

# IMPLANTAÇÃO DE MÓDULO IOT EM PÁS CARREGADEIRAS NA ARCELORMITTAL TUBARÃO\*

Leandro Ramos Rodrigues<sup>1</sup>  
Sérgio Valle Júnior<sup>2</sup>  
Jorge Emil João da Penha Júnior<sup>3</sup>

## Resumo

Esse artigo apresenta a estratégia adotada pela ArcelorMittal Tubarão referente extração de dados de pesagens realizadas pelas pás carregadeiras de forma automática. Para isto, foi desenvolvido um hardware que coleta e envia os dados por meio de uma comunicação de dados 3G/4G para o ambiente de nuvem da Microsoft por meio do produto *Microsoft Azure IoT Hub*, após isto os dados são enviados para o sistema de balanças rodoviárias da ArcelorMittal Tubarão. A Implantação da solução trouxe uma redução de 17% do tempo do recurso da contratada e 9% do tempo do recurso próprio, além de aumentar a confiabilidade do processo de consolidação dos dados de pesagem das pás carregadeiras.

**Palavras-chave:** Raspberry; Microsoft Azure; Pesagem; Pá Carregadeira; IOT; Gateway.

## IMPLEMENTATION OF IOT MODULE IN WHEEL LOADER IN ARCELORMITTAL TUBARÃO

### Abstract

This article presents the strategy adopted by ArcelorMittal Tubarão regarding the extraction of weighing data from the loaders automatically. For this, hardware was developed that collects and sends data through 3G / 4G data communication to the Microsoft cloud environment through the Microsoft Azure IoT Hub product, after which the data is sent to the scales system ArcelorMittal Tubarão. The implementation of the solution led to a reduction of 17% in the contractor's resource time and 9% of the time of the own resource, besides increasing the reliability of the consolidation process of the weighing data of the wheel loaders.

**Keywords:** Raspberry; Microsoft Azure; Weighting, Wheel Loader; IOT; Gateway.

<sup>1</sup> Bacharel em Ciência da Computação pela UFMG, Pós-Graduado em Redes de Telecomunicações, UFMG e Engenharia Elétrica com ênfase em Sistemas Inteligentes para Automação, IFES. Especialista Desenvolvimento em Automação e Instrumentação. ArcelorMittal Tubarão. Vitória.

<sup>2</sup> Bacharel em Engenharia Elétrica com Habilitação em Computação pela Faculdade Novo Milênio, Pós-Graduado em gerência de projetos, Universidade de Vila Velha e Engenharia Elétrica com ênfase em Sistemas Inteligentes para Automação, IFES. Especialista Desenvolvimento em Automação e Instrumentação. ArcelorMittal Tubarão. Vitória.

<sup>3</sup> Engenheiro da Automação e Controle, Especialista em Instrumentação e Analista de Sistemas, Etaire TI & Automação, Vitória, ES, Brasil.

## 1 INTRODUÇÃO

A ArcelorMittal Tubarão é uma usina de produção de aço integrada. Isso significa que a unidade realiza todas as etapas do processo: produção de coque, sinter e gusa, fabricação do aço líquido, lingotamento em placas e laminação.

A usina também produz toda a energia que consome, por meio de termelétricas que utilizam os vapores e gases gerados no próprio processo produtivo.

Como parte do dia a dia de uma usina integrada, a manipulação e movimentação de matérias primas e resíduos oriundos do processo produtivo são feitas pela maioria das vezes utilizando pás carregadeiras nos pátios de estocagem. Todo o processo de manipulação dos materiais é realizado por uma empresa terceira, onde seu pagamento está baseado no total de quilogramas manipulados por cada pá carregadeira durante o respectivo mês.

Para contabilização do total de quilogramas manipulados, foram instaladas balanças em cada pá carregadeira que necessitasse do gerenciamento da quantidade de material manipulado. O processo de pesagem destas balanças segue um procedimento sequencial que determina quando uma pesagem é iniciada e finalizada, possibilitando, assim, a medição de forma correta.

Os registros de pesagens são armazenados na memória interna de cada balança, só sendo possível a extração dos dados por meio de um dispositivo de memória *flash*, mais comumente conhecido como *pen drive*.

A necessidade de ter um processo de contabilização e extração de dados de forma mais robusta, segura e automática levou à ArcelorMittal Tubarão desenvolver junto às empresas do ramo de desenvolvimento de *software* e de eletrônica um mecanismo de IOT (*Internet of Things*).

Com a implantação deste dispositivo em cada pá carregadeira esperava-se: Maior confiabilidade dos dados tendo em vista que os dados de pesagens seriam extraídos de forma automática sem a necessidade de manipulação humana, redução do tempo de traslado do operador da pá carregadeira, em que o mesmo se deslocava fisicamente até o escritório para a entrega do pen drive.

### 1.1 Descrição do problema

As pás carregadeiras possuem um sistema de pesagem VEI, que por sua vez é composto de uma interface com operador *Millennium5*, similar ao apresentado na Figura 1.



Figura 1. IHM VEI Millennium5.

O referido equipamento é munido de uma porta *USB* que possibilita exportar um relatório de pesos no formato de uma planilha *CSV* (*Comma-separated values*). Atualmente, após a exportação dos dados, a planilha é encaminhada fisicamente para a área responsável e mensalmente a informação é consolidada e processada para medição. Os relatórios de pesos são gerados três vezes ao dia, sendo:

- Por volta de meia noite;
- Por volta das 07:00h (início do turno matutino);
- Por volta das 17:00h (fim do turno matutino ou início do turno noturno).

Por se tratar de uma planilha *CSV*, o arquivo não possui nenhum tipo de chave de acesso ou criptografia nas informações.

