:37.490
13	117	ESP8266_as_Gateway	23.98	67.1	50.39	-2	-3	63	2020-06-13 01:07:32.537
14	116	Raspberry_as_Gateway	24.01	67.01	2.929	0	1	64	2020-06-13 01:02:37.480
15	115	ESP8266_as_Gateway	24.02	66.9	50.49	3	-5	64	2020-06-13 01:02:32.640

Figura 5.24: Tabela *dbo.monitoring*

Esta tabela é constituída por nove colunas:

- coluna *ID* - incrementada cada vez que é inserido um novo valor;
- coluna *Device_ID* - corresponde ao dispositivo que obteve os valores armazenados na respetiva linha;
- colunas *Temperature* a *Acceleration_Z* - correspondem, respetivamente, aos valores

de temperatura, humidade, luminosidade, aceleração no eixo X, aceleração no eixo Y e aceleração no eixo Z obtidos pelos dispositivos;

- coluna *DateTime* - representa a data e hora de armazenamento dos valores.

Para realizar o armazenamento dos valores obtidos a partir da *Bosch IoT Things*, a Aplicação (Servidor) envia para a Base de Dados uma *query SQL*, como se pode verificar no diagrama apresentado na Figura 5.23. Essa *query SQL* permite armazenar os valores na tabela *dbo.monitoring* e tem a seguinte estrutura:

```
INSERT INTO [dbo].[monitoring] (Device_ID, Temperature, Humidity, Luminosity, Acceleration_X, Acceleration_Y, Acceleration_Z) VALUES (device_id, temp, humid, lumin, xAcc, yAcc, zAcc);
```

Menu da Interface Gráfica

De seguida apresenta-se o menu *Monitoring*, um dos menus da Interface Gráfica. Em conjunto com as funcionalidades da Aplicação (Servidor) descritas anteriormente, este menu permite visualizar graficamente os valores que estão a ser monitorizados pelos diferentes dispositivos e conferir o estado de funcionamento de cada dispositivo. O menu *Monitoring* é apresentado na Figura 5.25.

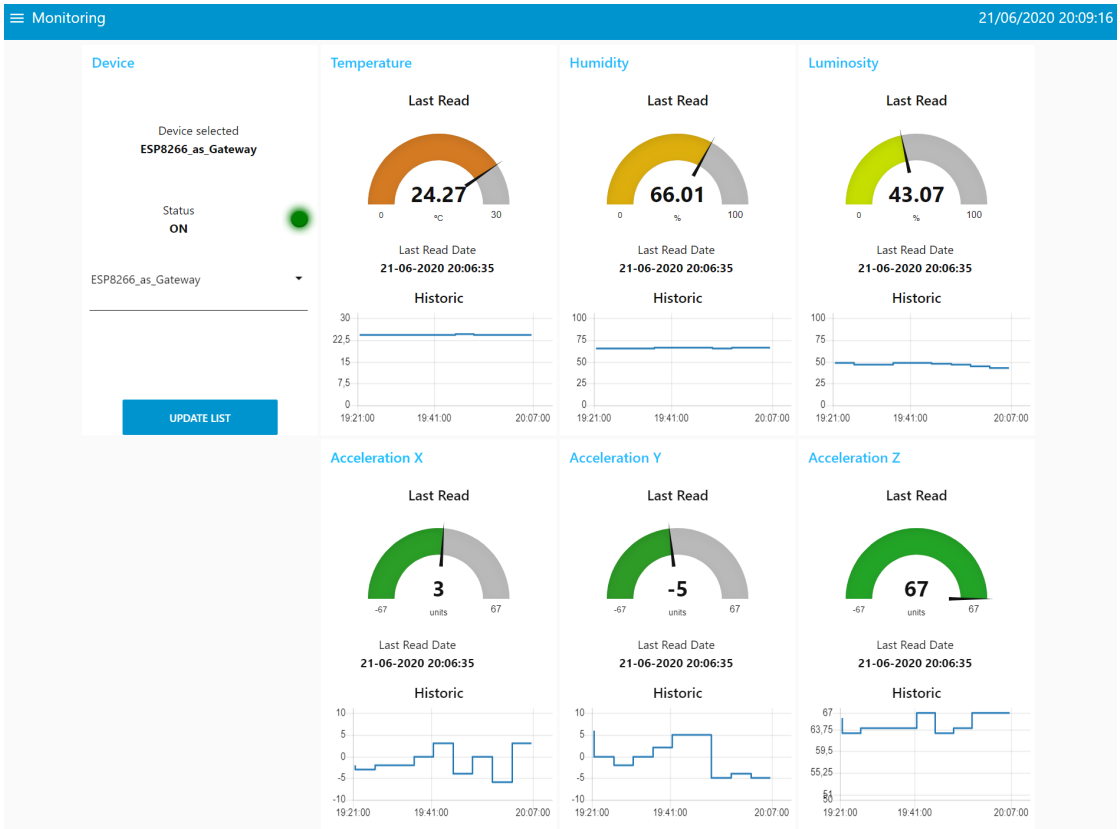

Figura 5.25: Menu para visualizar graficamente os valores monitorizados

Como se pode perceber pela Figura 5.25, o menu é composto por quatro colunas.

Na primeira coluna é escolhido o dispositivo para visualizar os valores que está a monitorizar e obter o seu estado de funcionamento. Uma vez que esses valores são selecionados a partir da tabela *dbo.monitoring*, através desta escolha define-se o *device_id* que está presente na *query SQL* para selecionar os últimos dez registos de um determinado dispositivo.

A *query SQL* para selecionar os valores da tabela é a seguinte:

```
SELECT TOP 10 * FROM [dbo].[monitoring] WHERE Device_ID = device_id ORD
ER BY ID DESC;
```

É ainda possível visualizar o botão *Update List*, o qual serve para atualizar a lista de dispositivos para selecionar. Através das funcionalidades do *Node-RED*, esta lista é preenchida automaticamente com os *device_id* de todos os dispositivos registados. Esses *device_id* são obtidos através do seguinte pedido HTTP:

```
GET /api/2/search/things HTTP/1.1
Host: https://things.eu-1.bosch-iot-suite.com
accept: application/json
Authorization: Bearer <Test Token>
```

A segunda, terceira e quarta coluna são divididas em dois grupos. Deste modo, existem seis grupos onde cada grupo corresponde a cada valor que está a ser monitorizado, ou seja, temperatura, humidade, luminosidade, acelerações segundo os eixos X, Y e Z. Em cada um desses grupos existe um gráfico com o último valor obtido pelo dispositivo, em cima, e um histórico com os últimos dez valores obtidos, em baixo.

Na Figura 5.26 é possível visualizar as interações que se estabelecem para visualizar os valores que estão a ser monitorizados.

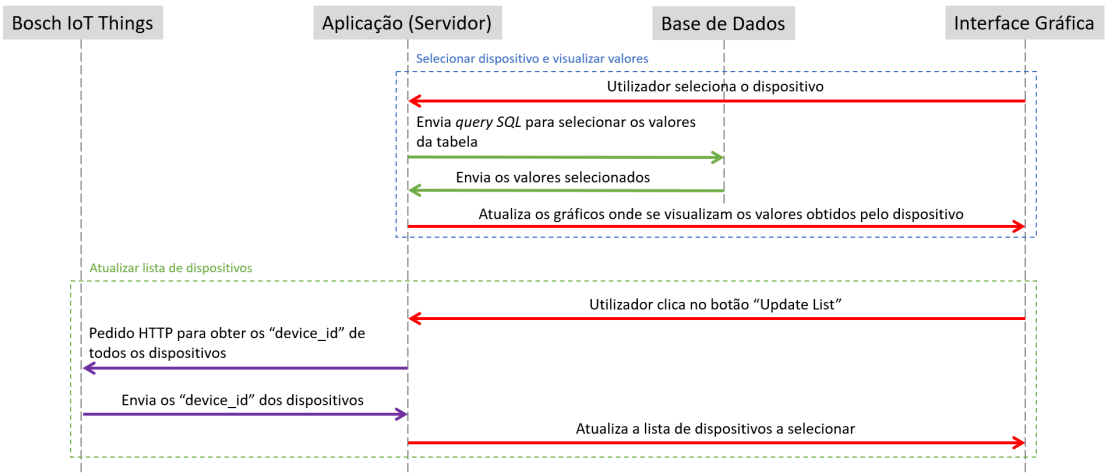

Figura 5.26: Diagrama de interações para efetuar a visualização dos valores obtidos

5.5 Gestão do *Firmware* ou *Software* de cada dispositivo

Outra parte das implementações é a que está relacionada com a gestão do *firmware* ou *software* instalado em cada dispositivo. Para efetuar esta gestão recorre-se a um Serviço da *Bosch IoT Suite*, o *Bosch IoT Rollouts*.

Uma vez que se trata de um serviço da *Bosch IoT Suite*, a sua subscrição é realizada através de uma conta na *Bosch IoT Suite*, que pode ser a mesma que se usa para subscrever o plano *Bosch IoT Suite for Asset Communication*. A criação de uma conta está explicada no Apêndice A.

Este serviço permite registar dispositivos, criar atualizações e efetuar a atualização do *firmware* ou *software* que cada dispositivo tem instalado. Permite também obter informações de uma maneira mais geral, isto é, sobre todos os dispositivos registados e atualizações criadas, ou então de uma maneira mais específica. Na maneira mais específica, permite obter os dados de um dispositivo em particular e os dados sobre a atualização atribuída a esse dispositivo.

Nesta Secção explica-se detalhadamente como é possível efetuar esta gestão através do *Bosch IoT Rollouts* e também a integração deste serviço com a Aplicação (Servidor).

5.5.1 Registar um dispositivo e criar uma atualização no *Bosch IoT Rollouts*

O Bosch IoT Rolouts disponibiliza duas maneiras de registar um dispositivo, criar uma atualização e atribuí-la a um dispositivo. Pode ser através da *Management UI* ou através da *Management API*. A *Management UI* é a interface gráfica que o serviço disponibiliza, enquanto a *Management API* pode ser usada por outras aplicações para efetuarem a integração com o serviço.

O primeiro passo para a utilização do serviço *Bosch IoT Rollouts* é a sua subscrição. Depois de subscrever o serviço, é possível registar dispositivos, criar atualizações e associar essas atualizações aos dispositivo registados. Para isso, os passos a efetuar são os seguintes:

- Registar um dispositivo;
- Criar um *Software Module*;
- Efetuar o *upload* do ficheiro com o novo *firmware* ou *software* e associá-lo ao *Software Module*;
- Criar um *Distribution Set*;
- Associar ao *Distribution Set* o *Software Module*;
- Atribuir o *Distribution Set* ao dispositivo.

5.5.1.1 *Management UI*

A subscrição do serviço *Bosch IoT Rollouts* e a utilização da *Management UI* para efetuar os passos descritos acima, estão explicadas detalhadamente no Apêndice B.

5.5.1.2 Integração da *Management API* com a Aplicação (Servidor)

No sentido de tornar a Aplicação (Servidor) cada vez mais completa, e de modo a concentrar nela grande parte dos recursos necessários à gestão remota de dispositivos, implementou-se uma alternativa à *Management UI* para registar dispositivos e criar atualizações. Esta alternativa consiste na integração da *Management API* com a Aplicação (Servidor), a qual é estabelecida através de pedidos HTTP.

À semelhança do que acontece com as APIs usadas anteriormente, também para aceder a esta API do *Bosch IoT Rollouts*, é necessária uma autorização. Neste caso, a autorização consiste num *Username*, com o formato *tenant\username*, e numa *Password*. Estas três credenciais são criadas automaticamente no momento da subscrição do serviço. Para a sua consulta basta entrar na conta da *Bosch IoT Suite*, aceder ao menu *Service Subscriptions*, procurar o serviço que se pretende usar e clicar em *Show Credentials*.

Através das funcionalidades do *Node-RED*, é possível efetuar a autenticação com as credenciais referidas atrás e efetuar os pedidos HTTP.

Posto isto, de seguida apresentam-se os pedidos HTTP realizados pela Aplicação (Servidor) para efetuar os passos que permitem registar um dispositivo, criar uma atualização e associar uma atualização a um dispositivo.

Registar dispositivo

No registo de um dispositivo existem duas etapas: efetuar o *pre-provisioning* do dispositivo e efetuar o seu registo. A primeira etapa é efetuada através de um pedido HTTP à *Management API* com a seguinte estrutura:

```
POST /rest/v1/targets HTTP/1.1
Host: https://api.eu1.bosch-iot-rollouts.com
Content-Type: application/json;charset=UTF-8
```

Payload:

```
"controllerId": "<device_id>"
"name": "<device_name>"
```

A segunda etapa é também efetuada através de um pedido HTTP a uma API, mas desta vez é a *Direct Device Integration API*, outra API do *Bosch IoT Rollouts*. O pedido tem a seguinte estrutura:

```
GET /<tenant>/controller/v1/<controllerId> HTTP/1.1
Host: https://device.eu1.bosch-iot-rollouts.com
Accept: application/hal+json
Authorization: TargetToken <securityToken>
```

Realizadas estas duas etapas, o dispositivo está registado.

Criar *Software Module*

Para criar uma atualização, a primeira etapa é criar um *Software Module*. Mais uma vez, com recurso a um pedido HTTP à API, é possível concretizar essa etapa. O pedido tem a seguinte estrutura:

```
POST /rest/v1/softwaremodules HTTP/1.1
Host:  https://api.eu1.bosch-iot-rollouts.com
Content-Type:  application/json;charset=UTF-8
```

Payload:

```
"name":  "<SoftwareModule_name>"
"type":  "Application"
"version":  "1.0"
```

Upload do ficheiro com o novo *firmware* ou *software*

Depois de criado o *Software Module*, a segunda etapa é selecionar o ficheiro que vai estar associado a ele e que constitui a atualização de *software* ou *firmware* a efetuar. Esta etapa é efetuada também através de um pedido HTTP à API.

No entanto, não foi possível realizar a sua implementação na Aplicação (Servidor), pelo que esta etapa é realizada através da *Management UI*.

Criar *Distribution Set*

A terceira etapa é criar o *Distribution Set*. A sua criação é também efetuada através de um pedido HTTP à API. O pedido tem a seguinte estrutura:

```
POST /rest/v1/distributionsets HTTP/1.1
Host:  https://api.eu1.bosch-iot-rollouts.com
Content-Type:  application/json;charset=UTF-8
```

Payload:

```
"requiredMigrationStep":  "false"
"name":  "<DistributionSet_name>"
"type":  "app"
"version":  "1.0"
```

Associar o *Software Module* ao *Distribution Set*

A quarta e última etapa para criar uma atualização, é a associação do *Software Module* ao *Distribution Set*. Tal como nas etapas anteriores, esta etapa é também executada através de um pedido HTTP à API. O pedido tem a seguinte estrutura:

```
POST /rest/v1/distributionsets/<DistributionSet_ID>/assignedSM HTTP/1.1
Host:  https://api.eu1.bosch-iot-rollouts.com
Content-Type:  application/json;charset=UTF-8
```

Payload:

```
"id":  "<SoftwareModule_ID>"
```

Atribuir o *Distribution Set* ao dispositivo

Registado o dispositivo e criada a atualização, falta apenas atribuir o *Distribution Set* ao dispositivo. Mais uma vez, através de um pedido HTTP à API, é possível efetuar essa atribuição. O pedido tem a seguinte estrutura:

```
POST /rest/v1/distributionsets/<DistributionSet_ID>/assignedTargets/ HTTP/1.1
Host:  https://api.eu1.bosch-iot-rollouts.com
Content-Type:  application/json;charset=UTF-8
```

Payload:

```
"id":  "<controllerId>"
"type":  "forced"
```

Na Figura 5.27 apresenta-se o diagrama com os pedidos HTTP descritos acima.

