

# Monitorando Dados e Gerenciando Alertas em um Sistema para Controle de Aparelhos de Ar Condicionado

Alan Fernandes, Felipe Rocha, Bruna Soares, Bruno Matias,  
Oswaldo de Oliveira, Sâmela Ferreira, Thais Batista, Everton Cavalcante

Universidade Federal do Rio Grande do Norte (UFRN)  
Natal-RN, Brasil

{alan.fernandes63, brunasdcosta, bmatias.ufrn}@gmail.com,  
felipebarbalho.95@gmail.com, samelabrunaferreira@hotmail.com,  
oswaldo\_oliveira@hotmail.com.br, thaisbatista@gmail.com  
everton@dimap.ufrn.br

**Resumo.** *Sistemas de Internet das Coisas (IoT) são tipicamente compostos por vários dispositivos que podem apresentar falhas de hardware, falta de bateria, indisponibilidade de conexão à rede, etc. Além disso, os dados coletados por esses dispositivos podem se tornar imprecisos ou sofrer anomalias, como indicativos de potenciais falhas ou má calibração de sensores. Para evitar indisponibilidade ou comportamento indesejável no sistema, é essencial monitorar o estado dos dispositivos, detectar anomalias e emitir alertas que permitam rápidas correções. Este artigo apresenta um módulo de alertas para um sistema real de IoT que realiza o controle automatizado de aparelhos de ar condicionado com vistas a contribuir com a redução do consumo de energia elétrica em uma universidade. A arquitetura desse módulo tem como base o modelo de referência MAPE-K para identificar anomalias através das etapas de monitoramento dos dados, análise, planejamento e execução, todas fazendo uso de uma base de conhecimento constituída de regras configuráveis. Experimentos computacionais realizados com o intuito de avaliar a eficácia do módulo de alertas desenvolvido apontaram uma elevada precisão na detecção das anomalias.*

**Abstract.** *Internet of Things (IoT) systems are typically made up of several devices that may undergo hardware failures, lack of battery, unavailability of network connection, etc. Moreover, data collected by these devices may become inaccurate or suffer from anomalies as indicatives of potential failures or poor calibration of sensors. To avoid unavailability or undesirable behavior in the system, it is essential monitoring the state of devices, detecting anomalies, and issuing alerts that allow for quick corrections. This paper presents an alert module for a real-world IoT system that automatically controls air conditioners towards contributing to reduce energy consumption in a university. The architecture of this module relies on the MAPE-K reference model to identify anomalies through data monitoring, analysis, planning, and execution phases, all supported by a knowledge base composed of configurable rules. Computational experiments carried out to evaluate the efficacy of the developed alert module pointed out a high precision in anomaly detection.*

## 1. Introdução

A Internet das Coisas (IoT) preconiza a interação entre múltiplos dispositivos físicos e outros recursos físicos e virtuais através da Internet, tanto provendo informações sobre o

ambiente onde estão inseridos quanto realizando ações sobre ele. Dentre diversos outros aspectos, esse paradigma prevê um número crescente de dispositivos físicos realizando requisições e transmitindo dados, alcançando a ordem de bilhões de unidades conectadas. Com isso, sistemas baseados em IoT necessitam assimilar todas essas características e funcionar corretamente, mesmo em situações de uso intenso.

Outro aspecto relevante nesse contexto diz respeito às condições altamente dinâmicas dos ambientes de IoT, nos quais dispositivos físicos podem se tornar indisponíveis por falhas de *hardware*, falta de bateria, indisponibilidade de conexão à rede, mobilidade de usuários, etc. Além disso, os dados coletados por esses dispositivos físicos podem se tornar imprecisos ou sofrer anomalias pelas mais diversas razões, a exemplo de indicativo de potenciais falhas ou má calibração dos sensores. Se essas questões não forem mitigadas, os dados recebidos pelas aplicações e/ou usuários finais serão errôneos, fazendo com que a tomada de decisão acabe não correspondendo à realidade.

