Programa Interdisciplinar de Pós-Graduação em

Computação Aplicada

Mestrado Acadêmico

Paulo César Albarello

Controle de Acesso Sensível ao Contexto Baseado na Inferência em Trilhas

São Leopoldo, 2013

UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
UNIDADE ACADÊMICA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA INTERDISCIPLINAR DE PÓS-GRADUAÇÃO
EM COMPUTAÇÃO APLICADA
NÍVEL MESTRADO

PAULO CÉSAR ALBARELLO

CONTROLE DE ACESSO SENSÍVEL AO CONTEXTO BASEADO NA INFERÊNCIA EM TRILHAS

São Leopoldo
2013

Paulo César Albarello

CONTROLE DE ACESSO SENSÍVEL AO CONTEXTO BASEADO NA INFERÊNCIA EM TRILHAS

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre, pelo Programa Interdisciplinar de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos – UNISINOS

Orientador: Dr. Jorge Luis Victória Barbosa

São Leopoldo

2013

A327c Albarello, Paulo César
Controle de acesso sensível ao contexto baseado na inferência em trilhas / por Paulo César Albarello. – São Leopoldo, 2013.

84 f. : il. color. ; 30 cm.

Dissertação (mestrado) – Universidade do Vale do Rio dos Sinos, Programa Interdisciplinar de Pós-Graduação em Computação Aplicada, São Leopoldo, RS, 2013.
Orientação: Prof. Dr. Jorge Luis Victória Barbosa, Ciências Exatas e Tecnológicas.

1.Computação ubíqua – Controle de acesso. 2.Computação móvel. 3.Arquitetura de computador – Trilhas. 4.Sensibilidade ao contexto. I.Barbosa, Jorge Luis Victória. II.Título.

CDU 004.75.057.5
004.22
004.89

Catalogação na publicação:
Bibliotecária Carla Maria Goulart de Moraes – CRB 10/1252

RESUMO

O desenvolvimento da computação ubíqua e o aumento da capacidade de processamento de dispositivos de comunicação móveis como *smartphones* e *tablets* vêm propiciando o aparecimento de aplicativos que façam o uso da sensibilidade ao contexto como forma de ajudar o usuário em seus trabalhos cotidianos. Esta dissertação propõe um modelo para o controle de acesso de entidades que estejam inseridas em um contexto. Neste modelo, um contexto suporta atividades (recursos) que poderão ser utilizadas mediante a verificação das permissões por parte de um servidor de controle de acesso. O modelo propõe uma alternativa ao controle padrão de acesso valendo-se do histórico de permissões anteriores concedidas para a entidade (trilhas). As trilhas de acesso são analisadas e segundo regras estabelecidas no modelo são concedidas novas permissões de acesso. Os resultados demonstram a viabilidade da proposta, permitindo que sistemas de controle de acesso possam utilizar uma forma alternativa de concessão de direitos de acesso.

Palavras-chave: Computação Móvel. Computação Ubíqua. Controle de Acesso. Sensibilidade ao Contexto. Trilhas.

ABSTRACT

The development of ubiquitous computing and the increasing processing power of portable devices such as smartphones and tablets have favored the emergence of applications that make use of context-aware systems to help the users in their daily work. This dissertation presents an access control model to entities that are inserted in a context. In this model, a context has activities (resources) that can be used checking the permissions from an access control server. This model proposes an alternative to default access control using the history of previous permissions granted to the entity (trails). The trails are analyzed according established rules in the model and new access permissions will be granted. The results demonstrate the feasibility of the proposal, allowing access control systems may use an alternative way to grant access rights.

Keywords*: Mobile Computing. Ubiquitous Computing. Access Control. Context-Aware. Trails.*

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

API	Application Programming Interface
AWARENESS	Context-Aware Mobile Networks and Services
CONON	Context Ontology
DAC	Discretionary Access Control
ECA	Event-Condition-Action
MAC	Mandatory Access Control
PC	Personal Computer
RBAC	Role-Based Access Control
SOCAM	Service-Oriented Context-Aware Middleware
UbiCOSM	Ubiquitous Context-based Security Middleware
UML	Unified Modeling Language
XML	Extensible Markup Language

SUMÁRIO

1 INTRODUÇÃO

Nos dias de hoje os ambientes de convivência, sejam eles de trabalho ou de lazer, disponibilizam uma grande variedade de sistemas computacionais que estão ao alcance do usuário. Entre os serviços existentes é possível citar o uso de máquinas fotográficas, computadores, *smartphones*, *notebooks*, *tablets* e tantos outros.