## 2 DESENVOLVIMENTO

### 2.1 Visão geral

Visando uma solução de baixo custo, a principal premissa deste trabalho é a criação de um mecanismo que seja possível se integrar com a infraestrutura já existente.

Uma primeira abordagem estudada consistia em realizar o uso de um computador de placa única e de tamanho reduzido (*Raspberry Pi 3*) que emulasse um *storage drive* e recebesse os arquivos exportados pelo sistema de pesagem. Entretanto, observou-se uma massiva complexidade nessa emulação, combinado com o fato de que, em fase de testes, os arquivos ainda seriam armazenados em um *pen drive* por garantia, então a ideia de emulação foi abandonada, mas o uso do *Raspberry Pi 3* foi mantido.

A ideia adotada consistiu no uso compartilhado de um *pen drive* pelo sistema de pesagem e pelo *Raspberry Pi 3*. Para possibilitar essa manobra, um circuito de comutação foi construído e ligado ao pino digital do *Raspberry Pi 3*, que controla a comutação. O empecilho dessa solução encontra-se apenas no fato de não poder realizar a exportação no instante de transmissão, entretanto, como esse tempo de transmissão é relativamente baixo, algo menor que 05 segundos, não se torna um problema crítico para a solução.

Uma vez com a posse dos relatórios de pesagens, o *Raspberry Pi 3* responsabiliza-se por transmiti-los, entretanto, se o fizesse diretamente para algum servidor interno, isto implicaria em uma falha de segurança. Sendo assim, a solução adotada consistiu em transmitir os relatórios para um ambiente *cloud*, em que foram utilizados os recursos de armazenamento da *Microsoft Azure*. O principal motivador dessa escolha se fez pelo já uso deste ambiente em outras soluções, o que permitiu o desenvolvimento e teste sem custo por esse serviço, além de suporte técnico. No entanto, a plataforma em nuvem, assim como outras alternativas no mercado, é repleta de recursos que, apesar de não serem usadas nessa solução especificamente, ficarão disponíveis para implementações e melhorias futuras. Alguns exemplos como: microserviços, vinculação com banco de dados (estruturados e não estruturados), fácil implementação de aplicações de monitoramento (*dashboard*), análise de padrões e treinamento de inteligência artificial para tomadas de decisão, entre outros recursos.

Um servidor intermediário localizado dentro da rede interna da ArcelorMittal Tubarão, devidamente configurado para garantir os critérios de segurança, realiza o *download* e retransmite os arquivos para o sistema de nível 2, mais conhecido como SISBAL, garantindo o isolamento desses computadores com a rede mundial de computadores, a internet.

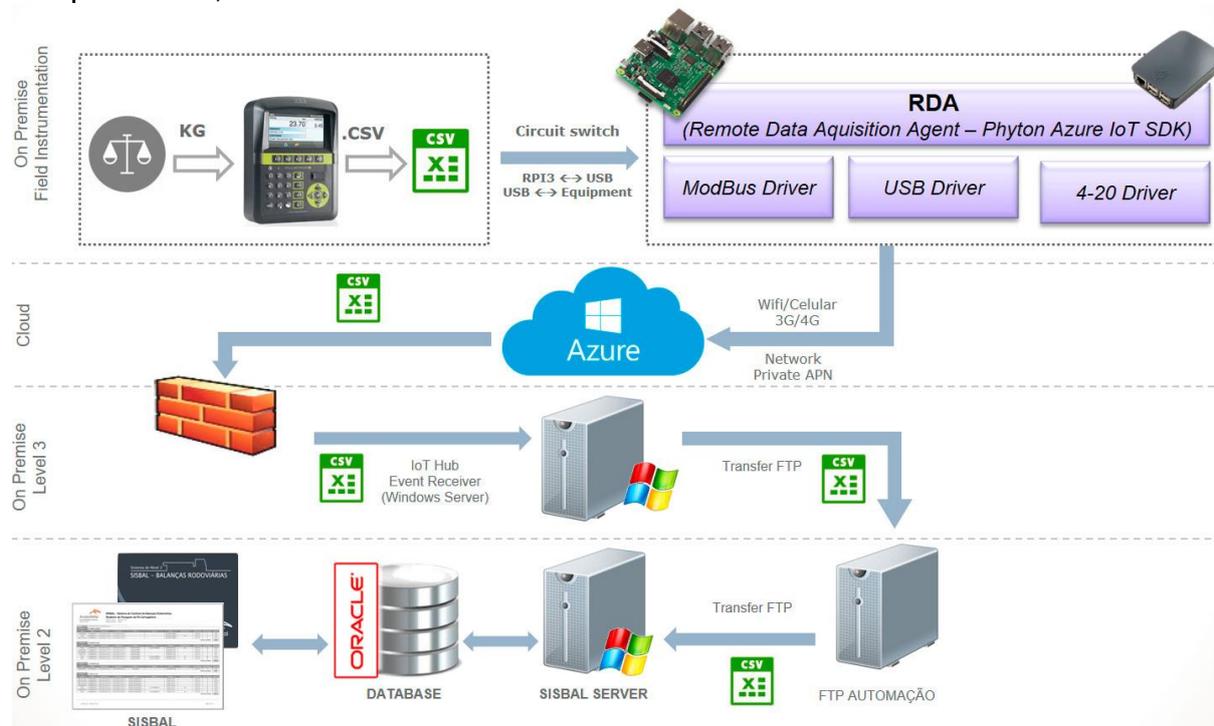


Figura 2. Visão geral da solução.