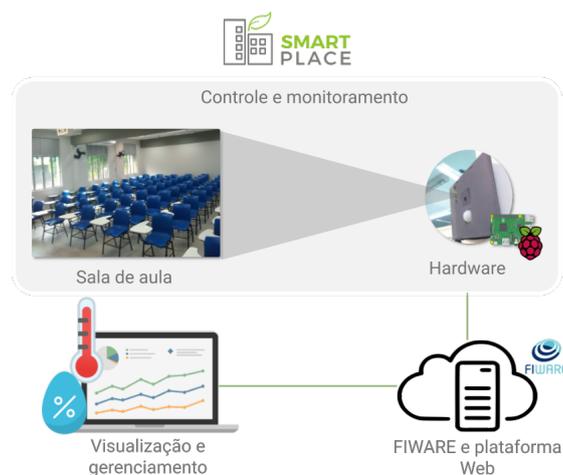
O *Smart Place* é um sistema de IoT desenvolvido com o propósito de contribuir com a redução do consumo de energia elétrica por meio do controle automático de aparelhos de ar condicionado [Rocha et al. 2019]. No *Smart Place*, sensores e câmeras de vídeo coletam dados referentes a temperatura, umidade e presença de pessoas nos ambientes monitorados. Esses dados são utilizados para realizar intervenções sobre os aparelhos de ar condicionado a fim de evitar que eles permaneçam ligados quando o ambiente em questão não está sendo utilizado. No entanto, o *Smart Place* apresenta atualmente lacunas com relação a como lidar com eventuais problemas, tais como desatualização no horário da unidade controladora do sistema, travamento do *software* de controle, falta de energia elétrica ou sensores com valores aferidos que estejam discrepantes das condições normais de operação. Esses problemas, se não resolvidos, podem implicar na gerência inadequada dos aparelhos de ar condicionado, como o acionamento sem a presença de pessoas no ambiente, ocasionando assim desperdício de energia elétrica.

Este artigo tem como objetivo apresentar o desenvolvimento de um Módulo de Alertas para o *Smart Place*, o qual pode inclusive ser adaptado para outros sistemas de IoT. Esse módulo tem como finalidade detectar automaticamente problemas que ocorrem no sistema e notificar os administradores acerca deles. Além do envio da notificação, o Módulo de Alertas persiste informações sobre o alerta em si e dispõe de uma interface que possibilita o gerenciamento dos alertas criados, permitindo aos usuários reportarem problemas que eles mesmo observem nos ambientes monitorados. Tal funcionalidade é essencial nesse contexto, pois podem surgir problemas que ainda não tenham sido mapeados nas regras do Módulo de Alerta.

O restante deste artigo está estruturado da seguinte forma. A Seção 2 apresenta brevemente o sistema *Smart Place*. A Seção 3 descreve o Módulo de Alertas do *Smart Place*, incluindo sua arquitetura e funcionamento. A Seção 4 apresenta uma avaliação do Módulo de Alertas quanto à eficácia na detecção de anomalias. A Seção 5 discute trabalhos relacionados. Por último, a Seção 6 contém considerações finais.

## **2. *Smart Place***

O *Smart Place* realiza o gerenciamento automático de aparelhos de ar condicionado com base nas informações obtidas por sensores de temperatura, umidade, movimento e uma câmera. O sistema liga os aparelhos de ar condicionado ao detectar a presença de pessoas no ambiente ou quando existe previsão de uso do local nos próximos 15 minutos; se essas



**Figura 1. Visão geral do *Smart Place*.**

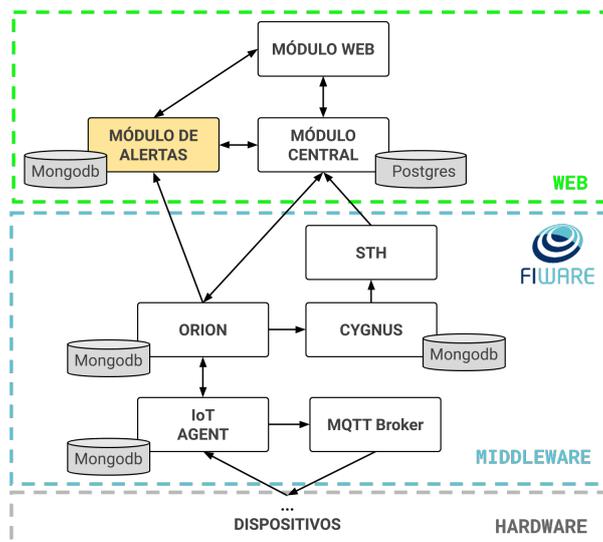
condições não forem satisfeitas, o aparelho é desligado. O *Smart Place* faz parte de um conjunto de aplicações voltadas para campi inteligentes na Universidade Federal do Rio Grande do Norte (UFRN), em Natal, Brasil, e se encontra atualmente implantado em 30 salas de aula da instituição, com perspectivas de expansão. Os resultados até então obtidos com a operação do *Smart Place* mostram que o sistema tem sido capaz de proporcionar de uma redução de aproximadamente 61,8% no consumo de energia elétrica em comparação ao controle realizado de forma manual [Rocha et al. 2020]. A Figura 1 mostra uma visão geral do *Smart Place*.

