UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Gabriel Rosa Goulart

CONTROLE DE ACESSO BASEADO EM PAPÉIS EM AMBIENTES ASSISTIDOS

FLORIANÓPOLIS
2018

Gabriel Rosa Goulart

CONTROLE DE ACESSO BASEADO EM PAPÉIS EM AMBIENTES ASSISTIDOS

Trabalho de Conclusão de Curso submetido à Universidade Federal de Santa Catarina, como requisito necessário para obtenção do grau de Bacharel em Ciências da Computação

Florianópolis, junho de 2018

GABRIEL ROSA GOULART

CONTROLE DE ACESSO BASEADO EM PAPÉIS EM AMBIENTES ASSISTIDOS

Este Trabalho de Conclusão de Curso foi julgado para a obtenção do Título de "Bacharel em Ciências da Computção", e aprovado em sua forma final pelo Curso de Bacharelado em Ciências da Computação

---

Coordenador(a): Prof. Dr. Eng. Rafael Luiz Cancian
Universidade Federal de Santa Catarina - UFSC

Banca Examinadora:

Orientador(a): Prof. Ph.D. Mário Antônio Ribeiro Dantas
Universidade Federal de Santa Catarina - UFSC

Prof. D. Alex Sandro Roschildt Pinto
Universidade Federal de Santa Catarina - UFSC

Prof. D. Frank Augusto Siqueira
Universidade Federal de Santa Catarina - UFSC

Este trabalho é dedicado aos meus pais Rogério e Maria de Fátima Goulart, e aos meus irmãos Aryel e Maria Helena Goulart, pois são imprescindíveis em minha vida e tudo que fiz e faço é buscando ser alguém melhor para dar orgulho a eles.

Agradecimentos

Primeiramente gostaria de agradecer a Deus e a todos os orixás que me auxiliaram e auxiliam não somente nesta jornada, mas em todos os momentos da minha vida. Gostaria de agradecer também o meu professor e orientador Mário Dantas, que em todos momentos que precisei se mostrou disponível e atencioso, me guiando de forma concisa e séria ao longo das etapas para conceber este trabalho.

Não poderia deixar de agradecer aos meus pais, Rogério Lindolfo Goulart e Maria de Fátima Rosa Goulart, e aos meus irmãos, Aryel Rosa Goulart e Maria Helena Rosa Goulart, que em todos os momentos, bons e ruins, estiveram ao meu lado me apoiando, e que para este trabalho tiveram papel fundamental, ajudando nos testes e dividindo todos os momentos comigo. Importante ressaltar que sem eles este trabalho não seria possível.

A minha namorada Sabrina Juttel Mendes, que desde o início me ajudou, ouvindo meus planos e ideias, e mesmo sem entendê-las me auxiliou na resolução de vários problemas. Agradeço também a minha amiga Tainá de Oliveira Johnson, que iniciou a vida acadêmica junto comigo e participou de todos os momentos, me dando muita força para seguir em frente.

Agradeço a minha família, que se mostraram interessados no meu trabalho e sempre estiveram disponíveis quando precisei. Ao meu amigo Matheus Rodrigues Aranha e ao seu sogro Marcos Gonçalves que me ajudaram no manuseio dos dispositivos utilizados neste trabalho.

Para finalizar, não poderia deixar de agradecer aos meus amigos, que de uma forma ou outra, ajudaram neste trabalho.

Resumo

O crescimento da população idosa em âmbito mundial já é uma realidade, e com o passar dos anos essa população tende a crescer mais, por diversos motivos, como por exemplo os avanços na medicina. Para suprir a demanda que esse crescimento irá produzir, soluções como a de ambientes assistidos tem se tornado uma alternativa plausível, pois utilizando diversos sensores corporais e de ambiente, proveem um monitoramento constante do indivíduo que necessita de assistência, ou seja, o próprio ambiente onde a pessoa vive, irá auxiliar no seu cuidado. Consequentemente uma preocupação que se tem é com a segurança desses ambientes, e nesse sentido este trabalho propõe um sistema de controle de acesso baseado em papéis de usuário e ambiente, utilizando informações de contexto para compor as regras de acessos. Para validar o funcionamento desse sistema, 4 indivíduos de diferentes faixas etárias realizaram testes durante o período de duas semanas, onde tags RFID foram utilizadas para identificar os usuários de maneira não intrusiva. Após os testes, identificou-se que todas as tentativas de acesso foram processadas corretamente, independentemente das mudanças dos papéis ou das regras de acesso, o que mostra que o sistema proposto de fato funciona.

Palavras-chave: Controle de acesso, Controle de Acesso baseado em papéis, Consciência de contexto, Ambientes Assistidos.

Abstract

The growth of the elderly population worldwide is already a reality, and over the years this population tends to grow more, for various reasons, such as advances in medicine. To meet the demand that this growth will produce, solutions such as that of ambient assisted living have become a plausible alternative, because using various body and environment sensors, provide a constant monitoring of the individual who needs assistance, that is, the environment where the person lives, will help in his care. Consequently, one concern is with the security of these environments, and in this sense this work proposes an access control system based on user and environment roles, using context information to compose the access rules. To validate the functioning of this system, 4 individuals from different age groups performed tests during the two-week period, where RFID tags were used to identify users in a non-intrusive manner. After the tests, it was identified that all access attempts were processed correctly, regardless of the changes of roles or access rules, which shows that the proposed system does indeed work.

Keywords: Access Control, Role Based Access Control, Context awareness, Ambient Assisted Living .

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

AAL	Ambient Assisted Living - Ambiente Assistido
AMI	Ambient Intelligence - Inteligência Ambiental
ANSI	American National Standards Institute - Instituto Nacional de Normas
DAC	Discretionary Access Control - Controle de Acesso Discricionário
IBM	International Business Machines
IOT	Internet Of Things - Internet Das Coisas
IP	Internet Protocol - Protocolo de Internet
JSON	JavaScript Object Notation - Notação de Objetos JavaScript
LDR	Light Dependent Resistor - Resistor Dependente de Luz
MAC	Mandatory Access Control - Controle de Acesso Obrigatorio
MQTT	Message Queuing Telemetry Transport - Transporte de Telemetria em Fila de Mensagens
NIST	National Institute of Standards and Technology - Instituto Nacional de Padrões e Tecnologia
RBAC	Role Based Access Control - Controle de Acesso Baseado em Papel
RFID	Radio Frequency Identification - Identificação por Radiofrequência
TCP	Transmission Control Protocol - Protocolo de Controle de Transmissão

Sumário

1 Introdução

Segundo uma pesquisa das Nações Unidas, a população idosa dobrará até 2050 (ONU, 2017). Com isso o consumo de serviços voltados para essa população também aumentará, porém se soluções inovadoras não forem encontradas e aplicadas, esses serviços sofrerão com um défice muito grande para suprir as necessidades da sociedade.

Nesse contexto soluções como a de ambientes assistidos são aplicados com o âmbito de fornecer uma ajuda, e complementar os serviços voltados a saúde e bem estar, não só dos idosos, mas de qualquer indivíduo que precise ser assistido. A utilização dessa solução provê um maior conforto para o indivíduo, pois ele poderá viver no seu ambiente domiciliar e mesmo assim continuar sendo acompanhado pelo o seu médico por exemplo, e se alguma anormalidade acontecer, imediatamente todos os envolvidos no cuidado do indivíduo serão notificados e as ações necessárias serão tomadas.

A abordagem de ambientes assistidos tem grandes possibilidades de ser amplamente aceita pelos usuários, principalmente pela população idosa, pois utiliza tecnologias que se inserem no ambiente e não necessitam de muitas interações explícitas com o usuário. Além de possibilitar que as pessoas que necessitam de assistência vivam em suas casas, sem que tenha uma ou mais pessoas fisicamente lhes acompanhando.

Com o crescimento e popularização dos ambientes assistidos, não se pode deixar de pensar na segurança desses ambientes, e por este motivo o controle de acesso é de extrema importância, pois garante acesso físico ao ambiente, apenas para pessoas previamente autorizadas, utilizando diversas abordagens como, por exemplo, a baseada em papéis.

Seguindo essa linha, este trabalho abordará o controle de acesso baseado em papéis, papéis de usuário e ambiente, juntamente com informações de contexto, ou seja, informações que o ambiente pode prover para o sistema, com o objetivo de realizar um controle de acesso inteligente e sensível ao ambiente.

1.1 Motivação

O número de trabalhos relacionados a ambientes assistidos tem aumentado gradativamente nos últimos anos no Brasil, porém, basta uma simples pesquisa, para se constatar que esse número ainda é muito menor do que o europeu. Visando contribuir para o crescimento de pesquisas voltadas para ambientes assistidos, este trabalho procura empregar uma solução para esse tipo de ambiente, focando na parte de segurança de acesso, pois são poucos os trabalhos que abordam essa parte tão importante em um ambiente.

1.2 Objetivos

1.2.1 Objetivos Gerais

Propor uma abordagem de baixo custo e que funcione corretamente, para realizar o controle de acesso físico em ambientes assistidos, utilizando de maneira não intrusiva os papéis dos usuários e ambientes, juntamente com as informações de contexto.

1.2.2 Objetivos Específicos

Para alcançar o objetivo principal descrito acima, uma série de objetivos específicos são esperados, como:

- Estudar conceitos básicos relativos a controle de acesso orientado a contexto;
- Pesquisar trabalhos relacionados ao tema;
- Fazer uma proposta inicial de ambiente;
- Implementar um protótipo;
- Testar e verificar a validade da proposta;
- Escrever um artigo sobre a contribuição;
- Apresentar artigo.

1.3 Delimitações do Trabalho

Neste trabalho os testes foram realizados em um ambiente controlado, onde não se levou em consideração problemas relacionados ao fornecimento de energia elétrica, problemas de conectividade entre os dispositivos, má utilização por parte dos usuários e problemas causados por desastres, sendo eles naturais ou não.

Com recursos financeiros limitados, este trabalho se limitou a utilizar dispositivos cujo custo-benefício fosse suficiente para mostrar a validade do sistema. Por este motivo apenas dois ambientes são monitorados.

Todos os dispositivos foram configurados de maneira manual, pois a configuração automática dos mesmos não é o foco deste trabalho. Importante ressaltar também que a identificação dos usuários é realizada por meio de tags RFID, onde se pode identificar quem é o usuário sem ferir a sua privacidade.

1.4 Metodologia

A metodologia empregada neste trabalho é de cunho exploratório e se baseia em: estudo dos conceitos relacionados ao tema; pesquisa para fundamentação teórica; modelagem, implementação e validação da proposta.

1.5 Organização dos capítulos

Este trabalho está organizado da seguinte forma: o capítulo 2 apresenta a fundamentação teórica importante para o entendimento deste trabalho; já no capítulo 3 é realizada uma análise dos trabalhos correlatos; o capítulo 4 apresenta a proposta deste trabalho, mostrando cada parte importante do sistema e como irá funcionar; no capítulo 5 são apresentados o ambiente e os testes que validam a proposta; e por fim o capítulo 6 apresenta a conclusão e as propostas de trabalhos futuros.

2 Embasamento Teórico

Neste capítulo serão abordados os conceitos essenciais para total entendimento da proposta deste trabalho.

2.1 Controle de Acesso

O conceito de controle de acesso não é recente, pois desde tempos antigos esse método é utilizado para permitir acesso a recursos somente para usuários previamente autorizados. Como citado por SANDHU; SAMARATI(1994) é importante deixar claro que o controle de acesso não é uma solução completa para a segurança de um sistema, seja ele físico ou digital. É necessário utilizá-lo em conjunto com outras soluções, como por exemplo, meios de autenticação e meios de auditoria.

Com os avanços tecnológicos, as maneiras de gerenciar e aplicar o controle de acesso vem mudando, porém as principais políticas desse método são bastante conhecidas e importantes para a implementação do mesmo. É importante ressaltar que essas políticas, originalmente, não foram propostas para o uso em ambientes físicos e inteligentes. Por este motivo, a política adotada nesse trabalho será adaptada para se adequar às necessidades de um ambiente real e inteligente.

A seguir serão tratados os aspectos relacionados às políticas clássicas de controle de acesso e a política adotada neste trabalho.

2.1.1 Controle de Acesso Discricionário

O controle de acesso discricionário, do inglês *Discretionary Access Control* (DAC), é uma das clássicas políticas de controle de acesso, onde o controle é baseado na identidade do usuário e autorização (ou regras), especificando para cada usuário ou grupo de usuários, e para cada recurso do sistema, os modos de acesso que serão permitidos para o usuário ou grupo em relação ao recurso. É importante enfatizar que a autorização para acessar um certo recurso é dada pelo próprio dono do recurso, ou seja, o controle de acesso é descentralizado.

Para cada requisição feita pelo usuário para acessar um recurso, é realizada uma avaliação sobre as autorizações do usuário para o recurso. Se existe uma autorização para acessar no modo especificado, então o acesso é garantido, caso contrário é negado.

Conforme SANDHU; SAMARATI(1994) o controle de acesso discricionário não provê uma real garantia no fluxo de informação no sistema, pois é fácil ignorar as restrições

de acesso. Para um melhor entendimento, o seguinte exemplo é utilizado no artigo: Um usuário que tem permissão para ler uma informação pode passá-la para outro usuário que não tem autorização para tal ação, sem que o dono da informação tenha conhecimento. Essa situação pode ocorrer porque o controle de acesso discricionário não impõe nenhuma restrição sobre o uso da informação, ou seja, a disseminação da informação não é controlada. Ao contrário do controle de acesso discricionário, o controle de acesso obrigatório controla a disseminação da informação, impedindo que situações como a apresentada acima ocorram.

2.1.2 Controle de Acesso Obrigatório

O controle de acesso obrigatório, do inglês *Mandatory Access Control* (MAC), também é uma das clássicas políticas de controle de acesso, e nessa abordagem o acesso é controlado através da classificação dos usuários e recursos envolvidos no sistema, ou seja, eles são associados a níveis de segurança. Em relação aos recursos, os níveis de segurança representam a sensibilidade da informação, isto é, o potencial dano que um acesso não autorizado ao recurso poderia causar. Já para os usuários, representam a confiança em não compartilhar recursos sensíveis com outros usuários não habilitados a acessar tal recurso.

Pode-se perceber que essa abordagem segue uma hierarquia. Por conta disso, essa política é muito popular em meios militares e governamentais, pois são meios onde as hierarquias ficam explícitas.

O modelo MAC traz consigo o controle de fluxo de informação através dos princípios *read* e *write*, que podem ser abordados de duas maneiras, *read down write up* e *read up write down*. Na primeira abordagem, para o usuário acessar o recurso (*read down*) , o nível de segurança associado a ele deve estar no mesmo nível ou em níveis superiores na hierarquia em relação ao nível de segurança do recurso. Já para criar ou modificar um recurso (*write up*), o nível de segurança associado ao usuário deve estar no mesmo nível ou em níveis inferiores ao nível de segurança do recurso. Na segunda abordagem o conceito é inverso ao da primeira, ou seja, para acessar um recurso (*read up*), o nível de segurança do usuário deve estar no mesmo nível ou em níveis inferiores ao nível de segurança do recurso, e para criar ou modificar (*write down*), o nível de segurança do usuário deve estar no mesmo nível ou em níveis superiores ao nível de segurança do recurso.

Além do controle de fluxo, um aspecto bastante relevante do MAC é que seu gerenciamento é centralizado, diferentemente do DAC. Sendo assim, todas as regras de acesso são criadas, editadas ou excluídas pelo administrador do sistema, ou seja, o usuário não tem mais domínio sobre o acesso, mesmo o usuário sendo o proprietário do recurso.

2.1.3 Controle de Acesso Baseado em Papéis

O controle de acesso baseado em papéis, também conhecido como *Role Based Access Control* (RBAC), foi proposto por volta de 1970 quando sistemas multiusuários e multiaplicações começaram a surgir. Sua padronização foi realizada em 2004 pelo *National Institute of Standards and Technology* (NIST). Atualmente o padrão ANSI para o RBAC é o INCITS 359-2012 .

Basicamente essa abordagem associa os usuários a papéis, e os papéis a permissões. Isso permite que os usuários possam ser associados a outros papéis facilmente, e que as permissões associadas aos papéis possam ser alteradas ou revogadas de uma maneira simples também, ou seja, com a incorporação de novas aplicações no sistema, o impacto para adequar os usuários e papéis é o mínimo possível.

O RBAC é definido por quatro componentes: o RBAC núcleo, RBAC hierárquico, Separação estática de relações de serviço e Separação dinâmica de relações de serviço. O componente mais importante do RBAC é o núcleo, o qual pode ser utilizado sem os outros componentes, pois traz consigo as funções essenciais do controle de acesso baseado em papéis. Em relação aos modelos apresentados anteriormente, o RBAC se assemelha ao MAC, pois sua administração é centralizada, assim como no controle de acesso obrigatório.

2.1.3.1 RBAC Núcleo

O núcleo inclui alguns elementos básicos para o controle de acesso. Esses elementos são: usuários (US), papéis (PA), objetos (OBJS), operações (OPS) e permissões (PERMS). Através da figura 1 se pode observar o conjunto de elementos e suas interações. A relação entre usuários e papéis (AU), e entre papéis e permissões (AP) é de muitos para muitos, o que permite que um usuário possa ser associado a vários papéis. Além disso o núcleo tem um elemento chamado sessão (SE), que por sua vez armazena para cada usuário os papéis ativos relacionados a ele.

Importante destacar que o elemento usuário não precisa ser necessariamente um humano. Esse elemento pode ser representado por um sistema computacional, por exemplo. Já o elemento objeto pode ser qualquer recurso do sistema que necessite de controle de acesso, onde se permite ou não realizar as operações OP.

