DESENVOLVIMENTO DE PROTÓTIPO DE UM SISTEMA EMBARCADO DE FECHADURA ELETRÔNICA PARA CONTROLE DE HOMOLOGAÇÃO E ACESSO

Alessandro Muras de Oliveira Pino[1]

Bruno de Sousa Monteiro[2]

Leonardo Augusto Casillo[3]

Resumo: o ambiente universitário é um lugar onde um considerável número de pessoas demandam acesso a diferentes salas de aula e laboratórios, e quando chaves convencionais são utilizadas, surgem situações de dificuldade no gerenciamento, acesso ao claviculário, controle de acesso e riscos de vulnerabilidade. Diante deste cenário, este trabalho tem como objetivo desenvolver um sistema distribuído de controle de acesso para trancas eletrônicas, que faça uso de um sistema embarcado de baixo custo. Para isto, contempla: desenvolver uma arquitetura entre seus componentes, permitir a comunicação segura entre os dispositivos por meio de criptografia em rede sem fio, desenvolver um subsistema para persistência de dados de acesso e histórico e, por fim, validar os artefatos produzidos com usuários potenciais do prédio de Laboratórios de Ciência da Computação da UFERSA.

Palavras-chave: sistemas embarcados; internet das coisas; RFID; fechadura; segurança.

1. INTRODUÇÃO

O ambiente da Universidade é um lugar onde um considerável número de pessoas demandam acesso a diferentes salas de aula e laboratórios. O uso de chaves convencionais gera dificuldades de gerenciamento destas, requer um controle eficiente dos responsáveis pelo claviculário, apresenta vulnerabilidade contra cópias e não provê por si só uma solução de controle de acesso.

Tomando como cenário o prédio de Laboratórios de Ciência da Computação da UFERSA (LCC), sabe-se que, no presente, o controle de entrega e recebimento das chaves aos docentes é desempenhado por funcionários terceirizados, tais como, auxiliares de serviços gerais (ASG) e vigilantes da instituição. Entretanto, há trocas de turno entre os funcionários mencionados, o que requer o gerenciamento das chaves por parte deles, atividade esta passível de riscos, tais como: extravio do objeto; ausência de logística de controle de localização momentânea das chaves; insciência por parte dos docentes e/ou responsáveis momentâneos, qual funcionário está em seu turno, designado a receber e acondicionar as chaves no claviculário da secretaria ao fim das atividades; carência de logística eficaz quanto à verificação de autorização de acesso de docentes e/ou discentes ao ambiente, em que este requisita entrada; e dificuldade de administração de cópias de chave de laboratórios de pesquisa, fornecidas a membros docentes/discentes desta, em que, caso ocorra seu desligamento, impossibilita a revogação do acesso deste, sendo indicado, por questões de segurança, a substituição da fechadura do ambiente.

Durante o estudo da literatura, foi possível verificar que diversas outras instituições se deparam com problemas semelhantes, e algumas soluções foram propostas. No estudo de caso realizado por SCANDOLARA JR (2018) para a Universidade Federal de Santa Catarina, o autor é motivado por problemas semelhantes aos discutidos no presente trabalho. Sua solução baseia-se em uma arquitetura simples em que são utilizados apenas microcontroladores para atender às demandas de acesso. Nos resultados apresentados, o protótipo desenvolvido mostrou-se funcional, entretanto, a limitação de memória do dispositivo de nó principal, que armazena as regras de permissão de acesso e processa as requisições de todos os outros microcontroladores, limita o protótipo nesse quesito. Além disso, o autor sugere como trabalho futuro a utilização de protocolos de segurança para a comunicação de dados, salientando que essa questão foi excluída do escopo de seu trabalho.

[1] Discente da Graduação em Ciência da Computação - Universidade Federal Rural do Semiárido. E-mail: alessandro@ufersa.edu.br

[2] Professor do Magistério Superior - Universidade Federal Rural do Semiárido. E-mail: brunomonteiro@ufersa.edu.br

[3] Professor do Magistério Superior - Universidade Federal Rural do Semiárido. E-mail: casillo@ufersa.edu.br

O protótipo desenvolvido por CARNIEL et al. (2015) teve como objeto de estudo o controle de acesso a ambientes, entretanto o referido trabalho limitou-se em desenvolver o dispositivo baseado nos preceitos da Internet das Coisas, termo proveniente do inglês *Internet of Things (IoT)*. Tal exclusão de uma infraestrutura robusta faz com que o processamento seja centralizado no dispositivo IoT, atribuindo um objetivo a este para o qual não foi feito. O protótipo tem uma dependência de infraestrutura de rede cabeada para interagir com a Internet, visto que em seu *hardware* é empregado um *shield* Ethernet. Há uma série de placas de prototipagem disponíveis com recursos de conexão sem fio no padrão IEEE 802.11b/g/n. Optar por utilizar um *shield* Ethernet requisita um cabo *Ethernet* disponível próximo ao protótipo, demanda um maior espaço para acomodar o *hardware* e poderá onerar o custo final de aquisição dos componentes, visto que são necessárias duas placas: Arduino e *shield*.

No estudo desenvolvido por VERMA E TRIPATHI (2010), eles propuseram um sistema de controle de acesso utilizando RFID passivo, motor de passo e utilização de um banco de dados para concentrar as informações de permissões de acesso e histórico. Seu protótipo apresentou-se funcional, entretanto, o escopo do trabalho limitou-se em desenvolver o *hardware*, sem detalhar a arquitetura de comunicação entre o protótipo e a aplicação de controle e o sistema de persistência dos dados, e preocupações com segurança da informação na troca de mensagens entre os componentes do sistema. Além disso, os autores mencionam a possibilidade de acessar o banco de dados por meio da Internet ou Intranet, o que pode se tornar um ponto de vulnerabilidade a ataques de intrusos, podendo comprometer a segurança do sistema de controle de acesso por meio da liberação de uma *tag* de usuário não autorizada.