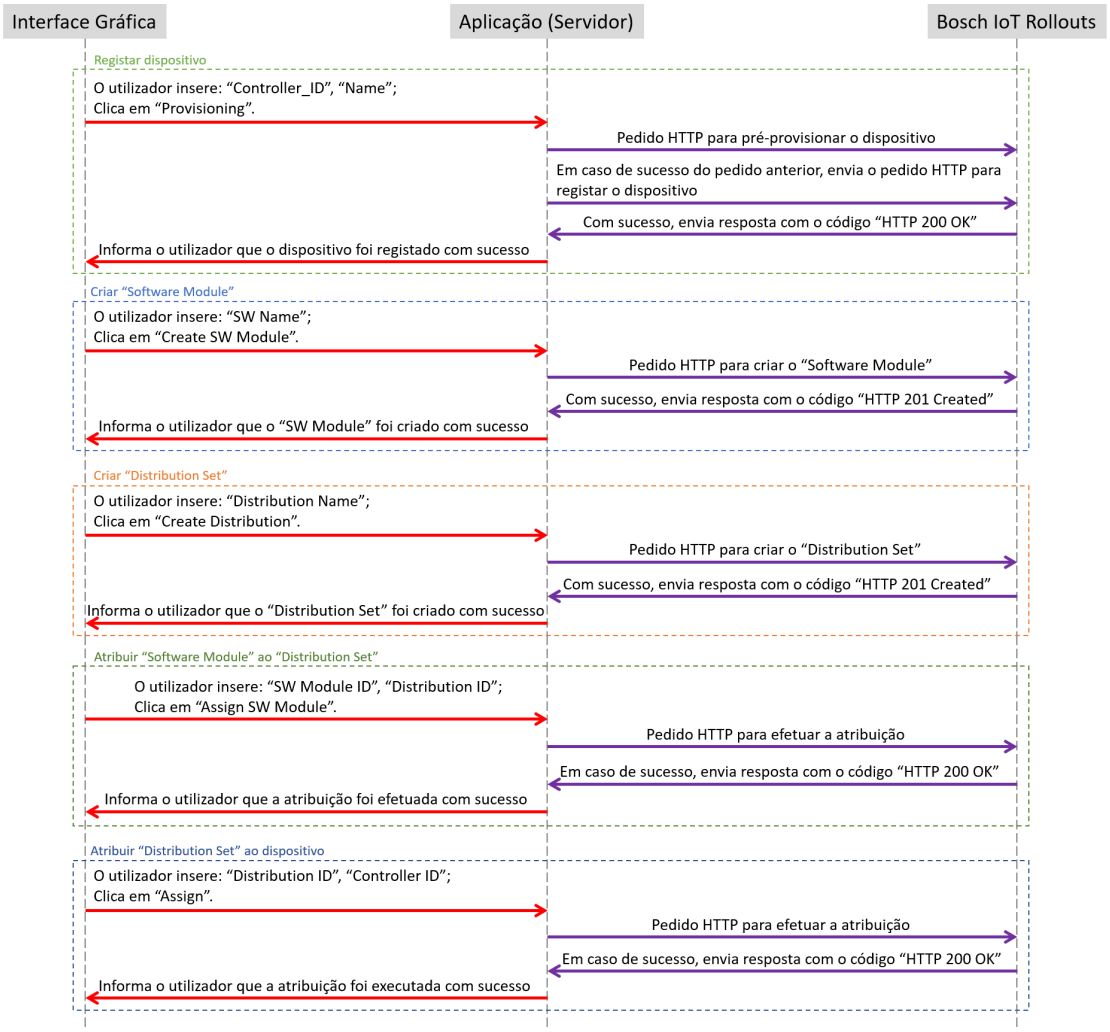

Figura 5.27: Diagrama de interações entre a Aplicação (Servidor), *Bosch IoT Rollouts* e Interface Gráfica

5.5.1.3 Menu da Interface Gráfica

O menu que se apresenta de seguida, juntamente com a integração da *Management API* com a Aplicação (Servidor), permitem registar um dispositivo no serviço *Bosch IoT Rollouts*, assim como criar uma atualização e atribuí-la a um dispositivo registado. Muitos dos parâmetros que são enviados nos pedidos HTTP efetuados e descritos anteriormente, são inseridos através deste menu da Interface Gráfica. Desta forma, o diagrama presente na Figura 5.27, ajuda a perceber a integração do menu da Interface Gráfica com a Aplicação (Servidor).

O menu pode ser observado na Figura 5.28.

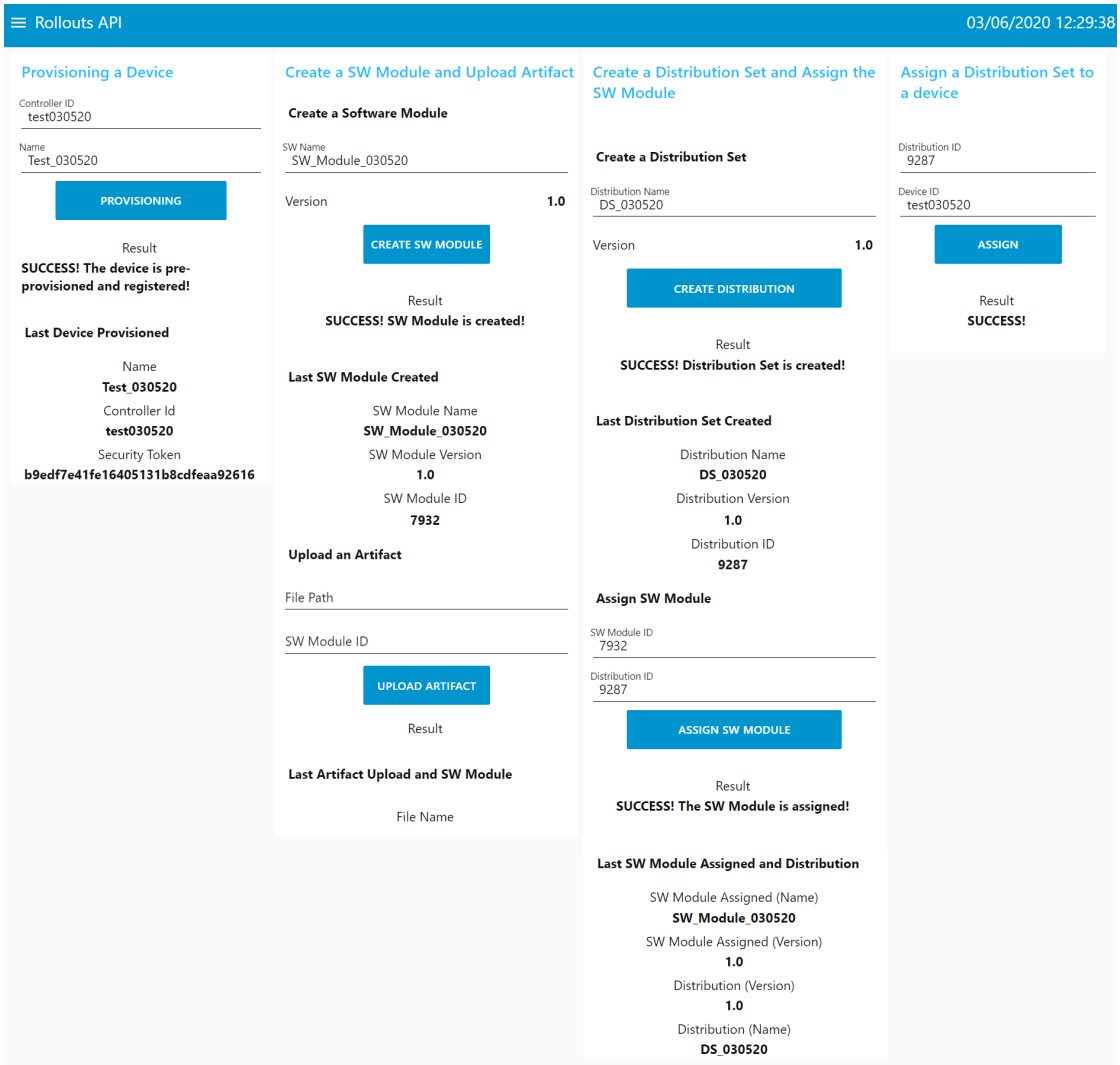

Figura 5.28: Menu para registar um dispositivo, criar uma atualização e atualizar um dispositivo no *Bosch IoT Rollouts*

O menu é composto por quatro colunas. A primeira coluna serve para registar um dispositivo, a segunda coluna para criar o *Software Module*, a terceira coluna para criar o *Distribution Set* e associar um *Software Module* a um *Distribution Set* e, por fim, a quarta coluna serve para associar um *Distribution Set* a um dispositivo registado.

Na primeira coluna inserem-se as credenciais *Controller ID* e *Name* e clica-se no botão *Provisioning* para efetuar o registo do dispositivo. Quando se clica no botão efetuam-se os pedidos HTTP às APIs. Visualiza-se o resultado e ainda breves informações sobre o dispositivo, caso este tenha sido registado com sucesso. Caso já exista um dispositivo com o *Controller ID* inserido, não é possível registar o dispositivo. Naturalmente, também não é válido clicar no botão e as credenciais estarem por preencher.

Na segunda coluna insere-se o nome do *Software Module* que se vai criar. Assumiu-se que os *Software Modules* têm apenas a versão 1.0. Para criar o *Software Module* clica-se no botão *Create SW Module*, e é realizado o pedido HTTP à API. À semelhança da coluna anterior também aqui se visualiza o resultado e breves informações sobre o *Software Module*, caso tenha sido criado com sucesso. Se já houver um *Software Module* criado com o nome inserido, ou o campo esteja por preencher, não é possível realizar a sua criação. É possível visualizar os campos a preencher e o botão para realizar o *upload* do *Artifact*, contudo esta etapa não foi implementada pelo que será realizada através da *Management UI*, como referido.

Na terceira coluna insere-se o nome do *Distribution Set* que se vai criar. Também neste caso se assumiu que os *Distribution Sets* têm apenas a versão 1.0. Para criar o *Distribution Set* clica-se no botão *Create Distribution*, sendo realizado o pedido HTTP à API. Depois de clicar no botão obtém-se o resultado, e caso a criação tenha sucesso, obtêm-se ainda breves informações sobre o *Distribution Set* criado. Se já houver um *Distribution Set* criado com o nome inserido ou o campo esteja por preencher, não é possível realizar a sua criação. Nesta coluna também é possível atribuir um *Software Module* a um *Distribution Set*. Para efetuar essa atribuição, é necessário inserir os IDs do *Software Module* e do *Distribution Set* e por fim clicar no botão *Assign SW Module*. Quando se clica no botão executa-se o pedido HTTP à API. Depois de clicar no botão visualiza-se o resultado e caso a atribuição tenha tido sucesso, visualizam-se também breves informações sobre o *Software Module* e *Distribution Set* em causa.

Na quarta e última coluna, insere-se o ID do *Distribution Set* que vai ser atribuído e o *Controller ID* do dispositivo ao qual se vai atribuir. Depois de preenchidos os campos, clica-se no botão *Assign* executando o pedido HTTP à API. Depois de clicar no botão obtém-se o resultado e, em caso de sucesso, a atribuição fica concluída.

5.5.2 Atualização do lado do dispositivo

Estando a atualização atribuída ao dispositivo, importa perceber o que se tem de realizar do lado do dispositivo. Para obter atualização que lhe está atribuída e realizar a sua instalação, o dispositivo deve efetuar os seguintes passos:

- Verificar se tem uma atualização atribuída;
- Obter informações sobre a atualização;
- Efetuar o *download* do *Artifact*;
- Instalar a atualização;
- Enviar o *feedback* da instalação para o *Bosch IoT Rollouts*.

Todos estes passos são efetuados através de pedidos HTTP à *Direct Device Integration API*, uma API do *Bosch IoT Rollouts*.

No diagrama presente na Figura 5.29, é possível visualizar as interações entre o dispositivo e o *Bosch IoT Rollouts*, mais concretamente os pedidos HTTP efetuados e as respetivas respostas.

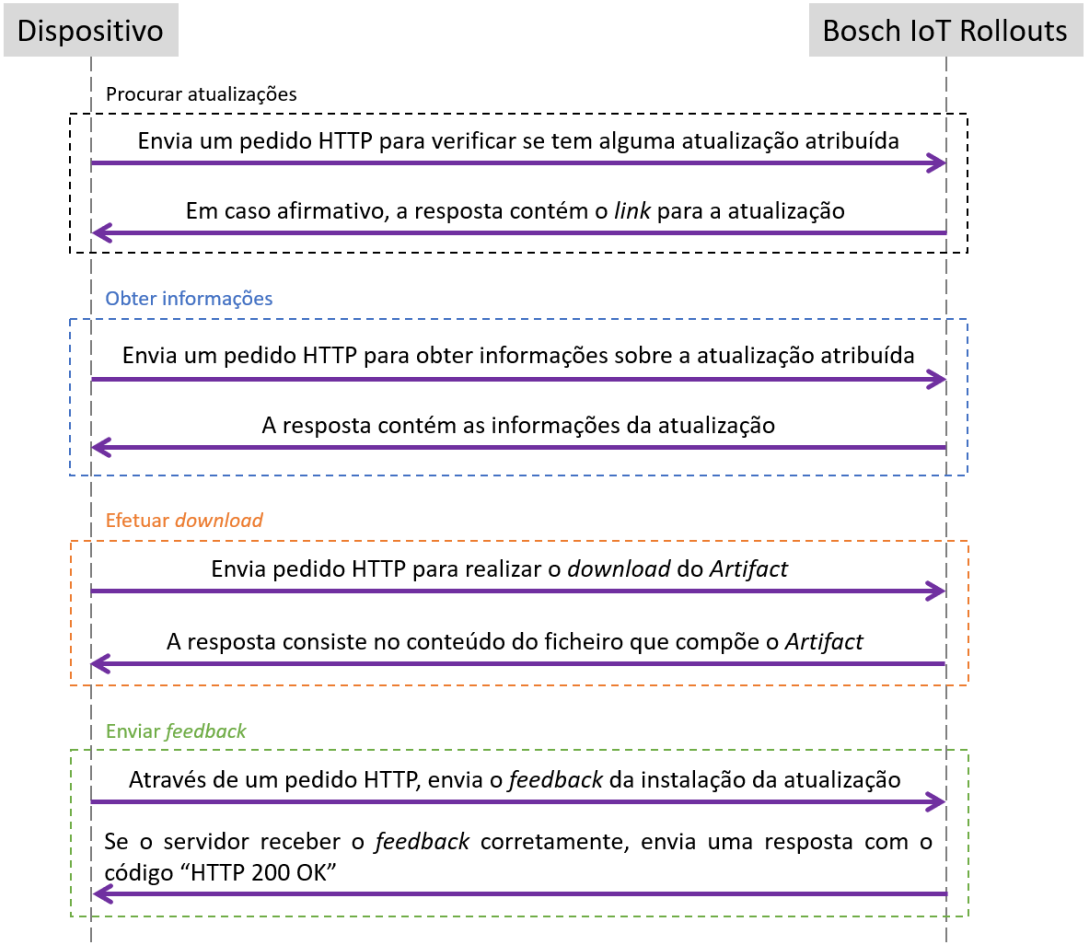

Figura 5.29: Diagrama de interações entre o dispositivo e a *Bosch IoT Rollouts*

De seguida, apresentam-se detalhadamente os pedidos HTTP efetuados.

Verificar se tem uma atualização atribuída

Para verificar se existe alguma atualização atribuída a si, o dispositivo consulta o serviço *Bosch IoT Rollouts*. O pedido HTTP para efetuar a consulta tem a seguinte estrutura:

```
GET /<tenant>/controller/v1/<controllerId> HTTP/1.1
Host:  https://device.eu1.bosch-iot-rollouts.com
Accept:  application/hal+json
Authorization:  TargetToken <Security Token>
```

Caso exista uma atualização atribuída ao dispositivo, a resposta ao pedido contém o *link* para a atualização.

Obter informações sobre a atualização

Efetuando um novo pedido HTTP utilizando o *link* obtido na resposta anterior, obtêm-se informações sobre a atualização, nomeadamente o *action_id*, o *link* para efetuar o *download* do *Artifact* e o nome do ficheiro que compõe o *Artifact*. O pedido tem a seguinte estrutura:

```
GET /<tenant>/controller/v1/<controllerId>/deploymentBase/<restante conteúdo
do link> HTTP/1.1
Host: https://device.eu1.bosch-iot-rollouts.com
Accept: application/hal+json
Authorization: TargetToken <Security Token>
```

Download do *Artifact*

O *download* do *Artifact* é realizado através do pedido HTTP composto pelo *link* obtido na resposta anterior. O pedido tem a seguinte estrutura:

```
GET /<tenant>/<restante conteúdo do link> HTTP/1.1
Host: https://device.eu1.bosch-iot-rollouts.com
Accept: application/hal+json
Authorization: TargetToken <Security Token>
```

Instalar a atualização

Relativamente à instalação da atualização, esta ocorre de forma diferente no ESP8266 e no Raspberry Pi.

No ESP8266, a instalação é efetuada através da biblioteca *ESPhttpUpdate* e de algumas *callbacks* associadas. Depois de instalar a atualização e enviar o *feedback*, o ESP8266 reinicia já com a nova atualização.

Quanto ao Raspberry Pi, a gestão das atualizações é realizada pelo *script Python* Atualização. Na etapa correspondente à obtenção de informações sobre a atualização, uma dessas informações é o nome do ficheiro que corresponde ao *Artifact*. Com base no nome obtido, o *script* verifica se já existe um ficheiro com esse nome no sistema. Caso haja, é reescrito com a informação presente no *Artifact*, caso contrário é criado de raiz. Depois de alterado ou criado o ficheiro, o *script* Atualização envia o *feedback* e ordena a execução do ficheiro.

Enviar *feedback* para o *Bosch IoT Rollouts*

Depois de efetuar o *download* e a instalação da atualização, o dispositivo tem de enviar para o *Bosch IoT Rollouts* o *feedback* sobre o sucesso ou não da instalação. O pedido HTTP para enviar o *feedback* tem a seguinte estrutura:

```
POST /<tenant>/controller/v1/<controllerId>/deploymentBase/<action_id>/
feedback HTTP/1.1
Host: https://device.eu1.bosch-iot-rollouts.com
Content-Type: application/json;charset=UTF-8
Accept: application/hal+json
```

```
Authorization:  TargetToken <Security Token>
```

Payload:

```
"id":  "<action_id>"
"status":  {
  "result":  {
    "finished":  "success"
  },
  "execution":  "closed",
  "details":  [ ]
}
```

5.5.3 Obter informações sobre cada dispositivo através da Aplicação (Servidor)

Continuando a caminhar no sentido de tornar a Aplicação (Servidor) mais completa, dotando-a de todas as funcionalidades necessárias para uma gestão remota de dispositivos mais eficiente, outra funcionalidade desenvolvida é a obtenção de informações sobre cada dispositivo registado no serviço *Bosch IoT Rollouts* subscrito.

A partir do momento em que existe um dispositivo registado, é possível obter informações sobre ele. Essas informações contemplam dados sobre o próprio dispositivo, e caso já lhe tenha sido atribuída e instalada uma atualização, é também possível obter dados sobre o *Artifact*, o *Software Module* e o *Distribution Set* que compõem essa atualização.