A Figura 2 ilustra a arquitetura do *Smart Place*. Essa arquitetura é composta por três elementos principais, a saber: (i) um dispositivo de *hardware* responsável por coletar dados do ambiente monitorado e enviar comandos de controle aos aparelhos de ar condicionado; (ii) uma infraestrutura de *middleware* que fornece serviços como gerenciamento de contexto, de dispositivos e de dados, e; (iii) uma plataforma Web através da qual usuários podem visualizar os dados coletados pelos dispositivos e gerenciar os ambientes e aparelhos monitorados.

O dispositivo de *hardware* é o elemento que monitora e controla os aparelhos de ar condicionado, sendo composto pelos sensores de temperatura, umidade e movimento, além da câmera, todos conectados a um Raspberry Pi. O Raspberry Pi também executa um algoritmo de decisão que determina a ação de atuação de acordo com os dados recebidos dos sensores e as imagens captadas pela câmera. Além disso, o dispositivo de *hardware* atua sobre os aparelhos de ar condicionado mediante a emissão de sinais infravermelho.

Em relação à infraestrutura de *middleware*, o *Smart Place* faz uso da FIWARE<sup>1</sup>, uma plataforma genérica de código aberto que fornece diversos componentes (chamados *generic enablers* – GEs) reutilizáveis e interoperáveis para facilitar o desenvolvimento de aplicações em diferentes domínios, além de adotar protocolos padronizados de comunicação. Como mostra a Figura 2, quatro GEs da FIWARE são usados no *Smart Place*, a saber, *Orion Context Broker*, *IoT Agent*, *STH Comet* e *Cygnus*. O *Orion Context Broker* é responsável pelo gerenciamento de entidades de contexto e subscrições de partes interessadas em mudanças de estado nessas entidades. O *IoT Agent* é responsável pela

<sup>1</sup><https://www.fiware.org/>



**Figura 2. Arquitetura do Smart Place.**

comunicação com os dispositivos que utilizam diferentes protocolos de comunicação, entre eles o MQTT, provido pelo *Broker MQTT*. O *Cygnus* vincula eventos de mudança de estado a bancos de dados responsáveis pelo armazenamento de dados dos eventos. Por sua vez, esses dados são geridos pelo *STH Comet* para permitir consultas com base em unidades de tempo (ano, mês, dia, hora, minuto, segundo).

A plataforma Web fornece aos usuários uma interface gráfica para gerenciar os ambientes monitorados e exibe gráficos e tabelas condensados a fim de permitir uma melhor análise das informações disponíveis. Os usuários podem definir parâmetros para a execução do sistema e acionar comandos que são enviados aos dispositivos, como ligar ou desligar um ar condicionado, configurar a temperatura ambiente desejada, etc. A Figura 3 mostra uma tela do *dashboard* do *Smart Place*, o qual exibe estimativas de consumo de energia elétrica ocasionadas pelo uso dos aparelhos de ar condicionado e seu respectivo custo financeiro. O *dashboard* também exibe a quantidade de aparelhos e salas que foram utilizados para fazer as estimativas, bem como os aparelhos que se encontram ligados no momento e os alertas em aberto, isto é, pendentes de resolução.



**Figura 3. Tela do dashboard visualizado na plataforma Web do Smart Place.**

### 3. Gerenciamento de Anomalias

O Módulo de Alertas do *Smart Place* tem como finalidade identificar comportamentos anormais na operação do sistema e alertar os administradores para que eles possam proceder com as correções necessárias. Na UFRN, a operação do *Smart Place* considera os dias e horário de funcionamento das salas de aula da instituição, informações essas que serviram de base para criação das regras de acionamento anormal do aparelho de ar condicionado. O funcionamento normal do acionamento dos aparelhos, de acordo com as informações disponibilizadas pela universidade, seria de segunda a sexta-feira entre 06h e 23h. Dessa forma, o acionamento de um aparelho de ar condicionado fora desse período de tempo deve ser considerado anormal. Além disso, também foi estabelecido um intervalo dos valores de temperatura e umidade considerados normais, conforme estabelecido pelos administradores do sistema através da plataforma Web, de modo que quaisquer dados que não estejam contidos nesse intervalo também deverão caracterizar um comportamento anormal. Além disso, também foram criadas regras que estipulam o limite de tempo para obter resposta dos dispositivos, sendo aplicadas através da verificação do período decorrido desde que os sensores de cada dispositivo enviaram a última atualização para o *Orion Context Broker*. Ultrapassado esse tempo limite, o dispositivo é considerado incapaz de responder, caracterizando assim mais um comportamento anormal.

O Módulo de Alertas foi desenvolvido em conformidade com o IBM MAPE-K [IBM 2003], um modelo de referência para o desenvolvimento de sistemas autônomos em diversos contextos. Conforme ilustrado na Figura 4, o MAPE-K possui quatro etapas, *monitoramento*, *análise*, *planejamento* e *execução*. Essas quatro etapas compartilham uma *base de conhecimento*, fundamental à tomada de decisão. A implementação dessas etapas possibilita que um sistema consiga se autogerenciar em tempo de execução.