Conforme descrito por FERRAIOLO; KUHN(2004) segue a especificação do núcleo do RBAC:

- USUÁRIOS, PAPÉIS, OPERAÇÕES e OBJETOS
- $AU \subseteq$ US X PA, relação de atribuição muitos para muitos entre usuário e papel.
- $usuários\_atribuidos : (p : PA) \rightarrow 2^{US}$, mapeamento do papel p em um conjunto de usuários. $Formalmente : usuários\_atribuidos(p) = u \in US | (u, p) \in AU$

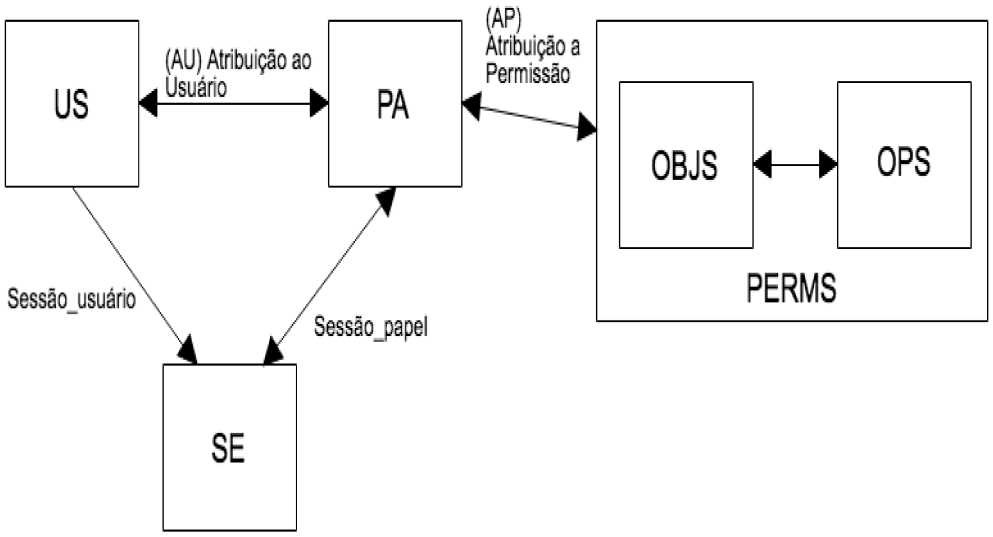

Figura 1 – Núcleo do RBAC

- $PERMS = 2^{(\text{OBJ X OP})}$, conjunto de permissões.
- $AP \subseteq$ PERMS X PA, relação de atribuição muitos para muitos entre permissões e papéis.
- $permissões\_atribuidas(r : PA) \rightarrow 2^{PERMS}$, mapeamento do papel r em um conjunto de permissões. $Formalmente : permissões\_atribuidas(r) = p \in PERMS|(p, r) \in AP$
- $Op(p : PERMS) \rightarrow op \subseteq OPS$, mapeamento da permissão para a operação, o qual dá o conjunto de operações associadas com a permissão p.
- $Ob(p : PERMS) \rightarrow ob \subseteq OBJS$, mapeamento da permissão para o objeto, o qual dá o conjunto de objetos associados com a permissão p.
- $SE =$ conjunto de sessões.
- $sessão\_usuário(s : SE) \rightarrow US$, mapeamento da sessão s em um usuário correspondente.
- $sessão\_papel(s : SE) \rightarrow 2^{PA}$, mapeamento da sessão s em um conjunto de papéis. $Formalmente : sessão\_papel(Si) \subseteq p \in PA|(sessão\_usuário(Si), p) \in AU$
- $sessão\_permissão\_disponível(s : SE) \rightarrow 2^{PERMS}$, permissão disponível para um usuário em uma $sessão = \bigcup_{p \in sessão\_papel(s)} permissões\_atribuidas(p)$

2.1.3.2 RBAC Hierárquico

Esse componente introduz o conceito de hierarquia de papel (HP). A Hierarquia representa naturalmente a estrutura de papéis, refletindo em uma linha de organização das autoridades e responsabilidades.

A hierarquia de papéis define a relação de herança entre papéis, ou seja, se um papel P1 agrega todas as permissões de um papel P2, diz-se que P1 herda P2. Obviamente, como P1 está em um nível de hierarquia maior que P2, ele pode conter mais permissões que P2. Na figura 2 se tem um exemplo prático de hierarquia de papéis, onde mostra simplificadamente a estrutura hierárquica dos papéis em um mercado .

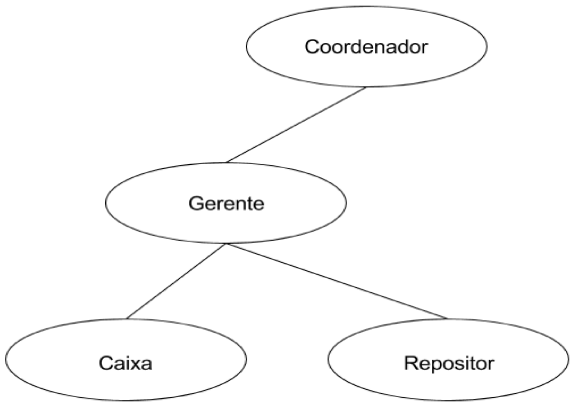

Figura 2 – Exemplo de Hierarquia de Papéis

Na figura 2 o papel Coordenador está no nível mais alto da hierarquia, logo ele possui todas as permissões associadas aos demais papéis. Já o papel Gerente só possui as permissões dos papéis Caixa e Repositor, e no caso, os dois últimos papéis no nível de hierarquia, só possuem as suas permissões.

2.1.3.3 Separação estática de relações de serviço

Com o crescimento do sistema, usuários vão sendo associados a novos papéis, e como um usuário pode se associar a mais de um papel, um problema que pode ocorrer é o conflito de interesses, ou seja, o usuário se associa a dois papéis que são conflitantes. Por exemplo, na figura 2 o usuário se associa ao papel Caixa e ao papel Gerente. Esse tipo de ação deveria ser proibida, pois o papel Caixa é subordinado do papel Gerente, e como que um usuário poderia ser subordinado dele mesmo? Para resolver essa situação, foi agregada ao RBAC a separação estática de relações de serviço, que consiste em restringir associações entre usuário e papéis, ou seja, antes de se associar um usuário a um papel é verificado se essa associação é permitida, caso seja, a associação é feita, caso contrário não.

2.1.3.4 Separação dinâmica de relações de serviço

Anteriormente foi discutido sobre separação estática de relações de serviço, porém conflitos podem ocorrer de maneira dinâmica, ou seja, durante a ativação de papéis para um certo usuário. A separação dinâmica de relações de serviço previne que dois papéis conflitantes estejam ativos ao mesmo tempo para um usuário, portanto para um papel P1 que tem conflito com um papel P2 ficar ativo para um usuário U, o papel P2 tem que estar inativo para o usuário U. Importante observar que assim como na separação estática, a solução desse problema se dá pelo uso de restrições, que são verificadas nas ativações de papéis.

2.2 Contexto

O termo contexto pode ter várias definições, e defini-lo de maneira correta é essencial para o desenvolvimento deste trabalho, pois é necessário saber o que é relevante e o que não é, principalmente quando se agrega a consciência de contexto a um sistema. A definição utilizada neste trabalho é a desenvolvida por Annd K. Dey em (DEY, 2001). A seguir a definição de contexto:

> Contexto é qualquer informação que possa ser usada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, lugar ou objeto que é considerado relevante para a interação entre um usuário e um aplicativo, incluindo o usuário e as próprias aplicações. (DEY, 2001)

A partir deste momento sempre que for falado em contexto, a definição utilizada será essa apresentada acima.

Outro conceito importante para se destacar aqui é o de consciência de contexto. Mais uma vez se utilizará a definição dada por DEY como segue:

> Um sistema é consciente de contexto se ele usa o contexto para prover informações e/ou serviços relevantes para o usuário, onde a relevância depende da tarefa do usuário. (DEY, 2001)

2.3 Internet Das Coisas

Nos últimos anos assuntos relacionadas a internet das coisas, do inglês *Internet Of Things* (IOT), vem se tornando bastante populares, principalmente pela vasta gama de possibilidades que surgem com a IOT. Tomando em consideração aspectos como a interdisciplinaridade que a internet das coisas traz consigo, BUCKLEY(2006) explicita que a IOT é o tema principal para a evolução da informação e comunicação nas próximas décadas.

Apesar de estar sendo muito estudada, a IOT não tem uma definição universal. Por este motivo, WHITMORE et al.(2015) definem como sendo o conceito principal da IOT, objetos do dia a dia que podem ser equipados com recursos de identificação, detecção, rede e processamento que lhes permitam se comunicar uns com os outros e com outros dispositivos e serviços pela Internet para atingir algum objetivo.

2.4 Ambientes Assistidos

Com o amadurecimento das tecnologias da informação, surgiram novos conceitos e paradigmas, os quais tem como objetivo auxiliar na saúde e bem estar dos seres humanos. Justamente com esses objetivos o conceito de ambientes assistidos, do inglês *Ambient Assisted Living* (AAL), surgiu. Sendo implementado utilizando-se um conjunto de tecnologias, atuando sobre o paradigma de *Ambient Intelligence* (AMI), inteligência ambiental traduzido livremente para o português.

De acordo com BROEK et al.(2010) ambientes assistidos podem ser produtos, serviços ou sistemas, que através do uso de tecnologias, melhoram o processo de envelhecimento do ser humano em sua casa, comunidade ou emprego, ou seja, melhoram a qualidade de vida, autonomia, a participação na vida social, habilidades e empregabilidade. Além de proporcionarem uma diminuição nos custos relacionados a cuidados com a saúde. BROEK et al.(2010) também explicitam que os AAL podem ser usados não apenas por idosos, mas também por outros grupos de pessoas que necessitam de alguma assistência, por exemplo pessoas com necessidades especiais.

Segundo KLEINBERGER et al.(2007) para um AAL alcançar os seus objetivos, os principais requisitos são:

1. Ser discreto e não intrusivo para alcançar uma maior aceitação.
2. Tem que se adaptar as mudanças de situações pessoais ou capacidades do indivíduo e do meio ambiente para atender às necessidades individuais.
3. Prover seus serviços de forma acessível para aumentar a usabilidade.

BROEK et al.(2010) também expõem de maneira mais abrangente os principais requisitos para um AAL alcançar os seus objetivos.

1. Ser embarcado (não invasivo ou dispositivos invisíveis).
2. Distribuído através do ambiente ou diretamente integrado nos aparelhos ou móveis.
3. Personalizado (tolerante as necessidades do usuário).

4. Adaptativo (responsivo para o usuário e o seu ambiente).

5. Antecipação (Antecipando os desejos do usuário na medida do possível sem mediação consciente).

Pode-se perceber que tanto BROEK et al.(2010) quanto KLEINBERGER et al.(2007) convergem em dois pontos quando relacionado a requisitos para um ambiente assistido. São eles: o sistema tem que ser não intrusivo e adaptativo. Isso mostra duas características bastante visíveis e importantes em um AAL.

Para entender ambientes assistidos é importante entender o paradigma de inteligência ambiental. AUGUSTO(2007) define AMI como sendo um ambiente digital que apóia as pessoas em suas vidas diárias, auxiliando-os de forma sensata. Vale destacar que o paradigma AMI não é apenas aplicado ao contexto saúde, mas também pode ser usado em outros contextos, como por exemplo, auxilio em transporte público ou serviços educacionais.

2.5 Protocolos de Comunicação

No mundo da computação existem diversos protocolos de comunicação, cada um com as suas características próprias para suprir determinada necessidade. Obviamente que para internet das coisas nem todos os protocolos se adequam, por isso ao longo dos anos protocolos foram desenvolvidos ou adequados para trabalhar com IOT.

ROTTA et al.(2017) trazem um levantamento dos principais protocolos de comunicação com foco em IOT. Interessante observar que esses tipos de ambientes, baseados em IOT, necessitam de protocolos leves, eficientes e que sejam capazes de trabalhar em um meio onde a capacidade de computação é limitada, como destacam MARTINS; ZEM(2014). A tabela 1 mostra os principais protocolos de comunicação, quando se trata de ambientes com foco em IOT, comparando os mesmos com as camadas TCP/IP e os protocolos que operam sobre elas.

Tabela 1 – Comparativo entre camadas TCP/IP e Protocolos para IOT

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

Fonte : ROTTA et al.(2017)

A seguir será discutido mais amplamente sobre o protocolo de comunicação MQTT, o qual será utilizado neste trabalho.

2.5.1 MQTT

Message Queuing Telemetry Transport (MQTT), é um protocolo desenvolvido pela IBM no final da década de 90, com o objetivo de ser utilizado principalmente nas áreas relacionadas ao petróleo e satélites. Utilizando o conceito de *publish/subscribe*, se tornou uma alternativa bastante interessante para se trabalhar com IOT, pois traz consigo diversas vantagens, como por exemplo ser um protocolo leve, e que trabalha com mensagens assíncronas. YUAN(2017) ainda destaca a possibilidade de usar o MQTT em hardware de dispositivo altamente restringido e em redes de largura da banda limitada e de alta latência, o que traduz com muitas propriedades um ambiente de IOT.

A figura 3 exemplifica de maneira simplificada a arquitetura do protocolo MQTT. Os dispositivos *publishers* são responsáveis por publicarem as informações para um determinado tópico, o *broker* (gerenciador), recebe essas publicações e repassa para todos os *subscribers*, de acordo com os tópicos de interesse de cada *subscriber*, ou seja, se um dispositivo *publish* publica alguma informação para o tópico X, todos os *subscribers* registrados no tópico X receberão a informação . Como se pode perceber o *broker* tem um papel fundamental para o funcionamento do protocolo, pois ele é o canal de comunicação entre os *publishers* e *subscribers*.

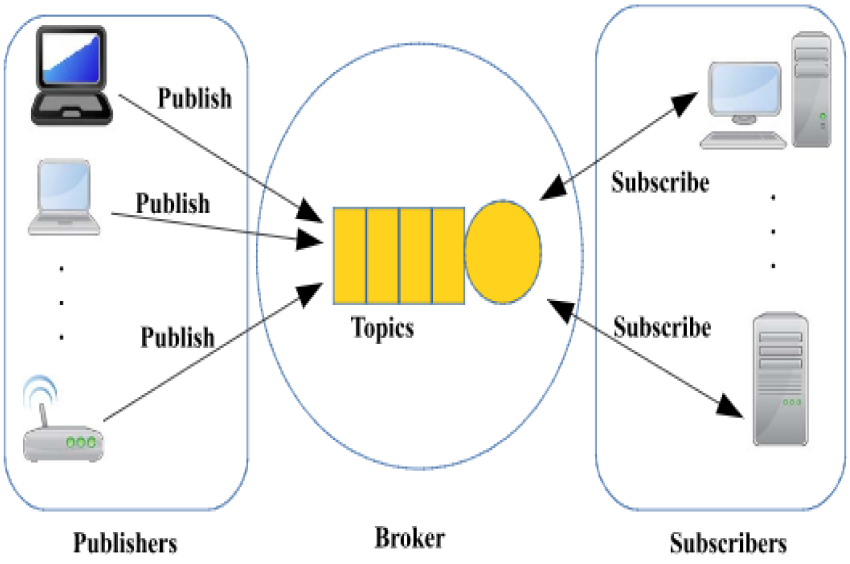

Figura 3 – Arquitetura MQTT

Fonte : AL-FUQAHA et al.(2015)

2.5.1.1 Eclipse Mosquitto Broker

Eclipse Mosquitto é um gerenciador de mensagens open source que implementa o protocolo MQTT. Mosquitto broker oferece uma implementação leve e eficiente, o que o faz uma escolha interessante quando se trabalha com MQTT e internet das coisas, principalmente porque o seu consumo de recursos é baixo. Por exemplo, a versão 1.4.12 consome por volta de 3 MB de memória com 1000 clientes conectados, segundo (ECLIPSE..., 2017b).

Importante citar que o projeto Eclipse Mosquitto faz parte do Eclipse IOT Working Group, que consiste em um grupo de companhias que investem e promovem projetos *open source* voltados para IOT.

3 Trabalhos Correlatos

Na comunidade acadêmica assuntos relacionado ao controle de acesso já vem sendo discutidos há bastante tempo, principalmente o baseado em papéis. Porém, ainda não se encontram em abundância trabalhos que foquem no uso do controle do acesso em ambientes assistidos, mesmo que nos últimos anos, assuntos relacionados a ambientes assistidos tenham estado em alta.

Nos trabalhos que foram analisados, formas de se aplicar o controle de acesso baseado em papéis foram encontradas, algumas levando em consideração o contexto, outras atribuindo papéis aos ambientes. Entretanto, a maioria dos trabalhos não deixa claro a sua aplicabilidade em ambientes assistidos, de maneira a controlar o acesso a um ambiente real, utilizando todos os recursos que tal ambiente pode oferecer.

3.1 Análise dos trabalhos correlatos

Como citado anteriormente, diversas alternativas para realizar o controle de acesso baseado em papéis são encontradas. ZHANG et al.(2004) utilizam máquinas de estados para fazer o controle de papéis ativos e permissões atribuídas aos papéis. Como a aplicação é consciente de contexto, um agente de contexto coleta as informações e gera eventos que disparam transições nas máquinas de estados.

Outra abordagem consciente de contexto é a de COVINGTON et al.(2001), que traz o conceito de papel de ambiente, o que não é um elemento do padrão RBAC. Sendo assim, com essa nova atribuição, as permissões são associadas tanto aos papéis de usuários quanto aos papéis de ambiente, provendo uma maior flexibilidade para o sistema como um todo.

PARK et al.(2006) utilizam o conceito de papel de contexto, o que se assemelha ao papel de ambiente apresentado por COVINGTON et al.(2001). Entretanto, elementos como datas e tempo são utilizados para formar o papel de contexto, que são associados aos papéis de usuários e assim formam a política de segurança. A seguir um exemplo do controle de acesso utilizando papel de contexto.