Para obter essas informações, a aplicação executa diversos pedidos HTTP à Managment API do *Bosch IoT Rollouts*. Tal como referido na Secção 5.5.1.2, a ligação da Aplicação (Servidor) com a API tem de ser autenticada. Mais uma vez, através das funcionalidades do *Node-RED* é possível efetuar a autenticação.

5.5.3.1 Obter as informações do dispositivo

O primeiro pedido HTTP é efetuado para obter as informações sobre o dispositivo. O pedido à API tem a seguinte estrutura:

```
GET /rest/v1/targets HTTP/1.1
Host:  https://api.eu1.bosch-iot-rollouts.com
```

Na resposta ao pedido obtém-se informação sobre todos os dispositivos registados no serviço subscrito. No entanto, através das funcionalidades do *Node-RED*, consegue-se extrair da resposta as informações relativas a um determinado dispositivo. Essas informações são as seguintes:

- Nome;
- Controller ID;
- Data da última procura por atualizações;
- Nome/código de quem efetuou a última alteração;
- Data da última alteração;

- Nome/código de quem criou;
- Data da criação;
- Link para obter as informações sobre o dispositivo;

5.5.3.2 Obter as informações da última atualização do dispositivo

Obtidas as informações sobre o dispositivo, importa agora obter as informações sobre a última atualização instalada no dispositivo.

Obter as informações do *Distribution Set* que compõe a última atualização

Tendo em conta as etapas para a criação de uma atualização, o próximo passo é obter os dados sobre o *Distribution Set* através da qual se efetuou a última atualização de *firmware* ou *software* desse dispositivo. Para obter esses dados fazem-se dois pedidos HTTP à API.

O primeiro pedido à API é composto pelo *link* que foi obtido na resposta anterior e tem a seguinte estrutura:

```
GET /rest/v1/targets/<controllerId> HTTP/1.1
Host:  https://api.eu1.bosch-iot-rollouts.com
```

Na resposta recebida obtêm-se informações sobre o dispositivo, tais como as que estão acima referidas. Além disso, obtém-se a data da instalação da atualização, e ainda um *link* para obter as informações sobre o *Distribution Set* que compõe a atualização instalada.

O segundo pedido é composto pelo *link* que está presente na resposta anterior e tem a seguinte estrutura:

```
GET /rest/v1/targets/<controllerId>/installedDS HTTP/1.1
Host:  https://api.eu1.bosch-iot-rollouts.com
```

Através das duas repostas recebidas, é possível obter as seguintes informações acerca do *Distribution Set*:

- Nome;
- Versão;
- Nome/código de quem efetuou a última alteração;
- Data da última alteração;
- Nome/código de quem criou;
- Data da criação;

Obter as informações do *Software Module* que compõe a última atualização

Agora que já foram obtidas as informações sobre o dispositivo e sobre o *Distribution Set* que compõe a última atualização instalada, importa obter também as informações sobre o *Software Module* associado ao *Distribution Set*.

O pedido HTTP efetuado para obter as informações sobre o *Distribution Set* também permite obter as informações sobre o *Software Module* associado a ele. Essas informações são as seguintes:

- Nome;
- Versão;
- Nome/código de quem efetuou a última alteração;
- Data da última alteração;
- Nome/código de quem criou;
- Data da criação;
- Link para obter informações sobre o *Software Module*.

Obter o nome do *Artifact* que compõe o *Software Module*

A última informação a obter é o nome do *Artifact* que está associado ao *Software Module*. Com o objetivo de obter essa informação, foram feitas mais dois pedidos HTTP à API. O primeiro pedido é composto pelo *link* obtido na resposta anterior e tem a seguinte estrutura:

```
GET /rest/v1/softwaremodules/<SoftwareModule_ID> HTTP/1.1
Host:  https://api.eu1.bosch-iot-rollouts.com
```

Na resposta obtêm-se informações sobre o *Software Module*, tais como as que estão descritas acima, e ainda o *link* para obter informações sobre os *Artifacts* associados ao *Software Module*.

O segundo pedido à API contempla o *link* que se obteve na resposta anterior e tem a seguinte estrutura:

```
GET /rest/v1/softwaremodules/<SoftwareModule_ID>/artifacts HTTP/1.1
Host:  https://api.eu1.bosch-iot-rollouts.com
```

Através da resposta recebida é possível obter o nome do ficheiro que corresponde ao *Artifact* associado ao *Software Module*.

5.5.3.3 Menu da Interface Gráfica

Também associado a esta funcionalidade da Aplicação (Servidor) existe um menu da Interface Gráfica. Na Figura 5.30, apresenta-se esse menu.

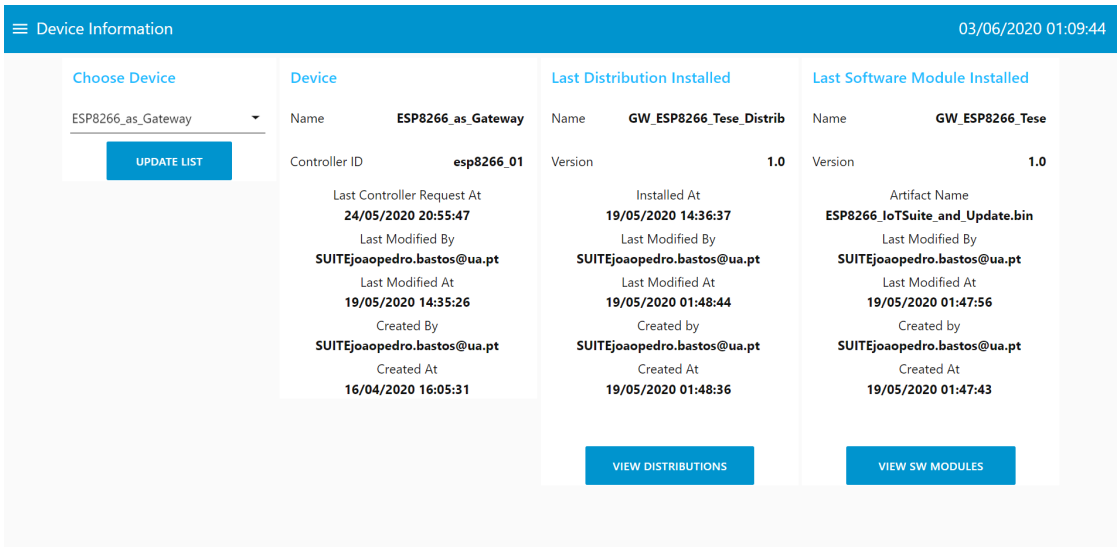

Figura 5.30: Menu para visualizar as informações sobre os dispositivos registados no serviço *Bosch IoT Rollouts*

Como se pode ver na figura anterior, o menu é constituído por quatro colunas.

Na primeira coluna é selecionado o dispositivo, permitindo assim visualizar as suas informações. Uma vez que efetuando o pedido HTTP descrito na Secção 5.5.3.1 obtêm-se informações sobre todos os dispositivos registados, através da seleção de um dispositivo e das funcionalidades do *Node-RED*, apresentam-se apenas as informações relacionadas com o dispositivo selecionado.

Na segunda coluna estão presentes as informações obtidas acerca do dispositivo selecionado.

Na terceira coluna visualizam-se as informações sobre o último *Distribution Set* instalado no dispositivo selecionado.

Por fim, na quarta coluna, apresentam-se as informações sobre o *Software Module* que estando associado ao *Distribution Set* fica também associado ao dispositivo selecionado.

No menu apresentado também é possível observar três botões. O botão *Update List*, o botão *View Distributions* e o botão *View SW Modules*.

Clicando no botão *Update List* atualiza-se a lista de dispositivos que se podem selecionar para visualizar as suas informações. Através das funcionalidades do *Node-RED*, esta lista é preenchida automaticamente com os nomes dos dispositivos registados. Os nomes dos dispositivos são obtidos efetuando um pedido HTTP à *Management API*. Clicando no botão efetua-se a autenticação e o pedido HTTP. O pedido efetuado é o seguinte:

```
GET /rest/v1/targets HTTP/1.1
Host:  https://api.eu1.bosch-iot-rollouts.com
```

Uma vez que a resposta obtida tem muito mais informação do que apenas os nomes dos dispositivos registados, através das funcionalidades do *Node-RED*, é possível retirar apenas os nomes e assim efetuar o preenchimento da lista.

Clicando nos botões *View Distributions* e *View SW Modules*, abrem-se outros menus da Interface Gráfica. Esses menus serão apresentados e explicados na próxima Secção, assim como as funcionalidades da Aplicação (Servidor) relacionadas com eles.

5.5.4 *Distribution Sets* e *Software Modules* criados

Na Secção anterior explica-se como obter informações sobre cada dispositivo e sobre o *Distribution Set*, o *Software Module* e o *Artifact* que constituem a sua última atualização.

No entanto, é importante obter informações sobre todos os *Distributions Sets* e *Software Modules* criados no serviço *Bosch IoT Rollouts* subscrito.

Sabendo todas essas informações, assim como as informações relativas aos dispositivos registados, torna-se muito mais fácil efetuar a gestão desses recursos.

Relativamente aos *Distribution Sets* criados, para obter informações acerca deles, o pedido HTTP à API é o seguinte:

```
GET /rest/v1/distributionsets HTTP/1.1
Host:  https://api.eu1.bosch-iot-rollouts.com
```

Através da resposta recebida é possível obter, entre outros, os seguintes dados:

- Nome;
- Versão;
- *Distribution Set* ID;
- Nome/código de quem efetuou a última alteração;
- Data da última alteração;
- Nome/código de quem criou;
- Data da criação;
- Nome e versão do *Software Module* associado.

Quanto aos *Software Modules* criados, as informações sobre eles são obtidas através do seguinte pedido HTTP à API:

```
GET /rest/v1/softwaremodules HTTP/1.1
Host:  https://api.eu1.bosch-iot-rollouts.com
```

Através da resposta recebida, é possível obter informações tais como:

- Nome;
- Versão;
- *Software Module* ID;

- Nome/código de quem efetuou a última alteração;
- Data da última alteração;
- Nome/código de quem criou;
- Data da criação;
- Nome do *Artifact* associado.

Menus da Interface Gráfica

Para permitir a visualização das informações que se adquirem através dos pedidos HTTP descritos anteriormente, existem dois menus na Interface Gráfica. Existe o menu *Distribution Sets* para visualizar as informações relativas aos *Distribution Sets* criados, o qual é apresentado na Figura 5.31 e existe o menu *Software Modules* para visualizar as informações relativas aos *Software Modules* criados, e que é apresentado na Figura 5.32.

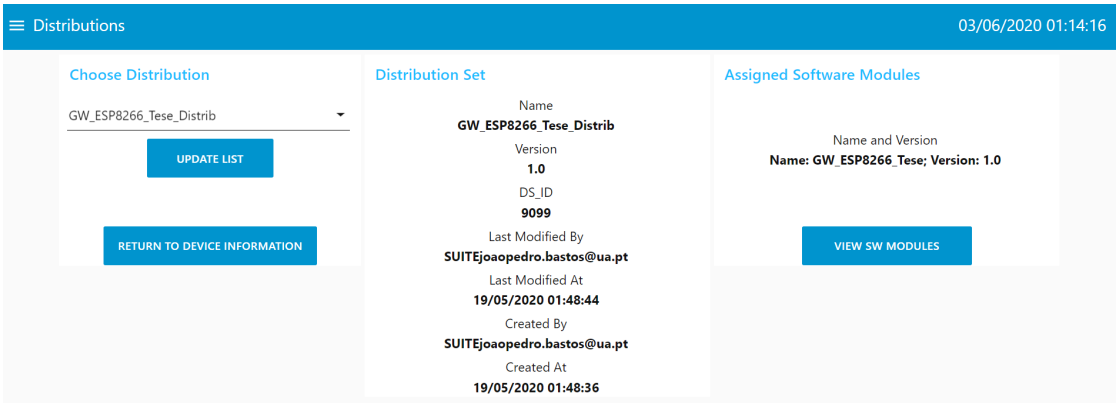

Figura 5.31: Menu para visualizar as informações sobre os *Distribution Sets* criados

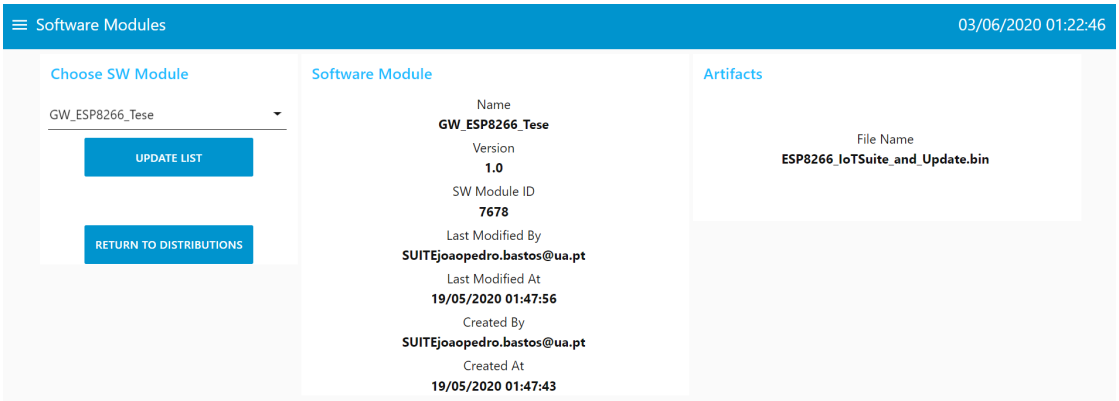

Figura 5.32: Menu para visualizar as informações sobre os *Software Modules* criados

Analisando o menu dos *Distribution Sets*, apresentado na Figura 5.31, observa-se que é composto por três colunas.

Na primeira coluna é selecionado o *Distribution Set* cujas informações se pretendem observar. A partir do pedido HTTP que se efetua e que se encontra descrito na Secção 5.5.4, obtém-se informações sobre todos os *Distribution Sets* criados. Contudo, com a seleção de um *Distribution Set* e através das funcionalidades do *Node-RED*, apresentam-se só as informações relacionadas com o *Distribution Set* selecionado.

Na segunda coluna estão presentes as informações obtidas acerca do *Distribution Set* selecionado.

Por fim, na terceira coluna visualizam-se breves informações sobre o *Software Module* associado ao *Distribution Set* selecionado.

Neste menu é também possível observar três botões. O botão *Update List*, o botão *Return to Device Information* e o botão *View SW Modules*.

Clicando no botão *Update List* atualiza-se a lista de *Distribution Sets* que se podem selecionar para visualizar as suas informações. Através das funcionalidades do *Node-RED*, a lista é preenchida automaticamente com os nomes dos *Distribution Sets* criados. Os nomes são obtidos efetuando um pedido HTTP à *Management API*. Clicando no botão efetua-se a autenticação e o pedido HTTP. O pedido efetuado é o seguinte:

```
GET /rest/v1/distributionsets HTTP/1.1
Host:  https://api.eu1.bosch-iot-rollouts.com
```

Uma vez que a resposta obtida tem muito mais informação do que apenas os nomes dos *Distribution Sets* criados, através das funcionalidades do *Node-RED*, é possível retirar apenas os nomes e assim efetuar o preenchimento da lista.

Clicando no botão *Return to Device Information* regressa-se ao menu *Device Information*, apresentado na Figura 5.26. Por fim, quando se clica no botão *View SW Modules*, acede-se ao menu *Software Modules*.

Analisando agora o menu *Software Modules*, que se pode visualizar na Figura 5.32, observa-se que é composto por três colunas.

Na primeira coluna é selecionado o *Software Module* cujas informações se pretendem visualizar. Como referido na Secção 5.5.4, o pedido HTTP efetuado obtém informações obre todos os *Software Modules* criados. No entanto, a partir da seleção de um *Software Module* e através das funcionalidades do *Node-RED*, apresentam-se só as informações relacionadas com o *Software Module* selecionado.

Na segunda coluna estão presentes as informações obtidas acerca do *Software Module* selecionado.

Na terceira coluna visualizam-se breves informações sobre o *Artifact* associado ao *Software Module* selecionado.

Neste menu é também possível observar dois botões. O botão *Update List* e o botão *Return to Device Information*.

De uma forma semelhante ao que acontece nos menus *Device Information* e *Distribution Sets*, clicando no botão *Update List* atualiza-se a lista, neste caso, de *Software Modules* que se podem selecionar para visualizar as suas informações. Mais uma vez, através das funcionalidades do *Node-RED*, a lista é preenchida automaticamente com os nomes dos *Software Modules* criados. Os nomes são obtidos efetuando um pedido HTTP à *Management API*. Clicando no botão efetua-se a autenticação e o pedido HTTP. O pedido efetuado é o seguinte:

```
GET /rest/v1/softwaremodules HTTP/1.1
Host:  https://api.eu1.bosch-iot-rollouts.com
```

Uma vez que a resposta obtida tem muito mais informação do que apenas os nomes dos *Software Modules* criados, através das funcionalidades do *Node-RED*, é possível retirar apenas os nomes e assim efetuar o preenchimento da lista.

Tal como no menu *Distribution Sets*, clicando no botão *Return to Device Information* regressa-se ao menu *Device Information*, apresentado na Figura 5.26.

Capítulo 6

Análise de Desempenho e Conclusões

6.1 Análise de Desempenho

Tendo em conta os requisitos definidos para a solução que permite efetuar uma gestão remota de dispositivos de uma forma simples e completa, efetua-se uma análise do desempenho de forma a verificar se a solução implementada cumpre esses requisitos.

A análise será feita tendo em conta duas funcionalidades distintas: uma parte para a aquisição, processamento e monitorização de valores por parte dos dispositivos e outra parte relacionada com gestão do *firmware* ou *software* dos dispositivos.