Como há grande tendência em aumentar a interconectividade entre os aparelhos nomeados e no seu acréscimo na taxa de transmissão de dados disponibilizada pela internet, pesquisas revelam que tais mudanças nos sistemas computacionais darão início a criação de um Ambiente Inteligente (HECKMANN, 2005).

Tais possibilidades estão tornando em realidade a ideia que *Mark Weiser* explanou em 1991 e a denominou de computação ubíqua, que tem como objetivo tornar a interação pessoa-máquina invisível, ou seja, integrar a informática com as ações e comportamentos naturais das pessoas (WEISER, 1991).

Constantemente o poder de processamento dos equipamentos computacionais vem aumentando e a miniaturização de componentes e dispositivos de processamento vem igualmente sendo feita. Estes dispositivos estão sendo conectados a aparelhos de uso pessoal, como *smartphones*, permitindo assim, que usuários possam utilizar uma rede de informações a qualquer momento e em qualquer lugar, levando em conta informações relevantes a respeito do contexto em que este usuário está inserido.

Dado que dispositivos móveis de computação poderão executar estes aplicativos em diversos ambientes, seria interessante que o controle de acesso do sistema buscasse através da análise da trilha do usuário as atividades que o mesmo possuía e possui acesso para delegar novas concessões. A análise da trilha dispensaria o trabalho de acessos extras para receber permissões em outros contextos.

1.1 Motivação

O tema “controle de acesso” (BALASUBRAMANIAN, 2007) tem sido utilizado inúmeras vezes por especialistas para definir formas de controle que possam identificar a

entidade que está inserida em um determinado contexto e abstrair então, quais seriam as atividades que poderiam ser concedidas a ela.

Um aplicativo é dito sensível ao contexto se este utilizar informações sobre o ambiente de entidades relevantes ao seu domínio para prover informações ou serviços a seus usuários (MIRANDA et al., 2006). Estas informações normalmente levam em consideração a localização, a identificação do usuário e a interpretação de regras de trabalho. Como exemplo prático desta técnica é possível citar um usuário necessitando imprimir um documento e o sistema localizar onde está a impressora mais próxima, que permita a ele executar o trabalho. Ou ainda o exemplo de um telefone celular que passa para o modo silencioso, automaticamente, ao entrar em um ambiente onde esteja ocorrendo uma reunião.

Para que o aplicativo possa oferecer serviços que sejam úteis ao usuário, uma alternativa seria o uso de trilhas, ou seja, armazenar e gerenciar em formato de histórico, todos os seus acessos para fornecer, posteriormente, permissões baseadas em atividades anteriores (SILVA et al., 2009).

Além do uso da trilha, é importante também que o sistema saiba quem é a entidade que está inserida no contexto. As entidades poderão ter cargos ou incumbências diferentes dentro de contextos adversos. Estes cargos contextuais serão caracterizados por “papéis” (LI et al., 2008).

Mediante a crescente disponibilização de dispositivos computacionais móveis, observa-se que grandes quantidades de informações, possivelmente, terão de ser administradas, podendo ocasionar em um aumento de trabalho a ser gerenciado. Devido a este fato, surge o conceito de Tecnologia Calma (WEISER et al.1996), onde a interação entre dispositivos/usuários ocorre na periferia da atenção destes usuários, desaparecendo da consciência dos mesmos e aparecendo somente quando necessário.

Saber identificar um usuário e quais são suas intenções é tão importante quanto saber em qual contexto este usuário está inserido. A partir da trilha por ele deixada, seria possível que o sistema pudesse mapear as atividades realizadas anteriormente e disponibilizar, novamente, o acesso. Este recurso tornaria o sistema autônomo por controlar o acesso a atividades que já haviam sido usadas.