- transação = $< papel\_usuário, papel\_contexto, permissão >$
- bit_permissão = permitir , negar
- regra da política = $<$ transação, $bit\_permissão >$, $exemplo <<$ criança, $(18h < T < 21h), TV\_Ligar >, permitir >$

KAYES et al.(2017) utilizam as informações de contexto para ativar o papel do usuário, semelhante aos trabalhos apresentados anteriormente. A utilização do contexto para ativar um papel de usuário é realizada através de expressões contextual, ou seja, uma composição de contextos, onde se pode utilizar informações como a localização do usuário, dias da semana ou até mesmo as escalas de trabalho, por exemplo. O gerenciamento dessas políticas de controle de acesso é realizado utilizando-se ontologias, o que facilita no processo de verificação das condições para ativar um papel, e também na expansão do sistema, como a criação de novas políticas de acesso ou até mesmo de papéis de usuário.

No processo de pesquisa limitou-se a pesquisar abordagens de controle de acesso baseadas em papéis, as quais fossem sensíveis ao contexto, para que assim houvesse uma maior proximidade com os objetivos deste trabalho, mesmo que a maioria dos trabalhos foquem em apresentar modelos, que muitas vezes não são aplicados em um ambiente assistido.

4 Proposta

Realizar o controle de acesso é extremamente importante quando se fala na segurança de um ambiente, e obviamente não seria diferente para um ambiente assistido. Com o objetivo de explorar essa questão, este trabalho propõe um sistema para controlar o acesso físico em ambientes assistidos de maneira não intrusiva, utilizando papéis de usuários e ambientes juntamente com as informações de contexto para construir regras de acesso.

A figura 4 apresenta um exemplo do funcionamento do sistema. Supondo que um usuário X desempenha o papel de usuário EMPREGADO e está tentando acessar um ambiente Y com papel de ambiente SALA DO CHEFE, e a regra de acesso associada ao papel de usuário e ambiente utilize as seguintes informações de contexto : horário, data e se o chefe se encontra em sua sala. Para que o usuário X consiga acessar a sala do chefe as informações de contexto precisam ser verdadeiras. Para um entendimento mais amplo, a figura 5 apresenta o diagrama de atividades do sistema de controle de acesso.

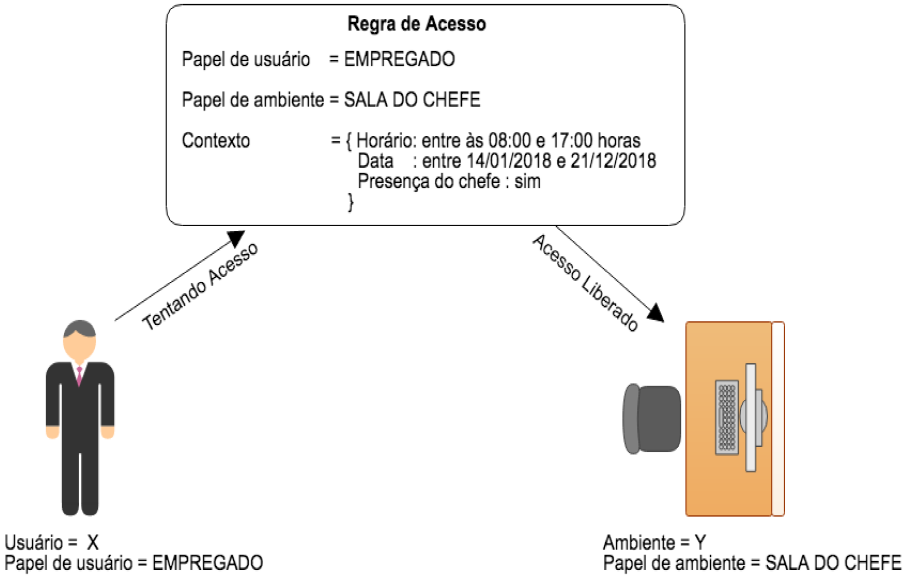

Figura 4 – Exemplo do Sistema

A identificação do usuário é realizada de maneira não intrusiva utilizando tags RFID semelhantes às apresentadas na figura 6. As tags são associadas a papéis de usuário, que por sua vez integram a regra de acesso. As informações e associações, como por exemplo entre usuário e papel de usuário, são armazenadas em um banco de dados MYSQL, cuja modelagem será apresentada na seção 4.4. Importante ressaltar que o sistema foi implementado utilizando as linguagens Python e C para Arduino junto com o protocolo de comunicação MQTT.

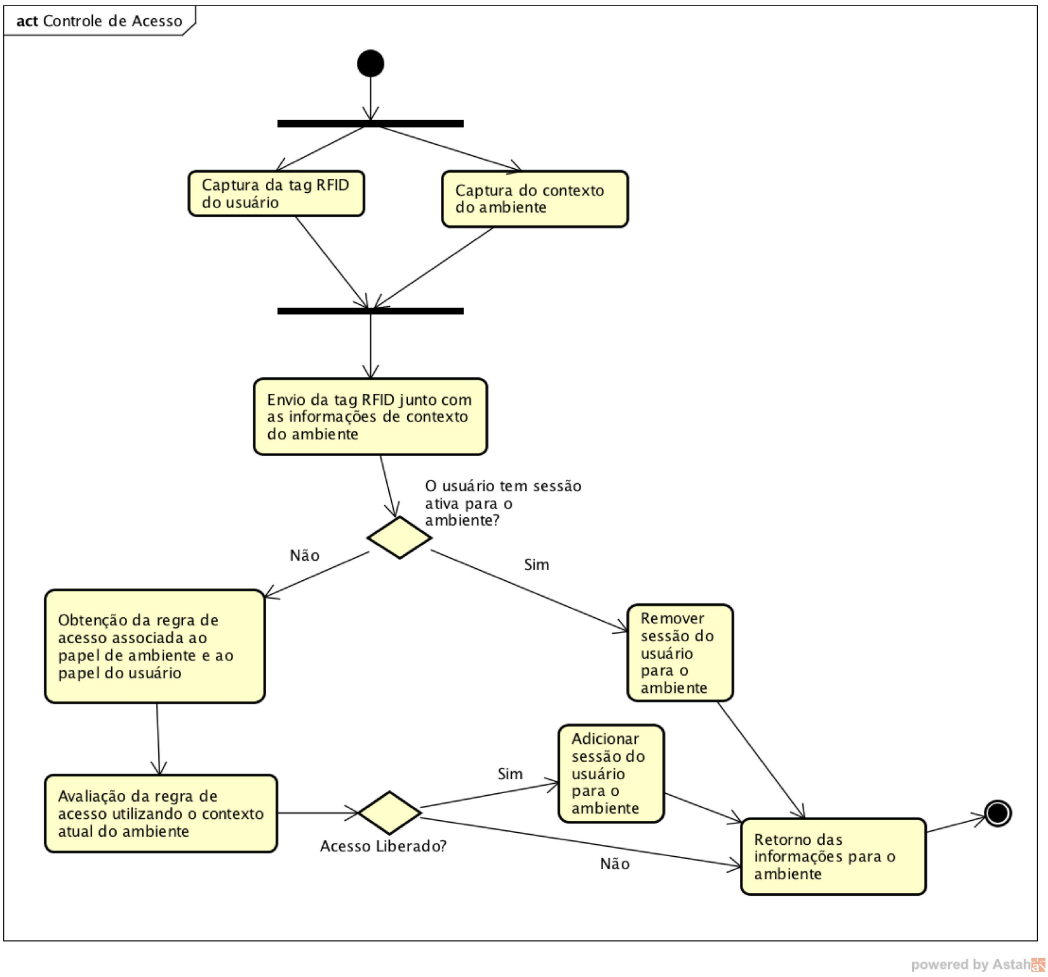

Figura 5 – Diagrama de Atividades do Sistema

Figura 6 – Exemplos de Tags RFID

4.1 Expansão do RBAC

Neste trabalho o controle de acesso utilizado será o baseado em papéis, o RBAC, porém só será utilizado o RBAC núcleo, o qual contempla as funções básicas para o funcionamento do controle de acesso conforme especificado na subseção 2.1.3.1.

Para que o controle de acesso aproveite todos os recursos que um ambiente assistido pode oferecer, se baseando na expansão do RBAC feita por COVINGTON et al.(2001), será adicionado ao modelo o conceito de ambiente e papel de ambiente, o que permitirá uma maior flexibilidade no sistema, pois a regra de acesso não dependerá apenas do papel do usuário, mas também do papel de ambiente. A figura 7 apresenta a expansão do RBAC núcleo que será utilizado neste trabalho.

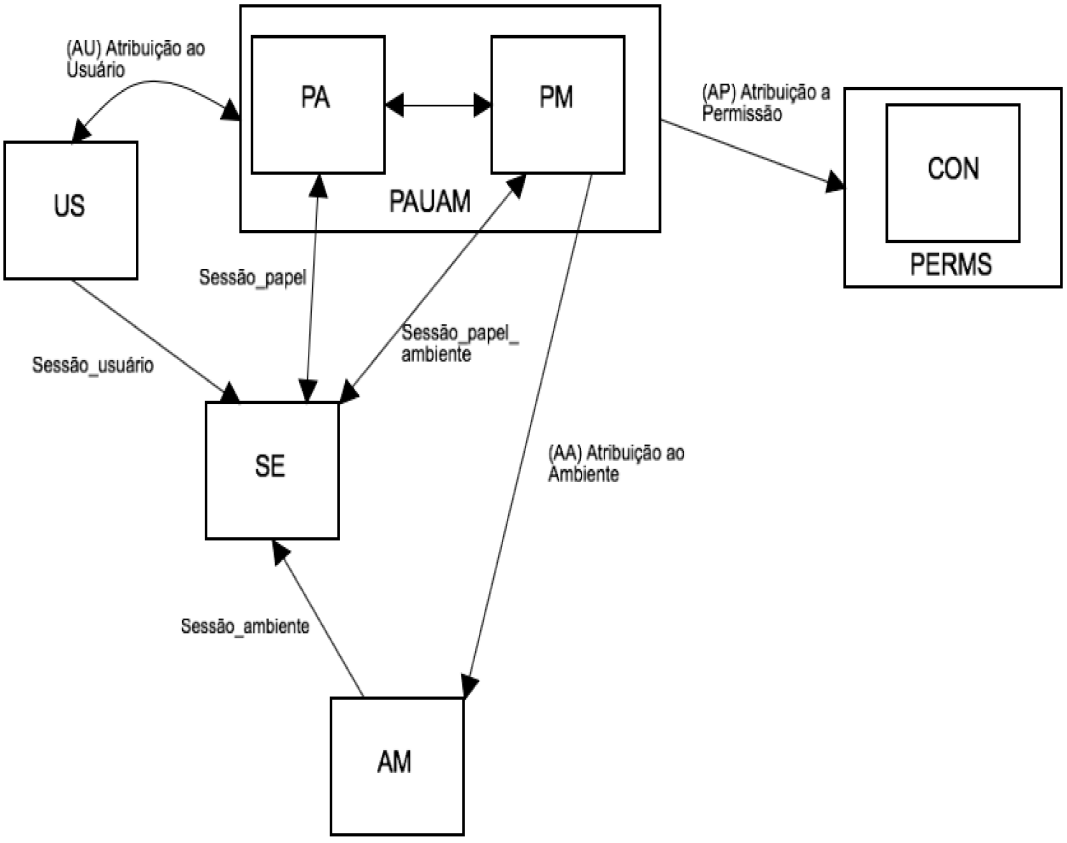

Figura 7 – Extensão do RBAC

Especificações para a expansão do RBAC:

- Usuário (US), Papel de Usuário (PA), Ambiente (AM) , Papel de Ambiente (PM), Sessão (SE), Permissão (PERMS), Contexto (CON).
- $PAUAM = 2^{(\text{PA X PM})}$, conjunto de papéis de usuário associados a papéis de ambiente.

- $AU \subseteq$ US X PAUAM, relação de atribuição muitos para muitos entre usuário e papéis de usuário relacionados com papéis de ambiente.

- $AA =$ relação de um para muitos entre ambientes e papéis de ambiente.

- $AP \subseteq$ PERMS X PAUAM, relação de atribuição um para muitos entre papéis de usuário associados a papéis de ambiente e permissões.

- $Sessão\_usuário(s : S) \rightarrow US$, mapeamento da sessão s em um usuário correspondente.

- $Sessão\_ambiente(s : SE) \rightarrow AM$, mapeamento da sessão s em um ambiente correspondente.

- $Sessão\_papel(s : S) \rightarrow 2^{PA}$, mapeamento da sessão s em um conjunto de papéis de usuário.

- $Sessão\_papel\_ambiente(s : SE) \rightarrow 2^{PM}$, mapeamento da sessão s em um conjunto de papéis de ambiente.

Outra adaptação importante a ser feita no RBAC, é que a sessão representará que o usuário está registrado em um ambiente, ou seja, representará que o usuário está no ambiente. Caso o usuário não tenha uma sessão ativa para um certo ambiente, significa que o usuário não está no ambiente.

4.2 Equipamentos

Levando em consideração as limitações financeiras, procurou-se uma boa relação custo/benefício para ser empregada no trabalho. Por este motivo equipamentos de baixo custo foram utilizados, além do uso do próprio computador do autor do trabalho, diminuindo assim os gastos.

A figura 8 apresenta o elemento responsável pela leitura de tags RFID do usuário. Esse leitor fica conectado a placa arduino UNO, figura 10, juntamente com o shield ETHERNET, figura 9, que é responsável pelo envio dos dados para o computador, o qual servirá como gateway do sistema.

A tabela 2 mostra os equipamentos utilizados neste trabalho e a figura 11 apresenta o módulo, o qual ficará montado próximo à entrada do ambiente.

Tabela 2 – Equipamentos utilizados no trabalho

Equipamento	Descrição
Leitor RFID Mfrc522 Mifare	Realiza a leitura de tags RFID.
Ethernet Shield W5100	Responsável pelo envio de dados.
Sensor Ultrassônico HC-SR04	Avalia a distância até um objeto que está à sua frente.
Sensor de luminosidade LDR	Avalia a luminosidade no ambiente.
Leds Difuso Vermelho	Acende se a solicitação de acesso foi negada.
Leds Difuso Verde	Acende se a solicitação de acesso foi aceita.
Arduino UNO R3	Recebe e envia dados do ambiente para o gateway.
MacBook pro - MacOs Versão 10.13.3	Gateway do sistema.

Figura 8 – Leitor RFID

Fonte : (FLOP, 2017c)

Figura 9 – Ethernet Shield

Fonte : (FLOP, 2017b)

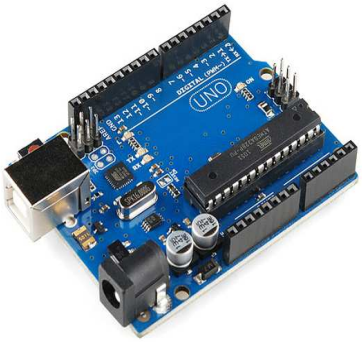

Figura 10 – Placa arduino UNO R3

Fonte : (FLOP, 2017a)

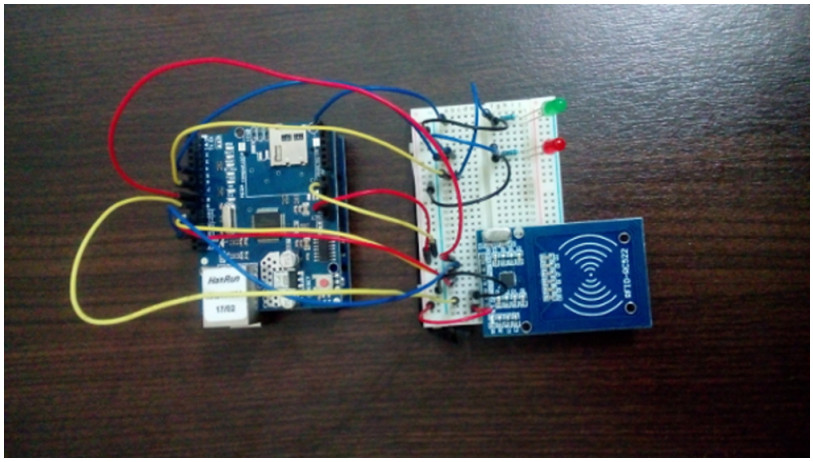

Figura 11 – Módulo ambiente montado

Com o objetivo de prover informações de contexto mais robustas, dois sensores serão utilizados. A figura 12 apresenta o sensor ultrassônico. Esse sensor é capaz de medir a distância até objetos que estão à sua frente. Basicamente ele funciona emitindo sinais ultrassônicos e esperando o seu retorno para calcular a distância. Sua precisão é de 3 milímetros, podendo ler distâncias de 2 centímetros até 4 metros. O segundo sensor, apresentado pela figura 13, é capaz de medir a luminosidade no ambiente. Tecnicamente ele é um resistor dependente de luz, *Light Dependent Resistor* (LDR), ou seja, quanto mais luz incide sobre ele, menor é o valor da resistência. Logo quanto menos luz está incidindo sobre o resistor, maior é o valor da resistência. Os valores variam de 0 até 1023, porém neste trabalho essa faixa de valores será convertida em uma faixa de valores que variam de 0 até 255, conforme sugerido por THOMSEN(2011). Portanto o valor 0 significa que o ambiente está bem iluminado, e 255 que o ambiente está escuro.

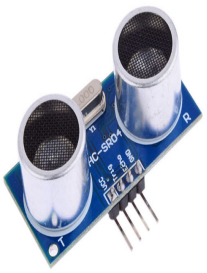

Figura 12 – Sensor ultrassônico

Fonte : (FLOP, 2017e)

Figura 13 – Resistor dependente de luz

Fonte : (FLOP, 2017d)

4.3 Módulos do Sistema

Para um melhor entendimento do sistema e facilitar a sua concepção, o sistema será dividido em dois módulos: módulo do ambiente,que entrará em contato com o usuário; e o módulo gateway, que contemplará o gerenciamento do sistema.