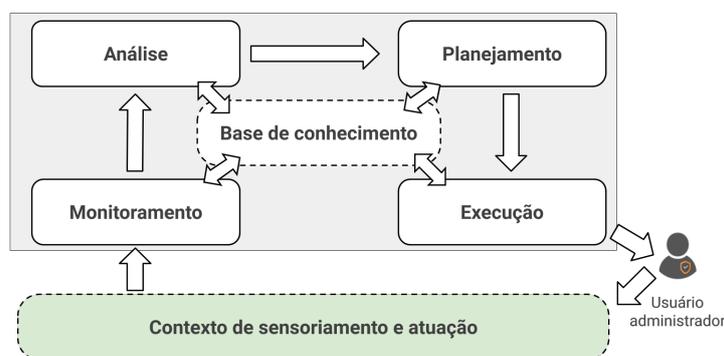


Figura 4. Fases e elementos do modelo de referência MAPE-K.

A etapa de monitoramento é responsável pela coleta de dados relativos ao contexto de sensoriamento e atuação, bem como pela aplicação das regras contidas na base de conhecimento sobre tais dados com a finalidade de verificar possíveis inconformidades. Em caso de suspeita de problemas, o fluxo segue para a etapa de análise, na qual são realizadas inspeções mais complexas sobre as possíveis anomalias e verifica-se se alguma regra está sendo violada. Em caso afirmativo, na etapa de planejamento são criados ou escolhidos os procedimentos (um comando simples ou um fluxo mais complexo de procedimentos) para realização de uma ou mais mudanças desejadas no contexto de sensoriamento e atuação. Finalmente, na etapa de execução, as ações planejadas são efetivamente executadas,

podendo-se valer da utilização de ferramentas que permitem o envio de notificações para os administradores.

Na arquitetura do *Smart Place* (ver Figura 2), o Módulo de Alertas posiciona-se junto ao Módulo Central, conectando-se à infraestrutura de *middleware*, ao Módulo Web e aos usuários. A Figura 5 apresenta a arquitetura geral do *Smart Place* sob a perspectiva do Módulo de Alertas. Na camada de *hardware* estão os elementos a serem monitorados, como sensores, atuadores e dispositivos que fazem parte da aplicação. Na camada de *middleware* estão os GEs da FIWARE, com destaque para o *Orion Context Broker*, que permite o Módulo de Alertas atuar de forma independente do Módulo Central. O Módulo Central contém informações completas dos prédios, salas, aparelhos de condicionado, sensores e atuadores presentes no *Smart Place*, as quais são consumidas pelo Módulo de Alertas. O Módulo Web é uma interface gráfica que possibilita que administradores (responsáveis pelo gerenciamento de informações e manutenções do sistema) e usuários comuns (que podem reportar problemas no sistema) tenham acesso às informações do *Smart Place* e através da qual pode-se executar operações disponibilizadas pelo Módulo Central e pelo Módulo de Alertas. Os componentes que constituem o Módulo de Alertas são análogos às fases do MAPE-K e são descritos a seguir.

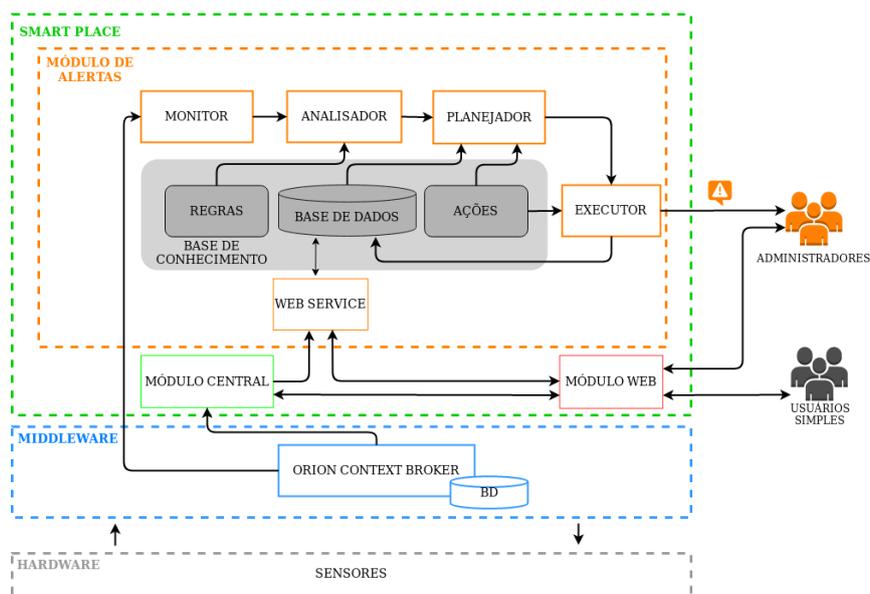


Figura 5. Arquitetura do *Smart Place* sob a perspectiva do Módulo de Alertas.