Além do uso do histórico de acessos (trilhas), outro ponto que pode ser explorado é a confiança que uma entidade possui a respeito de outra (GIANG et al. 2007). A confiança é um

fator de segurança ou firme convicção que uma entidade tem por outra. Também seria a presunção de idoneidade, onde a entidade possuiria determinados fins já conhecidos sobre o recurso a ser utilizado. Sendo assim, quando uma entidade afirma ter confiança em outra, o controle de acesso pode se valer desta informação para conceder permissões.

Enfocadas as definições, este trabalho propõe uma alternativa ao controle de acesso contextualizado registrando e utilizando uma trilha de acessos prévios feitos por uma entidade a um sistema e utilizando este histórico como critério para concessões.

1.2 Problema

Muitas pesquisas têm sido feitas no ramo da computação ubíqua no quesito controle de acesso (BALASUBRAMANIAN, 2007). Diferentes técnicas para o controle de acesso foram desenvolvidas.

Todas as pesquisas verificadas não fazem o uso do histórico de acessos anteriores como critério para conceder a permissão de acesso. Analisando estes fatos, este estudo pretende apresentar a criação de um modelo de controle de acesso sensível ao contexto, utilizando como critério alternativo de acesso a recursos, a análise da trilha de acessos anteriores do usuário.

Nesta trilha serão registradas, por exemplo, as atividades realizadas por uma entidade em um determinado contexto num formato sequencial. O modelo propõe também o uso de recursos de confiança entre entidades para permitir acessos temporários.

1.3 Questão de pesquisa

Seria possível desenvolver um modelo de controle de acesso a ambientes contextualizados que levasse em consideração, além dos papéis que a entidade estaria portando no momento do acesso, a análise do histórico de acessos anteriores (sua trilha) e ainda utilizar atributos de confiança entre entidades para conceder permissões?

1.4 Objetivos

O objetivo geral deste trabalho é especificar, codificar e validar um modelo para controle de acesso contextualizado baseado em trilhas, denominado *EasyConn4All*. As principais características do modelo são realizar o controle de acesso a recursos (atividades) que estejam inseridas em um contexto, valendo-se para isto do uso de papéis que a entidade esteja portando, do histórico de acessos anteriores ao contexto ou ainda da troca de concessões através de um atributo de confiança, tendo módulos responsáveis pela verificação das informações e concessão de atividades.

Os objetivos específicos deste trabalho são:

- permitir o gerenciamento de diferentes entidades / atividades;
- manter registros históricos de atividades permitidas (trilha);
- permitir concessões através de troca de confianças;
- disponibilizar atividades controladas.

1.5 Organização da Dissertação

A dissertação está organizada da seguinte forma. O capítulo dois descreve os conceitos envolvidos como a computação ubíqua, a definição e qualificação de contextos e o controle de acesso.

No capítulo três são expostos os trabalhos relacionados, apresentando um estudo comparativo entre eles. O capítulo quatro descreve o modelo proposto, detalhando sua arquitetura. No capítulo cinco são mostrados os detalhes sobre a implementação do protótipo.

O capítulo seis apresenta o uso do aplicativo baseado em cenários para avaliar o modelo. E no capítulo sete são apresentadas as considerações finais e os trabalhos futuros.

2 CONCEITUAÇÃO

Neste capítulo serão abordados os conceitos considerados relevantes para um melhor entendimento e captura do processo de construção deste trabalho.

2.1 Computação Ubíqua

Um grupo de pesquisadores do laboratório de Ciência Computacional *Xerox Palo Alto Research Center*, liderado por Mark Weiser relacionou tendências ligadas à computação que poderiam alterar o cenário em que os sistemas computacionais estivessem inseridos.

No cômputo destas ideias, foi constatado que o tamanho dos dispositivos computacionais estava diminuindo e o custo dos equipamentos igualmente estava ficando mais barato, tendência esta, que se mantém até hoje.