## 2.2 O Hardware

O *hardware* da solução se baseia basicamente no uso de um *Raspberry Pi 3*. No entanto, esse é alimentado por 5V. Considerando que a tensão operacional da Pá Carregadeira é de 24V, foi necessário realizar a conversão dessa tensão.



Figura 3. Primeira versão do hardware.

Além da conversão de tensão, o circuito de suporte ainda agrega duas funções fundamentais. A primeira é a de interface com o operador, pelo uso de um botão e dois *LEDs* de sinalização. A segunda, a de permitir a comutação do acesso ao *pen drive*, hora pela balança, hora pelo *Raspberry Pi 3*.



Figura 4. Hardware em funcionamento.

A Figura 5 apresenta um diagrama de blocos que esclarece o funcionamento desse circuito de suporte.

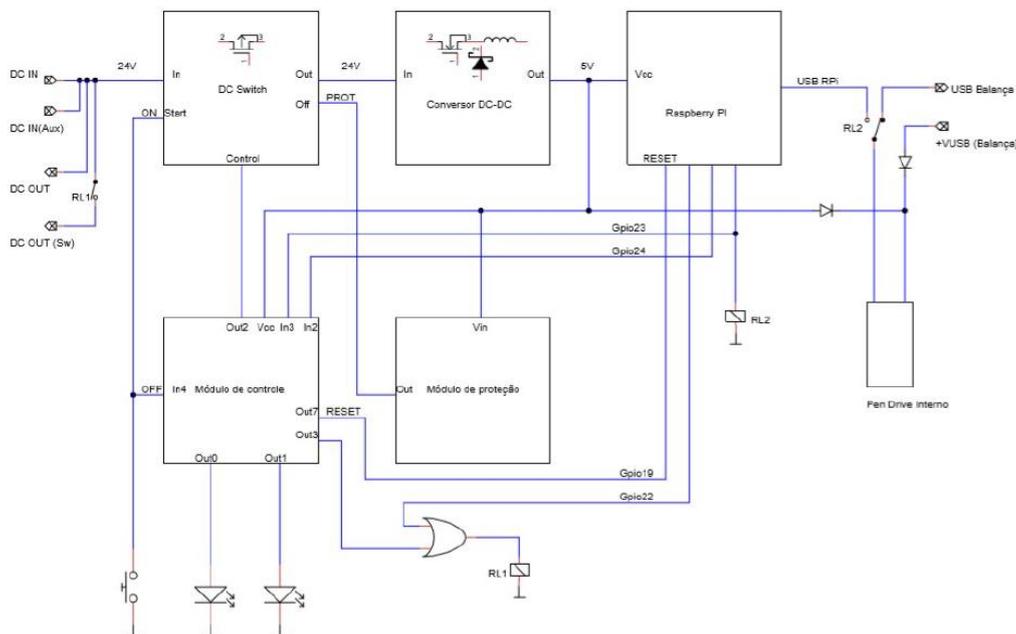


Figura 5. Esquemático do Circuito de Apoio.

As interfaces à esquerda do circuito fazem referência aos *plugs* presentes no entorno da caixa. O 'DC IN', localizado na parte superior da caixa, refere-se à entrada de alimentação 24V proveniente da pá carregadeira. O intuito consistiu em permitir o uso do mesmo cabo que já alimentava a balança.

Para possibilitar a alimentação da balança, já que sua alimentação passaria a ser utilizada pelo sistema embarcado da solução em questão, foi colocada uma

derivação direta de 'DC IN', levando um cabo responsável por alimentar a balança. Esse cabo é indicado no esquemático pela interface 'DC OUT' e localiza-se ao lado do *plug* 'DC IN' na parte superior da caixa.

Tanto para 'DC IN' como para 'DC OUT', foi feito o uso de um *plug* circular M12 de 4 vias similar ao da alimentação da balança, permitindo a compatibilidade. Como esse *plug* não é proprietário (uso/fornecimento exclusivo pelo fabricante da *Millennium5*), ele não é facilmente encontrado no mercado local. Para a construção do protótipo de teste, o *plug* usado foi removido de uma balança defeituosa.

O bloco identificado como 'DC Switch' consiste em um grupo de componentes eletrônicos que realiza a manobra da tensão de alimentação até o conversor DC-DC. Ele é excitado por um pulso em sua *interface* de "Start", proveniente do botão de interface com o usuário.

Quando o pulso é recebido, o sinal de 'In' é direcionado para 'Out', que alimenta o conversor DC-DC. Mesmo que dure poucos milissegundos, esse pulso já será o suficiente para que a saída do Conversor DC-DC alimente o módulo de controle que, por sua vez, enviará um sinal à *interface* "Control" de 'DC Switch', fechando o contato e garantindo, então, que o circuito permaneça ligado.

Para uma saída de 5V, a entrada não deve ser menor que 8V (apesar do *datasheet* do módulo informar um offset de 1,5V, a faixa de tensão entre 6,5V e 8V torna a saída bem ruidosa e por isso desaconselhável). A corrente máxima de saída é de 3A, mas operacionalmente é recomendado considerar como corrente máxima (constante) de 2A, lembrando que somente o *Raspberry Pi* já consome 1,5A, mesmo em *underclock*.

A maioria desses módulos são munidos de um resistor regulável (*trimpot*) para o ajuste da tensão de saída. No caso desse trabalho em específico, este componente foi substituído por um par de resistores. No caso da replicação em série dessa solução, é fortemente aconselhável que seja tomada a mesma medida, ou que seja utilizado um módulo conversor DC-DC com saída 5V (não ajustável).

O bloco chamado '*Raspberry Pi*' consiste em um computador de placa única de tamanho reduzido desse mesmo nome.