6.1.1 Aquisição, processamento e monitorização de valores

A aquisição, processamento e monitorização dos valores provenientes dos sensores foi implementada com sucesso e é possível visualizar na Figura 5.25.

Usando o *Use Case 1* como primeira validação, o ESP8266 recolhe com sucesso os valores dos sensores e comunica com a *Bosch IoT Suite*, mais concretamente com o *Bosch IoT Hub*. Este, por sua vez, comunica com a *Bosch IoT Things*. Através da integração da API da *Bosch IoT Things* com a Aplicação (Servidor), os valores são extraídos e guardados numa Base de Dados desenvolvida em SQL Express. A visualização dos valores é feita posteriormente recorrendo à Interface Gráfica associada à Aplicação (Servidor). A Aplicação (Servidor) e a Interface Gráfica são desenvolvidas através do *Node-RED*.

Relativamente ao *Use Case 2*, a comunicação com a *Bosch IoT Suite* é efetuada através do Raspberry Pi, sendo este o responsável por adquirir dados de diversos dispositivos. Neste *Use Case*, o ESP8266 envia para o Raspberry Pi os valores recolhidos pelos sensores, os quais são posteriormente enviados pelo Raspberry Pi para o *Bosch IoT Hub*.

Foi ainda implementada uma funcionalidade entre o Raspberry Pi e os sensores, a qual permite pedir uma recolha de dados acionada pelo utilizador.

Aquisição dos valores

As comunicações com os sensores BME280, MMA7455 e a leitura da sua entrada analógica, são realizadas com sucesso pelo ESP8266. Desta forma, é possível obter os valores de temperatura, humidade, acelerações e luminosidade.

No *Use Case 1*, implementou-se com sucesso o protocolo de comunicação MQTT, que permite estabelecer a comunicação com a *Bosch IoT Suite*.

Quanto ao *Use Case 2*, implementou-se também o protocolo de comunicação MQTT, para que seja estabelecida a comunicação entre o ESP8266 e o Raspberry Pi, permitindo que o ESP8266 envie para o Raspberry Pi os valores obtidos pelos sensores. Como neste *Use Case* o Raspberry Pi é que tem a função de comunicar com a *Bosch IoT Suite*, implementou-se também o protocolo MQTT de forma a estabelecer essa comunicação. Todas estas comunicações foram estabelecidas com sucesso.

Registo de dispositivos na *Bosch IoT Suite*

Relativamente aos dispositivos que comunicam com a *Bosch IoT Suite*, o ESP8266 no *Use Case 1* e o Raspberry Pi no *Use Case 2*, ambos têm de estar registados na *Bosch IoT Suite* para que possam estabelecer a comunicação com a *Bosch IoT Suite*.

O registo de dispositivos pode ser realizado de forma manual utilizando diretamente a API *Bosch IoT Suite - Device Provisioning*, através de um conjunto de ferramentas ou através da Aplicação (Servidor) desenvolvida que implementa a API referida.

Na Tabela 6.1 apresenta-se a comparação do número de locais a aceder, o número de cliques e o tempo necessário para registar um dispositivo.

	API	Vorto e Postman	Aplicação (Servidor)
Nº de locais	15	10	1
Nº de cliques	30	20	4
Tempo	5 min	3 min	20 seg

Tabela 6.1: Tabela que demonstra as vantagens do registo de um dispositivo na *Bosch IoT Suite* através da Aplicação (Servidor)

Ao registar um dispositivo na *Bosch IoT Suite*, este fica habilitado a utilizar os serviços *Bosch IoT Things* e o *Bosch IoT Hub*.

Aplicação (Servidor) e *digital twin*

Com o registo de um dispositivo na *Bosch IoT Suite*, cria-se também um *digital twin* na *Bosch IoT Things*. O *digital twin* é uma representação digital do dispositivo físico e do seu valor num determinado instante de tempo.

A comunicação entre o dispositivo físico e o *digital twin* é realizada através do *Bosch IoT Hub*. Assim, os dados que o dispositivo físico envia ficam armazenados na *Bosch IoT Things*, mais concretamente no *digital twin*.

Através da integração da *Bosch IoT Things - API v2* com a Aplicação (Servidor), é possível obter periodicamente os novos valores que os dispositivos físicos enviam para os *digital twins* sem qualquer intervenção do utilizador. Os valores obtidos são depois armazenados na Base de Dados criada.

Para permitir a visualização dos valores obtidos, e assim cumprir com um dos requisitos da solução, desenvolveu-se a Interface Gráfica. Aqui, o utilizador pode escolher o dispositivo cujos valores adquiridos e estado de funcionamento pretende verificar.

De uma forma geral, desde a aquisição dos valores até à sua visualização, passando pela seu armazenamento e pela obtenção do estado de funcionamento do dispositivo,

através da Aplicação (Servidor) todos estes processos ocorrem sem intervenção humana, o que não era até então possível devido à utilização das APIs. A intervenção humana surge apenas na Interface Gráfica, onde o utilizador pode escolher o dispositivo.

6.1.2 Gestão do *firmware* ou *software*

Quanto à parte da gestão do *firmware* ou *software* instalado nos dispositivos, foram desenvolvidas funcionalidades para que essa gestão possa ser feita remotamente.

As funcionalidades desenvolvidas aplicam-se no ESP8266 no *Use Case 1*, no Raspberry Pi no *Use Case 2* e na Aplicação (Servidor), e o seu desenvolvimento tem como base a comunicação com duas APIs do *Bosch IoT Rollouts*, a *Management API* e a *Direct Device Integration API*.

Ao contrário da *Bosch IoT Things*, o *Bosch IoT Rollouts* já oferece uma *User Interface* fácil de usar. A partir dela é possível registar dispositivos, criar atualizações, atualizar dispositivos e obter informações sobre os dispositivos registados e sobre as atualizações criadas. No entanto, este serviço também oferece diferentes APIs para efetuar a sua integração com outras aplicações.

Integração da Aplicação (Servidor) com o *Bosch IoT Rollouts*

A integração da Aplicação (Servidor) com o *Bosch IoT Rollouts* é estabelecida justamente através de uma dessas APIs, neste caso a *Management API*.

Decidiu-se fazer essa integração com a Aplicação (Servidor) para que esta se torne mais completa e capaz de oferecer todas as funcionalidades para efetuar a gestão remota de dispositivos de uma forma simples, rápida e eficiente. Deste modo, além das capacidades relacionadas com a monitorização dos valores, a Aplicação (Servidor) passa também a ter as capacidades que até aqui só estavam presentes na *User Interface* do *Bosch IoT Rollouts*.

Assim, com a Aplicação (Servidor) é possível registar dispositivos no *Bosch IoT Rollouts*, criar atualizações e atribuí-las aos dispositivos registados. Além disso, é possível obter informações sobre cada dispositivo registado, nomeadamente dados relacionados com o seu registo ou com as alterações efetuadas e qual a versão de *software* ou *firmware* que tem atualmente instalado. Por fim, também é possível obter informações sobre todas as atualizações criadas.

Contudo, durante a criação de uma atualização, o *upload* do ficheiro que contém o novo *firmware* ou *software* tem de ser feito através da interface gráfica que o serviço *Bosch IoT Rollouts* fornece.

Atualização do lado do dispositivo

Relativamente ao dispositivo, seja o ESP8266 ou o Raspberry Pi, através das funcionalidades desenvolvidas, este é capaz de comunicar com o *Bosch IoT Rollouts* através da *Direct Device Integration API*.

Através desta comunicação, o dispositivo verifica se tem alguma atualização atribuída. Caso tenha, obtém as informações acerca dela, nomeadamente o endereço para realizar o seu *download*. Depois de efetuado o *download*, o dispositivo instala a atualização e envia o *feedback* de sucesso para a API.

Através da implementação destas funcionalidades, é possível realizar uma gestão do *firmware* ou *software* recorrendo praticamente apenas à Aplicação (Servidor).

6.2 Conclusões

A dissertação Gestão de Remota de Dispositivos foi concluída com sucesso. Desenvolveu-se uma ferramenta *low cost* que incorpora todas as necessidades identificadas inicialmente para a gestão de dispositivos: monitorização e atualizações.

Tendo em consideração os objetivos para o desenvolvimento da ferramenta, o utilizador conseguiu ter acesso aos dados disponibilizados remotamente, conseguindo ainda controlar o *software/firmware* no dispositivo. A ferramenta foi desenvolvida para utilizar *software Open Source*, neste caso a *Bosch IoT Suite* (*Bosch IoT Hub*, *Bosch IoT Things* e *Bosch IoT Rollouts*), estando então ligada ao *working group* do *Eclipse IoT*. A nível de arquitetura, esta foi definida de forma a integrar o ecossistema de soluções da TT, estando ainda a contribuir para o repositório de soluções da equipa do TT/MFD. A interface gráfica foi desenvolvida tendo em consideração ferramentas já disponibilizadas pela Bosch, estando assim mais acessível aos seus futuros utilizadores. É possível ao utilizador registar dispositivos, escolher os valores a monitorizar, adaptar os próprios *dashboards* e ainda consegue atualizar remotamente o *software* ou pacote de *softwares* no seu dispositivo, reduzindo assim o número de tarefas repetitivas, o número de aplicações a utilizar e o tempo para as fazer.

Apesar de não ter sido possível a implementação da solução em ambiente industrial, o ambiente controlado em que foi testado permite verificar a robustez e a fiabilidade da solução.

Ao longo desta dissertação surgiram ainda alguns desafios como a utilização de *software* ainda em desenvolvimento, a utilização de linguagens de programação não conhecidas ou a utilização de novas tecnologias. No entanto, o desafio foi recebido com entusiasmo e concretizado.

Assim, desenvolveu-se uma solução como base para gestão dos dispositivos que compõe uma rede IoT de uma empresa.

6.3 Trabalhos futuros

Devido ao facto de não ter sido possível realizar a implementação do *upload* do *Artifact* através da Aplicação (Servidor) e da Interface Gráfica associada, um dos trabalhos futuros passa por ultrapassar essa barreira.

Esta dissertação permitiu a recolha de dados de múltiplos dispositivos no chão de uma fábrica. Assim, a implementação da ferramenta mas com análise de dados ligada a um desafio como a *condition monitoring* ou a manutenção preditiva seria um próximo passo interessante.

Tendo em conta a Aplicação (Servidor) em si e o local onde irá correr, importa monitorizar o servidor ou computador onde irá estar em execução. Essa monitorização deve ser realizada através da obtenção de métricas do computador ou servidor como RAM, CPU, entre outras, e da aplicação em si como por exemplo, se o serviço está a correr e o seu consumo específico ao longo do tempo. Também um conjunto de alarmes definidos por *thresholds* deveriam ser configurados.

Uma tarefa que ainda requer muita interação humana, é muito repetitiva e leva o seu tempo a realizar, é a subscrição de um serviço ou plano. Tendo isto em conta e pensando sempre em acrescentar novas funcionalidades à ferramenta, mais concretamente à Aplicação (Servidor), um desenvolvimento interessante seria tornar essa tarefa mais

simples e rápida. Um método para implementar essa funcionalidade poderá ser através de *bots* que replicassem as ações humanas juntamente com alguns desenvolvimentos na Aplicação (Servidor) e na Interface Gráfica associada.

Bibliografia

[1] Bosch. O Grupo Bosch no mundo; Disponível em: http://www.bosch.pt/a-nossa-empresa/o-grupo-bosch-no-mundo.

[2] Bosch. Bosch em Portugal; Disponível em: https://www.bosch.pt/a-nossa-empresa/bosch-em-portugal.

[3] Bosch. Bosch Termotecnologia S.A.; Disponível em: https://www.bosch.pt/a-nossa-empresa/bosch-em-portugal/aveiro/.

[4] Bosch. Nexeed Manufacturing Execution System; Disponível em: https://www.bosch-india-software.com/en/products-and-services/digital/factories-of-the-future/nexeed-mes/.

[5] Bosch ConnectedWorld Blog. Device management: the key to future-proof IoT deployments; Disponível em: https://blog.bosch-si.com/bosch-iot-suite/device-management-iot-deployments/.

[6] IoT For All. IoT Device Management: What is it and Why do You Need it?; Disponível em: https://www.iotforall.com/what-is-iot-device-management/.

[7] Lasi H, Fettke P, Kemper HG, Feld T, Hoffmann M. Industry 4.0. Business and Information Systems Engineering. 2014; Disponível em: https://link.springer.com/article/10.1007%2Fs12599-014-0334-4.

[8] Roblek V, Mesko M, Krapez A. A Complex View of Industry 4.0. SAGE Open. 2016; Disponível em: http://journals.sagepub.com/doi/pdf/10.1177/ 2158244016653987.

[9] Eletrical Technology. What is Industrial Automation | Types of Industrial Automation; Disponível em: https://www.electricaltechnology.org/2015/09/what-is-industrial-automation.html.

[10] Åkerman M. Implementing Shop Floor IT for Industry 4.0. 2018; Disponível em: https://www.researchgate.net/publication/326224890_Implementing_Shop_Floor_IT_for_Industry_40.

[11] Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. IEEE. 2015; Disponível em: https://ieeexplore.ieee.org/document/7123563.

[12] Reliableplant. Breaking Down Condition Monitoring; Disponível em: https://www.reliableplant.com/condition-monitoring-31760.

[13] Bosch ConnectedWorld Blog. Software updates in the IoT: an introduction to SOTA; Disponível em: https://blog.bosch-si.com/bosch-iot-suite/software-updates-in-the-iot-an-introduction-to-sota/.

[14] Bosch ConnectedWorld Blog. How to manage software updates in the IoT; Disponível em: https://blog.bosch-si.com/bosch-iot-suite/how-to-manage-software-updates-in-iot/.

[15] Opentext. Top 5 benefits of industrial IoT platforms; Disponível em: https://blogs.opentext.com/industrial-iot-platforms/.

[16] IoT For All. What is an IoT Platform?; Disponível em: https://www.iotforall.com/what-is-an-iot-platform/.

[17] Bosch. The core of the Bosch IoT Suite is based on open source; Disponível em: https://developer.bosch-iot-suite.com/open-source/.

[18] Bosch. Página inicial da Bosch IoT Suite; Disponível em: https://www.bosch-iot-suite.com/.

[19] Bosch. Capabilities of the Bosch IoT Suite; Disponível em: https://www.bosch-iot-suite.com/capabilities-bosch-iot-suite/.

[20] Bosch. Easy and secure device connectivity for the IoT; Disponível em: https://developer.bosch-iot-suite.com/service/hub/.

[21] Bosch. Managed inventory of digital twins for IoT device assets; Disponível em: https://developer.bosch-iot-suite.com/service/things/.

[22] Bosch. Manage and control software and firmware updates for IoT devices; Disponível em: https://developer.bosch-iot-suite.com/service/rollouts/.

[23] AWS. Página inicial da AWS IoT; Disponível em: https://aws.amazon.com/pt/iot/.

[24] AWS. AWS IoT para aplicações industriais; Disponível em: https://aws.amazon.com/pt/iot/solutions/industrial-iot/?c=isec=uc1.

[25] Microsoft Azure. Microsoft Azure IoT Suite – Connecting Your Things to the Cloud; Disponível em: https://azure.microsoft.com/pt-pt/blog/microsoft-azure-iot-suite-connecting-your-things-to-the-cloud/.

[26] IBM. Getting to know MQTT; Disponível em: https://developer.ibm.com/articles/iot-mqtt-why-good-for-iot/.

[27] Eclipse Foundation. MQTT and CoAP, IoT Protocols; Disponível em: https://www.eclipse.org/community/eclipse$_n ewsletter$/2014/$february/article$2$.php$.

[28] HiveMQ. Quality of Service 0,1 2 - MQTT Essentials: Part 6; Disponível em: https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/.

[29] Fielding R, Gettys J, Mogul J, Frystyk H, Masinter L, et al. Hypertext Transfer Protocol – HTTP/1.1. 2007; Disponível em: https://www.w3.org/Protocols/HTTP/1.1/rfc2616bis/draft-lafon-rfc2616bis-03.html.

[30] Cloudflare. What Is HTTPS?; Disponível em: https://www.cloudflare.com/learning/ssl/what-is-https/.

[31] Espressif Systems. ESP8266 Datasheet. 2020; Disponível em: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf.

[32] Last Minute Engineers. Insight Into ESP8266 NodeMCU Features & Using It With Arduino IDE. 2019; Disponível em: https://lastminuteengineers.com/esp8266-nodemcu-arduino-tutorial.

[33] Comunidade Raspberry Pi Portugal. Raspberry Pi; Disponível em: https://www.raspberrypiportugal.pt/raspberry-pi/.

[34] Raspberry Pi. Raspberry Pi 3 Model B+; Disponível em: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/.

[35] Comunidade Raspberry Pi Portugal. Raspberry Pi 3 Model B+; Disponível em: https://www.raspberrypiportugal.pt/novo-raspberry-pi-3-model-b-disponivel/.

[36] Robotics University. Introducing Raspberry Pi 3 Model B+; Disponível em: https://www.robotics-university.com/2018/06/introducing-raspberry-pi-3-model-bplus.html.

[37] Bosch. BME Datasheet. Disponível em: https://ae-bst.resource.bosch.com/media/$_t ech/media/datasheets/BST - BME280 - DS002.pdf$.

[38] Last Minute Engineers. Interface BME280 Temperature, Humidity Pressure Sensor with Arduino; Disponível em: https://lastminuteengineers.com/bme280-arduino-tutorial/.

[39] NXP. MMA7455 Datasheet; Disponível em: https://www.nxp.com/docs/en/data-sheet/MMA7455L.pdf.