Outro ponto destacado foi o grande número de computadores que estariam sendo desenvolvidos em comparação ao número de pessoas (muitas pessoas teriam acesso a sistemas computacionais móveis, como *smartphones*, GPS etc.) e para interligar estes aparelhos a ideia da comunicação sem fio (*wireless*), também foi destacada. Hoje a comunicação *wireless* é uma realidade vinda desta tendência.

Weiser e Brown (1996), concluíram que o grande número de dispositivos computacionais e o consequente aumento do acesso a estes dispositivos levariam a um acréscimo considerável da informação que seria manipulada.

Sobre os acessos aos dispositivos, Mark Weiser (1991), defende que no futuro estes sistemas estarão presentes nos mais diferentes objetos do cotidiano, como em etiquetas de roupas, máquinas de café, interruptores de luz, entre outros, de forma invisível aos que irão utilizá-la. Neste mundo idealizado por Weiser, seria necessário aprender a conviver com os sistemas computacionais e não apenas interagir com eles.

Na computação ubíqua se faz necessário o uso de pequenos computadores cujo preço não seja elevado e de tecnologias para conexão sem fio. Por exemplo, um domicílio controlado por dispositivos de computação ubíqua poderá ter o controle remoto da iluminação

da casa, do sistema de incêndios, dos sistemas de entretenimento, dos sistemas para monitorizar a saúde dos ocupantes da casa etc.

Satyanarayanan (2010) classifica a computação móvel como uma evolução da computação distribuída e a computação ubíqua como uma evolução da computação móvel. As tendências de cada uma destas tecnologias tiveram desafios que precisavam ser superados. Sobre estes desafios é possível destacar a comunicação remota, a tolerância a falhas, a alta disponibilidade, o acesso a informações remotas, a segurança distribuída, as redes móveis, o acesso a informações móveis, as aplicações adaptativas, os sistemas sensíveis à energia e a sensibilidade à localização.

A estes desafios, a computação ubíqua ainda acrescenta o uso de espaços inteligentes, a invisibilidade e a escalabilidade (COSTA et al. 2008).

Uma das premissas dos estudiosos da área é o de tornar a computação cada vez mais presente no cotidiano das pessoas de forma que a tecnologia seja capaz de armazenar informações que sejam pertinentes e aprender segundo as atividades feitas anteriormente. Atividades como o compartilhamento de informações se transformarão em tarefas mais fáceis. Sem que ela seja percebida, esta tecnologia deverá fazer parte do processo de evolução da humanidade.

2.2 Sensibilidade ao Contexto

Segundo a definição de Anind Dey, contexto seria "*qualquer informação relevante que possa ser utilizada para caracterizar entidades de uma interação usuário-computador*" (DEY,2001).

O termo sensibilidade trata da capacidade de um sistema perceber a ocorrência de eventos ou sensores existentes a sua volta (DEY,2001). O termo contexto trata de tudo que influencia o usuário na forma de interagir com o ambiente físico. Sendo assim, a sensibilidade ao contexto seria a capacidade de um sistema computacional estar ciente das mudanças do ambiente em relação ao usuário e assim reagir de forma proativa aos eventos que possam alterar este contexto.

A sensibilidade ao contexto torna possível a criação e utilização de serviços personalizados e específicos para uma determinada ocasião e momento (CAO, 2006, HISAZUMI, 2003)

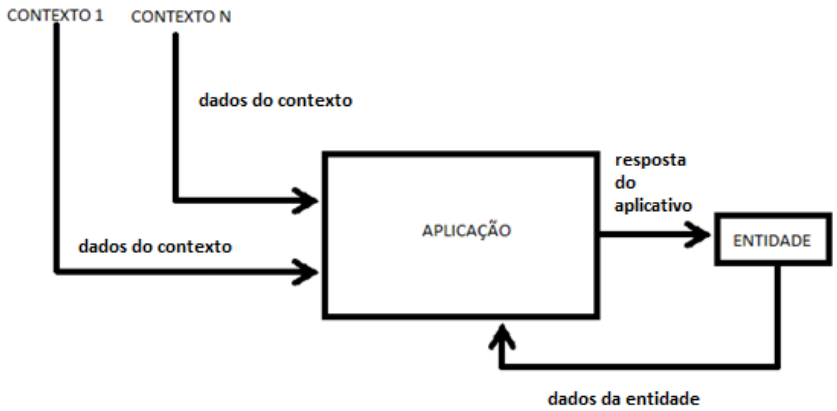
Anind Dey (2001) afirma ainda que o contexto pode ser caracterizado pelo estado físico, social, emocional ou informacional do usuário.