- Módulo do ambiente : a figura 11 apresenta um exemplo do módulo do ambiente. Ele será responsável por realizar a leitura das tags RFID do usuário e liberar ou não o acesso do mesmo, ou seja, ficará localizado próximo à entrada do ambiente, onde o controle de acesso está sendo realizado. O módulo é composto pela placa Arduino, o leitor RFID e o shield Ethernet.
- Módulo gateway : o módulo gateway será responsável por gerenciar todo o sistema, portanto ele recebe os dados do módulo do ambiente e os processa com o objetivo de verificar se o usuário pode ou não acessar o ambiente. O módulo se divide em três sub módulos, o gerenciador de mensagens, avaliador da política de acesso e persistência.
  - Gerenciador de mensagens : necessário para que o protocolo MQTT funcione. Conforme exposto na subseção 2.5.1, o gerenciador de mensagens utiliza o Eclipse Mosquitto *broker* como *broker* do protocolo MQTT. Esse sub módulo realiza toda a interação entre o módulo gateway e módulo do ambiente, ou seja, todas as mensagens trocadas entre os dois módulos são gerenciadas pelo gerenciador de mensagens.
  - Avaliador da política de acesso: basicamente esse sub módulo é responsável por avaliar a regra de acesso associada ao papel de usuário e ao papel de ambiente, e decidir se o usuário tem ou não o acesso garantido ao ambiente.
  - Persistência : sub módulo que realizará todas as operações de armazenamento, criação, atualização e busca de dados no sistema. Ressaltando que a persistência será realizada em um banco de dados MYSQL versão 5.7.21.

4.4 Banco de Dados

A figura 14 traz o modelamento do banco dados utilizado pelo sistema de controle de acesso. Observando atentamente percebe-se que o diagrama relacional traduz em termos práticos o RBAC adaptado, apresentado na seção 4.1. Importante destacar alguns pontos que são essenciais para o sistema.

O primeiro ponto importante para se destacar é a coluna tag na tabela Usuário. Essa coluna armazenará as tags RFID dos usuários, e será usada como chave para as associações. Ainda falando sobre a tabela Usuário, a coluna descrição serve meramente

para identificar o usuário de maneira genérica, como por exemplo "user1", o que facilitará na análise dos resultados.

A tabela Regra_De_Acesso armazenará as informações utilizadas para validar o acesso do usuário, como papel de usuário, papel de ambiente e contexto. Quando o valor do contexto for igual a NULL, significa que o usuário pode acessar o ambiente sem restrições, ou seja, a qualquer momento.

Outro ponto importante é a tabela Recurso. Ela armazenará os recursos disponíveis nos ambientes, que poderão integrar as regras de acesso. Um exemplo de recursos de um ambiente são os sensores que o mesmo dispõe.

A tabela Evento armazenará todos os eventos realizados no ambiente, possibilitando a validação do sistema de controle de acesso, através da análise dos eventos de entrada e saída de usuários nos ambientes controlados.

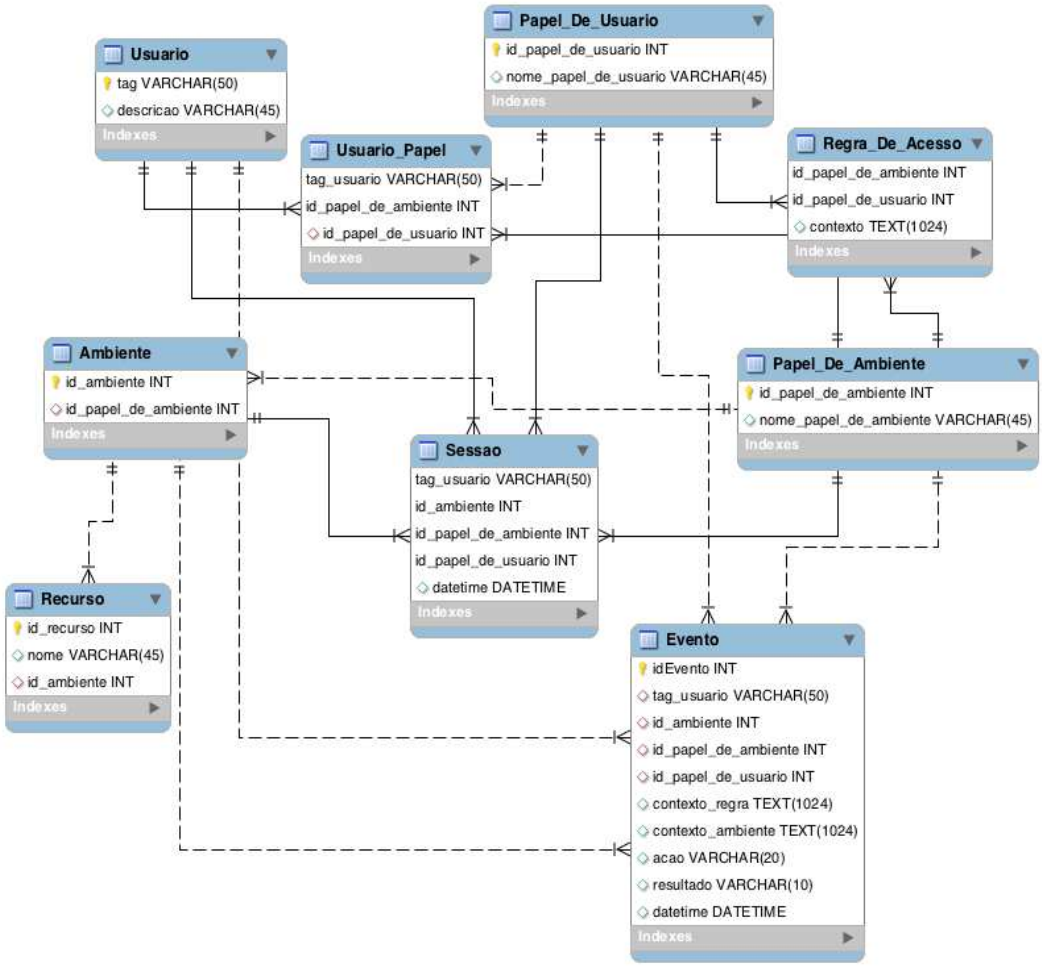

Figura 14 – Modelo relacional

4.5 Representação das Informações

As informações que serão trocadas entre os módulos do sistema, quase que plenamente estarão representadas no formato JSON, exceto o retorno da avaliação da regra de acesso por parte do módulo gateway para o módulo ambiente, que nesse caso será um boolean.

Por ser um formato de simples entendimento e de fácil utilização, o JSON se tornou uma escolha excelente para representar as informações no sistema. Além disso, diversas bibliotecas que trabalham com JSON são facilmente encontradas para diversas linguagens de programação.

Conforme apresentado na subseção 2.5.1, o protocolo MQTT trabalha com tópicos para estabelecer a comunicação entre os dispositivos que se comunicam. Logo o sistema de controle de acesso utilizará alguns tópicos, listados na tabela 3, com o objetivo de manter a comunicação entre os módulos gateway e do ambiente. A figura 15 apresenta o tópico TagIdentificada, o qual envia para o módulo gateway as informações do ambiente quando uma tag RFID é identificada. O módulo gateway se inscreve nesse tópico e o módulo do ambiente publica nesse mesmo tópico. As informações enviadas consistem em: identificador único do ambiente, valor da tag RFID do usuário que foi lida e as informações dos sensores que o ambiente possui.

Os tópicos idAmbiente-Ambiente-Acessando e idAmbiente-Ambiente-Saindo são utilizados para retornar o resultado da avaliação das regras de acesso para o módulo do ambiente. Em termos do protocolo MQTT, o módulo do ambiente se inscreve nos tópicos, onde idAmbiente representa o identificador único do ambiente, e o módulo gateway pública nesses tópicos de acordo com o evento que está acontecendo, acessando ou saindo, para o respectivo ambiente que realizou a leitura da tag RFID do usuário.

Tabela 3 – Tópicos MQTT do sistema

Tópico	Descrição
TagIdentificada	Envia informações do ambiente ao identificar uma tag RFID.
idAmbiente-Ambiente-Acessando	Recebe um boolean True ou False para o acesso ao ambiente.
idAmbiente-Ambiente-Saindo	Recebe um boolean True ou False para a saída do ambiente.

4.5.1 Representação do Contexto

Conforme explanado na seção 2.2, o contexto pode ser qualquer informação que possa ser utilizada para caracterizar a situação de uma entidade. Neste trabalho serão

```
{
    "ambiente": identificadorAmbiente,
    "usuario": tagRFIDUsuario,
    "contexto": {
        identificadorSensor: valorSensor
    }
}
```

Figura 15 – JSON do tópico TagIdentificada

considerados três tipos de informação, mostrados na tabela 4. A figura 16 apresenta a representação de contexto dentro da especificação da regra de acesso associada a um papel de usuário e papel de ambiente. O campo Tipo serve para expor qual o tipo de informação será utilizado. O campo Recurso traz o identificador do recurso do ambiente. Esse campo só será utilizado quando o campo Tipo for Recurso. O campo Valor contém a informação que representará que o contexto é verdadeiro, e o campo operação especifica a operação que será utilizada para verificar se o contexto é verdadeiro ou não em relação ao contexto atual do ambiente. A tabela 5 apresenta as operações consideradas neste trabalho e a figura 17 mostra um exemplo da avaliação de contexto realizada pelo sistema.

Os tipos de informação podem ser combinados para formarem contextos mais complexos e assim tornar a regra de acesso mais complexa e precisa. A figura 18 expõe um exemplo de combinação dos tipos de informação que podem ser utilizadas na regra de acesso.

Importante ressaltar que na avaliação da regra de acesso utilizando contexto com os tipos de informação combinados, conforme a figura 18, a relação lógica entre cada informação é AND, ou seja, caso alguma das informações seja falsa, o acesso ao ambiente não é autorizado.

```
[{
        "Tipo": tipoInformação,
        "Recurso": identificadorRecurso,
        "Valor": valorTipo,
        "Operacao": operaçãoRealizada
    }
]
```

Figura 16 – JSON da representação de contexto simples

Tabela 4 – Tipos de informação de contexto

Tipo
Data
Horário
Recurso

Tabela 5 – Operações com os valores de contexto

Operação	Descrição	Exemplo
Maior	Avalia se o valor do contexto atual do ambiente é maior que o valor do contexto da regra de acesso.	{ "Tipo": tipoInformação, "Recurso": identificadorRecurso,"Valor": valorTipo,"Operacao": "maior"}
Menor	Avalia se o valor do contexto atual do ambiente é menor que o valor do contexto da regra de acesso.	{ "Tipo": tipoInformação, "Recurso": identificadorRecurso,"Valor": valorTipo,"Operacao": "menor"}
Igual	Avalia se o valor do contexto atual do ambiente é igual ao valor do contexto da regra de acesso.	{ "Tipo": tipoInformação, "Recurso": identificadorRecurso,"Valor": valorTipo,"Operacao": "igual"}
Diferente	Avalia se o valor do contexto atual do ambiente é diferente do valor do contexto da regra de acesso.	{ "Tipo": tipoInformação, "Recurso": identificadorRecurso,"Valor": valorTipo,"Operacao": "diferente"}
Between	Avalia se o valor do contexto atual do ambiente está entre os valores da regra de acesso.	{ "Tipo": tipoInformação, "Recurso": identificadorRecurso,"Valor": "valorTipo1, valorTipo2","Operacao": "between"}

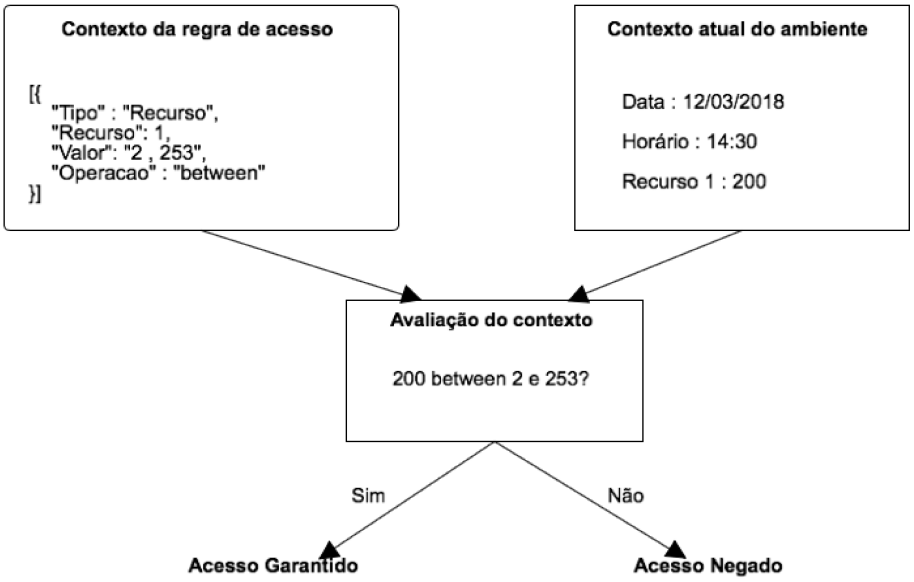

Figura 17 – Exemplo da avaliação de contexto

```
[{
        "Tipo": tipoInformação,
        "Recurso": identificadorRecurso,
        "Valor": valorTipo,
        "Operacao": operaçãoRealizada
    },
    {
        "Tipo": tipoInformação,
        "Recurso": identificadorRecurso,
        "Valor": valorTipo,
        "Operacao": operaçãoRealizada
    }
]
```

Figura 18 – JSON da representação de contexto complexo

5 Ambientes e Resultados Experimentais

Neste capítulo serão apresentados os resultados obtidos através dos experimentos realizados nos ambientes representados pela figura 19, onde os números na figura, 1 e 2, são os identificadores únicos de cada ambiente utilizados pelo sistema. A figura também mostra o posicionamento dos módulos do sistema e a localização dos sensores, os quais são utilizados pelo ambiente 1, conforme a tabela 6.

Para mostrar a validade do sistema de controle de acesso baseado em papéis, foram utilizados dois cenários, cada qual com o seu objetivo específico. Os testes foram realizados durante o período de duas semanas, onde quatro usuários, dois com idades entre 20 e 30 anos, e os outros dois com idades entre 40 e 50 anos, utilizaram o sistema.

Figura 19 – Planta baixa do ambiente de teste

Tabela 6 – Sensores (Recursos) do ambiente 1

Identificador	Descrição
1	Sensor de luminosidade
2	Sensor de distância ultrassônico

A tabela 7 apresenta os papéis de usuário cadastrados no sistema, papéis esses que são característicos de um ambiente domiciliar. Seguindo a mesma vertente, a tabela 8

contém os papéis de ambiente utilizados. Lembrando que os papéis são configuráveis, ou seja, podem ser adicionados, editados ou excluídos.

Tabela 7 – Papéis de usuário

Papel	Identificador
Filho	1
Pai	2
Mãe	3
Convidado	4

Tabela 8 – Papéis de ambiente

Papel	Identificador
Quarto Filho	1
Quarto Pais	2
Sala de Estar	3

5.1 Resultados Gerais

Ao longo de duas semanas de teste, resultados expressivos foram obtidos a fim de mostrar a validade do sistema. Em todas as tentativas de acesso, o sistema se comportou de acordo com o esperado, lendo a tag do usuário, capturando o contexto do ambiente, processando as regras de acesso e verificando se o usuário poderia ou não acessar o ambiente.

Os resultados que serão mostrados a seguir levam em consideração os dois cenários de teste, pois nesta seção o objetivo é mostrar os resultados gerais obtidos no sistema.

A figura 20 expõe o gráfico de utilização do sistema através da quantidade de eventos capturados. Esses eventos consistem em tentativas de acesso nos ambientes por parte dos usuários. Durante o período de testes foram registrados 284 eventos, sendo 240 destes acessos que foram garantidos pelas regras de acesso, e 44 que foram negados.

Já a figura 21 apresenta os eventos registrados no sistema agrupados por horário. Essa informação é importante, pois se consegue ter uma noção da utilização do sistema por horário, e assim obter conclusões importantes. Por exemplo, pode-se observar que por volta das 19 horas o sistema passou por um pico de eventos, o que significa que houveram muitas tentativas de acesso.

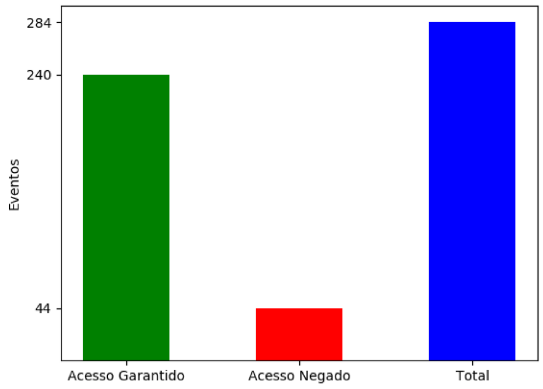

Figura 20 – Eventos no sistema

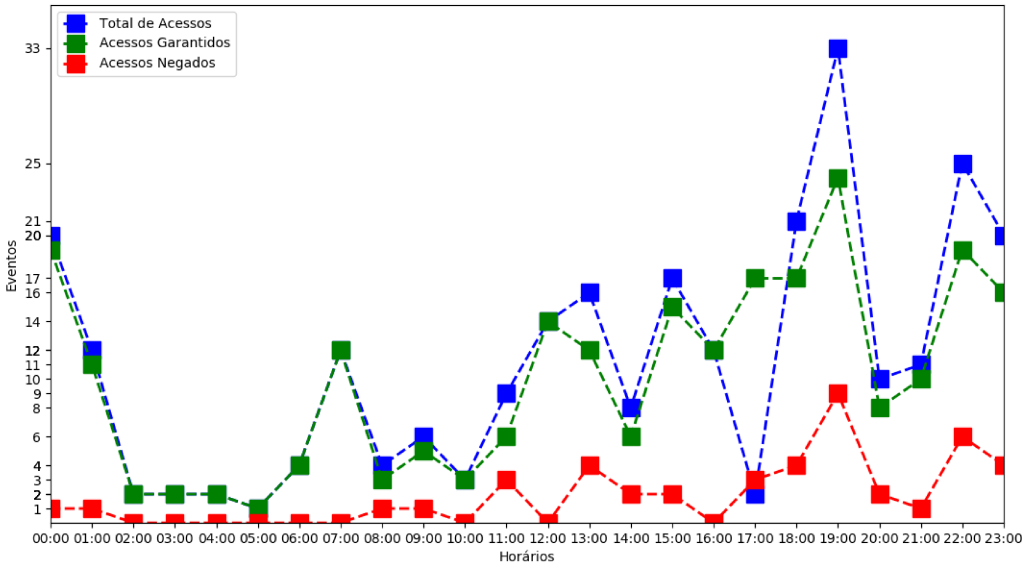

Figura 21 – Eventos no sistema por horário

5.2 Cenário 1

Neste primeiro cenário de teste o objetivo é mostrar que o sistema de fato realiza o controle de acesso baseado em papéis. A tabela 9 e a tabela 10 apresentam as configurações do cenário 1 para os papéis de ambiente Quarto Filho e Quarto Pais, sendo que o ambiente 1 desempenha o papel de ambiente Quarto Filho, e o ambiente 2 desempenha o papel de ambiente Quarto Pais. A tabela 11 mostra as regras de acesso utilizadas no cenário 1, ressaltando que as regras de acesso levam em consideração o papel de ambiente, papel de usuário e as informações de contexto.