[40] Parallax Inc. MMA7455 3-Axis Accelerometer Module; Disponível em: https://www.parallax.com/product/28526.

[41] Bosch. Command and control - request-response; Disponível em: https://docs.bosch-iot-suite.com/asset-communication/Command-and-control—request-response.html

Apêndice A

Tutorial *Bosch IoT Things*

Este apêndice tem como base o tutorial *Connect a simple device to the Bosch IoT Suite*, o qual pode ser acedido clicando aqui. Com este tutorial, pretende-se fornecer uma visão geral sobre os recursos mais importantes do serviço *Bosch IoT Things*. Para se efetuar essa visão geral, efetuam-se as seguintes etapas:

- Subscrever e configurar o plano *Bosch IoT Suite for Asset Communication*;
- Registar o dispositivo na *Bosch IoT Suite*;
- Gerar o *firmware* a instalar no dispositivo;
- Configurar o *firmware*;
- Modificar o *firmware* para enviar valores reais
- Modificar o *firmware* para ativar *command & control*

A.1 Subscrever e configurar o plano *Bosch IoT Suite for Asset Communication*

Para subscrever e configurara o plano, é necessário efetuar as seguintes etapas:

- Criar uma conta *Bosch IoT Suite*;
- Subscrever o plano mencionado;
- Configurar um *namespace* para os dispositivos;
- Criar o *OAuth2.0 Client*;
- Gerar um *Test Token*.

A.1.1 Criar uma conta na *Bosch IoT Suite*

Antes de começar, é preciso criar uma conta na *Bosch IoT Suite*. Os passos para a sua criação são os seguintes:

1 - Entrar no *site* da *Bosch IoT Suite*, clicar no icone *My Account* e de seguida em *Sign in*. Estes passos estão demonstrados na Figura A.1.
2 - Irá abrir-se um novo separador, onde se clica em *Ainda não está registado?*. (Figura A.2)

3 - De seguida, abre-se a página para efetuar o registo. Nessa página preenche-se o endereço de e-mail, define-se a palavra-passe, aceita-se os termos e condições, e a política de privacidade e, por fim, clica-se no botão *Registar*. (Figura A.3)

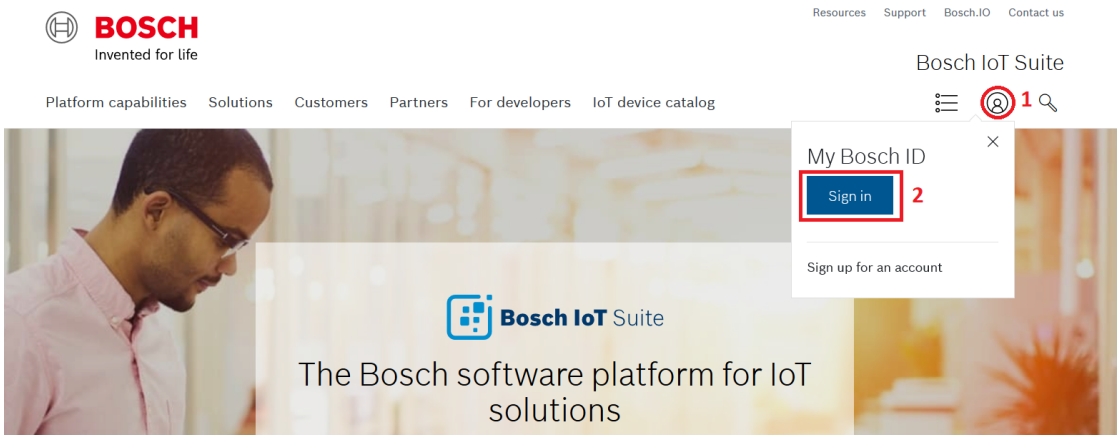

Figura A.1: Página inicial da *Bosch IoT Suite*

Figura A.2: Aceder à página para efetuar o registo na *Bosch IoT Suite*

Figura A.3: Preencher as credenciais e efetuar o registo

4 - Realizados estes passos, é enviado para o endereço de *e-mail*, um *e-mail* para ativar a conta. Nesse *e-mail* apenas é necessário clicar em *Ativar conta*.

5 - Depois da conta estar criada e ativada, efetuam-se as últimas configurações, tais como o nome e apelido do utilizador.

6 - Por fim, inicia-se sessão com as credenciais criadas.

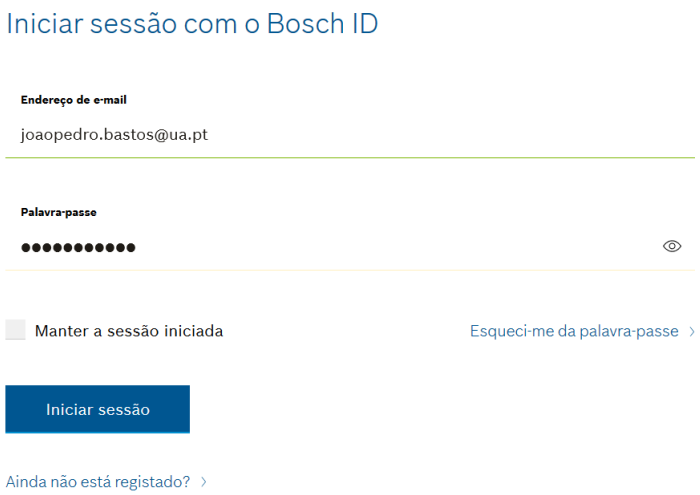

Figura A.4: Iniciar sessão com o Bosch ID criado

A.1.2 Subscrever o plano

Depois de criada a conta na *Bosch IoT Suite*, a próxima etapa é subscrever o plano *Bosch IoT Suite for Asset Communication*. Este plano inclui os serviços *Bosch IoT Hub* e *Bosch IoT Things*.

1 - Iniciada a sessão, na página inicial clicar em *To Subscriptions*.

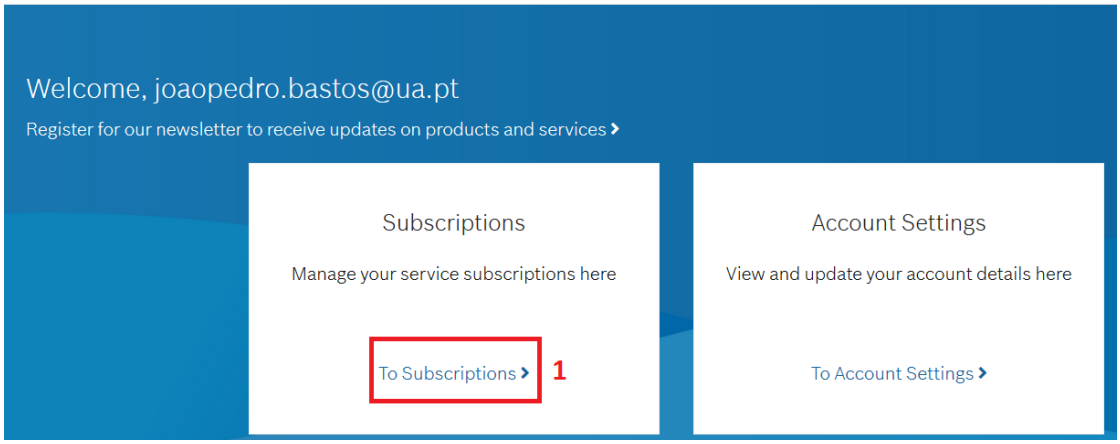

Figura A.5: Ambiente gráfico depois de iniciar sessão e primeiro passo para realizar a subscrição de um serviço

2 - No menu *Service Subscriptions*, clicar em em *New Subscription*.

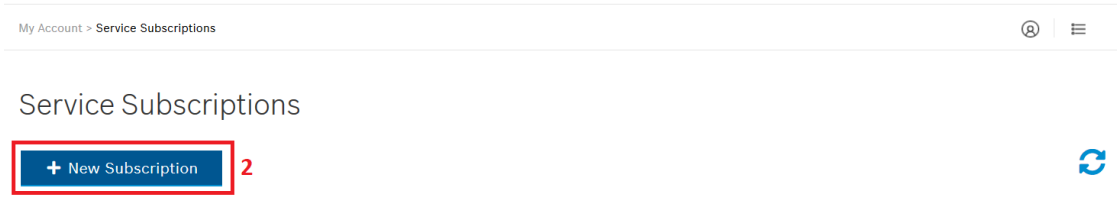

Figura A.6: Segundo passo para subscrever um serviço da *Bosch IoT Suite*

3 - Escolher o plano *Bosch IoT Suite for Asset Communication*.

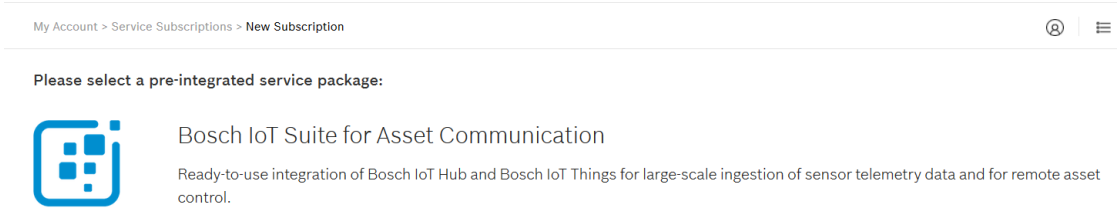

Figura A.7: Seleção do plano *Bosch IoT Suite for Asset Communication*

4 - Subscrever o serviço, selecionando o *Free Plan* de Frankfurt (EU-1), inserindo o *Instance Name* e clicando em *Subscribe*.

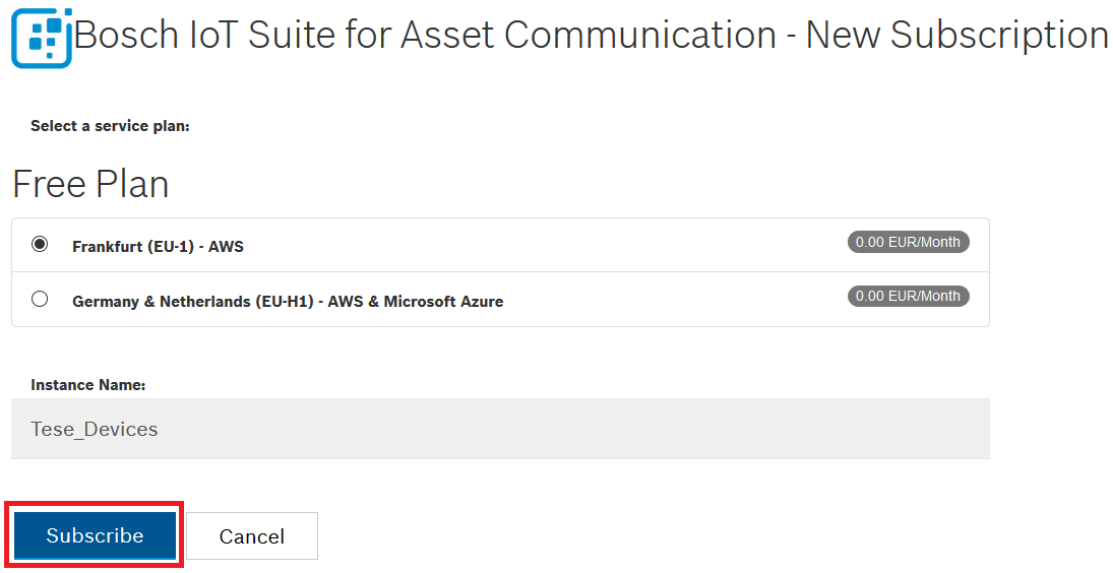

Figura A.8: Subscrição do plano *Bosch IoT Suite for Asset Communication*

Irá abrir-se uma nova janela para confirmar os detalhes da subscrição e o valor (0€). Para terminar efetivamente a subscrição do plano, clica-se em *Subscribe*.

A.1.3 Configurar um *namespace* para os dispositivos

O *namespace* é um prefixo exclusivo para os Things ID's que se irão criar posteriormente, tornando também exclusivo cada um deles.

A configuração do *namespace* efetua-se através dos seguintes passos:

1 - No menu *Service Subscriptions*, procurar o *Instance Name* do serviço subscrito anteriormente e clica-se em *Go to Things Dashboard*.

Figura A.9: Aceder ao plano subscrito no menu *Service Subscriptions*

2 - De seguida, clicar em *Namespace*.

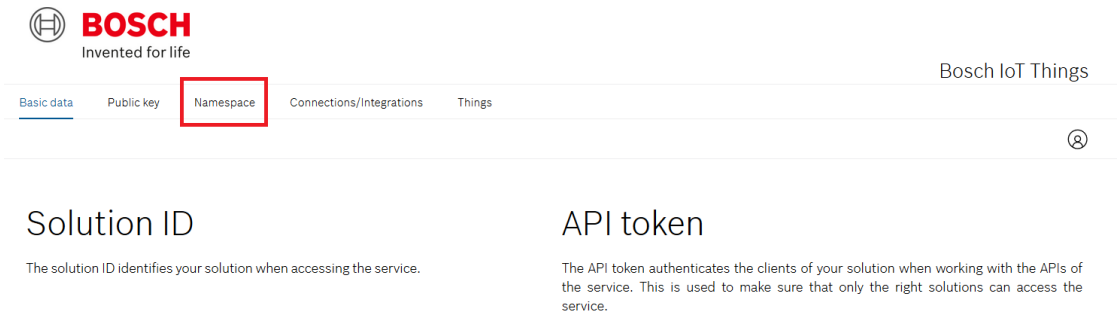

Figura A.10: Primeiro passo para a definição no *Namespace*

3 - Inserir um novo e único *Namespace* e clicar em *Submit Namespace*.

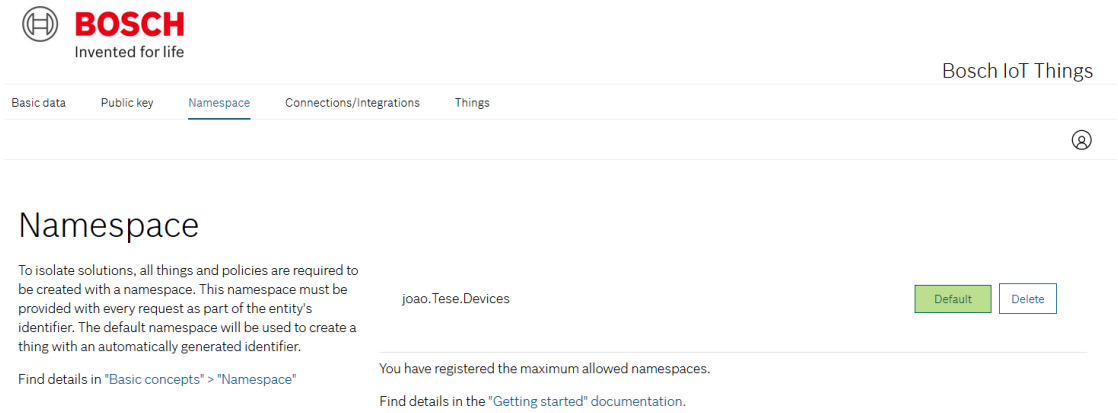

Figura A.11: Definição do *Namespace*

A.1.4 Criar o *OAuth2.0 Client*

O *OAuth2.0 Client* permite efetuar a autenticação e assim usar as APIs da *Bosch IoT Suite*. Os passos para a sua criação de um cliente com total acesso à *Bosch IoT Things* e *Bosch IoT Hub* são os seguintes:

1 - Clicar no ícone *My Account* e de seguida em *OAuth2 Clients*.

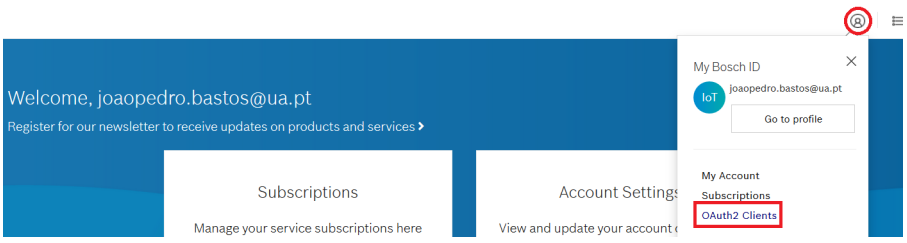

Figura A.12: Aceder ao menu *OAuth2 Clients* a partir do ambiente gráfico inicial

2 - De seguida, clicar em *New OAuth2 Client*.

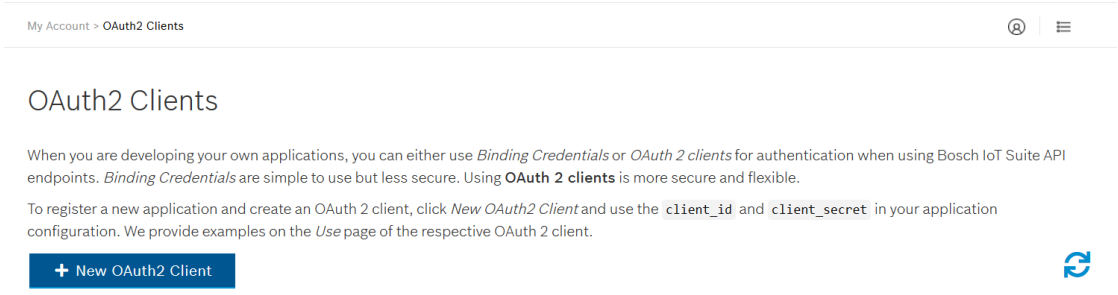

Figura A.13: Primeiro passo para a criação de um *OAuth2 Clients* novo

3 - Atribuir um nome ao *OAuth2 Client*, selecionar ambas as *checkbox* e clicar em *Create*.