Haja vista é possível inferir sobre quais seriam os tipos de informações sugeridas para a correta caracterização do contexto. Qualquer informação que possua algum critério que caracterize a situação de um usuário perante um ambiente qualquer pode ser caracterizada como contexto (DEY,2001).

Um sistema sensível ao contexto deve possuir algumas características:

- apresentação de informações e serviços para o usuário;
- execução automática de serviços para o usuário;
- classificação de informações de contexto para uma posterior recuperação de dados.

Figura 1: Adaptabilidade a diferentes contextos.



Fonte: Elaborado pelo autor

A Figura 1 mostra o funcionamento de uma aplicação sensível ao contexto. Na figura é possível ver que a aplicação interpreta as informações contextuais e gera as saídas referentes ao acesso controlado. As informações são constantemente validadas (tanto da entidade quanto do contexto), para verificar se continuam acessíveis ao usuário (entidade).

A sensibilidade ao contexto apresenta divisões com relação às variantes de contexto para análise pelo sistema (DEY, 2001). Nessa dimensão, é possível identificar elementos mais objetivos, como espaço (local), tempo (dia, hora), temperatura, ou elementos mais subjetivos, como objetivos, intenções, emoções, interesses, atividades de usuários.

2.3 Controle de Acesso

Em sistemas ubíquos, um dos requisitos principais para estabelecer níveis de segurança ao acesso de serviços seria o processo de determinar se uma entidade (seja ela um usuário, um aplicativo ou um processo), possui permissão para acessar serviços (uso de arquivos, serviços, sistema, etc.), com base em políticas de acesso predeterminadas (HULSEBOSCH et al. 2005, BALASUBRAMANIAN, 2007). Estas políticas de acesso são definidas como um conjunto de regras que determinam o comportamento da rede e do sistema em certas ocasiões. Algumas políticas de controle de acesso podem ser aplicadas juntas (JAJODIA, 2001).

Em se tratando de ambientes ubíquos, o controle de acesso merece um destaque especial devido à mobilidade dos dispositivos e o nível de acesso que cada um dos usuários poderá ter sobre o sistema (FRAINER, 2006).

O controle de acesso abordado neste trabalho envolve técnicas de autorização, autenticação e de auditoria em seus registros.

2.3.1 Controle de Acesso Discricionário

O controle de acesso discricionário (*discretionary access control ou DAC*) se resume a uma estratégia de política de controle de acesso que é definida pelo proprietário do recurso a ser disponibilizado. Este detentor do recurso a ser disponibilizado decide quais entidades terão acesso ao recurso disponibilizado e com qual nível de privilégio (LI et al., 2008).

Os dois conceitos importantes do DAC são:

- qualquer recurso disponibilizado pelo sistema deverá ter um proprietário. Esta política de acesso será determinada pelo proprietário do recurso. Um objeto sem um proprietário específico é considerado não protegido;
- direitos de acesso são estabelecidos pelo proprietário do recurso.

Duas implementações deste mecanismo são as Listas de Controle de Acesso (*Access Control Lists - ACL*) e Capacidades (*Capabilities*) (ANDERSON, 2001).

As ACL descrevem, para cada objeto, quais sujeitos possuem quais direitos sobre este objeto. Um modelo simples de ACL pode ser visto nos sistemas UNIX, em que os arquivos possuem listas de permissões de acesso (direitos) para o proprietário, o grupo e os demais usuários.

Similarmente, as Capacidades (*Capabilities*) são representadas como as linhas de uma matriz de acesso. Elas descrevem quais direitos um sujeito possui sobre os objetos. *Capabilities* podem ser vistas como o inverso das ACL.