Portanto, diante dos problemas e cenários analisados, o presente trabalho tem como **objetivo** desenvolver um sistema distribuído de controle de acesso para trancas eletrônicas, que faça uso de um sistema embarcado de baixo custo (em relação aos produtos comerciais análogos). Entre os objetivos específicos estão: a) conceber e desenvolver uma arquitetura com baixo acoplamento entre seus componentes de *hardware* e serviços que permita suportar requisições simultâneas e escalabilidade horizontal; b) permitir a comunicação segura entre os dispositivos, utilizando criptografia para assegurar a confidencialidade no envio e recebimento de mensagens via rede sem fio; c) desenvolver um subsistema para persistência dos dados da aplicação, tais como, dados de usuário, permissões de acesso, salas e histórico; d) validar os artefatos produzidos com usuários potenciais do LCC/UFERSA.

2. DESENVOLVIMENTO

A seguir, serão detalhadas todas as informações relevantes para o desenvolvimento do trabalho, organizadas nas seguintes subseções: a fundamentação teórica sobre as tecnologias envolvidas na solução do problema é apresentada na seção 2.1; e o método utilizado para alcançar a solução proposta é apresentado na seção 2.2.

2.1. Fundamentação teórica

A utilização do conceito de fechaduras tem sido usado desde a antiguidade. Há vestígios de objetos primitivos utilizados como fechadura em ruínas encontradas em cidades do Egito e da Pérsia. Atualmente, há uma grande diversidade de fechaduras, sendo estas frequentemente usadas para proteger os locais físicos (GOODRICH E TAMASSIA, 2013).

Conforme (COSTA, 2016, p. 9), "A IoT está fornecendo novas maneiras de reagir ao mundo físico por intermédio da capacidade de dispositivos eletrônicos avançados que coletam dados. À medida que novos cenários de aplicação são visualizados, com o auxílio de informações geradas por sensores, novos problemas e obstáculos surgirão". Os microcontroladores são comumente utilizados para prover inteligência em sistemas de controle de acesso, criando assim o conceito de fechaduras inteligentes. O controle de acesso é comumente feito por fechaduras com chaves convencionais. A utilização deste método de controle de acesso em ambientes com muito fluxo de pessoas pode vir a causar problemas, como o extravio das cópias das chaves (SCANDOLARA JR, 2018). Além desses dispositivos, (COELHO, 2017, p.68) destaca que "Cartões *contactless*, tags RFID, digitação de códigos ou impressão digital são formas possíveis de identificar um utilizador perante um sistema de controle de acessos".

Uma fechadura inteligente é um sistema gerenciado por um microcontrolador que, associado a um meio de entrada de informação, pode controlar a abertura de uma porta. (CARNIEL et al., 2015). A entrada de dados feita pelo usuário pode ser realizada por meio de uma identificação por rádio frequência (*Radio-Frequency IDentification* - RFID), sendo este um dos métodos de comunicação entre dispositivos. Uma razão favorável para a adesão de sistemas de automação nos mais diversos setores (industrial, residencial, predial etc.) em substituição à modelos convencionais é

o baixo custo do *hardware* empregado na construção do protótipo. Embora se tenha disponível no mercado tecnologias disponíveis em produtos comerciais aplicáveis ao controle de acesso e segurança predial baseado em RFID, os sistemas citados são predominantemente do tipo *standalone*, onde não dispõem de conectividade com um dispositivo central para o controle e homologação de acesso, sendo tipicamente necessária a atuação individual em cada fechadura de controle de acesso para liberação e/ou revogação de credenciais (SCANDOLARA JR, 2018).

Goodrich e Tamassia (2013, p.76) definem que:

> *A identificação por radiofrequência, ou RFID, é uma tecnologia cujo uso tem crescido rapidamente e que se apoia em pequenos transponders para transmitir informação de identificação via ondas de rádio. Como cartões inteligentes sem contato chips RFID utilizam um circuito integrado para armazenar informação e uma antena espiral para transmitir e receber um sinal de rádio. Como cartões inteligentes, identificadores RFID devem ser usados em conjunto com um leitor ou gravador à parte. Enquanto alguns identificadores RFID exigem uma bateria, muitos outros são passivos e não exigem. O alcance efetivo de RFID varia de uns poucos centímetros a vários metros, mas na maioria dos casos, visto que os dados são transmitidos por ondas de rádio, não é necessário que um identificador esteja na linha de visão do leitor. Essa tecnologia está sendo adotada em uma grande diversidade de aplicações. Como chips RFID operam usando ondas de rádio, eles podem liberar informação sem necessidade de contato físico.*

No contexto dos requisitos de segurança, intrínsecos à sistemas de controle de acesso, a encriptação é caracterizada como um artifício que possibilita a dois participantes estabelecer uma comunicação entre eles de forma confidencial sobre um canal potencialmente inseguro. O uso de criptografia tem como finalidade atender às metas de segurança. Os criptossistemas de chave pública têm como vantagem: evitar o problema do compartilhamento de chave única compartilhada entre o emissor e receptor; possibilidade de compartilhar a chave pública, mantendo apenas chaves privadas secretas; suportar de forma eficiente a comunicação entre vários usuários. As tecnologias *Secure Socket Layer (SSL) e Transport Layer Security (TLS)* utilizam o certificado para verificar a autenticidade do servidor e iniciar um canal de comunicação seguro. No sistema de criptografia RSA, a encriptação e a decriptação são feitas usando potenciação modular, e o Teorema de Euler é utilizado como base, assim como outras propriedades da aritmética modular. A configuração de RSA permite que um potencial recebedor de mensagem crie suas chaves públicas e privadas (GOODRICH E TAMASSIA, 2013).

Ainda no âmbito dos requisitos de segurança, criptografia é um requisito primordial em projetos e implementações de IoT. Os sistemas de criptografia são usados para proteger a comunicação. Dentre os criptossistemas existentes, destaca-se o uso da encriptação de chave pública. A criptografia assimétrica gerencia o sigilo de dados, autentica os participantes e força o não-repúdio (LEA, 2018).