**Monitor.** O papel do Monitor consiste na sua subscrição em tópicos de seu interesse (aparelho de ar condicionado, umidade, temperatura, movimento e câmera) no *Orion Context Broker* para receber as notificações das mudanças ocorridas nos elementos monitorados e repassá-las ao Analisador. O Monitor também realiza verificações periódicas com o intuito de detectar algum dispositivo ou sensor que esteja sem responder. Tais verificações são necessárias, pois existe uma classe de problemas decorrentes de falta de resposta dos dispositivos, como a possibilidade do aparelho de ar condicionado permanecer ligado por um período anormal, incorrendo em desperdício de energia.

**Analisador.** O Analisador realiza análises das informações provenientes do Monitor com o objetivo de detectar alguma violação de regra. Essas análises são baseadas

em regras que estão contidas na Base de Conhecimento. Caso seja detectada alguma inconformidade, ele repassará os detalhes do problema ao Planejador.

**Planejador.** O Planejador recebe os detalhes do problema que foi detectado pelo Analisador e verifica se é preciso tomar alguma ação. Para verificar se deve ou não tomar ações, ele verifica, na base de dados, se a sala e o problema que foi detectado já possuem alerta em aberto. Caso exista, ele não notificará os administradores e não salvará os dados do alerta para evitar o envio de notificações repetidas e a persistência de dados em duplicidade. Caso não haja alerta com essas condições, o Planejador escolherá as ações de envio de *e-mails* e persistência de alertas na base de dados e repassará essas ações junto com os detalhes recebidos ao Executor.

**Executor.** O Executor recebe os detalhes do problema que formarão o alerta junto com as ações escolhidas pelo Planejador e executa essas ações fornecendo os detalhes do problema. Com isso, dado que tenha sido gerado um alerta e a ação seja notificação dos administradores, o componente irá executar o método de enviar *e-mail* tomando como base o modelo do alerta.

**Base de Conhecimento.** A Base de Conhecimento é o componente que contém todas as regras de negócio e informações do Módulo de Alertas. A Base de Conhecimento é formada por uma *base de dados*, um conjunto de *regras* e um conjunto de *ações*. Na base de dados são persistidos modelos e dados como, por exemplo, os contextos das entidades das salas monitoradas. As regras determinam como o sistema deve se comportar, sendo formadas por árvores de decisão utilizadas pela análise para a detecção de anomalias. Por fim, as ações são medidas a serem tomadas pelo Planejador, incluindo notificar os administradores, criar um alerta e persistir suas informações na base de dados. Mais precisamente, criar um alerta consiste em agregar as informações tendo como base as informações vindas da análise e dados como data, horário, localização do dispositivo e informações dos sensores ou aparelhos de ar condicionado.

As regras atualmente implementadas no Módulo de Alertas são:

- R1: aparelho de ar condicionado com *status* ligado em horários incomuns;
- R2: aparelho de ar condicionado com *status* ligado nos fins de semana;
- R3: sensor de umidade com valores destoantes ou de temperatura com valores discrepantes;
- R4: dispositivo sem enviar atualizações por um período de tempo, e;
- R5: entidade de contexto do tipo sensor sem receber atualizações por um período de tempo.

Ao ocorrer uma mudança nas entidades de interesse do Monitor, o *Orion Context Broker* irá notificá-lo com as informações da mudança. Após o Monitor receber as informações contidas na notificação, ele as entrega ao Analisador e faz atualizações das informações da base de dados. O Monitor também repassa ao Analisador informações de verificações periódicas que ele realiza em sua base de dados. Após receber as informações, o Analisador aplica as regras nas informações. Caso alguma regra seja violada, ele transfere as informações que vieram junto com a regra que foi violada para o Planejador, que verifica se há necessidade de tomar ações. Em caso afirmativo, ele selecionará as ações a serem tomadas e irá repassá-las, junto com as informações do problema, ao Executor. Após receber as informações do problema e as ações recomendadas pelo Planejador, o Executor executará as ações considerando as informações do problema.

A Figura 6 mostra a interface gráfica pela qual usuários podem reportar problemas para os administradores do sistema, informando prédio, sala, data, hora e problema observado. A Figura 6 mostra ainda a tela de configurações, que permite que os administradores ativem e desativem regras e alterem seus parâmetros, como o tempo limite máximo em que um dispositivo pode ficar sem responder até que seja enviado um alerta.