O bloco chamado de 'módulo de proteção' consiste em um grupo de componentes eletrônicos responsáveis em mapear surtos na tensão de 5V e disparar um pulso na *interface* "Off" de 'DC Switch' na ocorrência de sobretensão. Resumidamente, é composto por um comparador analógico que tem uma tensão de referência regulada por meio de um resistor variável (*trimpot*). Essa tensão foi regulada para 5,20V, considerando que a documentação oficial do *Raspberry Pi 3* usado indica tensão máxima de alimentação em 5,3V.

O bloco chamado de 'Módulo de Controle' consiste em um microcontrolador PIC16F84 responsável em realizar a gestão de todo circuito de suporte. Uma vez energizado, imediatamente ele liga sua *interface* "Out2", enviando um sinal para a *interface* "Control" de 'DC Switch', garantindo o contato e o pleno funcionamento do circuito.

As interfaces “In4”, “Out0” e “Out1” do ‘módulo de controle’ encontram-se ligados aos componentes da tampa da caixa, sendo um botão, um LED verde e um LED vermelho.

Uma vez com o sistema em funcionamento, o LED verde irá piscar em uma frequência de 1Hz (80% Ativo). O LED vermelho sempre será mantido apagado, até que o *pen drive* seja comutado para acesso pelo *Raspberry Pi 3*, nessa situação o LED vermelho piscará na mesma frequência do LED Verde.

Caso o botão seja mantido pressionado por três segundos, o ‘Módulo de Controle’ enviará um sinal de controle ao *Raspberry Pi 3* fazendo com que esse seja desligado. Nesse período o LED verde piscará com frequência de 1Hz (50% ligado). Caso o botão seja mantido pressionado por oito segundos, o sinal entre “Out2” e “Control” será desativado, fazendo com que todo o circuito seja desligado instantaneamente.

Ainda analisando o ‘Módulo de Controle’, a interface “In3” monitora o estado da saída do *Raspberry* “GPIO23”, que é responsável em realizar a comutação de acesso ao *pen drive* pela atuação sobre o relê RL2. A interface “In2” monitora o estado da saída do *Raspberry* “GPIO24” que é usada como indicação de que o *Raspberry* está ligado. Dessa forma, esse sinal é usado para identificar que o *Raspberry* foi desligado e desligar o circuito de apoio após isso.

Mesmo na situação em que não foi solicitado seu desligamento, caso identificado que este ficou desligado por muito tempo, o ‘Módulo de Controle’ realiza o desligamento do circuito de apoio.

Finalizando a análise do ‘Módulo de Controle’, a interface “Out7” é responsável em enviar um pulso na entrada “GPIO19” do *Raspberry*, solicitando seu desligamento. E a interface “Out3” pode ser usada para atuação do relê RL1, sendo que esse trabalha paralelamente com a interface “GPIO22” do *Raspberry*, com precedência de ON, ou seja, qualquer um que envie um sinal ativo implicará na atuação do relê e por isso essa ligação é representada no esquemático por uma porta lógica OU.

O bloco chamado ‘*Pen Drive*’ consiste literalmente em um *pen drive*, ou mais especificamente em uma porta *USB* fêmea destinada a receber um *pen drive*. O relê RL2 atua na comutação dos pinos Data+ e Data- entre o *Raspberry* e a Balança. Lembrando que a Figura 5 é apenas um esquemático, vale frisar que o relê RL2 é do tipo DPDT.

Os diodos garantem que o *pen drive* sempre esteja energizado, evitando uma tensão reversa no sinal 5V do conversor *DC-DC* e na *USB* da balança. Para esses foram utilizados diodos de alta eficiência (*Schottky*) com queda de tensão na casa de 0,3V (máxima de 0,4V). Nos testes realizados, esse procedimento não gerou problemas, mas vale lembrar que a norma *USB* explicita tensão de alimentação de 5V. No caso dos *pen drives* usados, a queda de tensão dos diodos não causou problemas, mas em outras aplicações, esse pode ser um ponto de falha e deve ser testado.

Por fim, vale frisar que, apesar de não representado, todas as *interfaces* do *Raspberry* foram protegidas por meio de opto-acopladores, o que faz com que seus GPIO tenham sentido de fluxo, ou seja, as saídas não podem ser convertidas para entrada e as entradas não podem ser convertidas para saída. Além disso, vale frisar que a linha de 5V também recebeu diodos de proteção contra surto de tensão. Após validação do trabalho, resolvemos realizar a industrialização do *hardware* a fim de reduzir seu tamanho e aumentar a confiabilidade do mesmo. Em abril de 2019, o módulo de aquisição ganhou uma nova versão.

A nova versão do módulo de aquisição de dados conta com diversas melhorias, desde o dimensionamento físico até melhoria no kernel do sistema.

O dimensionamento físico do equipamento sofreu uma redução considerável comparado ao primeiro modelo, a necessidade de redução do espaço ocupado pelo equipamento dentro da cabine da pá carregadeira motivou o desenho de uma nova placa de circuito impresso com objetivo de otimizar a disposição dos componentes eletrônicos, proporcionando assim a redução do equipamento com um todo.

Além da parte física, melhorias no kernel do sistema foram realizadas, dentre tais podemos ressaltar a seguintes:

- Possibilidade de realizar o acesso ao sistema via rede *Wi-fi*, por meio de rede do tipo *Hotspot*;
- Aprimoramento de exceções causadas por falta de alimentação elétrica;
- Aprimoramento das configurações do software junto ao sistema operacional do *Raspberry*.