Os sistemas publicar-assinar têm como característica fundamental exercer a função de serviço intermediário, assim como assegurar que as mensagens geradas pelos publicadores sejam recebidas pelos assinantes, de forma eficiente. Também conhecidos como sistemas baseados em eventos distribuídos, estes apresentam um estilo de programação mais desacoplado e reativo oferecido pelos eventos. Os sistemas publicar-assinar tem como atividade vincular as assinaturas às publicações e garantir a entrega das notificações de evento. Estes sistemas são aplicados em uma grande variedade de domínios de aplicação. Entre os exemplos, cita-se: suporte para computação ubíqua, incluindo o gerenciamento de eventos provenientes de infraestrutura ubíqua. Os sistemas publicar-assinar têm duas características principais: heterogeneidade, onde os sistemas podem ser usados para conectar componentes heterogêneos; assíncronos, as notificações são enviadas de forma assíncrona, pelos publicadores que geram eventos, a todos os assinantes que expressaram interesse neles, para evitar que os publicadores precisam estar sincronizados com os assinantes, e os publicadores e os assinantes precisam estar desacoplados. Os sistemas distribuídos apresentam alguns desafios, dentre eles, a tolerância à falhas. Essas falhas podem ser parciais, ou seja, alguns componentes podem falhar, enquanto os demais continuam em execução. O mascaramento de falhas ocorre quando, posterior à sua detecção, esta pode ser ocultada ou tornar-se menos séria (COULOURIS et al., 2013).

2.2. Método

Para alcançar os objetivos estabelecidos, foram realizadas as seguintes atividades (Figura 1):

1. Análise de concorrentes e estado da arte: foram feitas pesquisas bibliográficas a fim de compreender melhor o problema e conceitos relacionados, analisar os principais casos de uso empregados no desenvolvimento de sistemas semelhantes ao proposto neste artigo, e as tecnologias emergentes no contexto da IoT;
2. Análise das tecnologias candidatas: com os resultados da atividade anterior, foi possível eleger as principais tecnologias, evidenciando suas vantagens e desvantagens e, levando em conta o fator custo x benefício, foram definidas as tecnologias adotadas para o desenvolvimento do sistema proposto;
3. Definição da arquitetura: com base nos objetivos específicos definidos, foi desenvolvido um projeto de arquitetura para atender aos requisitos do sistema proposto;
4. Implementação e integração dos componentes: figurou a execução prática das atividades de programação e montagem dos componentes de *hardware* e *software* para atender aos requisitos definidos na etapa anterior de definição da arquitetura;
5. Validação da solução: para validar os artefatos produzidos foram realizadas entrevistas semi-estruturadas com um professor e dois alunos bolsistas, pois são usuários potenciais das trancas eletrônicas.

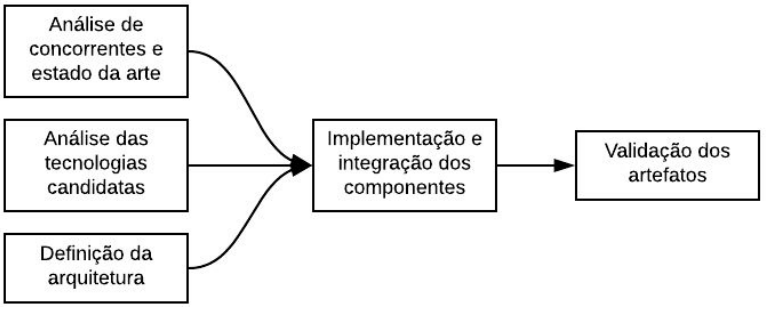

Figura 1. Fluxograma com as atividades realizadas.

3. RESULTADOS

A seguir, serão detalhados os resultados alcançados com a execução do método para contemplar os objetivos deste trabalho. As informações estão organizadas nas seguintes subseções: a arquitetura do sistema desenvolvido é apresentada na seção 3.1; os materiais utilizados na prototipagem do dispositivo IoT são apresentados na seção 3.2, e por fim, a validação do protótipo é apresentada na seção 3.3.

3.1 Arquitetura do sistema

O desenvolvimento do sistema de controle de homologação e acesso distribuído com RFID foi decomposto em quatro subsistemas (Figura 2):

a) **Dispositivo IoT**: foi necessário implementar sua lógica utilizando a linguagem *Wiring*. Trata-se de uma linguagem baseada em C, adaptada para a programação de microcontroladores. Também foi necessária a montagem do *hardware*, bem como verificar a alimentação dos diferentes componentes, onde cada um destes tem uma tensão e corrente elétrica específicas, e um terra (GND) em comum, fazendo com que todo o circuito seja alimentado corretamente, evitando eventuais problemas de origem elétrica;

b) **Mediador MQTT**: também chamado de *broker* MQTT, este atua como servidor distribuído do tipo publicar-assinar, baseado em tópicos, utilizando o protocolo MQTT. É responsável por tratar o recebimento das requisições dos dispositivos IoT de cada sala, e repassar aos clientes inscritos no tópico que, no presente sistema, o servidor de controle está inscrito, bem como encaminhar a resposta do processamento do servidor de controle para o dispositivo IoT, onde este está inscrito em seu próprio tópico;

c) **Controlador**: este atua na comunicação tanto com o Mediador MQTT, realizando o processamento das requisições feitas pelos Dispositivos IoT, comunicando-se com o Banco de Dados (BD) e posteriormente fechando o ciclo de requisição-resposta, enviando mensagens de resposta para os Dispositivos IoT por meio do Mediador MQTT;

d) **Banco de Dados**: este é responsável pela persistência dos dados de usuário, permissões de acesso, histórico de tentativas de acesso, dentre outros dados pertencentes ao modelo do negócio.

Vale ressaltar que a apresentação dos componentes na Figura 2 denota uma visão lógica do sistema. Os servidores explicitados podem estar fisicamente isolados, virtualizados ou sendo executados em uma mesma máquina. Decisões relacionadas à infraestrutura não pertencem ao escopo do estudo deste artigo.