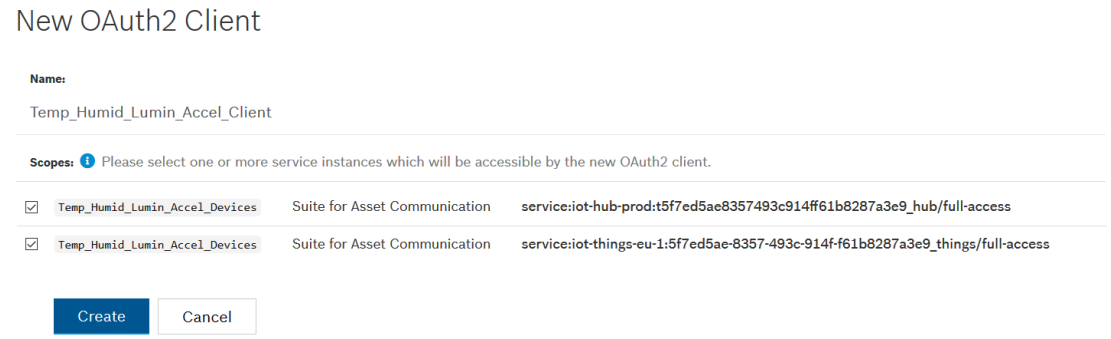

Figura A.14: Nomear e criar o novo *OAuth2 Clients*

A.1.5 Gerar um *Test Token*

O *Test Token* permite que o cliente realize a sua autenticação em todas as APIs da *Bosch IoT Suite*. Cada *Test Token* gerado tem a validade de 59 minutos.

Os passos para gerar o *Test Token* são os seguintes:

1 - Clicar no ícone *My Account* e de seguida em *OAuth2 Clients*.

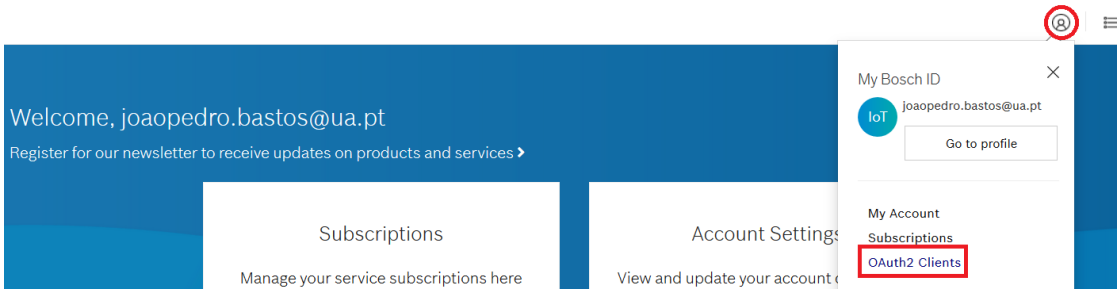

Figura A.15: Aceder ao menu *OAuth2 Clients* a partir do ambiente gráfico inicial

2 - No *OAuth2 Client* que se pretende usar, clicar em *Use*.

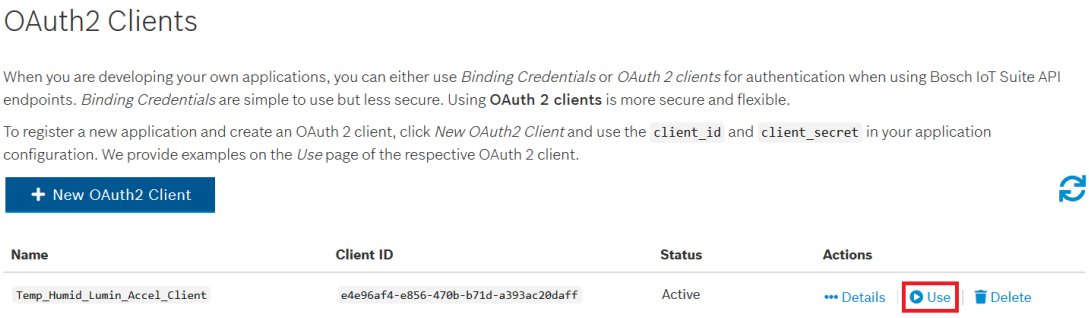

Figura A.16: Ativar o *OAuth2 Client* que se pretende usar

3 - Por fim, clicar em *Copy to clipboard* para copiar o *Test Token*.

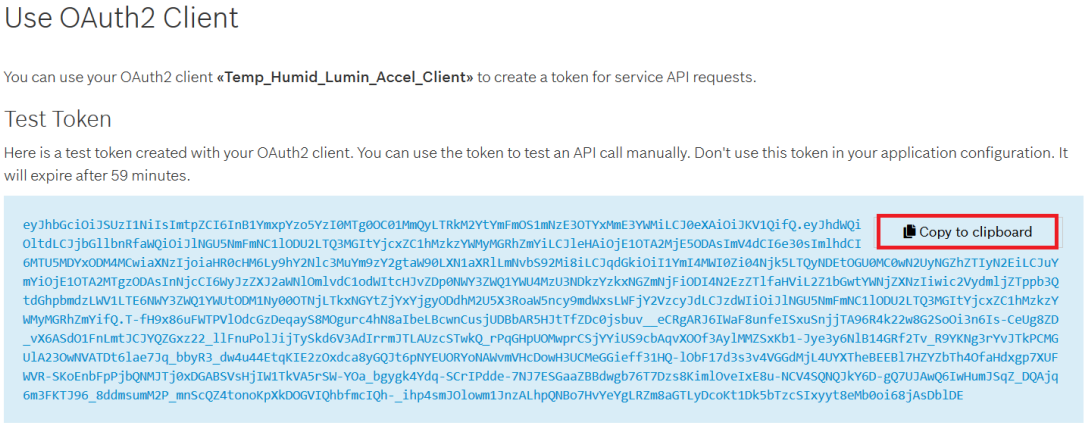

Figura A.17: Copiar o *Test Token* criado

A.2 Registar o dispositivo na *Bosch IoT Suite*

Existem duas maneiras de registar um dispositivo na *Bosch IoT Suite*:

- Através da *Device Provisioning API*;
- Através do *Vorto Repository* e do *Postman*;

A.2.1 Usando a Device Provisioning API

Para registar o dispositivo no plano *Bosch IoT Suite for Asset Communication*, recorre-se à Device Provisioning API (*link* para a API) através dos seguintes passos:

1 - Clicar em *Authorize*.

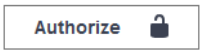

Figura A.18: Autenticação do cliente (1/2)

2 - Abre-se uma caixa de diálogo e aí cola-se o *Test Token* como valor da secção *bearerAuth(http, Bearer)*. De seguida clicar em *Authorize* e fechar a janela de diálogo clicando em *Close*.

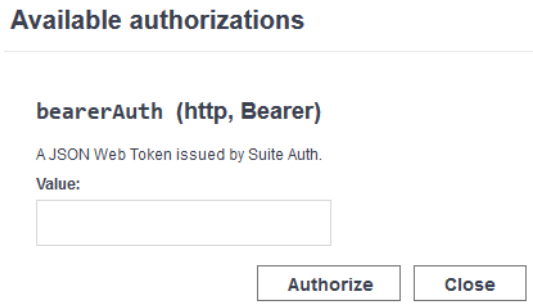

Figura A.19: Autenticação do cliente (2/2)

3 - Expandir a secção *provisioning*, clicar no método *POST /service-instance-id/devices* e clicar em *Try it out* (Figura A.20).

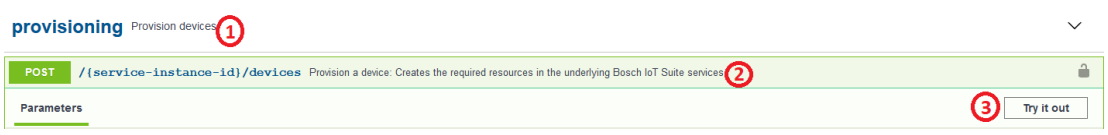

Figura A.20: Primeiros passos para usar a API

4 - Adicionar o *Service Instance ID*.

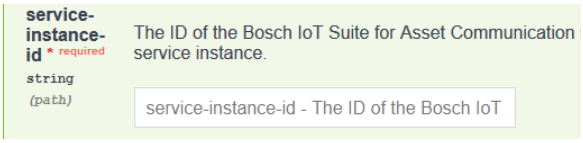

Figura A.21: Campo para inserir o *Service Instance ID*

Para obter o *Service Instance ID* acede-se ao menu *Service Subscriptions* e na subscrição que se está a usar clica-se em *Show Credentials*.

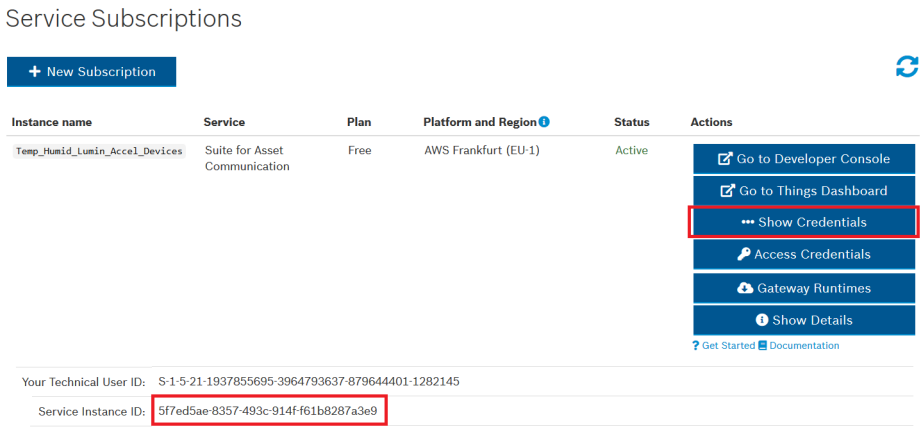

Figura A.22: Obter o *Service Instance ID* a partir do menu *Service Subscriptions*

5 - Na caixa de texto presente na API, num formato *JSON*, são definidas as informações do dispositivo, entre as quais as credenciais do dispositivo (ID e Password), atributos, *features*. O ID do dispositivo é constituído pelo *namespace* definido anteriormente, e pelo nome do dispositivo que se define aqui pela primeira vez. Terá um aspeto semelhante a <namespace:nome_dispositivo>.

6 - Depois de preenchido o campo *Edit Value*, clicar em *Execute*.

Os últimos dois passos são demonstrados na Figura A.23.

Figura A.23: Preencher o campo com o *JSON* e terminar o registo

A resposta obtida deve conter o código *HTTP 201 Created*, e o seu conteúdo é referente ao dispositivo que se acabou de registar.

A.2.2 Usando o *Vorto Repository* e o *Postman*

Uma alternativa a recorrer diretamente à *Device Provisioning API*, é utilizar o *Vorto Repository* e o *Postman* (requer instalação). O *Vorto Repository* gera o código e permite a sua transferência, auxiliando no processo de registo do dispositivo através do textitPostman, bem como no desenvolvimento do *firmware* que será instalado.

A.2.2.1 Obter os recursos do *Vorto Repository*

Para obter os recursos que são utilizados para registar o dispositivo ou então como seu *firmware*, efetuam-se os seguintes passos:

1 - Efetuar o *Login* no *Vorto Repository* (*link* para o *site*). O *Login* pode ser efetuado com o Bosch ID. De seguida, clicar em *Create Model*.

2 - Na criação do modelo, escolher o tipo *Information Model* e depois clicar em *Next*.

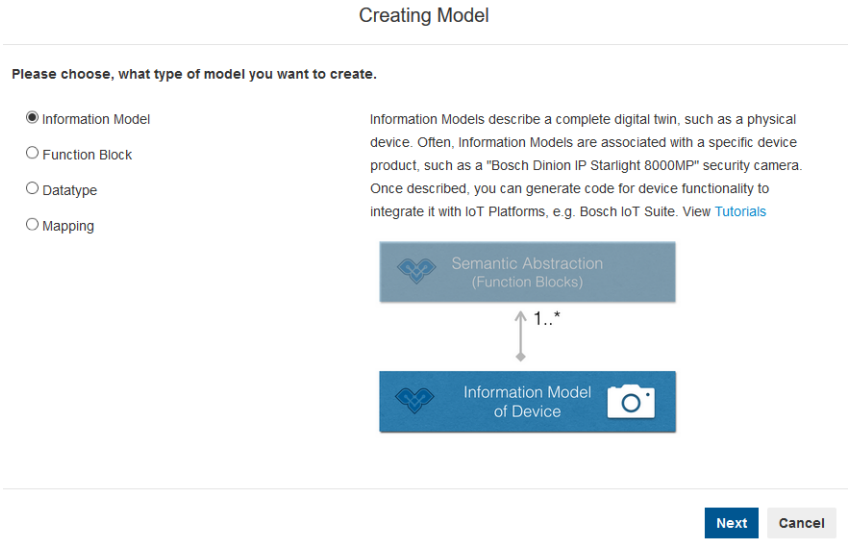

Figura A.24: Escolher o tipo *Information Model*

3 - Especificar o *namespace* (pode-se escolher o sugerido), o nome e a versão do modelo. Por fim, clicar em *Next*.

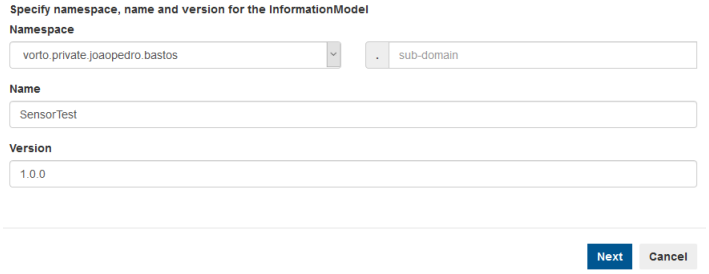

Figura A.25: Definir o *namespace*, o nome e a versão do modelo

4 - Escolher as propriedades, selecionando e adicionando uma de cada vez clicando em *Add*, e finalizar clicando em *Create*.

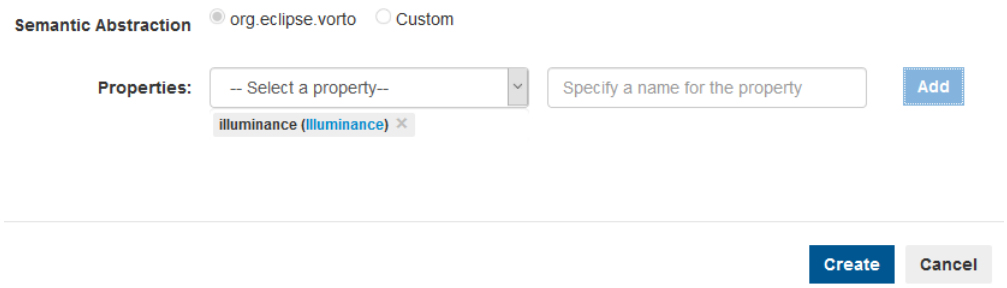

Figura A.26: Escolher as propriedades do modelo

5 - Escolher o *Plugin* da *Bosch IoT Suite* na lista de *Official Plugins* à direita.

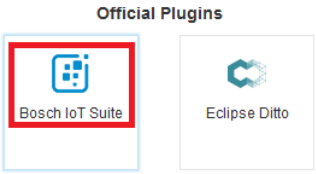

Figura A.27: Selecionar o *Plugin* da *Bosch IoT Suite*

6 - Na secção *Provision device*, clicar em *Source Code*. Será transferido um ficheiro *JSON*, que posteriormente será usado no *Postman* para efetuar o registo do dispositivo. Ainda neste passo, na secção *Connect device with Arduino/C* ou *Connect device with Python* (depende do dispositivo físico que se está a usar), clicar em *Source Code* e desta forma efetuar o *download* do *firmware*.

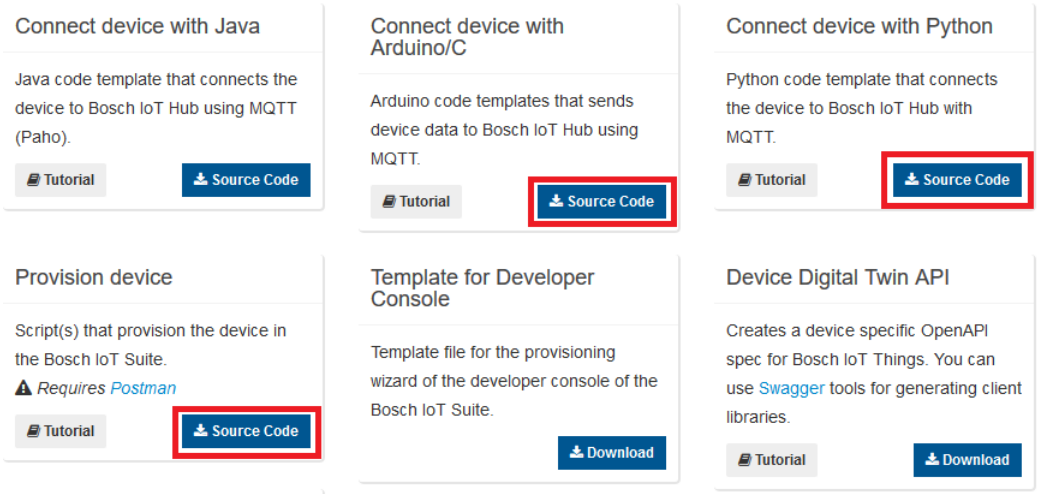

Figura A.28: Realizar o *download* do *JSON* e do *firmware*

A.2.2.2 Utilização do *Postman*

Como já referido, para usar esta ferramenta é necessário efetuar a sua instalação. O *download* pode ser realizado neste *site*.