The image shows two side-by-side screenshots of a web application interface. The left screenshot is a form for reporting a problem. It includes dropdown menus for 'Nome do prédio' (Setor III) and 'Nome da sala' (3H4), a date field (04/03/2020), and a time field (15:30). There is a 'Causa' (Cause) dropdown menu with options like 'Dispositivo off-line' and 'Ar condicionado ligado em horário anormal'. Below are optional fields for 'Sensor' and 'Ar condicionado'. A green 'Salvar' button is at the bottom. The right screenshot is a configuration page titled 'Valores das configurações'. It has two columns for 'Tempo limite máximo em horas de um dispositivo offline' and 'Tempo limite máximo em horas de um sensor offline', both set to 1. It also has fields for 'Valor máx de umidade' (90) and 'Valor máx de temperatura' (40). Below is a section 'Ativação/Desativação de regras' with a list of rules, each with a checked checkbox: 'Dispositivo offline', 'Sensor offline', 'Sensor de umidade com valor alto', 'Sensor de temperatura com valor alto', 'Ar-condicionado ligado em horário anormal', and 'Ar-condicionado ligado no final de semana'. A green 'Salvar' button is at the bottom.

**Figura 6. Tela para usuários reportarem problemas (esquerda) e tela para configurações do módulo de alertas (direita).**

A Figura 7 mostra a tela de listagem de alertas. Através dela, os administradores podem visualizar os alertas com *status* de aberto, resolvido e pendente, bem como alterar o *status* do alerta para solucionado ou pendente de solução.

Abertos		Resolvidos	Pendentes
<b>Prédio</b> Setor III	<b>Sala</b> 3D6	<b>Abertura</b> 2020-03-03 15:47:51	^
<b>Causa</b> Dispositivo off-line	<b>Sensor</b>	<b>Arcondicionado</b>	
✓	ⓘ		
<b>Prédio</b> Instituto Metropole Digital	<b>Sala</b> B406	<b>Abertura</b> 2020-03-03 15:47:51	▼
<b>Prédio</b> DIMAp	<b>Sala</b> CONSISTE	<b>Abertura</b> 2020-03-03 15:47:51	▼
<b>Prédio</b> Setor III	<b>Sala</b> 3H1	<b>Abertura</b> 2020-03-03 15:47:51	▼
<b>Prédio</b> Setor III	<b>Sala</b> 3B1	<b>Abertura</b> 2020-03-03 15:47:51	▼

**Figura 7. Tela de listagem de alertas**

#### 4. Avaliação

O Módulo de Alertas do *Smart Place* foi avaliado sob a perspectiva da precisão na detecção das anomalias considerando um conjunto de entrada de dados gerados pelos dispositivos. Nos experimentos computacionais realizados, foi construído um cenário experimental com dados simulados<sup>2</sup>, gerados aleatoriamente e previamente classificados como dados normais ou anômalos dentre as regras R1, R2 e R3 apresentadas na Seção 3. As

<sup>2</sup>A implementação da simulação foi realizada na linguagem de programação Python e está disponível através da seguinte URL: <https://bit.ly/2XWw9hj>.

regras R4 e R5 não foram avaliadas no estudo em questão por não dependerem de requisitos dos dispositivos e sim da falta de informações acerca deles dentro de um período limite. Um conjunto crescente de instâncias, entre 100 e 1.000 notificações foi submetido ao Módulo de Alertas. Para cada uma dessas configurações, foram realizadas 30 execuções a fim de conferir relevância estatística ao experimento. Ao final da execução, foram aferidas medidas de precisão, falsos positivos, falsos negativos, verdadeiros positivos, verdadeiros negativos, se foi emitido alerta ou se ele foi ignorado e dados do tempo de processamento dos alertas. Os experimentos para execução do Módulo de Alertas fizeram uso de um computador com processador Intel® Core™ i7, 8 GB de RAM e sistema operacional Linux Ubuntu 19.04.

A Tabela 1 apresenta o resultado das execuções para as métricas aferidas. É possível observar que não há uma variação significativa de nenhuma das métricas com o crescimento das instâncias, com exceção do número de alertas emitidos e ignorados. Esse crescimento se deve ao fato de que ocorreram mais anomalias e também repetições da mesma anomalia no mesmo ambiente monitorado, as quais são ignoradas pelo sistema para não haver uma sobreposição desnecessária do mesmo problema.