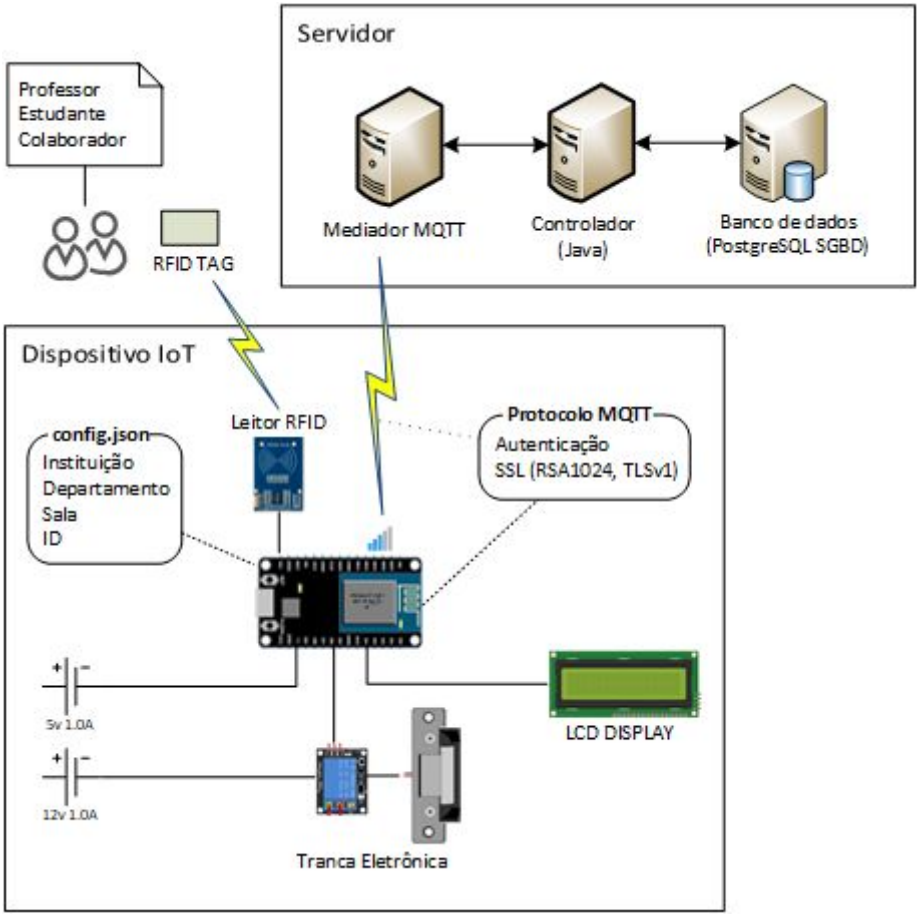

Figura 2. Arquitetura do sistema

A divisão do sistema em subsistemas almejou maior eficiência na implementação, bem como a estratégia de desagregação entre as diferentes tecnologias e linguagens de programação adotadas em cada um destes, tendo em vista promover o baixo acoplamento entre os componentes de *software*, de forma análoga à arquitetura orientada a microsserviços. Esse padrão arquitetural permite que cada componente do sistema seja construído segundo a tecnologia e a infraestrutura que for necessário para ele, não ocasionando qualquer impacto para as decisões arquiteturais e tecnológicas dos demais microsserviços (LUCAS, 2017).

3.1.1 Hardware IoT

O dispositivo IoT é composto por diversos componentes de *hardware* e módulos independentes interligados, originando um protótipo incubido de efetuar o controle físico de acesso às salas. Ao ser energizado por uma fonte de alimentação de corrente contínua de 5V 1A, o sistema embarcado é inicializado, carregando o sistema de arquivos SPIFFS (*SPI Flash File System*). Trata-se de um sistema de arquivos projetado para baixo uso de memória RAM em sistemas embarcados microcontrolados, permitindo criar arquivos e simular diretórios (BATRINU, 2018). O SPIFFS lê o arquivo *config.json* previamente salvo na memória do microcontrolador por uma biblioteca de autoria própria desenvolvida para esta finalidade. O arquivo *config.json* carrega as informações de identificação do dispositivo IoT e localidade em que este está instalado, em um objeto único descrito no formato JSON (*JavaScript Object Notation*), conforme apresentado na Figura 3.

```
{
  "organization": "UFERSA",
  "department": "LCC",
  "name": "Lab Inf 1",
  "id": "001"
}
```

Figura 3. Exemplo de conteúdo do arquivo *config.json*

As informações contidas no arquivo serão utilizadas durante as requisições de acesso feitas ao Mediador MQTT. Após o carregamento do arquivo *config.json*, o protótipo conecta-se à rede sem fio local pré-configurada no dispositivo, viabilizando a comunicação via rede com os demais subsistemas. Posteriormente, é estabelecida uma conexão com o Mediador MQTT, em nível *QoS zero,* ou seja, nesta configuração, a mensagem será enviada uma vez pelo cliente MQTT e não será armazenada (BATRINU, 2018). Por fim, caso a conexão com o Mediador MQTT tenha obtido sucesso, o leitor RFID é inicializado e o protótipo permanece em estado de espera, aguardando a aproximação de uma tag RFID de um usuário, para executar a requisição ao Mediador MQTT, onde este transmitirá a requisição ao servidor de controle inscrito no tópico, a fim de exercer a verificação de autorização de acesso. Caso houver falha na conexão com o Mediador *MQTT*, o protótipo entra em estado de *loop*, diligenciando estabelecer a conexão com tentativas de acesso em intervalos de tempo de cinco segundos.

Ao aproximar uma tag RFID de frequência 13,56 MHz (padrão adotado neste projeto), o dispositivo IoT lê o campo de memória da *tag* RFID de 4 bytes, onde está localizado o valor de *User Identification* (UID) com tamanho de quatro bytes. O UID é retido em memória temporária do dispositivo, e uma requisição no formato JSON é gerada, conforme padrão apresentado na Figura 4a. Tal requisição é enviada ao Mediador MQTT. O dispositivo IoT retorna ao estado de leitura de nova tag RFID, pois o fluxo do recebimento de resposta é uma função *callback* do tipo não bloqueante. Ao receber a resposta gerada pelo servidor de controle, a função *callback* é invocada e decorre o processamento da mensagem, no formato JSON, recebida. A mensagem de resposta dispõe de mais de um padrão, apresentados na Figura 4b e Figura 4c. Caso o atributo *cod_op* (refere-se ao código da operação) seja do tipo *ACK*, o dispositivo IoT entende que a *tag* RFID lida tem permissão de acesso à sala. Um sinal elétrico é enviado ao módulo relé, que fechará o circuito alimentado por uma fonte de corrente contínua de 12V 1A, direcionado ao acionamento da fechadura eletrônica. O módulo relé, incubido de liberar a abertura da porta, executa a sua função por um tempo de 2,5 segundos, tornando a travar a fechadura decorrido o tempo citado. A interface LCD exibe o nome do usuário que teve o acesso liberado no sistema, no instante em que a fechadura eletrônica é destravada, permanecendo por 3 segundos no visor. A informação do nome do usuário é obtida na mensagem de resposta, no atributo *opt*. O referido atributo é exclusivo de mensagens de resposta do tipo *ACK*. Outrocaso, se a mensagem de resposta informar no atributo *cod_op* o valor NACK, o dispositivo IoT entende que a *tag* RFID lida não tem permissão de acesso à sala. O campo *opt* é omitido da mensagem, pois não há informações de usuário a serem exibidas, e a interface LCD exibe uma mensagem informando ao usuário que o acesso foi negado.

```
{
  "id": "001",
  "id_tag": "CC7F5107",
  "cod_op": "REQ"
}
```