Realizada a transferência e a instalação do *Postman*, os passos a realizar são os seguintes:

1 - Importar o ficheiro *JSON* transferido do *Vorto Repository*, clicando em *Import*. Irá abrir-se uma janela, para a qual se pode arrastar o ficheiro ou então clicar em *Upload Files* e procurar e selecionar o ficheiro. Feito o *upload* do ficheiro, clica-se em *Import* e este passa a estar presente na coluna à esquerda no menu *Collections*.

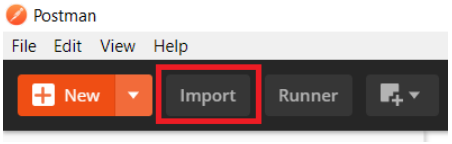

Figura A.29: Importar para o *Postman* o *JSON* transferido

2 - Na coluna à esquerda clicar sobre o ficheiro e de seguida sobre *Provision Request*.

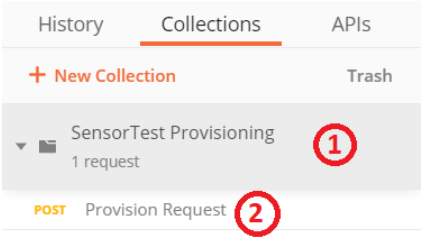

Figura A.30: Primeiros passos para registar o dispositivo através do *Postman*

3 - O próximo passo é inserir os dados necessários para efetuar o registo do dispositivo. Esses dados serão introduzidos em dois menus: o menu *Headers* e o menu *Pre-request Script*.

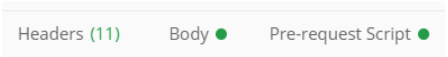

Figura A.31: Os três menus usados para realizar o registo

No menu *Headers* coloca-se o *Test Token*. Aqui, substitui-se *ADD_BEARER_TOKEN_HERE* pelo valor do *Test Token*. A forma de gerar o *Test Token* está explicada na Secção A.1.5.

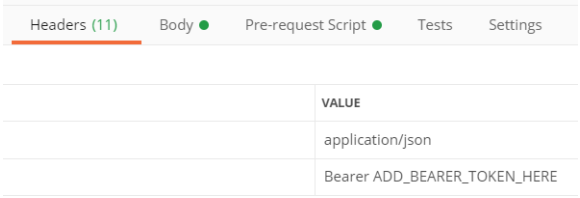

Figura A.32: Adicionar o *Test Token* no menu *Headers*

No menu *Pre-request Script* definem-se três credenciais: o *Service Instance ID*, a *password* do dispositivo, assim como o seu *id*. Importa relembrar que a forma de obter o *Service Instance ID* está representada na Figura A.22.

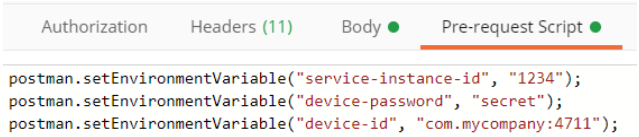

Figura A.33: Definir as três credenciais no menu *Pre-request Script*

O *Postman* depois tem a função de associar estas credenciais a variáveis presentes no *JSON* que pode ser analisado no menu *Body*.

No menu *Body* é necessário eliminar uma linha do *JSON* lá presente. A linha em causa é a *definition* da *thing* que se vai criar.

Figura A.34: Editar o *JSON* presente no menu *Body*

4 - Depois de inserido o *Test Token*, o *Service Instance ID*, a *password* e o *device_ id*, o último passo é clicar em *Send*.

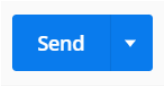

Figura A.35: Clicar em *Send* e concluir o registo do dispositivo

A resposta obtida deve conter o código *HTTP 201 Created*, e o seu conteúdo é referente ao dispositivo que se acabou de registar.

A.3 Gerar o *firmware* a instalar no dispositivo

Esta etapa já foi abordada quando se efetua o registo de um dispositivo através do *Vorto Repository* e do *Postman*.

Como referido no ponto 6 da Secção A.2.2.1, através do *Vorto Repository* também é possível obter o *firmware* que se vai instalar no dispositivo, seja em *Arduino/C* ou em *Python*. Como se pode observar na Figura A.28, clicando em *Source Code* obtém-se o *firmware* para o dispositivo.

No caso de um dispositivo em que a linguagem do seu *firmware* seja Arduino/C, é também necessário configurar o Arduino IDE. Informações detalhadas podem ser encontradas no tutorial *Connect a simple device to the Bosch IoT Suite*, o qual pode ser acedido clicando aqui.

A.4 Configurar o *firmware*

Para que o dispositivo estabeleça ligação com a *Bosch IoT Things* através do *Bosch IoT Hub*, é necessário realizar algumas configurações no *firmware*, nomeadamente relacionadas com o plano *Bosch IoT Suite for Asset Communication* subscrito, com as credenciais de registo do dispositivo e com a ligação *Wi-Fi*. Essas configurações estão explicadas e demonstradas detalhadamente no tutorial *Connect a simple device to the Bosch IoT Suite*, o qual pode ser acedido clicando aqui.

A.5 Modificar o *firmware* para enviar dados reais

As próximas modificações a efetuar no *firmware* são para que o dispositivo estabeleça ligação com os sensores. Assim, os valores que o dispositivo envia para a *Bosch IoT Suite* são reais. Mais uma vez, informações mais detalhadas sobre as modificações no *firmware* podem ser encontradas no tutorial *Connect a simple device to the Bosch IoT Suite*, o qual pode ser acedido clicando aqui.

Terminadas estas alterações, compila-se e efetua-se o *upload* do *firmware* para o dispositivo.

Importa agora verificar se os valores estão a ser enviados corretamente para o *digital twin* do dispositivo físico. Para realizar essa verificação, utiliza-se a *Bosch IoT Things - API v2*.

As etapas da verificação são:

1 - Aceder à API clicando aqui.
2 - Para aceder à API é necessário efetuar a autenticação. Mais uma vez, tal como explicado na Secção A.2.1, clica-se em *Authorize* e de seguida cola-se o *Test Token* na autorização *bearerAuth (http, Bearer)*. A obtenção do *Test Token* está explicada na Secção A.1.5.
3 - Expandir a secção *Things*, clicar no método *GET /things/thingId* e clicar em *Try it out*.

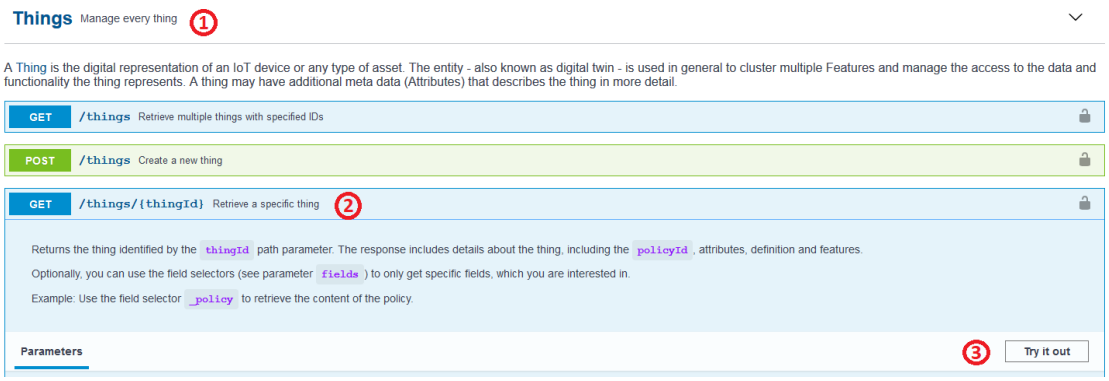

Figura A.36: Secção *Things* da *Bosch IoT Things - API v2*

4 - Adicionar o *Thing ID (namespace:device_id)*.

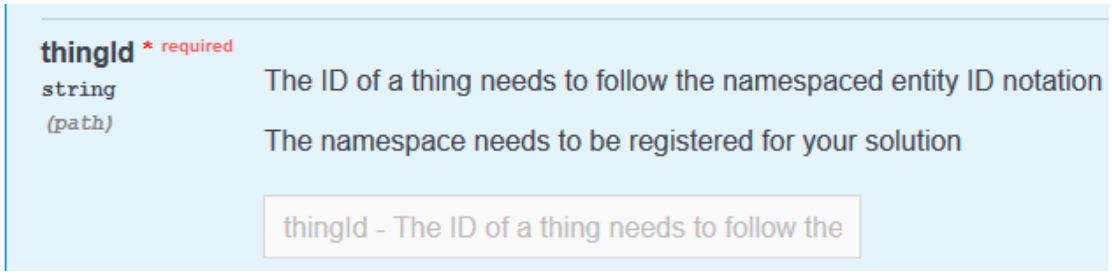

Figura A.37: Preencher o campo *thingId*

5 - Navegar para baixo na página e clicar em *Execute* para efetuar o pedido HTTP à API.

A resposta obtida deve conter o código *HTTP 200 OK*, e o conteúdo contém os valores que foram recolhidos pelos sensores.

A.6 Modificar o *firmware* para ativar *command & control*

Na Secção acima, os dados foram enviados no dispositivo para a *Bosch IoT Things* através do *Bosch IoT Hub*. No entanto, também é possível enviar comandos para o dispositivo a partir da *Bosch IoT Things*. Além de receber os comandos, o dispositivo também deve ser capaz processar esse comando e enviar uma resposta.

Para que o dispositivo receba esses comandos e envie as respostas, é necessário efetuar novas alterações no *firmware*, ao nível da comunicação com o *Bosch IoT Hub*. Informações mais detalhadas sobre essas alterações podem ser encontradas no tutorial *Connect a simple device to the Bosch IoT Suite*, o qual pode ser acedido clicando aqui.

Realizadas as alterações, compila-se e efetua-se o *upload* do *firmware* para o dispositivo.

Novamente, importa perceber se as alterações efetuadas no *firmware* estão a efetuar as suas funções. Para verificar se tudo funciona corretamente, usa-se mais uma vez a *Bosch IoT Things - API v2*.

As etapas da vrificação são as seguintes:

1 - Aceder à API clicando aqui.

2 - Como já referido, para aceder à API é necessário efetuar a autenticação. Mais uma vez, tal como explicado na Secção A.2.1, clica-se em *Authorize* e de seguida cola-se o *Test Token* na autorização *bearerAuth (http, Bearer)*. A obtenção do *Test Token* está explicada na Secção A.1.5.

3 - Expandir a secção *Messages*, clicar no método *POST /things/thingId/inbox/messages/messageSubject* e clicar em *Try it out*.

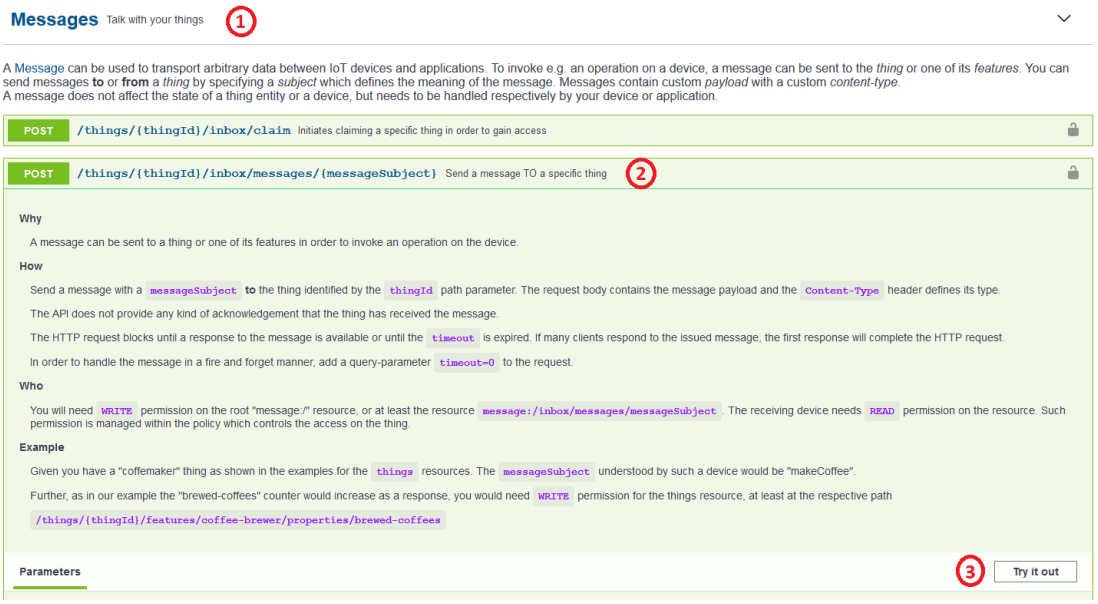

Figura A.38: Secção *Messages* da *Bosch IoT Things - API v2*

4 - Adicionar o *Thing ID (namespace:device_id)* e o *messageSubject*. Para que o dispositivo receba o comando, efetue o seu processamento e por fim envie uma resposta, o *messageSubject* aqui inserido tem de estar de acordo com a configuração do *firmware*. Ou seja, o dispositivo tem de estar preparado para receber mensagens com esse *messageSubject*.

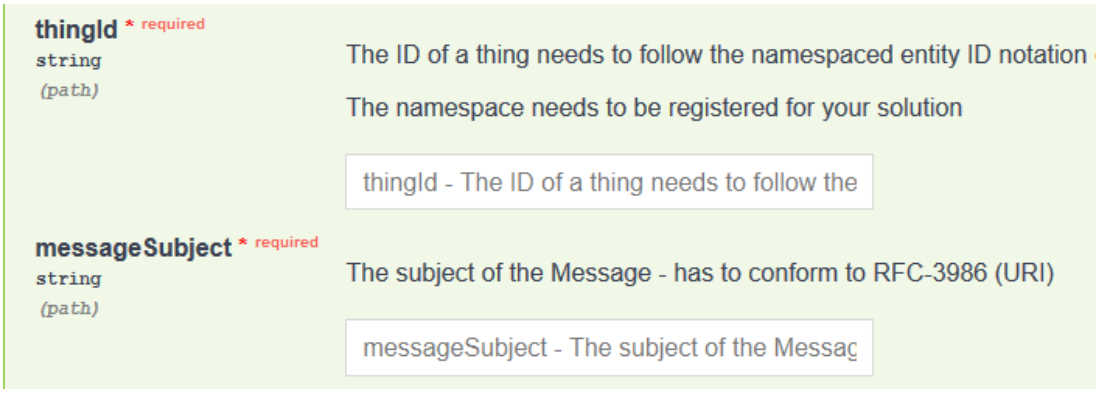

Figura A.39: Preencher os campos *thingId* e *messageSubject*

5 - No campo destinado ao conteúdo da mensagem enviada, inserir um *JSON* vazio, tal como {}.

6 - Por fim, clicar em *Execute* para efetuar o pedido HTTP à API.

A resposta obtida deve conter o código *HTTP 200 OK*, e o conteúdo contém a resposta enviada pelo dispositivo.

Apêndice B

Tutorial *Bosch IoT Rollouts*

Este apêndice destina-se a fornecer uma visão geral dos recursos mais importantes do serviço *Bosch IoT Rollouts*. Para se efetuar essa visão geral, seguem-se as seguintes etapas:

- Obter o serviço;
- Provisionar um dispositivo;
- Criar uma atualização;
- Atualizar um dispositivo

B.1 Obter o serviço

Uma vez que se trata de um serviço da *Bosch IoT Suite*, é necessário ter uma conta criada. A criação de uma conta na *Bosch IoT Suite* é descrita no Apêndice A.

Criada a conta, no menu *Service Subscriptions*, clicar em *New Subscription* e subscrever o serviço *Bosch IoT Rollouts*.

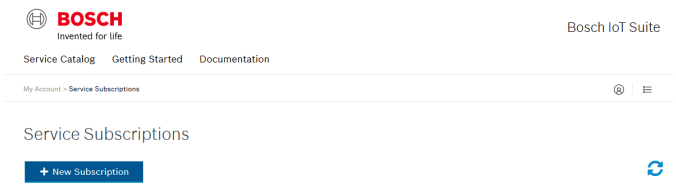

Figura B.1: Subscrever um novo serviço a partir do menu *Service Subscriptions*

Figura B.2: Selecionar o serviço *Bosch IoT Rollouts*

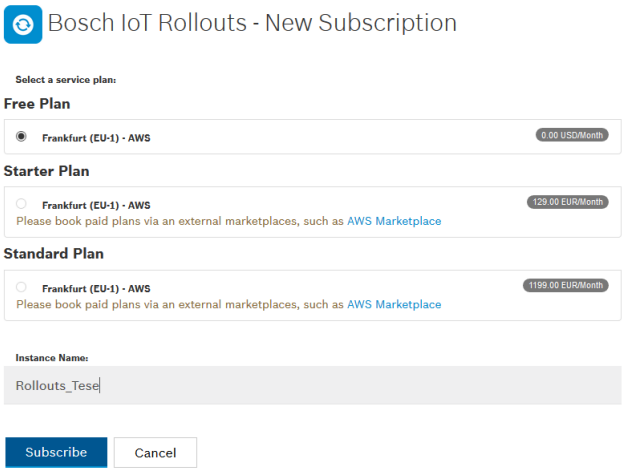

Figura B.3: Subscrever o plano grátis do serviço

De seguida selecionar o *Free Plan*, atribuir o *Instance Name* e clicar em *Subscribe*. Por fim, verificar os detalhes da subscrição e o custo e clicar em *Subscribe*.