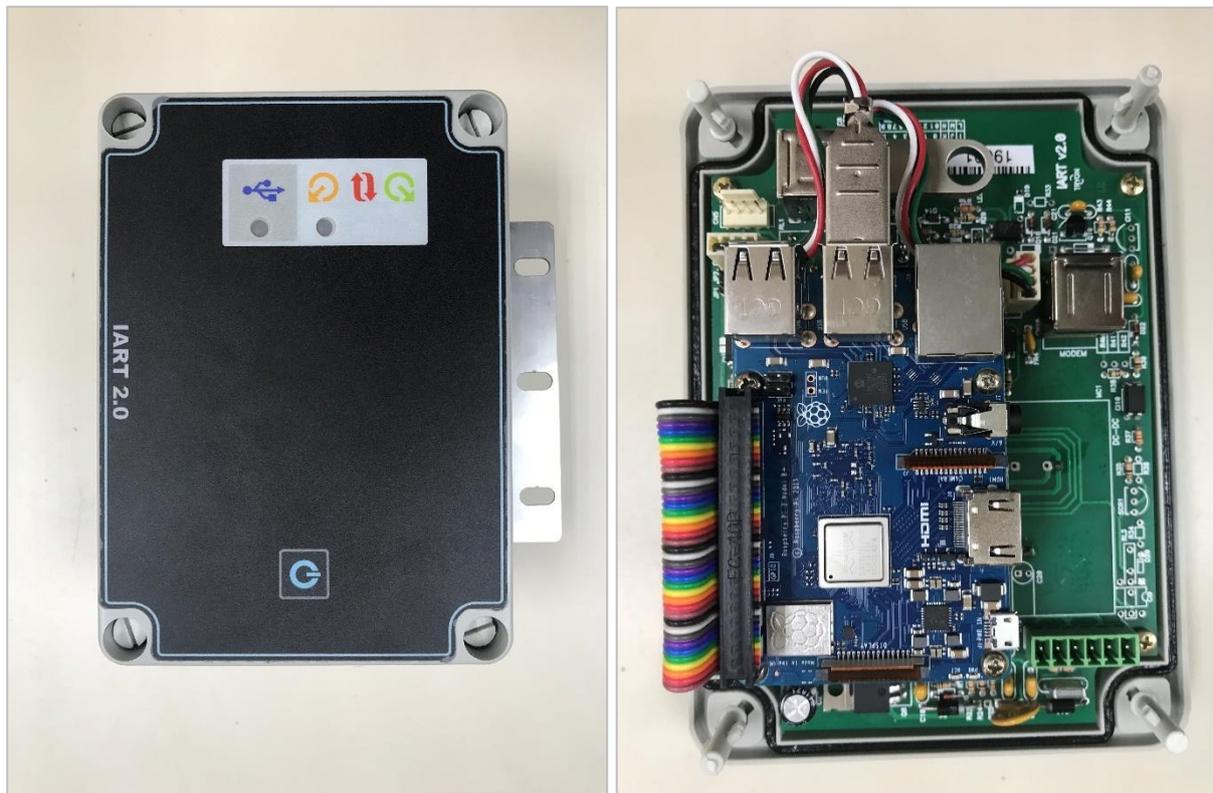


Figura 6. Versão 2 do Hardware.

### 2.3 O Raspberry e o sistema de coleta

A plataforma *Raspberry Pi* utilizado neste trabalho trata-se de um computador minimizado. É uma plataforma SoC (*system on chip*) que originalmente foi concebido para estudos acadêmicos e para promover o ensino de Ciência da Computação básica em escolas. No entanto, sua robusteza tem se mostrado tal que a plataforma tem ganhado nos últimos anos aplicações no mercado industrial.

Neste trabalho foi utilizado o *Raspberry Pi3 model B+*, na qual a plataforma é baseada no microprocessador BCM2837B0 da *Broadcom*, *chip 64bits* de quatro núcleos de 1,4GHz, *córtex A53*, 1GB de memória RAM, Wi-Fi (2,4 GHz e 5GHz), *Bluetooth 4.2*, além de outros recursos.

Como memória de massa, a plataforma faz uso de um cartão SD. Os mantenedores da plataforma recomendam o uso de cartões de 16GB, mas os cartões de 8GB já suportam o sistema. Entretanto, é de extrema importância o uso de cartões Classe 10 para garantir a estabilidade do sistema.

Para o projeto em questão, foi utilizada a imagem do sistema operacional *Raspbian Stretch lite* (versão de 27/06/2018). O *Raspbian* trata-se de um sistema operacional Linux baseado em *Debian*, adaptado para a plataforma *Raspberry Pi*, sendo que a versão *Stretch lite* faz uso dos recursos mínimos e operação via *shell* (não dispõe de *interface gráfica de usuário - GUI*).

Após configuração do *Raspberry*, foram desenvolvidos diversos programas em *Python*, a fim de realizarem e copiarem os arquivos do pen drive e transmiti-los ao ambiente *Microsoft Azure* por meio de uma comunicação de dados 3G/4G.

### 2.4 Microsoft Azure IoT Hub

Parte da arquitetura deste trabalho, foi feito o uso da arquitetura de computação em nuvem padrão (*Microsoft Azure IoT Hub*) que promove um equipamento para integração corporativa de maneira coordenada e segura.