a) requisição

```
{
  "id": "001",
  "id_tag": "CC7F5107",
  "cod_op": "ACK",
  "opt": "Alessandro Pino"
}
```

b) resposta ACK

```
{
  "id": "001",
  "id_tag": "CC7F5107",
  "cod_op": "NACK"
}
```

c) resposta NACK

Figura 4. Exemplos de mensagens (de requisição e de resposta) no formato JSON

3.1.2 Mediador MQTT

O Mediador MQTT é um sistema baseado em eventos distribuídos (MUHL et al., 2006). Propicia maior desacoplamento oferecido pelo eventos. Um sistema publicar-assinar, implementado pelo servidor *Mosquitto* utilizado nesse projeto, é um sistema em que publicadores lançam eventos para um serviço e assinantes inscrevem-se em eventos desejados. A tarefa desse tipo de sistema é vincular as assinaturas cadastradas com as respectivas publicações destinadas, e certificar a entrega das notificações geradas pelo evento de publicação (COULOURIS et al., 2013). No

sistema proposto descrito nesse artigo, o Mediador *MQTT* em execução é o *Mosquitto v3.1.1*. Trata-se de uma implementação de código aberto (sob a licença EPL/EDL) de um Mediador MQTT, implementado na linguagem C e, dentre as suas funcionalidades, destaca-se a tolerância à falhas, por meio da característica de *Quality of Service* (QoS) implementada no protocolo MQTT. Este gerencia a retransmissão de mensagens e garante a entrega, tornando a comunicação em redes sem fio mais confiável. O *Mosquitto* foi previamente configurado, por intermédio de um arquivo de extensão CONF, e nele foram inseridas referências para os arquivos de chave pública e privada gerados com o *openssl* (uma implementação de código aberto dos protocolos SSL e TLS), para conferir o requisito de segurança objetivado na arquitetura do sistema. Por fim, devidamente configurado, o Mediador MQTT foi posto em execução. Os dispositivos IoT de cada sala se conectam ao *Mosquitto* e são configurados para enviar mensagens de requisição e inscreverem-se no seu respectivo tópico. O modelo padronizado de tópico é apresentado na Figura 5.

<organização>/<departamento>/<id>

ufersa/lcc/001

Figura 5. Estrutura do tópico do protocolo MQTT

3.1.3 Servidor de controle

O servidor de controle, por sua vez, inscreve-se em um tópico curinga, apresentado na Figura 6, para receber todas as mensagens produzidas. Assim, em um padrão *single thread*, o servidor de controle é suficiente para receber todas as requisições dos dispositivos IoT pertencentes à rede da organização e enviar respostas específicas para o tópico de origem, onde apenas o dispositivo inscrito no tópico em questão receberá a resposta referente à sua requisição. O servidor foi implementado na linguagem Java. A Figura 7 ilustra a atividade de um sistema distribuído do tipo publicar-assinar. Vale ressaltar, que um mesmo cliente do Mediador MQTT pode tanto atuar como publicador quanto assinante quando inscrito em um tópico, dependendo apenas da implementação de ambas as funcionalidades.

<organizaçao>/<departamento>/#

ufersa/lcc/#

Figura 6. Tópico curinga, inscrito em todos os subtópicos correspondentes

O servidor de controle é incubido de receber as requisições de todos os dispositivos IoT advindas do Mediador MQTT, processá-las, constituir uma resposta no formato JSON, e emitir essa informação ao tópico de origem, onde o Mediador MQTT se encarregará de transmitir ao tópico. O servidor de controle, para processar as requisições de acesso, conecta-se ao SGBD PostgreSQL, adotado nesse projeto, e executa uma consulta no banco de dados, armazena de forma temporária para composição da mensagem de resposta, e subsequentemente executa uma inserção no banco de dados, a fim de persistir uma tabela de histórico dos eventos.

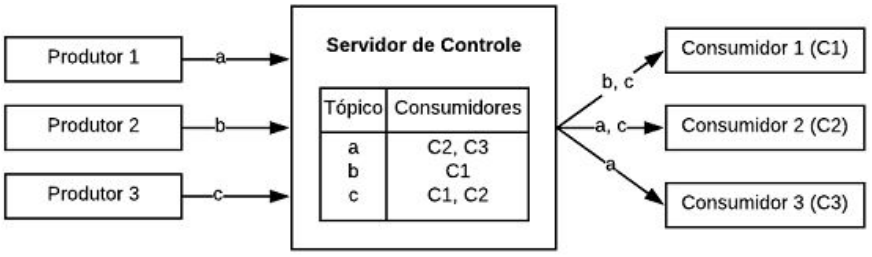

Figura 7. Ilustração de um sistema distribuído do tipo publicar-assinar

3.1.4 Sistema de Gerenciamento de Banco de Dados

O Sistema de Gerenciamento de Banco de Dados (SGBD) adotado nesse projeto foi o PostgreSQL.Trata-se de um poderoso SGBD objeto-relacional de código-fonte aberto que usa e estende a linguagem SQL (PostgreSQL, 2019). O modelo de banco de dados escolhido foi o Modelo Relacional. Há uma tendência de uso de banco de dados NoSQL para IoT devido a diversos casos de uso em que são geradas uma grande quantidade de dados. Embora FÁTIMA E

WASNIK (2016) salientem que o processamento de muitas transações constitui um problema ao lidar com sistemas clássicos de gerenciamento de dados SQL, o cenário relatado não se aplica ao projeto desenvolvido. A modelagem lógica das entidades e relacionamentos desse projeto, apresentado na Figura 8, demonstra um aspecto característico do modelo relacional. O referido modelo de gerenciamento de dados se apresenta na liderança entre os modelos de banco de dados mais utilizados mundialmente. O PostgreSQL, SGBD adotado no projeto, é *open-source*, encontra-se na quarta posição entre os SGBD relacionais mais utilizados, e continua em ascensão (DB-ENGINES Ranking, 2019).