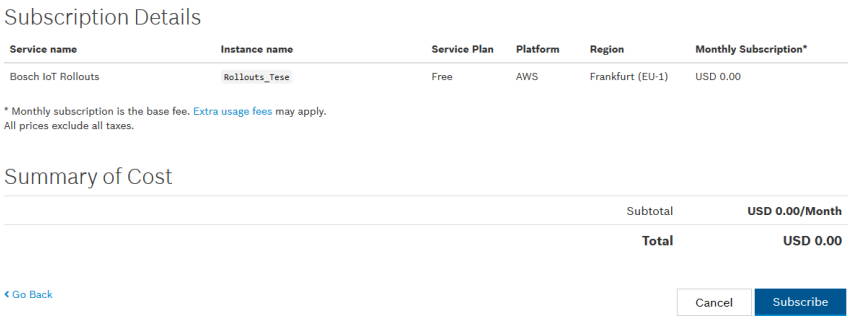

Figura B.4: Verificar o custo e terminar a subscrição

Alguns momentos depois, o serviço fica disponível para usar.

Quando o serviço já estiver em funcionamento, no menu *Service Subscriptions* procura-se o serviço em causa e clica-se em *Go to Dashboard*.

Figura B.5: Aceder ao serviço subscrito a partir do *Service Subscriptions*

B.2 Provisionar um dispositivo

Para provisionar um dispositivo, é necessário efetuar as seguintes etapas:

- Pre-provisionar o dispositivo;
- Registar o dispositivo.

B.2.1 Pre-provisionar o dispositivo

O *Bosch IoT Rollouts* efetua a gestão dos dispositivos, neste caso *Targets*, através do menu *Deployment Management*. Para se provisionar um novo dispositivo clica-se no ícone + na tabela *Targets*.

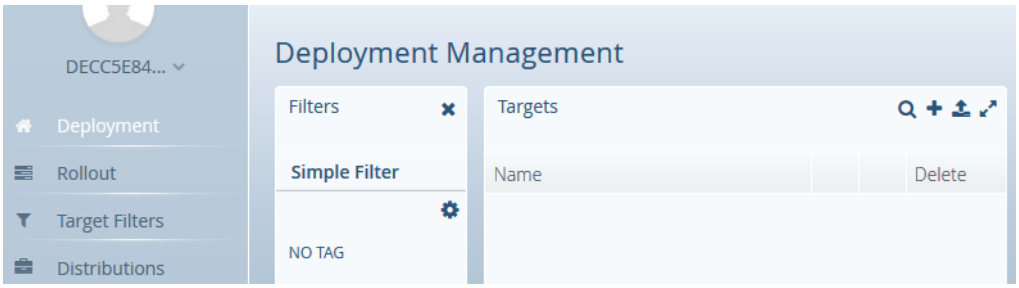

Figura B.6: Pre-provisionar um dispositivo (1/2)

Clicando neste icon abre-se uma janela onde se insere o *Controller ID*, o nome e uma breve descrição do dispositivo. Por fim clica-se em *Save*. No fim desta etapa, o estado do dispositivo é *UNKNOWN*.

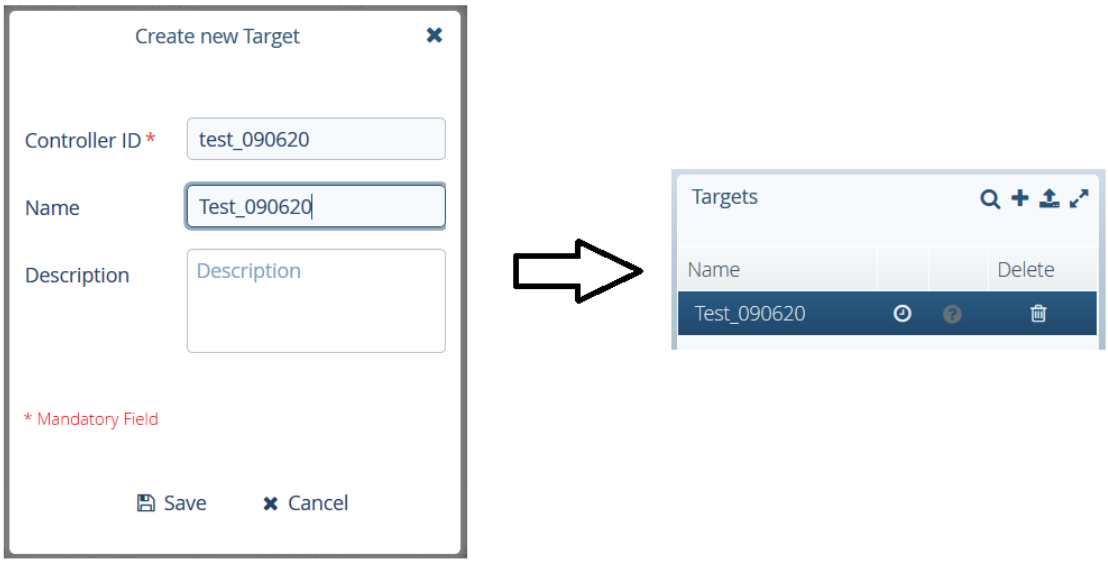

Figura B.7: Pre-provisionar um dispositivo (2/2)

B.2.2 Registar o dispositivo

Depois de pre-provisionar o dispositivo, é necessário registá-lo através de um pedido HTTP à *Direct Device Integration API*.

No entanto, é necessária uma autenticação para aceder à API. Essa autenticação é efetuada através do *Security Token* do dispositivo. É também necessário permitir que o dispositivo use o seu *Security Token* para efetuar a autenticação. Essa permissão é realizada através do menu *System Config*, selecionando a *checkbox Allow targets to authenticate directly with their target security Token* na secção *Authentication Configuration*.

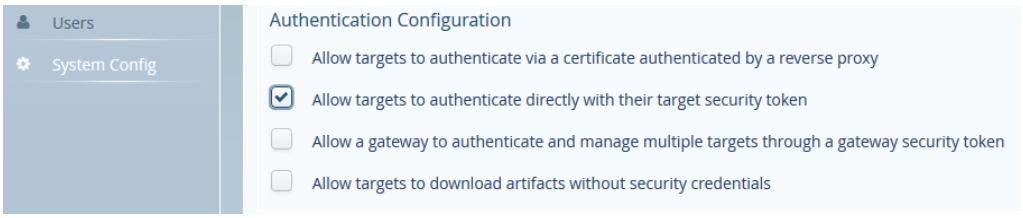

Figura B.8: Permitir utilização do *Security Token* para realizar a autenticação

Por fim, para salvar as alterações efetuadas, clica-se no ícone (disquete) no fim da página.

O *Security Token* encontra-se no fim da tabela *Targets* nos *Details* do dispositivo selecionado.

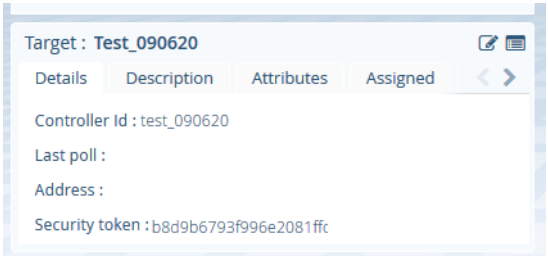

Figura B.9: Obtenção do *Security Token*

O pedido HTTP à DDI API tem a seguinte estrutura:

```
GET /<tenant>/controller/v1/test_090620 HTTP/1.1 HTTP/1.1
Host:  https://device.eu1.bosch-iot-rollouts.com
Accept:  application/hal+json
Authorization:  TargetToken <Security Token>
```

O *tenant* é uma credencial do serviço subscrito. Assim, no menu *Service Subscriptions* procura-se o serviço em causa, clica-se em *Show Credentials* e procura-se pela credencial *tenant*.

Figura B.10: Obter a credencial *tenant* a partir do menu *Service Subscriptions*

Depois de efetuado o registo, o estado do dispositivo mudou de *UNKNOWN* para *REGISTERED*.

Figura B.11: Dispositivo registado

B.3 Criar uma atualização

Para criar uma atualização, efetuam-se as seguintes etapas:

- Criar um *Software Module*;
- Fazer *upload* de um *Artifact* (ficheiro);
- Criar um *Distribution Set*;
- Atribuir o *Software Module* ao *Distribution Set*.

B.3.1 Criar um *Software Module*

Para criar o *Software Module* começa-se por escolher o menu *Upload Management*, clicando-se depois no ícone +.

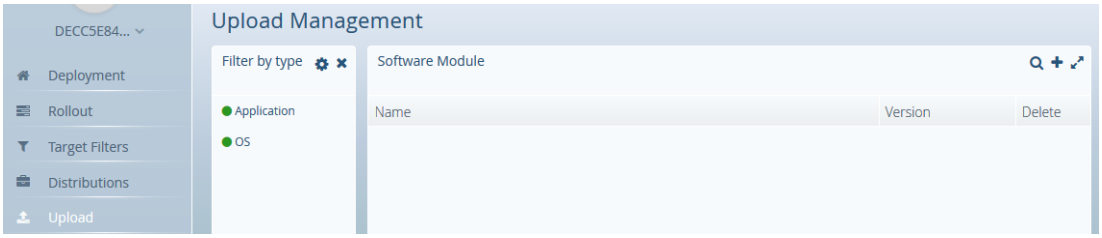

Figura B.12: Criação de um *Software Module* (1/2)

Clicando no ícone +, abre-se uma janela e aí seleciona-se o tipo do *Software Module* e insere-se o seu nome e a versão. Adicionalmente pode também inserir-se o fornecedor e uma breve descrição do *Software Module*.

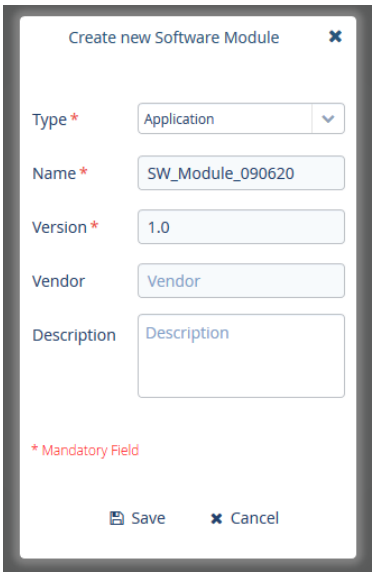

Figura B.13: Criação de um *Software Module* (2/2)

De seguida, faz-se o *upload* do ficheiro para o *Software Module*, clicando no botão *Upload File* e escolhendo o ficheiro ou arrastando-o para a área destinada.

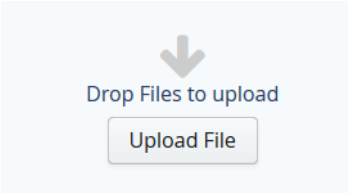

Figura B.14: Realizar o *upload* do ficheiro que contém a atualização

B.3.2 Criar um *Distribution Set*

Para a criação do *Distribution Set* escolhe-se o menu *Distributions Management*, clicando-se depois no ícone +.

Figura B.15: Criação de um *Distribution Set* (1/2)

Clicando no ícone +, abre-se uma janela e aí seleciona-se o tipo do *Distribution Set* e insere-se o seu nome e a versão. Adicionalmente pode também inserir-se uma breve descrição.

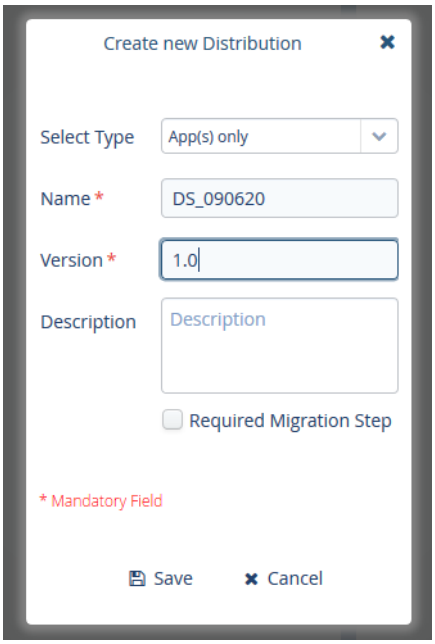

Figura B.16: Criação de um *Distribution Set* (2/2)

B.3.3 Atribuir o *Software Module* ao *Distribution Set*

Ainda dentro do mesmo menu, atribui-se o *Software Module* ao *Distribution Set*. Para realizar a atribuição apenas é necessário arrastar o *Software Module* para o *Distribution Set* e soltar.

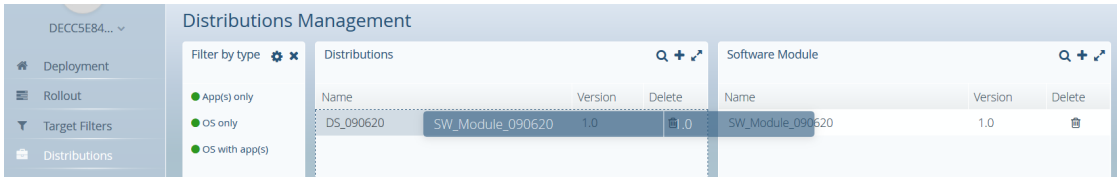

Figura B.17: Atribuir o *Software Module* ao *Distribution Set*

Irá abrir-se uma janela para confirmar a atribuição. Para terminar a atribuição clica-se em *OK*.

Clicando num *Distribution Set* é possível ver a verde os *Software Modules* associados a ele.

Figura B.18: Verificar os *Software Modules* atribuídos ao *Distribution Set*

B.4 Atualizar um dispositivo

A atualização de um dispositivo é composta por duas etapas:

- Atribuir um *Distribution Set* a um dispositivo;
- Efetuar a atualização do lado dispositivo.

B.4.1 Atribuir um *Distribution Set* a um dispositivo

A atribuição do *Distribution Set* ao dispositivo é realizada no menu *Deployment Management*. Para efetuar a atribuição apenas é necessário arrastar o *Distribution Set* para o dispositivo e soltar.

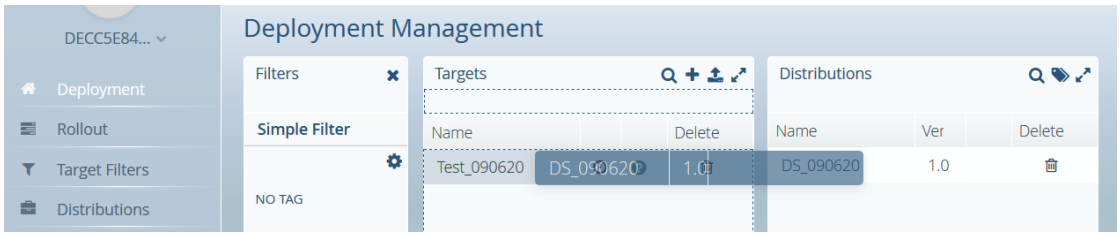

Figura B.19: Atribuir um *Distribution Set* a um dispositivo;

Irá abrir-se uma janela para confirmar a atribuição. Para terminar a atribuição escolhe-se a opção *Forced* e clica-se em *OK*.

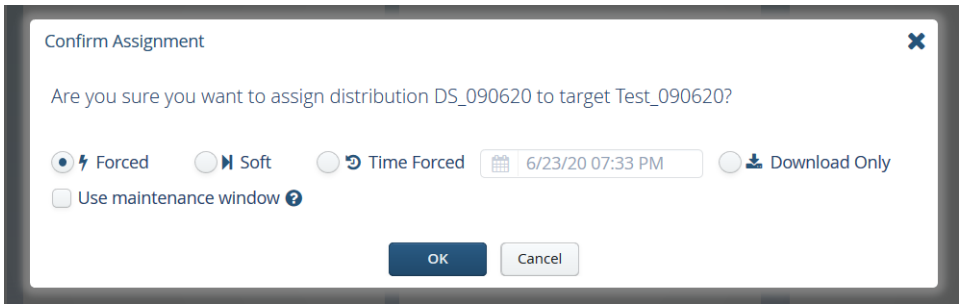

Figura B.20: Definir algumas características da atribuição e efetuá-la

Terminada a atribuição, o estado do dispositivo passa de *REGISTERED* para *PENDING*.

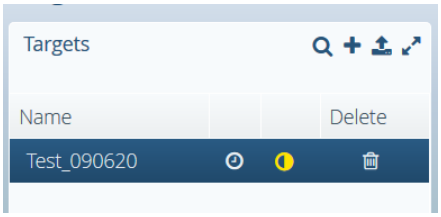

Figura B.21: Estado do dispositivo depois da atribuição

B.4.2 Efetuar a atualização do lado do dispositivo

A atualização do lado do dispositivo é composta por 4 etapas:

- Verificar se existem atualizações atribuídas a ele;
- Obter informações sobre a atualização;
- Efetuar o *download* do *Artifact*;
- Enviar *feedback* com o sucesso ou não da instalação.

Todas estas etapas são realizadas através de pedidos HTTP à *Direct Device Integration API*.

Estas etapas e pedidos estão explicados detalhadamente na Secção 5.3.2.

Caso a instalação tenha tido sucesso, o dispositivo fica atualizado e o seu estado passa de *PENDING* para *IN_SYNC*.

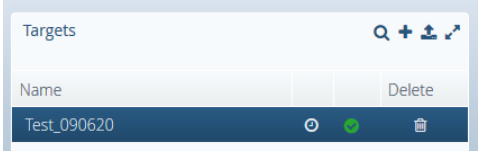

Figura B.22: Estado do dispositivo depois de efetuar a atualização