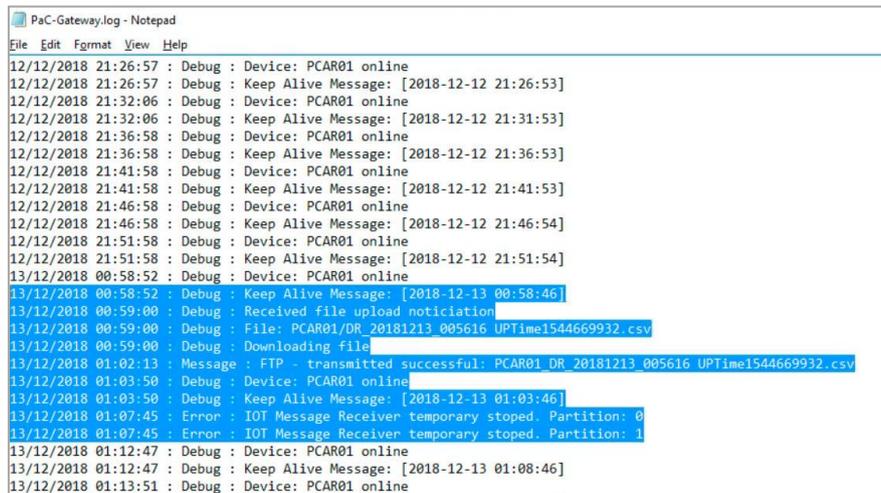
São características do ambiente da *Microsoft Azure IoT Hub*:

- Custos flexíveis de uso da computação em nuvem (crescimento gradual de recursos de acordo com a demanda).
- Promove o monitoramento de variáveis de processos contínuos presentes em diferentes tipos de equipamentos, suportando o acompanhamento da logística interna, a manutenção do transporte ferroviário, operações de carregadeiras e com alto potencial para serem utilizados em demandas adicionais.
- Suporte *multi-hardware* nos dispositivos Edge (x86 / ARM): Digi Router, PCs Industrial *Advantech* e *Raspberry Pi3*.
- O uso de APN privado está sendo implantado para redes 3G / 4G.
- API REST desenvolvida para promover uma interface de ingestão de dados padrão para o sistema PIMS.
- Uso do *Microsoft Azure IoT Hub SDK* (C # .Net e Python) e dos principais recursos de IoT fornecidos pela plataforma do Azure:
  - Gerenciamento de segurança.

- Protocolos de Mensagens (AMQP / MQTT) e Roteamento de Eventos.
- Armazenamento de Blobs do Azure para oferecer suporte a atualizações de software e arquivos.
- Recursos de encaixe do IOT Edge (sob avaliação com suporte da Microsoft).
- Conceitos de Device Twin e Direct Methods, permitindo configurações OTA (over-the-air) e atualizações de software.

## 2.5 Gateway

A aplicação baseada na linguagem C#, foi configurada para rodar em uma máquina virtual dentro da rede interna da ArcelorMittal Tubarão. Ela tem a finalidade de baixar os relatórios do ambiente da *Microsoft Azure*, à medida em que as notificações de upload são recebidas, e encaminhar os mesmos para o servidor de FTP da rede de nível 2 (Automação).



```
PaC-Gateway.log - Notepad
File Edit Format View Help
12/12/2018 21:26:57 : Debug : Device: PCAR01 online
12/12/2018 21:26:57 : Debug : Keep Alive Message: [2018-12-12 21:26:53]
12/12/2018 21:32:06 : Debug : Device: PCAR01 online
12/12/2018 21:32:06 : Debug : Keep Alive Message: [2018-12-12 21:31:53]
12/12/2018 21:36:58 : Debug : Device: PCAR01 online
12/12/2018 21:36:58 : Debug : Keep Alive Message: [2018-12-12 21:36:53]
12/12/2018 21:41:58 : Debug : Device: PCAR01 online
12/12/2018 21:41:58 : Debug : Keep Alive Message: [2018-12-12 21:41:53]
12/12/2018 21:46:58 : Debug : Device: PCAR01 online
12/12/2018 21:46:58 : Debug : Keep Alive Message: [2018-12-12 21:46:54]
12/12/2018 21:51:58 : Debug : Device: PCAR01 online
12/12/2018 21:51:58 : Debug : Keep Alive Message: [2018-12-12 21:51:54]
13/12/2018 00:58:52 : Debug : Device: PCAR01 online
13/12/2018 00:58:52 : Debug : Keep Alive Message: [2018-12-13 00:58:46]
13/12/2018 00:59:00 : Debug : Received file upload notification
13/12/2018 00:59:00 : Debug : File: PCAR01/DR_20181213_005616_UPTime1544669932.csv
13/12/2018 00:59:00 : Debug : Downloading file
13/12/2018 01:02:13 : Message : FTP - transmitted successful: PCAR01 DR 20181213_005616_UPTime1544669932.csv
13/12/2018 01:03:50 : Debug : Device: PCAR01 online
13/12/2018 01:03:50 : Debug : Keep Alive Message: [2018-12-13 01:03:46]
13/12/2018 01:07:45 : Error : IOT Message Receiver temporary stopped. Partition: 0
13/12/2018 01:07:45 : Error : IOT Message Receiver temporary stopped. Partition: 1
13/12/2018 01:12:47 : Debug : Device: PCAR01 online
13/12/2018 01:12:47 : Debug : Keep Alive Message: [2018-12-13 01:08:46]
13/12/2018 01:13:51 : Debug : Device: PCAR01 online
```

Figura 7. Datalog do Gateway – Recebimento de Relatório

Quando inicializado, diversas configurações são realizadas, além de realizar a conexão com a *Microsoft Azure*, mais especificamente ao recurso *IoTHub*. Esse recurso foi configurado em nuvem com duas partições, ou seja, dois subprocessos rodam em nuvem com a finalidade de mapear/atuar em determinadas condições. Neste trabalho, esses subprocessos “escutam” as mensagens dos *devices* e redirecionam as mesmas ao *gateway*.