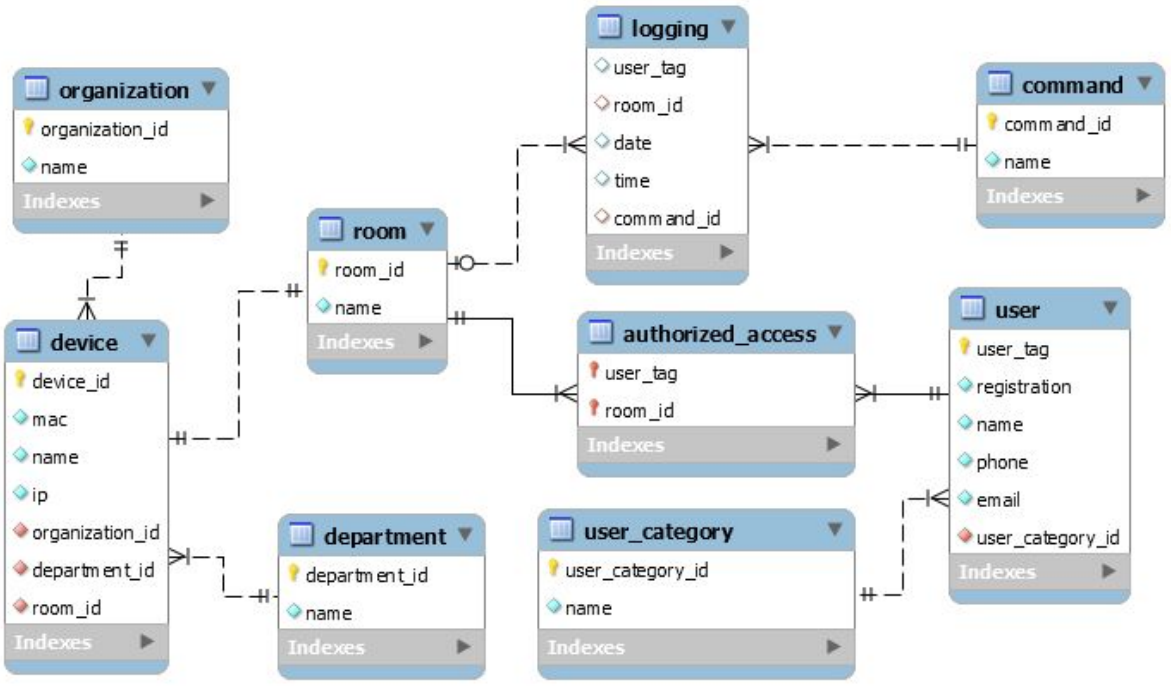

Figura 8. Modelo lógico do banco de dados relacional implementado

3.2 Equipamentos

3.2.1 Placa de prototipagem eletrônica

A placa de prototipagem eletrônica adotada nesse projeto foi a NodeMCU v3. É constituída pelo microcontrolador ESP8266. O microcontrolador em questão dispõe de tecnologia de comunicação wireless padrão IEEE 802.11b/g/n. Em contraste com a placa padrão da plataforma Arduino, o Arduino UNO, o ESP8266 apresentou-se viável por ter maior poder de processamento que o Arduino UNO, o que contribui para o processamento de algoritmos de criptografia sem se tornar um gargalo durante a sua execução. O ESP8266 também facilita para a criação de um sistema de arquivos e diretórios em sua memória interna (podendo dedicar entre 1MB e 4MB), utilizando a tecnologia SPIFFS, utilizada no projeto para armazenar o arquivo de configurações do dispositivo IoT *config.json*. Em contrapartida, o Arduino Uno apresenta uma memória EEPROM de 1KB, de difícil utilização, posto que a biblioteca de leitura/gravação da memória EEPROM preceitua manipular um byte por vez. Ademais, para o Arduino Uno atender ao requisito de comunicação wireless, é imprescindível um módulo ou *shield* para desempenhar tal função. O componente wireless vinculado ao Arduino Uno precisaria de um maior espaço físico para ser acondicionado, bem como oneraria o custo dos componentes. Os demais modelos de placas da plataforma Arduino foram descartados do estudo em questão, pois são de categorias diferentes de placa, com custos incompatíveis para o projeto. A Figura 9 corresponde a placa NodeMCU v3.

Figura 9. Microcontrolador ESP8266 acoplado à placa de prototipagem NodeMCU v3

3.2.2 Leitor de tags RFID

O leitor de tags RFID adotado foi o modelo MIFARE MFRC-522 antena 13,56MHz, compatível com tags da frequência citada. As tags utilizadas no projeto para testes foram dois cartões e dois chaveiros MIFARE 1KB 13,56MHz, todos passivos. A Figura 10 exibe o leitor RFID, à esquerda, seguido pelas tags do tipo cartão, ao centro, e as tags do tipo chaveiro, à direita.

Figura 10. Leitor RFID (MFRC-522); cartões RFID passivos (MIFARE 1KB); chaveiros RFID passivos

3.2.3 Interface Display LCD

O *display* utilizado como interface para o usuário é do tipo LCD (16x2), 16 colunas por 2 linhas, acoplado ao controlador I2C. Esse controlador permite que o *display* seja controlado por meio do protocolo I2C que, dentre as suas vantagens, destaca-se a utilização de apenas dois fios para comunicação (excluindo os fios de alimentação). A Figura 11 apresenta o display LCD utilizado no projeto.