A figura 22 deixa explícito o funcionamento do controle de acesso para o cenário 1, mostrando que as regras de acesso definidas na tabela 11 funcionaram de acordo com o esperado, o que era o objetivo a ser alcançado para este cenário de teste. Para reforçar a validade do sistema, a tabela 12 apresenta a quantidade de eventos para cada papel de usuário, levando em consideração os dois ambientes.

Tabela 9 – Configuração para o papel de ambiente Quarto Filho

Usuário	Papel de usuário
Usuário 1	Filho
Usuário 2	Filho
Usuário 3	Pai
Usuário 4	Mãe

Tabela 10 – Configuração para o papel de ambiente Quarto Pais

Usuário	Papel de usuário
Usuário 1	Filho
Usuário 2	Convidado
Usuário 3	Pai
Usuário 4	Mãe

Tabela 11 – Regras de acesso do cenário 1

Papel de Ambiente	Papel de usuário	Regra
Quarto Filho	Filho	Sem Regra (Acesso garantido)
Quarto Filho	Pai	Valor sensor 2 maior que 80; Valor data entre 05/03/2018 e 09/03/2018
Quarto Filho	Mãe	Valor sensor 1 entre 1 e 253; Valor data diferente de 10/03/2018; Valor horário entre 13:00 e 21:00 horas
Quarto Pais	Filho	Valor horário menor que 22:00
Quarto Pais	Convidado	Valor data entre 05/03/2018 e 09/03/2018; Valor horário entre 14:00 e 18:00 horas
Quarto Pais	Pai	Sem Regra (Acesso garantido)
Quarto Pais	Mãe	Sem Regra (Acesso garantido)

Tabela 12 – Total de eventos por papel de usuário

Papel de usuário	Total de eventos	Acesso garantido	Acesso negado
Filho	106	104	2
Convidado	8	0	8
Pai	34	27	7
Mãe	75	63	12

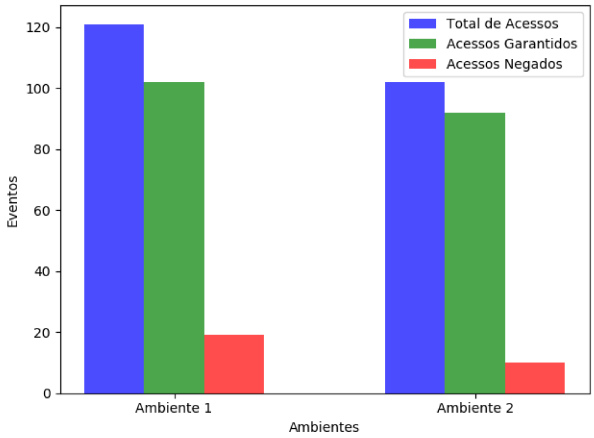

Figura 22 – Eventos nos ambientes do cenário 1

5.3 Cenário 2

O segundo cenário de teste tem como objetivo mostrar que o sistema de controle de acesso continua funcionando, mesmo quando as configurações dos ambientes são alteradas, ou seja, que o sistema funciona de maneira dinâmica, se adaptando às configurações atuais dos ambientes controlados.

Para mostrar que o objetivo foi alcançado, duas etapas foram utilizadas. Na etapa 1 foram utilizadas as configurações apresentadas pela tabela 13, enquanto na etapa 2 foram usadas as configurações contidas na tabela 14. Nesse cenário o papel de ambiente do ambiente 1 foi alterado para o papel de ambiente 3 (Sala de Estar). As associações entre usuários e papéis de usuário para este cenário estão apresentadas na tabela 15.

Tabela 13 – Regras de acesso do cenário 2 - Etapa 1

Papel de Ambiente	Papel de usuário	Regra
Sala de Estar	Filho	Valor horário entre 14:00 e 21:00 horas;
Sala de Estar	Pai	Valor sensor 1 entre 1 e 253; Valor sensor 2 menor que 100;
Sala de Estar	Mãe	Sem Regra (Acesso garantido)
Sala de Estar	Convidado	Valor data entre 14/03/2018 e 18/03/2018; Valor horário entre 14:00 e 18:00 horas

Tabela 14 – Regras de acesso do cenário 2 - Etapa 2

Papel de Ambiente	Papel de usuário	Regra
Sala de Estar	Filho	Valor horário entre 12:00 e 23:00 horas;
Sala de Estar	Pai	Sem Regra (Acesso garantido)
Sala de Estar	Mãe	Sem Regra (Acesso garantido)
Sala de Estar	Convidado	Valor sensor 1 menor que 254; Valor horário entre 13:00 e 20:00 horas

Tabela 15 – Configuração para o papel de ambiente Sala de Estar

Usuário	Papel de usuário
Usuário 1	Filho
Usuário 2	Convidado
Usuário 3	Pai
Usuário 4	Mãe

A figura 23 apresenta os resultados obtidos no cenário 2, destacando a dinamicidade do sistema, que era o objetivo esperado para este cenário. Pode-se perceber que o número de acessos negados é reduzido na segunda etapa. Isso acontece, pois as regras de acesso são alteradas, ou seja, se tornam menos restritivas para alguns usuários, por exemplo o usuário 3. Por este motivo mais acesso foram garantidos, mostrando que o sistema realiza o controle de acesso de forma correta, mesmo quando as configurações são alteradas.

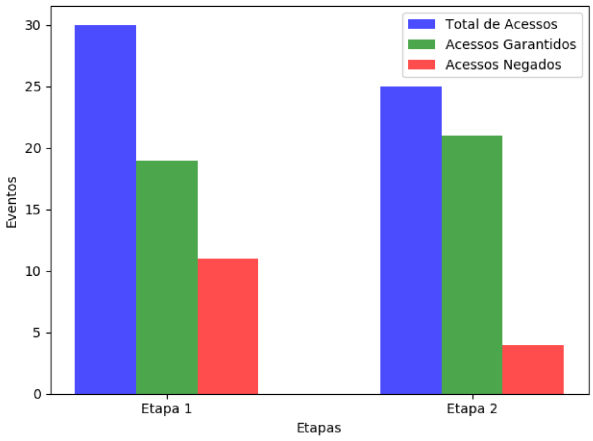

Figura 23 – Eventos do cenário 2 - Etapa 1 e 2

5.4 Comparando os cenários 1 e 2

Procurando deixar mais explícito o funcionamento e a dinamicidade do sistema de controle de acesso, as configurações do ambiente 1 para o cenário 1 e 2 serão comparadas. Lembrando que para o cenário 1 o ambiente 1 desempenhou o papel de ambiente 1 (Quarto Filho), e para o cenário 2 o papel de ambiente 3 (Sala de Estar).

Na figura 24 se pode observar os acessos agrupados por horário, o que comprova que a mudança de papel de ambiente também alterou a forma que o ambiente é utilizado, pois em horários, como por exemplo, entre as 2 e 5 da manhã, desempenhando o papel de ambiente 1, houveram por volta de 6 acessos, enquanto que desempenhando o papel de ambiente 3 foram 0 acessos.

Para continuar reforçando que o sistema de fato funciona, a figura 25 apresenta os totais de acessos (ou seja, acessos garantidos e negados) para o ambiente 1 desempenhando os dois papéis. Pode-se concluir que, ao desempenhar o papel de ambiente 1 (Quarto Filho), o ambiente 1 obteve uma taxa maior de utilização, ou seja, os usuários utilizaram mais o ambiente 1 quando era Quarto Filho do que quando era Sala de Estar.

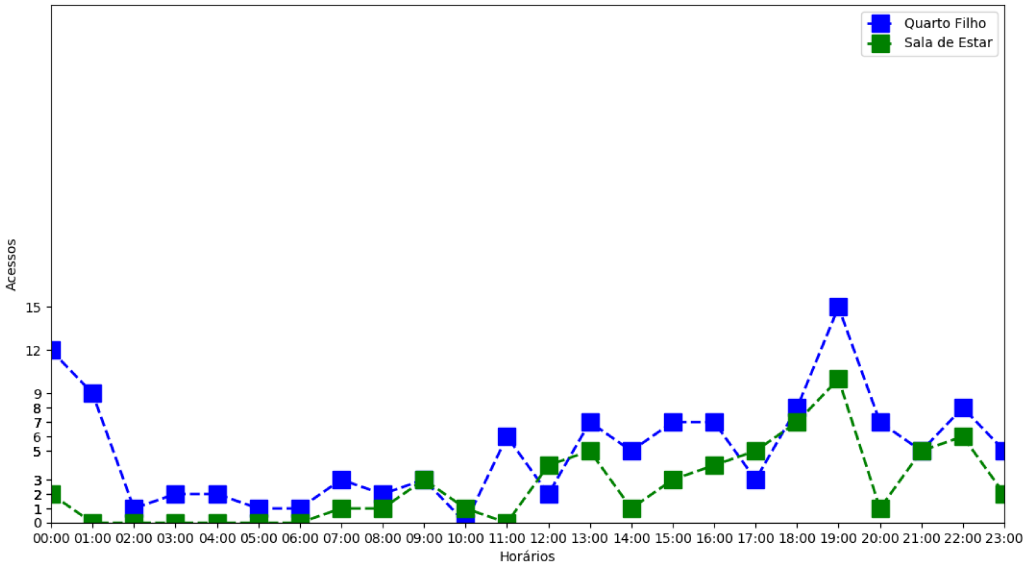

Figura 24 – Acessos no ambiente 1 por horário e por papel de ambiente

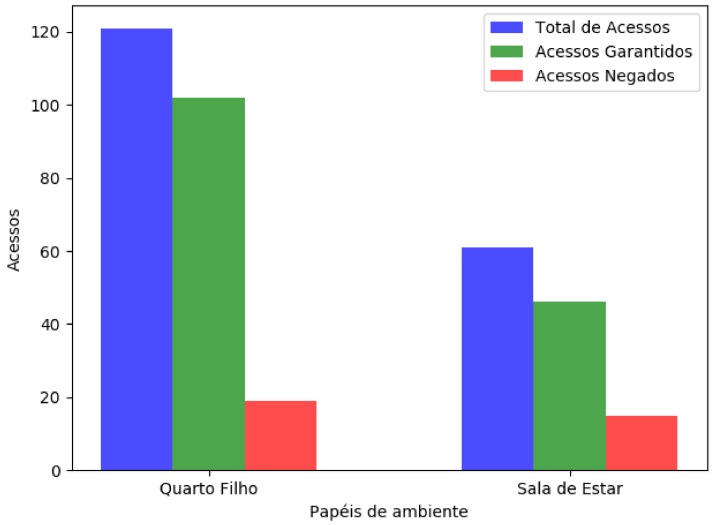

Figura 25 – Acessos no ambiente 1 por papel de ambiente

6 Conclusão e Trabalhos Futuros

Com este trabalho foi possível analisar diversos aspectos relacionados à área de controle de acesso. Sendo assim identificou-se que a abordagem baseada em papéis utilizando informações de contexto, é pouco empregada e explorada em ambientes assistidos, e por este motivo decidiu-se utilizar essa abordagem neste trabalho, implementando um sistema capaz de controlar o acesso, se baseando em papéis de usuário e ambientes em conjunto com informações de contexto.

Durante a etapa de testes e análise dos dados gerados pelo sistema, o controle de acesso apresentou informações extremamente importantes sobre o ambiente monitorado. Além das informações básicas, como quantidade de acesso, acessos negados e garantidos, as quais foram utilizadas para compor os resultados obtidos neste trabalho. O sistema de controle de acesso expôs uma série de informações que são obtidas implicitamente quando se analisa os eventos do sistema. Por exemplo, através dos eventos se pode ter uma noção da utilização do ambiente, e assim mensurar o consumo dos recursos em certo horário, ou taxa de ocupação do ambiente. Outra informação interessante que foi observada durante a análise dos dados gerados pelo sistema de controle de acesso, é que se pode obter a localização atual ou a localização em certo momento de um ou mais usuários.

Conforme discutido no capítulo 5. mostrou-se que o sistema proposto neste trabalho de fato realiza o controle de acesso, o que confirma que o objetivo principal foi alcançado. Ressaltando que o sistema funcionou de maneira dinâmica, ou seja, se adaptando às mudanças de papéis, e concisa, garantido e negando o acesso aos usuários. Consequentemente se conclui que o controle de acesso baseado em papéis é uma abordagem aplicável e eficaz na segurança de ambientes assistidos.

Para finalizar serão propostos alguns tópicos para a evolução do sistema de controle de acesso:

- Conforme explicado na subseção 4.5.1, as informações de contexto que compõem um contexto complexo na regra de acesso, são interligadas utilizando a operação lógica E, sendo assim todas as informações de contexto precisam ser verdadeiras para que o acesso seja garantido. Portanto se propõe implementar no sistema as outras operações lógicas, como por exemplo as operações OU e NEGAÇÃO.

- Para que o sistema se torne mais completo, além da utilização do RBAC núcleo, é importante adicionar ao sistema os outros módulos do padrão de controle de acesso baseado em papéis, ou seja, o RBAC Hierárquico, e as separações estática e dinâmica de relações de serviço. Módulos esses que foram discutidos na subseção 2.1.3.

- Outro aspecto importante no controle de acesso é ter um sistema capaz de gerenciá-lo, pois assim se torna mais fácil a inclusão, remoção e edição dos papéis, regras de acesso e outros componentes que compõem o controle de acesso.

Referências

AL-FUQAHA, A. et al. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys Tutorials*, v. 17, n. 4, p. 2347–2376, Fourthquarter 2015. ISSN 1553-877X. 28

AUGUSTO, J. C. Ambient Intelligence: the Confluence of Ubiquitous/Pervasive Computing and Artificial Intelligence . *Intelligent Computing Everywhere*, n. January 2007, p. 1–259, 2007. 27

BROEK, G. V. D.; CAVALLO, F.; WEHRMANN, C. *AALIANCE Ambient Assisted Living Roadmap*. Amsterdam, The Netherlands, The Netherlands: IOS Press, 2010. ISBN 1607504987, 9781607504986. 26, 27

BUCKLEY, J. From RFID to the Internet of Things: Pervasive networked systems. *Proceedings of the Pervasive Networked Systems Conference*, n. March, p. 32, 2006. 25

CONTROLE de luz utilizando LDR. 2013. Disponível em: ⟨https://www.arduinoecia.com.br/2013/09/controle-de-luz-utilizando-ldr.html⟩. Acesso em: 22 de fevereiro de 2018.

COVINGTON, M. J. et al. Securing context-aware applications using environment roles. *Proceedings of the sixth ACM symposium on Access control models and technologies*, n. January, p. pp. 10–20, 2001. Disponível em: ⟨http://doi.acm.org/10.1145/373256.373258⟩. 30, 34

DEY, A. K. Understand and using context. 2001. 25

ECLIPSE Mosquitto Home Page. 2017. Disponível em: ⟨https://mosquitto.org/⟩. Acesso em: 01 de fevereiro de 2018.

ECLIPSE Mosquitto Overview. 2017. Disponível em: ⟨https://projects.eclipse.org/projects/technology.mosquitto⟩. Acesso em: 01 de fevereiro de 2018. 29

FERRAIOLO, D. F.; KUHN, R. D. Role based access control. American National Standards Institute, Inc, 2004. 22

FLOP, F. *Arduino UNO3*. 2017. Disponível em: ⟨https://www.filipeflop.com/wp-content/uploads/2017/07/Arduino_Uno_R3.png⟩. Acesso em: 01 de fevereiro de 2018. 37

FLOP, F. *Ethernet Shield*. 2017. Disponível em: ⟨https://www.filipeflop.com/wp-content/uploads/2017/07/1000701-1.jpg⟩. Acesso em: 01 de fevereiro de 2018. 37

FLOP, F. *rfid imagem*. 2017. Disponível em: ⟨https://www.filipeflop.com/wp-content/uploads/2017/07/450xN-33.jpg⟩. Acesso em: 01 de fevereiro de 2018. 36

FLOP, F. *Sensor Luminosidade*. 2017. Disponível em: ⟨https://www.filipeflop.com/wp-content/uploads/2017/07/450xN-7.jpg⟩. Acesso em: 01 de fevereiro de 2018. 38

FLOP, F. *Sensor Ultrasônico*. 2017. Disponível em: ⟨https://www.filipeflop.com/wp-content/uploads/2017/07/58594_16826.jpg⟩. Acesso em: 01 de fevereiro de 2018. 38

GERSHENFELD, N.; KRIKORIAN, R.; COHEN, D. *The Internet of Things*. [S.l.: s.n.], 2004. v. 291. 76–81 p. ISSN 0036-8733. ISBN 00368733.

HOSSAIN, M. A. et al. From Sensing to Alerting: A Pathway of RESTful Messagin in Ambient Assisted Living. 2016.

HUNKELER, U.; TRUONG, H. L.; STANFORD-CLARK, A. MQTT-S - A Publish/Subscribe Protocol For Wireless Sensor Networks.

JIH, W.-r. et al. Context-aware Access Control in Pervasive Healthcare. *Context*, n. February 2014, p. 2–9, 2005. Disponível em: ⟨http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.61.772⟩.