2.3.2 Controle de Acesso Obrigatório

No modelo de controle de acesso obrigatório (*mandatory access control ou MAC*) a norma de acesso será definida não pelo proprietário do recurso, mas sim pelo sistema. Este tipo de controle deve ser utilizado em sistemas onde as informações estejam em um nível de segurança bastante alto, como informações governamentais e militares (LI et al., 2008).

Algumas características do controle de acesso obrigatório são as seguintes:

- Rótulos de sensibilidade: Nos sistemas de controle de acesso obrigatório, os objetos devem possuir rótulos. Um rótulo de sensibilidade de um sujeito estabelece o seu nível de confiança, que é o nível de confiança necessário para acessar o objeto pretendido. Este rótulo de sensibilidade deve ser de nível igual ou superior ao requisitado pelo objeto.
- Importação e exportação de dados: A importação e exportação de informações para outros sistemas ou dispositivos é uma função crítica de um sistema baseado em MAC. Deve ser garantido que os rótulos de sensibilidade sejam mantidos e implementados de maneira conveniente, onde a informação sensível seja protegida a todo o momento.

2.3.3 Controle de Acesso Baseado em Papéis

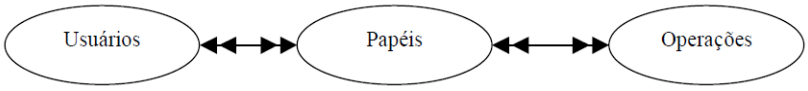
Um controle baseado em papéis (*Role-Based Access Control* ou *RBAC*) (SANDHU, 1996) é um modelo de acesso utilizado para restringir a permissão apenas a usuários

autorizados. Esta abordagem é uma alternativa aos sistemas de controles de acesso do tipo MAC e DAC.

No modelo RBAC, os usuários são feitos membros de “papéis”, os quais têm direitos de acesso. O modelo usa o conceito de “operações”, representando as ações que um papel pode executar sobre os objetos compartilhados da rede (HULSEBOSCH et al. 2005), e, o conceito de “usuários” que representam os membros associados aos papéis .

Na Figura 2, está exemplificado o funcionamento do modelo. Usuários são representados por papéis que possuem restrições de acesso a determinadas operações. As setas duplas indicam a multiplicidade decorrente de vários usuários poderem possuir vários papéis e os papéis por sua vez podem ter associados a si várias operações.

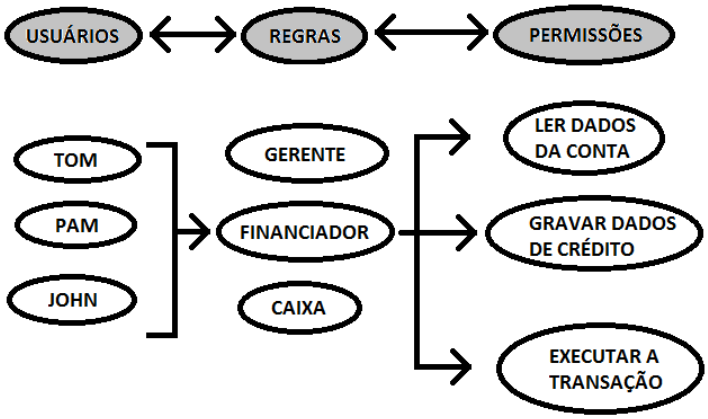
Figura 2: Visão do RBAC.



Fonte: NABHEN, 2003

As operações podem ser associadas ou excluídas dos papéis objetivando atender às eventuais alterações das atividades desempenhadas e representadas por um papel dentro da organização. Os papéis podem ser atualizados sem a necessidade da modificação individual de cada usuário ou grupos de usuários. Estas alterações passam automaticamente a ser refletidas para todos os usuários que possuam associação com o papel.

Figura 3: Modelo RBAC em exemplo.



Fonte: Adaptado de YAMADA et al. 2005

A Figura 3 exemplifica uma situação de validação do modelo. Nela, aparecem três usuários (Tom, Pam e John) que possuem para o modelo o papel de “financiador”. Para este papel, as permissões atribuídas são “ler dados da conta”, “gravar dados de crédito” e “executar a transação”.