**Tabela 1. Resultado da média de execuções das instâncias**

Configuração	Precisão	Alertas	Alertas ignorados	Tempo médio (ms)
100	100,00%	17,73	2,40	10,28
200	100,00%	32,73	8,93	7,70
300	100,00%	43,43	16,53	7,12
400	100,00%	52,00	27,07	6,81
500	100,00%	59,60	40,67	6,43
600	100,00%	64,80	55,30	6,42
700	100,00%	72,60	68,37	6,38
800	100,00%	74,63	81,77	6,18
900	100,00%	77,40	99,87	5,93
1000	100,00%	81,90	121,20	5,97

A Tabela 2 detalha a precisão com os percentuais de erros categorizados. É importante destacar que os falsos positivos (*FP*) indicam o percentual de dados classificados erroneamente como anomalia, enquanto que os falsos negativos (*FN*) indicam o percentual de dados anômalos não identificados pelo Módulo de Alertas. Por sua vez, os verdadeiros positivos (*VP*) apontam os dados anômalos corretamente detectados e os verdadeiros negativos (*VN*) indicam os dados sem anomalias também identificados corretamente. Dado que a precisão pode ser definida como a razão entre *VP* e a soma de *VP* e *FP*, a precisão média observada alcança 100% devido à ausência de falsos positivos detectados.

Com base nos resultados obtidos, é possível concluir que o Módulo de Alertas do *Smart Place* apresenta máxima eficiência (precisão média de 100%), o que aponta que o sistema mostra-se capaz de identificar os problemas que ocorrem nos dispositivos. Como mostra a Tabela 1, os problemas podem ser rapidamente identificados (6,92 ms, em média), possibilitando a notificação automática da equipe responsável pela manutenção. Apesar de o estudo realizado ter mapeado apenas os problemas conhecidos, é importante destacar que o *Smart Place* oferece um mecanismo manual de cadastro para novas ocorrências, o que possibilita a formulação de novas regras a serem incorporadas ao sistema.

**Tabela 2. Detalhamento da precisão das execuções das instâncias**

Configuração	Precisão	FP	FN	VP	VN
100	100,00%	0,00%	0,00%	20,13%	79,87%
200	100,00%	0,00%	0,00%	20,83%	79,17%
300	100,00%	0,00%	0,00%	19,99%	80,01%
400	100,00%	0,00%	0,00%	19,77%	80,23%
500	100,00%	0,00%	0,00%	20,05%	79,95%
600	100,00%	0,00%	0,00%	20,02%	79,98%
700	100,00%	0,00%	0,00%	20,14%	79,86%
800	100,00%	0,00%	0,00%	19,55%	80,45%
900	100,00%	0,00%	0,00%	19,70%	80,30%
1000	100,00%	0,00%	0,00%	20,31%	79,69%

## 5. Trabalhos Relacionados

Na literatura existem vários trabalhos que definem mecanismos de monitoramento a fim de melhorar, otimizar e evitar falhas em sistemas de IoT. Esta seção discute brevemente alguns desses trabalhos.

O sistema de monitoramento *BigPanDa* [Alekseev et al. 2018] é uma aplicação Web criada para análises de erros em tempo real. O sistema trabalha com cerca de 35 mil solicitações diárias e as classifica em três categorias, erros internos, erros externos e solicitações supérfluas. Na mesma linha, [Renita and Edna Elizabeth 2017] apresentam um sistema de análise e monitoramento de anomalias de servidor através do *Nagios*, uma ferramenta capaz de monitorar máquinas quanto serviços em tempo real, notificando o administrador da rede via SMS ou *e-mail* em caso de ocorrências de falhas.

[Chen et al. 2016] propõem um sistema para monitoramento e detecção de ciberataques em sistemas ciber-físicos críticos, como, por exemplo, sistemas médicos, sistemas de fabricação automatizadas, sistemas de rede elétrica etc. O trabalho proposto é composto por: (i) agentes de monitoramento, implantados nos dispositivos físicos para coletar informações, enviando-as para a nuvem da estrutura; (ii) a nuvem, responsável por armazenar e processar os dados recebidos, e; (iii) o centro de operação, que atualiza as políticas que detectam ameaças e monitora a segurança do sistema. Ao detectar algum tipo de ameaça, o sistema alerta automaticamente os administradores via *e-mail* e exibe uma notificação na página Web da aplicação.

Sob uma perspectiva de infraestrutura, [Stiawan et al. 2016] propõem um sistema para IoT de monitoramento e detecção de anomalias multiplataforma com suporte a dispositivos heterogêneos. O sistema é capaz de determinar as circunstâncias da anomalia em tempo real e quais dessas anomalias são dos perfis de dispositivos de rede, além de oferecer painéis para visualização do tráfego de entrada e saída e exibição de todos os dispositivos com gráficos de gerenciamento visual e estado dos dispositivos. Por sua vez, a proposta de [Yamnuhal et al. 2017] consiste em um sistema proativo para detectar falhas em sistemas distribuídos e notificar os administradores sobre sua existência, monitorando dados de processamento, memória, rede e disco rígido. Para fazer a detecção de falhas, os autores realizaram testes com três tipos de algoritmos de Aprendizado de Máquina, alcançando precisão de 90% para o módulo de detecção de falhas.