JUNIOR, V. A.; SANTIN, A. O. Modelo de ativação multi-domínios de papéis RBAC usando controle de acesso baseado em atributos. *XV Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg)*, 2015.

KAYES, A. S. M. et al. A Policy Model and Framework for Context-Aware Access Control to Information Resources. 2017. 31

KLEINBERGER, T. et al. Ambient intelligence in assisted living: Enable elderly people to handle future interfaces. In: *Proceedings of the 4th International Conference on Universal Access in Human-computer Interaction: Ambient Interaction*. Springer-Verlag, 2007. (UAHCI'07). ISBN 978-3-540-73280-8. Disponível em: ⟨http://dl.acm.org/citation.cfm?id=1763296.1763308⟩. 26, 27

LACERDA, F.; LIMA-MARQUES, M. Da necessidade de princípios de Arquitetura da Informação para a Internet das Coisas. 2015.

MARTINS, I. R.; ZEM, J. L. Estudo dos protocolos de comunicação MQTT e CoAP para aplicações Machine-To-Machine e Internet das Coisas. 2014. 27

ONU. *World Population Ageing*. 2017. Disponível em: ⟨http://www.un.org/en/development/desa/population/publications/pdf/ageing/WPA2017_Infochart.pdf⟩. Acesso em: 04 de fevereiro de 2018. 17

PARK, S.; HAN, Y.; CHUNG, T. Context-role based access control for context-aware application. *High Performance Computing and Communications*, p. 572–580, 2006. ISSN 16113349. 30

PRIYA, P. et al. Context-Aware Architecture for User Access Control. v. 2, n. 3, p. 201–204, 2014.

ROTTA, G.; CHARãO, A.; DANTAS, M. Um Estudo sobre Protocolos de Comunicacao para Ambientes de Internet das Coisas. 2017. 27

SAINT-EXUPERY, A. Internet of Things Strategic Research Roadmap. 2009.

SANDHU, R. S. et al. Role-Based Access Control Models. *IEEE Computer*, v. 29, n. 2, p. 38–47, 1995. ISSN 00189162.

SANDHU, R. S.; SAMARATI, P. Access control - principles and pratice. IEEE Communications Magazine, 1994. 20

SILVA, M. P. D. et al. Implementação da IOT para o Monitoramento das Variáveis Meteorológicas num AAL. 2016.

TANIKAWA, T. C. V. Reconhecimento e Localização de Indivíduos com Utilização de Sensores no Suporte aos Ambientes Assistidos. 2016.

THOMSEN, A. *Como conectar o Sensor Ultrassônico HC-SR04 ao Arduino*. 2011. Disponível em: ⟨https://www.filipeflop.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/⟩. Acesso em: 20 de fevereiro de 2018. 38

THOMSEN, A. *Controle de Acesso usando Leitor RFID com Arduino*. 2014. Disponível em: ⟨https://www.filipeflop.com/blog/controle-acesso-leitor-rfid-arduino/⟩. Acesso em: 20 de fevereiro de 2018.

TORĞUL, B.; ŞAğBANşUA, L.; BALO, F. B. Internet of Things: A Survey. *International Journal of Applied Mathematics, Electronics and Computers*, n. December 2016, p. 104–104, 2016. ISSN 2147-8228.

TRNKA, M.; CERNY, T. Context-aware Role-based Access Control Using Security Levels. *Proceedings of the 2015 Conference on Research in Adaptive and Convergent Systems*, p. 280–284, 2015.

WHAT is MQTT? Disponível em: ⟨http://mqtt.org/faq⟩. Acesso em: 01 de fevereiro de 2018.

WHITMORE, A.; AGARWAL, A.; Da Xu, L. The Internet of Things—A survey of topics and trends. *Information Systems Frontiers*, v. 17, n. 2, p. 261–274, 2015. ISSN 15729419. 26

YUAN, M. *Conhecendo o MQTT*. 2017. Disponível em: ⟨https://www.ibm.com/developerworks/br/library/iot-mqtt-why-good-for-iot/index.html⟩. Acesso em: 01 de fevereiro de 2018. 28

ZHANG, G. et al. Context-aware dynamic access control for pervasive applications. *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference*, p. 21–30, 2004. 30

ZHOU, Z.; WU, L.; HONG, Z. Context-Aware Access Control Model for Cloud Computing. *International Journal of Grid and Distributed Computing*, v. 6, n. 6, p. 1–12, 2013. ISSN 2005-4262.

Apêndices

APÊNDICE A – Códigos Fonte

A.1 Módulo do Ambiente

```
#include <MFRC522.h>
#include <SPI.h>
#include <Ethernet.h>
#include <PubSubClient.h>
#include <Ultrasonic.h>

// configurações de rede
byte mac[]    = {  0xDE, 0xED, 0xBA, 0xFE, 0xFE, 0xED }; // endereço MAC
IPAddress ip(192, 168, 1, 120); // ip da interface de rede

// configurações MQTT
#define MQTT_BROKER  "192.168.1.100"  //URL do broker MQTT
#define MQTT_PORT  1883  // Porta do Broker MQTT
#define MQTT_ID  "ambiente_1"
#define MQTT_TOPICO_SUBSCRIBE_ACESSO "1-Ambiente-Acessando"     //escuta
↪  o que é publicado para o topica
#define MQTT_TOPICO_SUBSCRIBE_SAIDA "1-Ambiente-Saindo"     //escuta o
↪  que é publicado para o topica
#define MQTT_TOPICO_PUBLISH   "TagIdentificada"      //publica no
↪  topico
char message_buff[100];

// configurações RFID
constexpr uint8_t RST_PIN = 9;
constexpr uint8_t SS_PIN = 8;
MFRC522 mfrc522(SS_PIN, RST_PIN);  // instancia  MFRC522
// configurações SENSORES
int portaLDR = A5;

// configuração LEDS
int portaLEDVerde = 7;
int portaLEDVermelho = 5;

```

```
31 // configuração do sensor ultrasônico
32 #define pino_trigger 2
33 #define pino_echo 3
34 Ultrasonic ultrasonic(pino_trigger, pino_echo); //Inicializa o sensor nos
   ↪  pinos definidos acima
35
36 EthernetClient ethClient;
37 PubSubClient mqttClient(ethClient);
38
39 void setup() {
40   // put your setup code here, to run once:
41   Serial.begin(9600); // Inicia a serial
42   SPI.begin();// Inicia  SPI bus
43   pinMode(portaLEDVerde, OUTPUT);
44   pinMode(portaLEDVermelho, OUTPUT);
45   setupRfid();
46   setupMqtt();
47   setupEthernet();
48   Serial.println("* Aproxime o cartão *");
49 }
50 // inicia o RFID
51 void setupRfid(){
52   mfrc522.PCD_Init(); // Inicia MFRC522
53  // mfrc522.PCD_DumpVersionToSerial();
54 }
55 // inicia o MQTT
56 void setupMqtt(){
57   Serial.println("* MQTT");
58   mqttClient.setServer(MQTT_BROKER, MQTT_PORT);
59   mqttClient.setCallback(callbackMqtt);
60 }
61
62 // inicia a rede
63 void setupEthernet(){
64   Serial.println("* ETHERNET ");
65   Ethernet.begin(mac, ip);
66   Serial.println(Ethernet.localIP());
67 }
68
```

```
69 // refaz a conexao com o broker
70 void reconectMqtt(){
71   while (!mqttClient.connected())
72     {
73         Serial.println("* Tentando se conectar ao Broker MQTT: ");
74         Serial.println(MQTT_BROKER);
75         if (mqttClient.connect("1"))
76         {
77             Serial.println("Conectado com sucesso ao broker MQTT!");
78             mqttClient.subscribe(MQTT_TOPICO_SUBSCRIBE_ACESSO);
79             mqttClient.subscribe(MQTT_TOPICO_SUBSCRIBE_SAIDA);
80         }
81         else
82         {
83             Serial.println("Falha ao reconectar no broker.");
84             Serial.println("Havera nova tentatica de conexao em 2s");
85             delay(2000);
86         }
87     }
88 }
89
90 // recebe os dados do broker
91 void callbackMqtt(char* topic, byte* payload, unsigned int length) {
92   Serial.println("MENSAGEM RECEBIDA");
93   String resposta = "";
94   for (int i=0;i<length;i++) {
95     resposta += (char)payload[i];
96   }
97
98   if (resposta == "True"){
99     Serial.println("True");
100    acendeLedVerde();
101  }else{
102    Serial.println("False");
103    acendeLedVermelho();
104  }
105 }
106
107 // envia as informações para o broker
```

```
108 void enviaInformacoes(String msg){
109   Serial.println("MENSAGEM ENVIADA");
110   msg.toCharArray(message_buff, msg.length()+1);
111   mqttClient.publish(MQTT_TOPICO_PUBLISH,message_buff);
112 }
113
114 // retorna o valor do sensor de luminosidade (quanto maior o valor menos
    ↪  luminosidade tem no ambiente)
115 float getValorLuminosidade(){
116   int valor = analogRead(portaLDR);
117   return map(valor,0,1023,0,255);
118 }
119
120 // pega o valor do sensor ultrasônico em centímetros
121 float getValorSensorUltrasonico(){
122     long microsec = ultrasonic.timing();
123     return ultrasonic.convert(microsec, Ultrasonic::CM);
124 }
125
126 // acende o led verde por 3 segundos
127 void acendeLedVerde(){
128   digitalWrite(portaLEDVerde, HIGH);
129   delay(3000);
130   digitalWrite(portaLEDVerde, LOW);
131 }
132
133 // acende o led vermelho por 3 segundos
134 void acendeLedVermelho(){
135   digitalWrite(portaLEDVermelho, HIGH);
136   delay(3000);
137   digitalWrite(portaLEDVermelho, LOW);
138 }
139
140 // acende todos os leds
141 void acendeAllLeds(){
142   digitalWrite(portaLEDVermelho, HIGH);
143   digitalWrite(portaLEDVerde, HIGH);
144   delay(1000);
145   digitalWrite(portaLEDVermelho, LOW);
```

```
146   digitalWrite(portaLEDVerde, LOW);
147
148 }
149 void loop() {
150   // verificando a conexao com o mqtt
151   if (!mqttClient.connected()) {
152     reconectMqtt(); // reconeccao do mqtt
153   }
154
155   // Look for new cards
156   if (mfrc522.PICC_IsNewCardPresent() and mfrc522.PICC_ReadCardSerial())
157   {
158     String conteudo= "";
159     byte letra;
160     for (byte i = 0; i < mfrc522.uid.size; i++)
161     {
162        //Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : "");
163        //Serial.print(mfrc522.uid.uidByte[i], HEX);
164        conteudo.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? "0" : ""));
165        conteudo.concat(String(mfrc522.uid.uidByte[i], HEX));
166     }
167     acendeAllLeds();
168     float valorSensorUltrasonico = getValorSensorUltrasonico();
169     Serial.println(valorSensorUltrasonico);
170     float valorSensorLuminosidade = getValorLuminosidade();
171     Serial.println(valorSensorLuminosidade);
172     String pubInfo = "{\"ambiente\": \"1\",\"usuario\": \""+conteudo+"\" ,
      ↪ \"contexto\" :
      ↪ {\"1\":"+valorSensorLuminosidade+",\"2\":"+valorSensorUltrasonico+"}}";
173     enviaInformacoes(pubInfo);
174     delay(5000);
175   }
176   mqttClient.loop();
177 }
```

A.2 Módulo Gateway

A.2.1 Classe Gateway

Principal classe do sistema

```
1  #import classes
2  import json
3  import time
4  import pickle
5
6  # import classe Grenciador
7  from gerenciador.Gerenciador import Gerenciador
8  # import classe Avaliador
9  from avaliador.Avaliador import Avaliador
10 # import classe Persistencia
11 from persistencia.Persistencia import Persistencia
12
13 class Gateway:
14
15   def __init__(self):
16
17     # sub módulo de avaliação da politica de acesso
18     self.avaliador = Avaliador(self)
19
20     # sub módulo de persistência
21     self.persistencia = Persistencia(self)
22
23     # sub módulo de gerenciamento
24     self.gerenciador = Gerenciador(self)
25     self.gerenciador.start()
26
27   def tagIdentificada(self, info):
28     resultadoAvaliacao = False
29     retorno = {}
30     contextoAtual = None
31     acao = None
32
33     # deserializando o json que vem do ambiente
34     infoDeserializada = json.loads(info)
```

```
35
36      # infomacoes de acesso (que vem da sessao se o usuário estiver no
        ↪ ambiente)
37      informacoesAcesso =
        ↪ self.persistencia.temSessao(infoDeserializada['usuario'],
        ↪ infoDeserializada['ambiente'])
38
39      # verificando se o usuario tem uma sessao para o ambiente que esta
        ↪ tentando acessar
40      if informacoesAcesso == None :
41        print("##### Usuário entrando no ambiente #####")
42        acao = "acessando"
43        # contem todas as informações relacionadas ao acesso (usuario,
          ↪ ambiente, papel usuario, papel ambiente, regra de acesso)
44        informacoesAcesso =
          ↪ self.persistencia.getInformacoesAcesso(infoDeserializada['usuario'],
          ↪ infoDeserializada['ambiente'])
45        print("Informações de Acesso: %s" % str(informacoesAcesso))
46        # verificando se retornou informacoes do banco de dados
47        if informacoesAcesso != None:
48          # montando o contexto atual
49          contextoAtual = {}
50          contextoAtual['Data'] = time.strftime('%d/%m/%Y')
51          contextoAtual['Horario'] = time.strftime('%H:%M')
52          contextoAtual['Recurso'] = infoDeserializada['contexto']
53          #print(infoDeserializada['contexto']['3'])
54          resultadoAvaliacao =
            ↪ self.avaliador.avaliar(contextoAtual,informacoesAcesso['contexto'])
55
56          if resultadoAvaliacao == True:
57            # registrando a sessao para o usuario no ambiente
58            if self.persistencia.criarSessao(informacoesAcesso) == False:
59              resultadoAvaliacao = False
60        else:
61          resultadoAvaliacao = False
62
63        print("RESULTADO : " + str(resultadoAvaliacao))
64
65      else:
```

```
66        print("##### Usuário saindo no ambiente #####")
67        acao = "saindo"
68        # deletando sessao
69        if self.persistencia.deletarSessao(infoDeserializada['usuario'],
          ↪  infoDeserializada['ambiente']) == False:
70          resultadoAvaliacao = False
71        else:
72          resultadoAvaliacao = True
73
74      # enviando o resultado de volta para o ambiente
75      retorno['ambiente'] = infoDeserializada['ambiente']
76      retorno['acao'] = acao
77      retorno['resultado'] = resultadoAvaliacao
78      # registrando evento
        ↪    self.registrarEvento(informacoesAcesso,contextoAtual,acao,resultadoAval
79      # retornando para o ambiente
80      self.retornarAcesso(retorno)
81
82    def retornarAcesso(self, info):
83      self.gerenciador.enviarMensagem(info)
84
85    def registrarEvento(self, informacoesAcesso, contextoAtual, acao,
      ↪  resultado):
86      print("EVENTO")
87      usuario = "ERRO"
88      ambiente = "ERRO"
89      id_papel_de_ambiente = "ERRO"
90      id_papel_de_usuario = "ERRO"
91      contexto_regra = "SEM REGRA"
92      contexto_ambiente = "SEM REGRA"
93      resultadoAvaliacao = "ERRO"
94      if informacoesAcesso != None:
95        usuario = informacoesAcesso['usuario']
96        ambiente = informacoesAcesso['ambiente']
97        id_papel_de_ambiente = informacoesAcesso['id_papel_de_ambiente']
98        id_papel_de_usuario = informacoesAcesso['id_papel_de_usuario']
99        resultadoAvaliacao = resultado
100
101       # verificando informações de contexto ao acessar o ambiente
```

```
102         if acao == "acessando":
103           if informacoesAcesso['contexto'] != None:
104             contexto_regra = str(json.loads(informacoesAcesso['contexto']))
105           if contextoAtual != None:
106             contexto_ambiente = str(contextoAtual)
107           else:
108             contexto_ambiente = ""
109       # registrando evento
110       if self.persistencia.registrarEvento(usuario, ambiente,
          ↪ id_papel_de_ambiente,id_papel_de_usuario, contexto_regra,
          ↪ contexto_ambiente,acao,resultadoAvaliacao) != False:
111         return True
112       return False
```

A.2.2 Classe Gerenciador

Classe responsável pelo gerenciamento das mensagens trocadas entre os módulos

```
1  import paho.mqtt.client as mqtt
2  #import configs
3  from Config import MQTT_INFO,MQTT_TOPICOS_GERAIS
4  #import classes
5  import Gateway
6  class Gerenciador:
7
8      def __init__(self, gateway):
9          self.gateway = gateway
10         self.MQTT_ADDRESS = MQTT_INFO['host']
11         self.MQTT_PORT =  MQTT_INFO['porta']
12         self.MQTT_TIMEOUT =  MQTT_INFO['timeout']
13
14     def start(self):
15         self.client = mqtt.Client()
16         self.client.on_connect = self.on_connect
17         self.client.on_subscribe = self.on_subscribe
18         self.client.on_message = self.on_message
19         self.client.connect(self.MQTT_ADDRESS, self.MQTT_PORT,self.
           ↪ MQTT_TIMEOUT)
20         self.client.subscribe('TagIdentificada')
21         self.client.loop_forever()
```

```
22
23      def enviarMensagem(self,info):
24          print('##### Enviando Mensagem para o ambiente #####')
25          print ('Informação: %s' % str(info))
26          result, mid = self.client.publish(info['ambiente']+"-"+
            ↪  MQTT_TOPICOS_GERAIS[info['acao']], info['resultado'])
27          print('Mensagem enviada ao canal: %d' % mid)
28
29      # MQTT METODOS
30      def on_connect(self,client, userdata, flags, rc):
31          print('Conectado. Resultado: %s' % str(rc))
32
33      def on_subscribe(self,client, userdata, mid, granted_qos):
34          print('Inscrito no tópico: %d' % mid)
35
36      def on_message(self,client, userdata, msg):
37          print('##### MENSAGEM RECEBIDA #####')
38          print('Tópico: %s' % msg.topic)
39
40          if msg.topic == 'TagIdentificada':
41              info = msg.payload.decode('utf-8')
42              print("Mensagem recebida do módulo de ambiente: %s" % info)
43              self.gateway.tagIdentificada(info)
44          else:
45              print('Tópico desconhecido.')
```