2.4 Trilhas

Cada vez mais a personalização de tarefas, atividades ou aplicativos utilizados pelo usuário estão se proliferando entre os sistemas computacionais. O uso das redes sociais como alternativa de entretenimento vem adotando as políticas de personalização de componentes para simplificar a experiência que o usuário venha a ter no sistema. Sendo assim, um dos pontos que se destaca no controle de acesso é o monitoramento do histórico, que o próprio usuário está tendo no sistema e utilizá-lo nos acessos futuros para permitir que determinadas funções fiquem disponíveis, com o simples fato de o sistema reconhecer que um usuário, já conhecido, está retornando ao sistema.

A esta tarefa de monitorar os acessos a atividades que o usuário esteja realizando no sistema, dá-se o nome de trilha (SILVA et al., 2009, SILVA et al., 2010). Uma trilha consiste no histórico de atividades dos contextos visitados por um usuário, durante um período. Este “histórico” de acessos feitos pelo usuário poderá ser registrado, servindo de “guia” para realizar as futuras permissões/negações a um usuário.

Embora a trilha geralmente esteja vinculada à localização de um usuário, ela possui um conceito mais amplo, que inclui as atividades realizadas, aplicações usadas, conteúdos acessados – entre outros – dentro de um contexto e um período de tempo específico (SILVA et al., 2009).

A cada novo contexto visitado, o registro de sua trilha permite ao sistema posteriormente verificar seus gostos, suas preferências e suas permissões no contexto.

De acordo com Driver e Clarke (2004), a pesquisa na área de trilhas é relevante para o campo do desenvolvimento de aplicações ubíquas, uma vez que a utilização de trilhas vai ao encontro do desafio central da área de computação ubíqua, isto é, o desenvolvimento de aplicações transparentes ao usuário.

2.5 Considerações sobre o Capítulo

Este capítulo teve o objetivo de descrever tópicos relevantes ao desenvolvimento do modelo *EasyConn4All*.

As técnicas apresentadas irão permitir que as tarefas relacionadas ao funcionamento do modelo possam ser implementadas permitindo que as regras para o controle de acesso contextualizado possam ser seguidas.

O sistema ser sensível ao contexto o torna específico em determinado ambiente, tendo suas atividades de uso distinto para o contexto em que o usuário estiver inserido.

O controle de acesso fará com que usuários diferentes tenham suas permissões controladas de acordo com as diretivas fornecidas ao modelo.

O registro da trilha de acessos de uma entidade servirá como referência parcial para controlar o acesso futuro da entidade. O registro da trilha e sua verificação a cada novo acesso é o diferencial deste trabalho.

O entendimento de cada um dos conceitos deste capítulo deixará estruturada a tarefa de gerenciar o contexto, pois existirão estruturas no sistema que analisarão as informações que caracterizam o contexto e se basearão na inferência em trilhas para determinar os níveis de acesso.

3 TRABALHOS RELACIONADOS

Neste capítulo estão descritos trabalhos relacionados ao controle de acesso em ambientes sensíveis ao contexto.

Os trabalhos estudados possuem em sua estrutura recursos que permitem o controle de acesso em ambientes sensíveis ao contexto. A maioria dos trabalhos relacionados possuem particularidades e focos de atuação em diversas áreas. Estes critérios foram levados em consideração no momento da escolha dos trabalhos.

Este capítulo possui uma breve explanação sobre os trabalhos, abordando sua arquitetura e a sua funcionalidade. Ao concluir a explanação dos trabalhos, é feita uma comparação entre eles, utilizando critérios que serão explicados no final do capítulo.