De modo geral, os trabalhos mencionados propõem sistemas de monitoramento com o objetivo de alertar usuários e desenvolvedores sobre comportamentos anômalos, analisando os dados em tempo real e classificando erros. Tais trabalhos assemelham-se bastante com o Módulo de Alertas do *Smart Place* visto que ambos agem na detecção de problemas por meio do envio e análise de dados a fim de gerar notificações detalhadas para os desenvolvedores visando a solução das anomalias. No entanto, um diferencial apresentado pelo Módulo de Alertas é a possibilidade de interação manual com o usuário, visto que não é possível mapear a totalidade de problemas que venham ocorrer em qualquer sistema computacional, bem como haver um *feedback* dos administradores acerca do *status* do problema até a sua resolução.

## 6. Conclusão

Este trabalho apresentou a arquitetura e implementação de um módulo de alertas para o *Smart Place*, um sistema de IoT que realiza monitoramento automático de aparelhos de ar condicionado. O módulo de alertas tem como função detectar e notificar automaticamente problemas que ocorrem no sistema. Além disso, esse módulo fornece uma interface Web que permite que os administradores gerenciem alertas e que usuários finais reportem problemas que ocorram nos ambientes monitorados pelo *Smart Place*. A arquitetura do módulo de alertas é baseada no MAPE-K, um modelo de referência para sistemas autônomicos, que permite o gerenciamento contínuo das medições de sensores e atuadores realizadas pelos dispositivos de hardware do *Smart Place*, identificando anomalias através da aplicação de regras e notificando os administradores do sistema para que fiquem cientes o mais rápido possível e possam atuar na correção. A interface Web permite ainda que as regras sejam configuradas pelos usuários, facilitando o que permite que o módulo de alertas seja facilmente adaptado para outros contextos. Os resultados dos experimentos computacionais realizados para o avaliar o módulo de alertas com relação a sua eficácia mostraram que o sistema de fato atende ao seu propósito, com precisão de 100% na detecção de problemas.

Como trabalho futuro, será desenvolvido um estudo acerca dos problemas que precisam de mais atenção, ainda não mapeados pelo módulo de alertas. Com isso, será incorporado ao sistema um mecanismo de correção de forma complementar à identificação de problemas realizada pelo módulo de alertas, para que seja possível o sistema identificar e automaticamente aplicar as correções necessárias. Além disso, serão elaboradas análises utilizando técnicas de Aprendizado de Máquina e Ciência de Dados para extração de informações importantes que podem auxiliar na resolução de problemas.

## Referências

- Alekseev, A., Korchuganova, T., Padolski, S. (2018). The BigPanDA self-monitoring alarm system for ATLAS. In *Proceedings of the VIII International Conference “Distributed Computing and Grid-technologies in Science and Education”*, volume 2267 of *CEUR Workshop Proceedings*, páginas 86–90.
- Chen, Z. et al. (2016). A Cloud Computing based network monitoring and threat detection system for critical infrastructures. *Big Data Research*, 3:10–23.
- IBM (2003). An architectural blueprint for Autonomic Computing. Technical report, IBM.

- Renita, J. Edna Elizabeth, N. (2017). Network's server monitoring and analysis using Nagios. In *Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking*, páginas 1904–1909, USA. IEEE.
- Rocha, F. et al. (2020). Energy efficiency in smart buildings: An IoT-based air conditioning control system. In Casaca, A., Katkoori, S., Ray, S., Strous, L., editors, *Proceedings of the Second IFIP International Cross-Domain Conference on Internet of Things*, volume 574 of *IFIP Advances in Information and Communication Technology*, páginas 21–35. Springer, Switzerland.
- Rocha, F., Santos, L. F., Gameleira Neto, J., Fernandes, A., Batista, T., Cavalcante, E. (2019). Um sistema de gerenciamento e automação de climatização para eficiência energética. In *Anais do XLVI Seminário Integrado de Software e Hardware*, páginas 81–92, Brasil. SBC.
- Stiawan, D., Idris, M. Y., Malik, R. F., Nurmaini, S., Budiarto, R. (2016). Anomaly detection and monitoring in Internet of Things communication. In *Proceedings of the 8th International Conference on Information Technology and Electrical Engineering*, USA. IEEE.
- Yamnuat, K., Phunchongharn, P., Achalakul, T. (2017). Failure detection through monitoring of the scientific distributed system. In *Proceedings of the 2017 International Conference on Applied System Innovation*, páginas 568–571, USA. IEEE.