A.2.3 Classe Avaliador

Realiza a verificação da regra de acesso

```
1  # import classes
2  import json
3  # import Gateway
4  import Gateway
5  # import classe de tipo
6  from avaliador.Tipo import Tipo
7  # import classe de operação
8  from avaliador.Operacao import Operacao
9
10 class Avaliador:
```

```
11
12      def __init__(self, gateway):
13          self.gateway = gateway
14          self.tipo = Tipo()
15          self.operacao = Operacao()
16
17      def avaliar(self, contextoAtual, contextoRegra):
18          print("##### Avaliando contexto #####")
19          print("Contexto Atual : %s" % str(contextoAtual))
20          print("Contexto da Regra : %s" % str(contextoRegra))
21
22          # verifica se o usuario tem acesso garantido independente de
            ↪  contexto
23          if contextoRegra == None:
24              return True
25          else:
26              avaliacao = False
27              contextoRegra = json.loads(contextoRegra)
28              for c in contextoRegra:
29
30                  # convertendo os valores de acordo com o seu tipo
31                  valorConvertidoRegra,valorConvertidoAtual =
                    ↪  getattr(self.tipo,c['Tipo'])(c['Valor'],
                    ↪  contextoAtual[c['Tipo']], c['Recurso'])
32
33                  # avaliando os valores de acordo com a operacao
34                  avaliacao =
                    ↪  getattr(self.operacao,c['Operacao'])(valorConvertidoRegra,valorConve
35                  print("Avaliação : " + str(avaliacao))
36                  if avaliacao == False:
37                      break
38              return avaliacao
```

A.2.4 Classe Operação

Responsável por realizar as operações disponíveis para avaliação das regras de acesso

```
1  class Operacao:
2
```

```
3       def igual(self, valorRegra, valorAtual):
4           print("Operação: IGUAL")
5           if valorRegra == valorAtual:
6               return True
7           else:
8               return False
9
10      def diferente(self, valorRegra, valorAtual):
11          print("Operação: DIFERENTE")
12          if valorRegra != valorAtual:
13              return True
14          else:
15              return False
16
17      def maior (self, valorRegra, valorAtual):
18          print("Operação: MAIOR")
19          if valorRegra > valorAtual:
20              return True
21          else:
22              return False
23
24      def menor (self, valorRegra, valorAtual):
25          print("Operação: MENOR")
26          if valorRegra < valorAtual:
27              return True
28          else:
29              return False
30
31      def between(self, valorRegra, valorAtual):
32          print("Operação: BETWEEN")
33          if valorAtual >= valorRegra[0] and valorAtual <= valorRegra[1]:
34              return True
35          else:
36              return False
```

A.2.5 Classe Tipo

Realiza o parse dos dados de acordo com o seu tipo

```
#import classes
import time

class Tipo:

   # convertendo as datas
    def Data(self, *args):
        print("#### Tipo: DATA ######")
        dataRegra = args[0]
        dataAtual = args [1]
        if ',' in dataRegra:
            dataRegra1, dataRegra2 = dataRegra.split(",")
            dataRegra1 = time.strptime(dataRegra1,'%d/%m/%Y')
            dataRegra2 = time.strptime(dataRegra2,'%d/%m/%Y')

            dataRegra = [dataRegra1,dataRegra2]
            dataAtual = time.strptime(dataAtual,'%d/%m/%Y')
            return dataRegra,dataAtual
        else:
            dataRegra = time.strptime(dataRegra,'%d/%m/%Y')
            dataAtual = time.strptime(dataAtual,'%d/%m/%Y')
            return dataRegra,dataAtual

    # convertendo os tempos
    def Horario(self, *args):
        print("#### Tipo: Horario ######")
        tempoRegra = args[0]
        tempoAtual = args[1]
        if ',' in tempoRegra:
            tempoRegra1, tempoRegra2 = tempoRegra.split(",")
            tempoRegra1 = time.strptime(tempoRegra1,'%H:%M')
            tempoRegra2 = time.strptime(tempoRegra2,'%H:%M')

            tempoRegra = [tempoRegra1,tempoRegra2]
            tempoAtual = time.strptime(tempoAtual,'%H:%M')
            return tempoRegra,tempoAtual
        else:
            tempoRegra = time.strptime(tempoRegra,'%H:%M')
            tempoAtual = time.strptime(tempoAtual,'%H:%M')
```

```
40              return tempoRegra,tempoAtual
41
42      def Recurso(self, *args):
43          print("#### Tipo: RECURSO ######")
44          recursoRegra = args[0]
45          recursoAtual = args[1]
46          idRecurso = args[2]
47
48
49          if type(recursoRegra) == str:
50
51              recursoAtual = recursoAtual[idRecurso]
52              if ',' in recursoRegra:
53                  recursoRegra1, recursoRegra2 = recursoRegra.split(",")
54                  recursoRegra =
                    ↪  [float(recursoRegra1),float(recursoRegra2)]
55                  recursoAtual = float(recursoAtual)
56
57              return recursoRegra, recursoAtual
58          else:
59              recursoAtual = recursoAtual[idRecurso]
60              return float(recursoRegra), float(recursoAtual)
```

A.2.6 Classe Persistência

Realiza toda a persistência de dados do sistema

```
1  import pymysql
2  #import configs
3  from Config import BD_INFO
4  #import classes
5  import Gateway
6  class Persistencia:
7
8      def __init__(self, gateway):
9          self.gateway = gateway
10         # conexao com o banco de dados
11         self.bd =
           ↪  pymysql.connect(BD_INFO['host'],BD_INFO['usuario'],BD_INFO['senha'],BD_
12         self.cursor = self.bd.cursor()
```

```
13
14      # obtem as informacoes associadas ao usuario e ao ambiente
15      def getInformacoesAcesso(self, usuario, ambiente):
16          retorno = {}
17          retorno['ambiente'] = ambiente
18          retorno['usuario'] = usuario
19
20          query = "SELECT pa.id_papel_de_ambiente,
            ↪  pa.nome_papel_de_ambiente, up.id_papel_de_usuario,
            ↪  pu.nome_papel_de_usuario, ra.contexto FROM Regra_De_Acesso ra
            ↪  INNER JOIN Ambiente a ON a.id_ambiente = "+ambiente+" INNER
            ↪  JOIN Papel_De_Ambiente pa ON pa.id_papel_de_ambiente =
            ↪  a.id_papel_de_ambiente INNER JOIN Usuario_Papel up ON
            ↪  up.tag_usuario = \""+usuario+"\" AND up.id_papel_de_ambiente
            ↪  = pa.id_papel_de_ambiente INNER JOIN Papel_De_Usuario pu ON
            ↪  pu.id_papel_de_usuario = up.id_papel_de_usuario WHERE
            ↪  ra.id_papel_de_ambiente = pa.id_papel_de_ambiente AND
            ↪  ra.id_papel_de_usuario = pu.id_papel_de_usuario;"
21          try:
22              print("##### Buscando informações da regra de acesso no banco
                ↪  de dados #####")
23              self.cursor.execute(query)
24              result = self.cursor.fetchone()
25              if result == None :
26                  retorno = None
27              else:
28                  retorno['id_papel_de_ambiente']     = result[0]
29                  retorno['nome_papel_de_ambiente']   = result[1]
30                  retorno['id_papel_de_usuario']      = result[2]
31                  retorno['nome_papel_de_usuario']    = result[3]
32                  retorno['contexto']                 = result[4]
33                  retorno['usuario']                  = usuario
34                  retorno['ambiente']                 = ambiente
35          except:
36              print("##### Buscando informações da regra de acesso no banco
                ↪  de dados - ERRO #####")
37              retorno = None
38          return retorno
39
```

```
40      def temSessao(self, usuario, ambiente):
41          retorno = None
42          query = "SELECT * FROM Sessao WHERE tag_usuario = \""+usuario+"\"
            ↪  AND id_ambiente = "+ambiente
43
44          try:
45              print("##### Buscando informações da sessão no banco de dados
                ↪  #####")
46              self.cursor.execute(query)
47              result = self.cursor.fetchone()
48
49              if result == None :
50                  retorno = None
51              else:
52                  retorno = {}
53                  retorno['usuario']                  = result[0]
54                  retorno['ambiente']                 = result[1]
55                  retorno['id_papel_de_ambiente']     = result[2]
56                  retorno['id_papel_de_usuario']      = result[3]
57
58
59          except:
60              print("##### Buscando informações da sessão no banco de dados
                ↪  - ERRO #####")
61          return retorno
62
63      def criarSessao(self, info):
64          query = "INSERT INTO Sessao (tag_usuario, id_ambiente,
            ↪  id_papel_de_ambiente, id_papel_de_usuario, datetime)  VALUES
            ↪  ( \""+info['usuario']+"\", "+str(info['ambiente'])+",
            ↪  "+str(info['id_papel_de_ambiente'])+",
            ↪  "+str(info['id_papel_de_usuario'])+", NOW()) "
65
66          try:
67              print ("##### Registrando sessão para o usuário no ambiente
                ↪  #####")
68              self.cursor.execute(query)
69              self.bd.commit()
70              return True
```

```
71          except:
72              print ("##### Registrando sessão para o usuário no ambiente -
                ↪  ERRO #####")
73              self.bd.rollback()
74              return False
75
76      def deletarSessao(self, usuario, ambiente):
77          query = "DELETE FROM Sessao WHERE tag_usuario = \""+usuario+"\"
            ↪  AND id_ambiente = "+str(ambiente)
78
79          try:
80              print ("##### Deletando sessão do usuário no ambiente #####")
81              self.cursor.execute(query)
82              self.bd.commit()
83              return True
84          except:
85              print ("##### Deletando sessão do usuário no ambiente - ERRO
                ↪  #####")
86              self.bd.rollback()
87              return False
88
89      def registrarEvento(self, *args):
90          tag_usuario = args[0]
91          id_ambiente = args[1]
92          id_papel_de_ambiente = args[2]
93          id_papel_de_usuario = args[3]
94          contexto_regra = args[4]
95          contexto_ambiente = args[5]
96          acao = args[6]
97          resultado = args[7]
98
99          query = "INSERT INTO Evento (tag_usuario, id_ambiente,
            ↪  id_papel_de_ambiente,
            ↪  id_papel_de_usuario,contexto_regra,contexto_ambiente,acao,resultado,
            ↪  datetime)  VALUES ( \""+tag_usuario+"\", "+str(id_ambiente)+",
            ↪  "+str(id_papel_de_ambiente)+",
            ↪  "+str(id_papel_de_usuario)+",\""+contexto_regra+"\",\""+contexto_ambiente+"\
            ↪  NOW()) "
100         try:
```

```
101             print ("##### Registrando evento #####")
102             self.cursor.execute(query)
103             self.bd.commit()
104             return True
105         except:
106             print ("##### Registrando evento - ERRO #####")
107             self.bd.rollback()
108             return False
```

APÊNDICE B – Artigo do TCC

CONTROLE DE ACESSO BASEADO EM PAPÉIS EM AMBIENTES ASSISTIDOS

Gabriel R. Goulart[1]

[1]Departamento de Informática e Estatística (INE) - Universidade Federal de Santa Catarina (UFSC)

gabriel.r.goulart94@gmail.com

Abstract. *This work presents an access control system based on user and environment roles, using context information to compose the access rules. To validate the functioning of this system, 4 individuals from different age groups performed tests during a two-week period in a home-based environment, where RFID tags were used to identify users in a non-intrusive way. After the tests, it was identified that all access attempts were processed correctly, regardless of the changes of roles or access rules, which shows that the proposed system does indeed work.*

Resumo. *Este trabalho apresenta um sistema de controle de acesso baseado em papéis de usuário e ambiente, utilizando informações de contexto para compor as regras de acessos. Para validar o funcionamento desse sistema, 4 indivíduos de diferentes faixas etárias realizaram testes durante o período de duas semanas em um ambiente assistido domiciliar, onde tags RFID foram utilizadas para identificar os usuários de maneira não intrusiva. Após os testes, identificou-se que todas as tentativas de acesso foram processadas corretamente, independentemente das mudanças dos papéis ou das regras de acesso, o que mostra que o sistema proposto de fato funciona.*

1. Introdução

Segundo uma pesquisa das Nações Unidas, a população idosa dobrará até 2050 [ONU 2017]. Com isso o consumo de serviços voltados para essa população também aumentará, porém se soluções inovadoras não forem encontradas e aplicadas, esses serviços sofrerão com um défice muito grande para suprir as necessidades da sociedade.

Nesse contexto soluções como a de ambientes assistidos são aplicados com o âmbito de fornecer uma ajuda, e complementar os serviços voltados a saúde e bem estar, não só dos idosos, mas de qualquer indivíduo que precise ser assistido. A utilização dessa solução provê um maior conforto para o indivíduo, pois ele poderá viver no seu ambiente domiciliar e mesmo assim continuar sendo acompanhado pelo o seu médico por exemplo, e se alguma anormalidade acontecer, imediatamente todos os envolvidos no cuidado do indivíduo serão notificados e as ações necessárias serão tomadas.

A abordagem de ambientes assistidos tem grandes possibilidades de se tornar popular e aceita pela população, principalmente pela população idosa, pois utiliza tecnologias que se inserem no ambiente e não necessitam de muitas interações explícitas com o usuário. Além de possibilitar que pessoas que necessitam de assistência, vivam em suas casas, sem que tenha uma ou mais pessoas fisicamente acompanhando elas.

Com o crescimento e popularização dos ambientes assistidos, não se pode deixar de pensar na segurança desses ambientes, e por este motivo o controle de acesso é de extrema importância, pois garante acesso ao ambiente, acesso físico, apenas para pessoas previamente autorizadas, utilizando diversas abordagens como por exemplo a baseada em papéis.

Seguindo essa linha, este trabalho abordará o controle de acesso baseado em papéis, papéis de usuário e ambiente, juntamente com informações de contexto, ou seja, informações que o ambiente pode prover para o sistema, com o objetivo de realizar um controle de acesso inteligente e sensível ao ambiente.

Este trabalho está dividido da seguinte forma: na seção 2 é realizado uma análise dos trabalhos correlatos, na seção 3 é apresentado a proposta deste trabalho, já na seção 4 os resultados são discutidos, e por fim a conclusão e trabalhos futuros, apresentados na seção 5.

2. Trabalhos Correlatos

Na comunidade acadêmica assuntos relacionado ao controle de acesso já vem sendo discutidos há bastante tempo, principalmente o baseado em papéis, porém ainda não se encontram em abundância, trabalhos que foquem no uso do controle do acesso em ambientes assistidos, mesmo que nos últimos anos, assuntos relacionados a ambientes assistidos tem estado em alta.

Nos trabalhos que foram analisados, formas de se aplicar o controle de acesso baseado em papéis foram encontrados, algumas levando em consideração o contexto, outras atribuindo papéis aos ambientes. Entretanto a maioria dos trabalhos não deixam claro a sua aplicabilidade em ambientes assistidos, de maneira a controlar o acesso a um ambiente real, utilizando todos os recursos que tal ambiente pode oferecer.

2.1. Análise dos Trabalhos Correlatos

Como citado anteriormente diversas alternativas para realizar o controle de acesso baseado em papéis são encontradas, [Zhang et al. 2004] utilizam máquinas de estados para fazer o controle de papéis ativos e permissões atribuídas aos papéis. Como a aplicação é consciente de contexto, um agente de contexto coleta as informações e gera eventos que disparam transições nas máquinas de estados.

Outra abordagem consciente de contexto é a de [Covington et al. 2001], eles trazem o conceito de papel de ambiente, o que não é um elemento do padrão RBAC. Sendo assim, com essa nova atribuição, as permissões são associadas tanto aos papéis de usuários quanto aos papéis de ambiente, provendo uma maior flexibilidade para o sistema como um todo.

[Park et al. 2006] utilizam o conceito de papel de contexto, o que se assemelha ao papel de ambiente apresentado por [Covington et al. 2001], entretanto elementos como datas e tempo são utilizados para formar o papel de contexto, que são associados aos papéis de usuários e assim formam a política de segurança. A seguir um exemplo do controle de acesso utilizando papel de contexto.

- transação $= < papel\_usuário, papel\_contexto, permissão >$
- bit_permissão = permitir , negar

- regra da política $=<$ transação, $bit\_permissão >$, $exemplo <<$ criança, $(18h < T < 21h), TV\_Ligar >, permitir >$

[Kayes et al. 2017] utilizam as informações de contexto para ativar o papel do usuário, semelhante aos trabalhos apresentados anteriormente. A utilização do contexto para ativar um papel de usuário é realizada através de expressões contextual, ou seja, uma composição de contextos, onde se pode utilizar informações como a localização do usuário, dias da semana ou até mesmo as escalas de trabalho, por exemplo. O gerenciamento dessas políticas de controle de acesso são realizadas utilizando-se ontologias, o que facilita no processo de verificação das condições para ativar um papel, e também na expansão do sistema, como a criação de novas políticas de acesso ou até mesmo papéis de usuário.

No processo de pesquisa limitou-se a pesquisar abordagens de controle de acesso baseadas em papéis, as quais fossem sensíveis ao contexto, para que assim houvesse uma maior proximidade com os objetivos deste trabalho, mesmo que a maioria dos trabalhos foquem em apresentar modelos, que muitas vezes não são aplicados em um ambiente assistido.

3. Proposta

Realizar o controle de acesso é extremamente importante quando se fala na segurança de um ambiente, obviamente não seria diferente para um ambiente assistido. Com o objetivo de explorar essa questão, este trabalho propõe um sistema para controlar o acesso físico em ambientes assistidos de maneira não intrusiva, utilizando papéis de usuários e ambientes juntamente com as informações de contexto para construir regras de acesso.