## 2.6 Disponibilização dos dados

Após processamento dos arquivos pelo *Gateway*, os arquivos são enviados para o SISBAL, sistema de balanças rodoviárias da ArcelorMittal Tubarão, que será o responsável por processar, armazenar e exibir de forma amigável os dados aos usuários finais.

Registro	Tipo	Modo de Pesagem	Data e Hora Início da Carga	Data e Hora Fim da Carga	Cliente	Produto	Nome Target	Peso Total (kg)	Peso Target (kg)	Proc
1	ACENDIMENTO	INCREMENTAL	13/05/2019 17:01:52	13/05/2019 17:01:52	39.ETL	54.LAMA.AF				
2	ACENDIMENTO	INCREMENTAL	13/05/2019 17:13:46	13/05/2019 17:13:46	39.ETL	54.LAMA.AF				
3	ACENDIMENTO	INCREMENTAL	13/05/2019 18:01:20	13/05/2019 19:01:20	39.ETL	54.LAMA.AF				
4	ACENDIMENTO	INCREMENTAL	13/05/2019 19:20:40	13/05/2019 19:20:40	39.ETL	54.LAMA.AF				
5	ACENDIMENTO	INCREMENTAL	14/05/2019 06:22:45	14/05/2019 06:22:45	39.ETL	54.LAMA.AF				
6	ACENDIMENTO	INCREMENTAL	14/05/2019 07:00:34	14/05/2019 07:00:34	39.ETL	54.LAMA.AF				
7	TOTAL	INCREMENTAL	14/05/2019 07:25:38	14/05/2019 09:23:48	39.ETL	53.LAMA.AC		273.640		
8	ACENDIMENTO	INCREMENTAL	14/05/2019 11:15:41	14/05/2019 11:15:41	39.ETL	54.LAMA.AF				
9	TOTAL	INCREMENTAL	14/05/2019 09:24:52	14/05/2019 16:24:16	39.ETL	54.LAMA.AF		363.860		

Figura 8. Exemplo de arquivo gerado pela balança

O SISBAL possui um processo que é executado de forma periódica a fim de verificar se existem arquivos a serem processados. Caso haja, o sistema realiza a leitura dos dados existentes nos mesmos e os armazena em uma base de dados Oracle.

Em paralelo, foi desenvolvida uma interface gráfica para que o usuário realize consultas e emita relatórios conforme desejar.

Ordem Reg. Arquivo	Tipo	Modo	Data Inicial	Data Final	Cliente	Produto	Nome Target
9	TOTAL	INCREMENTAL	2019-05-14 09:24:52.0	2019-05-14 16:24:16.0	39.ETL	54.LAMA.AF	
8	ACENDIMENTO	INCREMENTAL	2019-05-14 11:15:41.0	2019-05-14 11:15:41.0	39.ETL	54.LAMA.AF	
7	TOTAL	INCREMENTAL	2019-05-14 07:25:38.0	2019-05-14 09:23:48.0	39.ETL	53.LAMA.AC	
6	ACENDIMENTO	INCREMENTAL	2019-05-14 07:00:34.0	2019-05-14 07:00:34.0	39.ETL	54.LAMA.AF	
5	ACENDIMENTO	INCREMENTAL	2019-05-14 06:22:45.0	2019-05-14 06:22:45.0	39.ETL	54.LAMA.AF	
4	ACENDIMENTO	INCREMENTAL	2019-05-13 19:20:40.0	2019-05-13 19:20:40.0	39.ETL	54.LAMA.AF	
3	ACENDIMENTO	INCREMENTAL	2019-05-13 18:01:20.0	2019-05-13 18:01:20.0	39.ETL	54.LAMA.AF	
2	ACENDIMENTO	INCREMENTAL	2019-05-13 17:13:46.0	2019-05-13 17:13:46.0	39.ETL	54.LAMA.AF	
1	ACENDIMENTO	INCREMENTAL	2019-05-13 17:01:52.0	2019-05-13 17:01:52.0	39.ETL	54.LAMA.AF	
11	TOTAL	INCREMENTAL	2019-05-13 14:56:38.0	2019-05-13 16:45:40.0	39.ETL	54.LAMA.AF	
10	ACENDIMENTO	INCREMENTAL	2019-05-13 14:55:29.0	2019-05-13 14:55:29.0	39.ETL	53.LAMA.AC	
9	ACENDIMENTO	INCREMENTAL	2019-05-13 14:49:30.0	2019-05-13 14:49:30.0	39.ETL	53.LAMA.AC	
8	POWER OFF	INCREMENTAL	2019-05-13 14:49:15.0	2019-05-13 14:49:15.0	39.ETL	53.LAMA.AC	
7	ACENDIMENTO	INCREMENTAL	2019-05-13 11:22:53.0	2019-05-13 11:22:53.0	39.ETL	53.LAMA.AC	
6	ACENDIMENTO	INCREMENTAL	2019-05-13 10:49:38.0	2019-05-13 10:49:38.0	39.ETL	53.LAMA.AC	
5	ACENDIMENTO	INCREMENTAL	2019-05-13 07:22:25.0	2019-05-13 07:22:25.0	39.ETL	54.LAMA.AF	
4	ACENDIMENTO	INCREMENTAL	2019-05-13 06:52:12.0	2019-05-13 06:52:12.0	39.ETL	54.LAMA.AF	
3	ACENDIMENTO	INCREMENTAL	2019-05-13 06:24:22.0	2019-05-13 06:24:22.0	39.ETL	54.LAMA.AF	
2	POWER OFF	INCREMENTAL	2019-05-13 06:24:07.0	2019-05-13 06:24:07.0	39.ETL	54.LAMA.AF	
1	ACENDIMENTO	INCREMENTAL	2019-05-13 06:19:49.0	2019-05-13 06:19:49.0	39.ETL	54.LAMA.AF	
9	TOTAL	INCREMENTAL	2019-05-12 10:36:25.0	2019-05-12 10:36:25.0	39.ETL	54.LAMA.AF	
8	TOTAL	INCREMENTAL	2019-05-12 09:38:44.0	2019-05-12 13:04:11.0	39.ETL	53.LAMA.AC	
7	ACENDIMENTO	INCREMENTAL	2019-05-12 11:08:58.0	2019-05-12 11:08:58.0	39.ETL	53.LAMA.AC	
6	TOTAL	INCREMENTAL	2019-05-12 08:18:48.0	2019-05-12 08:37:21.0	39.ETL	54.LAMA.AF	
5	ACENDIMENTO	INCREMENTAL	2019-05-12 08:43:45.0	2019-05-12 08:43:45.0	39.ETL	53.LAMA.AC	
4	ACENDIMENTO	INCREMENTAL	2019-05-12 08:13:23.0	2019-05-12 08:13:23.0	39.ETL	53.LAMA.AC	
3	ACENDIMENTO	INCREMENTAL	2019-05-11 17:20:55.0	2019-05-11 17:20:55.0	39.ETL	53.LAMA.AC	
2	ACENDIMENTO	INCREMENTAL	2019-05-11 16:48:16.0	2019-05-11 16:48:16.0	39.ETL	53.LAMA.AC	