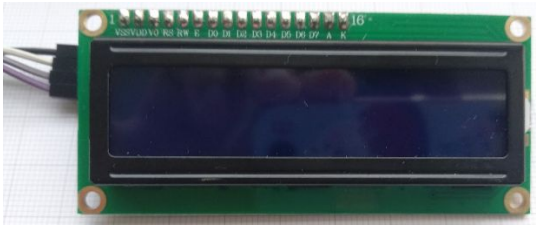

Figura 11. Display LCD 16x2 com módulo I2C

3.2.4 Fechadura Elétrica

O *hardware* para acionamento da fechadura é composto por um módulo relé de um canal, um fecho eletromagnético FE12 Amelco e uma fonte de alimentação de 12V 1A. O módulo relé ao receber um sinal elétrico proveniente do microcontrolador, fecha o circuito, permitindo que o fecho eletromagnético seja alimentado pela corrente proveniente da fonte de alimentação e libere a abertura da porta. A Figura 12 apresenta o módulo relé, à esquerda, e o fecho eletromagnético, à direita.

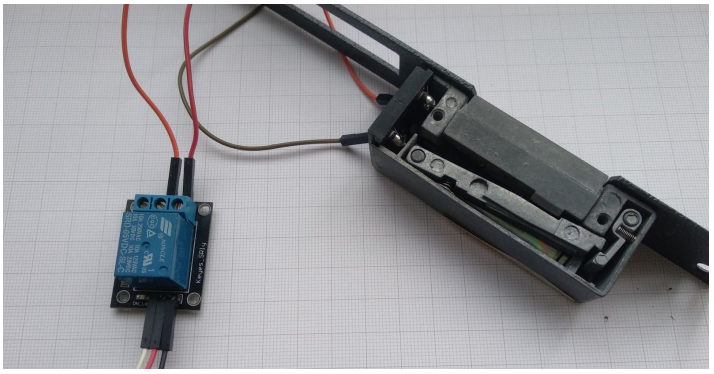

Figura 12 Módulo Relé de um canal; fecho eletrônico Amelco FE12

3.2.5 Computador Servidor

Os três servidores (Mediador *MQTT*, Servidor de Controle e Servidor de BD) foram instalados e executados em uma única máquina com Sistema Operacional Windows 10 Home Single Language, 64 bits, com processador Core i7-7700HQ 2.80GHz, 16GB de memória RAM DDR4 2400MHz, disco SSD M.2 Corsair MP500 NVMe 240GB. O computador em questão apresenta configurações superiores ao requisitos mínimos necessários para execução dos serviços. Este foi adotado por questões de disponibilidade do autor para executar os cenários de teste do sistema desenvolvido.

Os requisitos mínimos para execução dos servidores seguem o maior entre os mínimos requeridos pelo conjunto, que são: CPU Dual Core 64 bits, Sistema Operacional de 64 bits, 1 GB memória RAM (recomendado 2GB), 2 GB de espaço livre em disco. Vale ressaltar que os requisitos de *hardware* foram levantados levando em consideração um único servidor com todos os serviços nele instalados e em execução, desconsiderando os requisitos do sistema operacional instalado na máquina, caso o sistema exigir requisitos superiores ao mencionado. Os servidores podem ser instalados em uma mesma máquina, em máquinas diferentes, ou utilizando o recurso de virtualização de sistemas. Há versões de cada serviço compatíveis com os ambientes Microsoft Windows e GNU/Linux. Por se tratar de decisões relacionadas à infraestrutura, o estudo da melhor configuração não faz parte do escopo deste trabalho.

3.2.6 Custo dos equipamentos

A seguir, na Tabela 1, estão relacionados os itens de *hardware* para montagem do Dispositivo IoT e seu respectivo preço médio no mercado nacional e internacional, respectivamente. No mercado internacional, foi adotado a cotação do dólar no dia 17 de março de 2019, com o valor de R$3,82. Eventuais custos de frete por componentes adquiridos de lojas físicas localizadas em outras cidades não foram considerados.

Componente	Preço de mercado nacional	Preço de mercado internacional
NodeMCU v3 (ESP8266)	R$25,00	R$10,00
Leitor RFID MIFARE MFRC522 com duas tags	R$25,00	R$4,00
Display LCD 16x2 com módulo I2C	R$25,00	R$6,00
Módulo relé um canal	R$10,00	R$5,00
Fechadura elétrica 12V	R$75,00	R$20,00
Fonte alimentação 5V	R$10,00	R$7,00
Fonte alimentação 12V	R$10,00	R$8,00
TOTAL	**R$180,00**	**R$60,00**

Tabela 1. Custo dos equipamentos

Valor total dos componentes para montagem do protótipo, no mercado nacional: R$180,00. Valor total dos componentes, no mercado internacional: R$60,00. Ressalta-se que, comparando os valores total de frete no cenário nacional em relação ao internacional, o frete nacional apresenta um valor superior ao frete internacional (já somadas as taxas de importação), comparação feita em Real. A compra internacional traz como desvantagem o tempo total de espera pelo recebimento dos componentes eletrônicos.

3.3 Validação do sistema

O protótipo do Dispositivo IoT e o computador servidor foram montados e postos em execução no LCC/UFERSA Campus Mossoró, tornando o sistema funcional para testes. O sistema foi testado e validado por dois discentes bolsistas e um docente do curso de Ciência da Computação. Todos os entrevistados aprovaram o sistema quanto aos seguintes quesitos: relevância para o ambiente da Universidade; viabilidade da solução para o problema levantado neste trabalho e facilidade na utilização do sistema. Os questionamentos presentes na pesquisa de validação do sistema foram: "Você acha importante a universidade adotar esse tipo de controle de acesso a ambientes físicos? Por que?"; "Encontrou alguma dificuldade ou surgiu alguma dúvida em como utilizar o sistema? Quais?"; "Quais problemas atualmente você enfrenta para acesso aos laboratórios?"; "A interface com o usuário é amigável?"; "Quais quesitos você melhoraria no sistema apresentado?"; "Em que outro cenário você pensaria em aplicar essa solução tecnológica?"; "Quais benefícios esse sistema pode trazer se implantado no LCC?"; Os relatos produzidos pelos participantes foram analisados e um resumo dos relatos mais significativos é apresentado na Tabela 2.