A Figura 1 apresenta um exemplo do funcionamento do sistema. Supondo que um usuário X desempenha o papel de usuário EMPREGADO e está tentando acessar um ambiente Y com papel de ambiente SALA DO CHEFE, e a regra de acesso associada ao papel de usuário e ambiente utilize as seguintes informações de contexto : horário, data e se o chefe se encontra em sua sala. Para que o usuário X consiga acessar a sala do chefe as informações de contexto precisam ser verdadeiras.Para um entendimento mais amplo, a Figura 2 apresenta o diagrama de atividades do sistema de controle de acesso.

A identificação do usuário é realizada de maneira não intrusiva utilizando tags RFID. As tags são associadas a papéis de usuário, que por sua vez integram a regra de acesso. As informações e associações, como por exemplo entre usuário e papel de usuário, são armazenadas em um banco de dados MYSQL.Importante ressaltar que o sistema foi implementado utilizando as linguagens Python e C para arduino junto com o protocolo de comunicação MQTT.

3.1. Controle de Acesso Baseado em Papéis

Neste trabalho o controle de acesso utilizado será o baseado em papéis, o RBAC, porém só será utilizado as funções básicas, as quais garantem o funcionamento do controle de acesso.

Para que o controle de acesso aproveite todos os recursos que um ambiente assistido pode oferecer, se baseando na expansão do RBAC feita por [Covington et al. 2001], será adicionado ao modelo o conceito de ambiente e papel de ambiente, o que permitirá

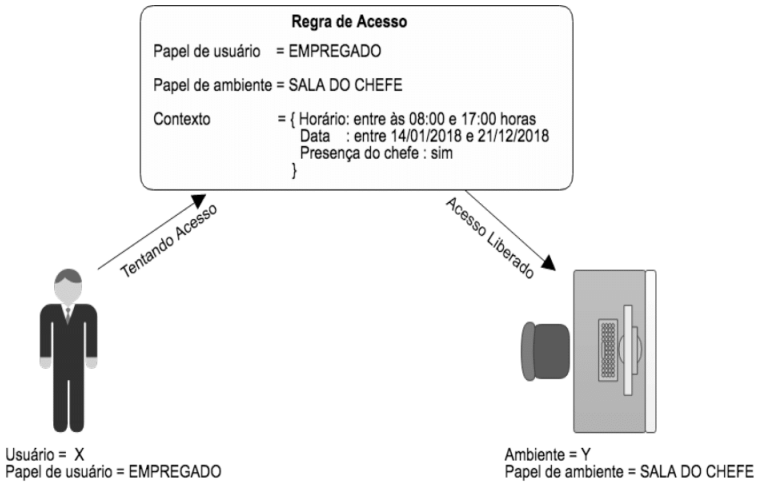

Figura 1. Exemplo do Sistema

uma maior flexibilidade no sistema, pois a regra de acesso não dependerá apenas do papel do usuário, mas também do papel de ambiente. A Figura 3 apresenta a expansão do RBAC que será utilizado neste trabalho.

Especificações para a expansão do RBAC:

- Usuário (US), Papel de Usuário (PA), Ambiente (AM) , Papel de Ambiente (PM), Sessão (SE), Permissão (PERMS), Contexto (CON).
- $PAUAM = 2^{(\text{PA X PM})}$, conjunto de papéis de usuário associados a papéis de ambiente.
- $AU \subseteq$ US X PAUAM, relação de atribuição muitos para muitos entre usuário e papéis de usuário relacionados com papéis de ambiente.
- $AA =$ relação de um para muitos entre ambientes e papéis de ambiente.
- $AP \subseteq$ PERMS X PAUAM, relação de atribuição um para muitos entre papéis de usuário associados a papéis de ambiente e permissões.
- $Sessão\_usuário(s : S) \rightarrow US$, mapeamento da sessão s em um usuário correspondente.
- $Sessão\_ambiente(s : SE) \rightarrow AM$, mapeamento da sessão s em um ambiente correspondente.
- $Sessão\_papel(s : S) \rightarrow 2^{PA}$, mapeamento da sessão s em um conjunto de papéis.
- $Sessão\_papel\_ambiente(s : SE) \rightarrow 2^{PM}$, mapeamento da sessão s em um conjunto de papéis de ambiente.

Outra adptação importante a ser feita no RBAC, é que a sessão representará que o usuário está registrado em um ambiente, ou seja, representará que o usuário está no ambiente. Caso o usuário não tenha uma sessão ativa para um certo ambiente, significa que o usuário não está no ambiente.

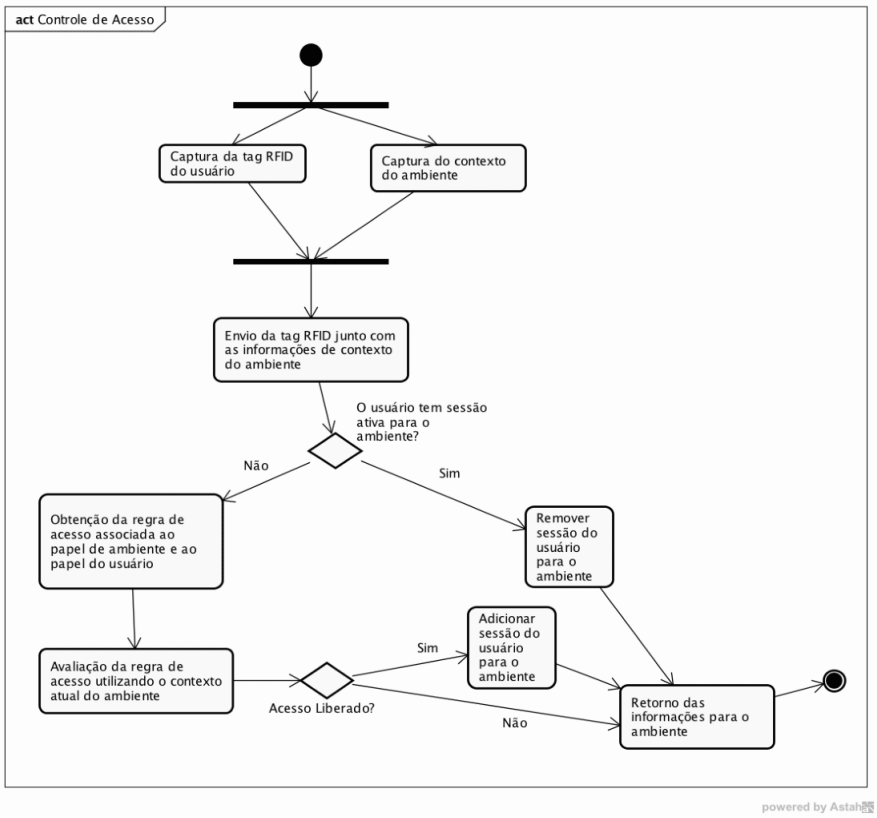

Figura 2. Diagrama de Atividades do Sistema

4. Ambiente e Resultados

Nesta seção serão apresentados os resultados obtidos através dos experimentos realizados nos ambientes representados pela Figura 4, onde os números na figura, 1 e 2, são os identificadores únicos de cada ambiente utilizados pelo sistema. A figura também mostra o posicionamento dos módulos do sistema e a localização dos sensores, os quais são utilizados pelo ambiente 1, conforme a tabela 1.

Para mostrar a validade do sistema de controle de acesso baseado em papéis, testes foram realizados durante o período de duas semanas, onde quatro usuários, dois com idades entre 20 e 30 anos, e os outros dois com idades entre 40 e 50 anos, utilizaram o sistema.

Tabela 1. Sensores (Recursos) do ambiente 1

Identificador	Descrição
1	Sensor de luminosidade
2	Sensor de distância ultrassônico

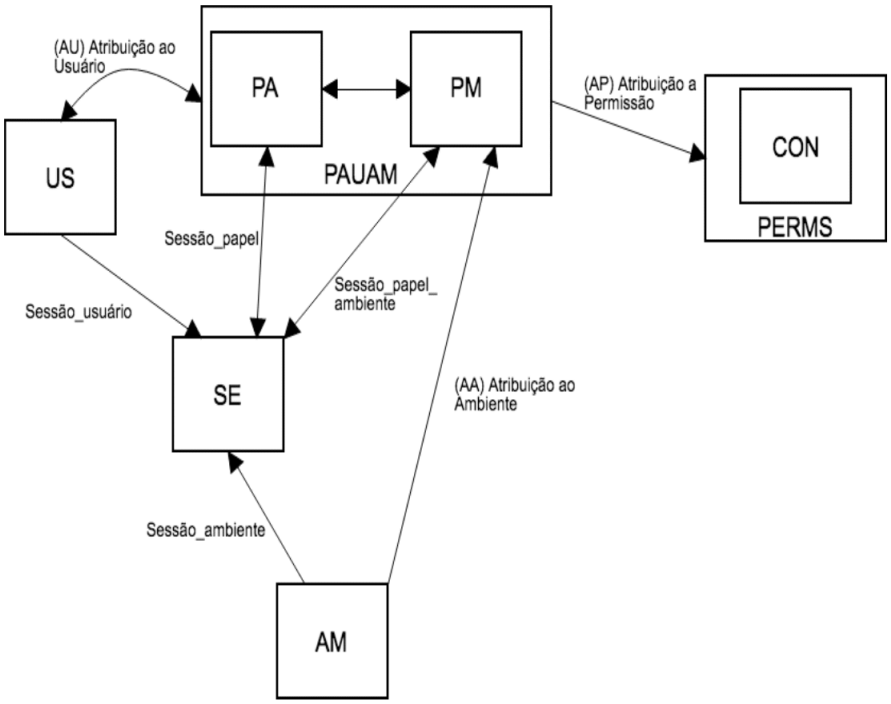

Figura 3. Extensão do RBAC

A tabela 2 apresenta os papéis de usuário cadastrados no sistema, papéis esses que são característicos de um ambiente domiciliar. Seguindo a mesma vertente, a tabela 3 contém os papéis de ambiente utilizados. Lembrando que os papéis são configuráveis, ou seja, podem ser adicionados, editados ou excluídos.

Tabela 2. Papéis de usuário

Papel	Identificador
Filho	1
Pai	2
Mãe	3
Convidado	4

4.1. Resultados

Ao longo de duas semanas de teste, resultados expressivos foram obtidos a fim de mostrar a validade do sistema. Em todas as tentativas de acesso, o sistema se comportou de acordo com o esperado, lendo a tag do usuário, capturando o contexto do ambiente, processando as regras de acesso e verificando se o usuário poderia ou não acessar o ambiente.

A figura 5 expõe o gráfico de utilização do sistema através da quantidade de eventos capturados. Esses eventos consistem em tentativas de acesso nos ambientes por parte

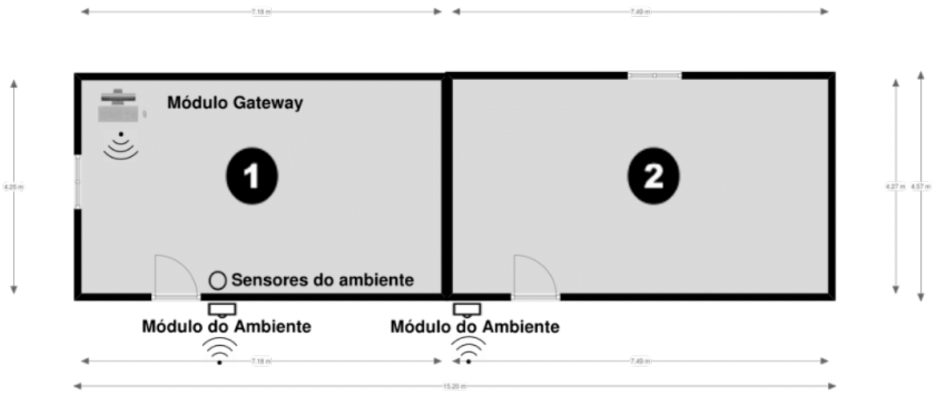

Figura 4. Planta baixa do ambiente de teste

Tabela 3. Papéis de ambiente

Papel	Identificador
Quarto Filho	1
Quarto Pais	2
Sala de Estar	3

dos usuários. Durante o período de testes foram registrados 284 eventos, sendo 240, acessos que foram garantidos pelas regras de acesso, e 44 que foram negados.

Já a figura 6 apresenta os eventos registrados no sistema agrupados por horário. Essa informação é importante, pois se consegue ter uma noção da utilização do sistema por horário, e assim obter conclusões importantes. Por exemplo, pode-se observar que por volta das 19 horas o sistema passou por um pico de eventos, o que significa que houveram bastantes tentativas de acesso.

Importante destacar que o sistema foi submetido a diversas configurações, ou seja, as regras de acesso e papéis foram alterados durante o período de teste, e em todas as configurações o sistema se comportou corretamente, avaliando as regras de acesso e garantindo ou negando o acesso ao ambiente.

5. Conclusão e Trabalhos Futuros

Com este trabalho foi possível analisar diversos aspectos relacionados à área de controle de acesso. Sendo assim identificou-se que a abordagem baseada em papéis utilizando informações de contexto, é pouco empregada e explorada em ambientes assistidos, e por

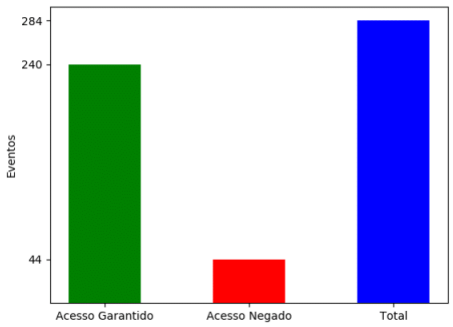

Figura 5. Eventos no sistema

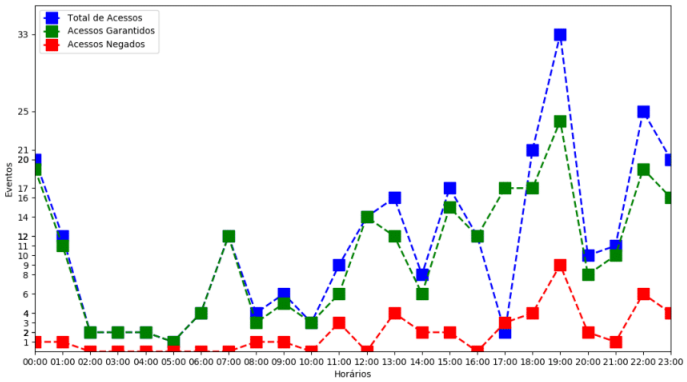

Figura 6. Eventos no sistema por horário

este motivo decidiu-se utilizar essa abordagem neste trabalho, implementando um sistema capaz de controlar o acesso, se baseando em papéis de usuário e ambientes em conjunto com informações de contexto.

Durante a etapa de testes e análise dos dados gerados pelo sistema, o controle de acesso apresentou informações extremamente importantes sobre o ambiente monitorado. Além das informações básicas, como quantidade de acesso, acessos negados e garantidos,as quais foram utilizadas para compor os resultados obtidos neste trabalho. O sistema de controle de acesso expôs uma série de informações que são obtidas implicitamente quando se analisa os eventos do sistema. Por exemplo, através dos eventos se pode ter uma noção da utilização do ambiente, e assim mensurar o consumo dos recursos em certo horário, ou taxa de ocupação do ambiente. Outra informação interessante que foi observada durante a análise dos dados gerados pelo sistema de controle de acesso, é que se pode obter a localização atual ou a localização em certo momento de um ou mais usuários.

Conforme discutido na seção 4. mostrou-se que o sistema proposto neste tra-

balho de fato realiza o controle de acesso, o que confirma que o objetivo principal foi alcançado. Ressaltando que o sistema funcionou de maneira dinâmica, ou seja, se adaptando às mudanças de papéis, e concisa, garantido e negando o acesso aos usuários. Consequentemente se conclui que o controle de acesso baseado em papéis é uma abordagem aplicável e eficaz na segurança de ambientes assistidos.

Para finalizar serão propostos alguns tópicos para a evolução do sistema de controle de acesso:

- Para que o sistema se torne mais completo, além da utilização das funções básicas do RBAC, é importante adicionar ao sistema os outros módulos do padrão de controle de acesso baseado em papéis.
- Outro aspecto importante no controle de acesso é ter um sistema capaz de gerenciá-lo, pois assim se torna mais fácil a inclusão, remoção e edição dos papéis, regras de acesso e outros componentes que compõem o controle de acesso.

Referências

Covington, M. J., Long, W., Srinivasan, S., Dev, A. K., Ahamad, M., and Abowd, G. D. (2001). Securing context-aware applications using environment roles. *Proceedings of the sixth ACM symposium on Access control models and technologies*, (January):pp. 10–20.

Dey, A. K. (2001). Understand and using context.

Ferraiolo, D. F. and Kuhn, R. D. (2004). Role based access control.

Jih, W.-r., Cheng, S.-y., Hsu, J. Y.-j., and Tsai, T.-m. (2005). Context-aware Access Control in Pervasive Healthcare. *Context*, (February 2014):2–9.

Kayes, A. S. M., Han, J., Rahayu, W., Islam, M. S., and Colman, A. (2017). A Policy Model and Framework for Context-Aware Access Control to Information Resources.

ONU (2017). World population ageing.

Park, S., Han, Y., and Chung, T. (2006). Context-role based access control for context-aware application. *High Performance Computing and Communications*, pages 572–580.

Sandhu, R. S. and Samarati, P. (1994). Access control - principles and pratice.

Trnka, M. and Cerny, T. (2015). Context-aware Role-based Access Control Using Security Levels. *Proceedings of the 2015 Conference on Research in Adaptive and Convergent Systems*, pages 280–284.

Van Den Broek, G., Cavallo, F., and Wehrmann, C. (2010). *AALIANCE Ambient Assisted Living Roadmap*. IOS Press, Amsterdam, The Netherlands, The Netherlands.

Zhang, G., Zhang, G., Parashar, M., and Parashar, M. (2004). Context-aware dynamic access control for pervasive applications. *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference*, pages 21–30.