Figura 9. Visualização dos dados no SISBAL.

### 3 CONCLUSÃO

Até o momento da elaboração deste artigo (19 de maio de 2019), já haviam sido processados e transferidos um total de 438 arquivos, correspondendo a aproximadamente 5.200 registros em um período de nove meses para uma única pá carregadeira.

A área operacional já deu aprovação para a industrialização de 10 novos dispositivos a serem implementados em 6 máquinas em 2019.

O desenvolvimento do módulo de captura e transferência dos dados possibilitou à ArcelorMittal Tubarão garantir os seguintes ganhos qualitativos:

- Aumento da confiabilidade dos dados, tendo em vista que não há mais manipulação humana dos arquivos de pesagens;

- Redução do risco de perda dos dados de pesagens;
- Disponibilização dos dados de forma amigável por meio de relatórios;
- Possibilidade do controle de acesso aos dados de pesagens via sistema SISBAL e a possibilidade de utilizar a solução para outras aplicações.

Além dos ganhos qualitativos, podemos mencionar ganhos quantitativos como:

- Redução de aproximadamente 17% do tempo gasto pela empresa contratada (equivalente a uma hora e vinte minutos) para realizar o procedimento de extração dos dados em um pen drive e leva-lo fisicamente ao escritório responsável por contabilizar as informações.
- Redução de aproximadamente 9% do tempo gasto pelo efetivo próprio (equivalente a quarenta e dois minutos) para realizar o procedimento de recebimento e consolidação dos dados de pesagem.

Para os próximos passos, evoluções no hardware e no software estão sendo estudadas melhorias como por exemplo:

- Realização de configuração dos dispositivos de forma remota;
- Atualização do “firmware”/software OTA remotamente;
- Monitoramento dos equipamentos utilizando ferramenta Zabbix;
- Inclusão do módulo de GPS para rastreamento do equipamento;

## Agradecimentos

A todas as empresas que participaram no processo de desenvolvimento deste trabalho e aos seus respectivos colaboradores, em especial à TRYON por todo apoio dado durante este trabalho, pela especificação e desenvolvimento do hardware de aquisição de dados (IART - Interface de Acesso Remoto Tryon) e à Etaure no desenvolvimento de todos os aplicativos implantados na solução.

## REFERÊNCIAS

- 1 Introdução ao armazenamento de Blobs do Azure: Tamra Myers. Microsoft Azure. 2019 [acesso em 02 abr. 2018]; Disponível em: <https://docs.microsoft.com/pt-br/azure/storage/blobs/storage-blobs-introduction>.
- 2 Criar uma conta de armazenamento: Tamra Myers. Microsoft Azure. 2019 [acesso em 02 abr. 2018]; Disponível em: <https://docs.microsoft.com/pt-br/azure/storage/common/storage-quickstart-create-account?tabs=azure-portal>.
- 3 Início Rápido: Usar .NET para criar um blob no armazenamento de objeto: Mark Hopkins. Microsoft Azure. 2019 [acesso em 02 abr. 2018]; Disponível em: <https://docs.microsoft.com/pt-br/azure/storage/blobs/storage-quickstart-blobs-dotnet?tabs=windows>.
- 4 O que é o Hub IoT do Azure?: Nicole Berdy. Microsoft Azure. 2019 [acesso em 02 abr. 2018]; Disponível em: <https://docs.microsoft.com/pt-br/azure/iot-hub/about-iot-hub>.
- 5 Início Rápido: Controlar um dispositivo conectado a um hub IoT (.NET): Robin Shahan. Microsoft Azure. 2019 [acesso em 02 abr. 2018]; Disponível em: <https://docs.microsoft.com/pt-br/azure/iot-hub/quickstart-control-device-dotnet>.
- 6 Handling Mode-Switching USB Devices on Linux. [acesso em 02 abr. 2018]; Disponível em: [http://www.draisberghof.de/usb\\_modeswitch](http://www.draisberghof.de/usb_modeswitch).
- 7 Raspberry Pi Documentation: RASPBERRY PI FOUNDATION. [acesso em 02 abr. 2018]; Disponível em: <https://www.raspberrypi.org/documentation/>.