	Relevância	**Viabilidade**	**Facilidade de uso**
Estudante bolsista 1	Estabelecer um controle eficiente de acesso ao laboratório, eliminar a responsabilidade do funcionário terceirizado por tal incubência. O histórico de acesso torna-se importante para a segurança do patrimônio.	Sistema viável, por ser de baixo custo e funcional. Levantado questionamento acerca de problemas no fluxo de entrada e saída de discentes durante o acontecimento das aulas, devido ao acesso ser liberado pelo docente.	Interface com o usuário aprovada, simples e funcional.
Estudante bolsista 2	Facilitar o acesso dos estudantes aos laboratórios de pesquisa, principalmente nos fins de semana	Sistema viável e facilitador do acesso aos ambientes de estudo e pesquisa. Ressalva para tolerância à falhas em caso de pane na conectividade com o servidor de controle.	Interface com o usuário eficiente, entretanto foi sugerido inserção de funcionalidade de emissão sonora para cada estado, para assegurar a acessibilidade a portadores de deficiência visual.
Professor	Solucionar o problema dos docentes não saberem a quem se reportar e onde encontrar o responsável para entregar a chave da sala após a aula	Sistema viável, e de importância significativa para melhoria do acesso às salas no ambiente da Universidade. Sugestão de aplicação para controle de presença de discentes.	Interface com o usuário conveniente, apresentou sugestões de melhoria para utilização de um *display* com tamanho maior, podendo exibir mais informações, e incrementar o número de mensagens ao usuário acerca de todos os possíveis estado de sucesso ou erro.

Tabela 2. Resumo dos relatos dos participantes

4. CONCLUSÕES

Conforme os objetivos estabelecidos para este trabalho, foram desenvolvidos o protótipo de um Dispositivo IoT e um conjunto de servidores para executar as funções de controle e persistência dos dados, apresentando assim como uma solução viável e válida para o problema levantado no presente trabalho, atendendo aos requisitos de desacoplamento, escalabilidade, segurança da informação e baixo custo.

O sistema implementado é composto por um sistema distribuído de entrega de mensagens, um servidor de controle e interface com os dados persistentes, um SGBD, um dispositivo IoT. Foi desenvolvido um protótipo de *hardware* de baixo custo de uma fechadura eletrônica inteligente capaz de conectar-se à infraestrutura de rede local sem fio, executar o envio de requisições e o processamento das respostas com o servidor de controle e gerenciamento de acesso. Vale enfatizar que o sistema desenvolvido pode ser adaptado para solver questões relacionadas à controle de acesso em outros cenários de automação, sendo um sistema de cunho e competência comercial.

Como contribuições deste trabalho, é importante destacar que o protótipo desenvolvido evidencia um potencial para tornar-se um produto comercial pois o sistema embarcado de fechadura eletrônica para controle de homologação e acesso mostrou ser uma solução robusta e viável para gerenciar o controle de acesso de uma organização. O protótipo cumpriu o desígnio de baixo custo dos componentes, destacando o mercado internacional como melhor opção para aquisição do *hardware*, apresentando um custo significativamente menor.

Para trabalhos futuros, que possam ser realizados por pesquisadores desta ou de outras instituições, pode-se destacar o estudo e desenvolvimento de uma solução de detecção de clonagem de tags RFID; medir a eficiência de outros tipos de banco de dados na execução de consultas simultâneas, contrastando com o modelo relacional; utilização de outros tipos de tecnologias RFID, como o NFC presente em *smartphones*, por exemplo; medição do consumo de energia do Dispositivo IoT e modelos para baixar o consumo do *hardware;* implementação de uma aplicação Web e Mobile para gerenciar os acessos e acessar os históricos.

REFERÊNCIAS

BATRINU, Catalin. **Projetos de Automação Residencial com ESP8266:** aproveite a potência deste minúsculo chip Wi-Fi para construir incríveis projetos de casas inteligentes. Packt Publishing Ltd, 2018.

CARNIEL, Guilherme et al. **Projeto e Desenvolvimento de Fechadura Eletrônica controlada pela Internet.** Anais do Computer on the Beach, 2015.

COELHO, Pedro. **Internet das Coisas**: Introdução Prática. Lisboa: FCA, 2017.

COSTA, Pedro Miguel Machado Monteiro. **IoT for Efficient Data Collection from Real World Resources.** 2016. 103p. Tese de Doutorado – Universidade Nova de Lisboa, Lisboa, 2016.

COULOURIS, George et al. **Sistemas Distribuídos:** Conceitos e Projeto. Bookman Editora, 2013.

DB-ENGINES Ranking. Disponível em: <https://db-engines.com/en/ranking_trend> Acesso em: 2 mar 2019.

FATIMA, Haleemunnisa; WASNIK, Kumud. **Comparison of SQL, NoSQL and NewSQL databases for internet of things.** In: 2016 IEEE Bombay Section Symposium (IBSS). IEEE, 2016. p. 1-6.

GOODRICH, Michael T.; TAMASSIA, Roberto. **Introdução à segurança de computadores**. Bookman, 2013.

LEA, Perry. **Internet of Things for Architects:** Architecting IoT solutions by implementing sensors, communication infrastructure, edge computing, analytics, and security. Packt Publishing Ltd, 2018.

LUCAS, M. C. et al. **Proposta de adoção de microsserviços em IoT**. Conferências Ibero-Americanas WWW/Internet e Computação Aplicada, 2017.

MÜHL, Gero; et al. **Distributed event-based systems**. Springer Science & Business Media, 2006.

PostgreSQL: about. Disponível em: <https://www.postgresql.org/about/> Acesso em: 9 mar 2019.

SCANDOLARA JR, Everaldo Selau. **Sistema eletrônico integrado para controle de acesso**: estudo de caso para a universidade federal de santa catarina. 2018. 72p. Trabalho de conclusão de curso (graduação) - Universidade Federal de Santa Catarina, Araranguá, 2018.

VERMA, Gyanendra K.; TRIPATHI, Pawan. **A digital security system with door lock system using RFID technology.** International Journal of Computer Applications, v. 5, n. 11, p. 6-8, 2010.