3.1 Infraware

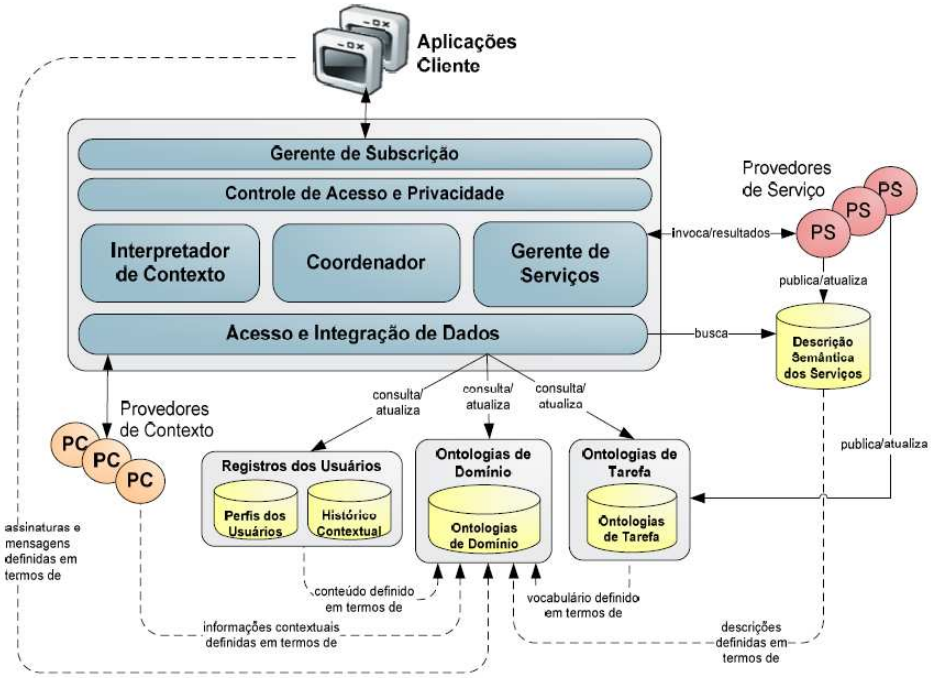
A Infraware disponibiliza serviços baseados na tecnologia de *Web Services* para apoio ao desenvolvimento de aplicações móveis sensíveis ao contexto, possuindo controle de acesso. Esta tecnologia é uma evolução da plataforma WASP (COSTA, 2003), um *middleware context-aware* desenvolvido pela *University of Twente, Telematica Instituut* e *Ericsson* (PEREIRA FILHO J. G.; PESSOA, 2006).

A Infraware utiliza novos componentes para tratamento de questões relacionadas à interpretação semântica de contexto, à obtenção e integração de dados contextuais heterogêneos e distribuídos, à gerência integrada de serviços com descrição semântica, à resolução de conflitos e coordenação entre aplicações e ao tratamento de questões relacionadas à privacidade e segurança. Além disso, a Infraware utiliza fortemente conceitos e tecnologias da *Web* Semântica na sua concepção. Ontologias são usadas para especificar modelos formais extensíveis que descrevem não somente o domínio das aplicações, mas também os serviços. Essa abordagem diferenciada da arquitetura provê meios de configurar as interações entre as aplicações e a plataforma em tempo de execução (*run-time*) (PEREIRA FILHO J. G.; PESSOA, 2006).

A Infraware possui um nível para o recebimento e o tratamento das subscrições das aplicações à plataforma e trata do controle de acesso e privacidade através de um módulo direcionado a tal propósito. A plataforma também soluciona o problema do acesso e

integração de dados heterogêneos através de uma infraestrutura dedicada, e, é capaz de manipular, derivar e interpretar, semanticamente, informações de contexto de domínios variados. Além disso, aborda o problema da resolução de conflitos entre aplicações de maneira diferenciada através de um componente coordenador. A Figura 4 ilustra a arquitetura geral da plataforma.

Figura 4: Plataforma Infraware.



Fonte: (PESSOA et al. 2006)

Os principais componentes da Infraware são os seguintes:

- Gerente de Subscrição: Interpreta e gerencia as requisições enviadas pelas aplicações, através das subscrições que são capazes de exporem suas requisições e necessidades à plataforma, sendo assim o gerente de subscrição permite que as aplicações removam, adicionem ou atualizem pedidos de subscrições. Este gerente também pode validar as mensagens trocadas entre a aplicação e a plataforma, verificar a linguagem de subscrição, verificar a existência de escopo da subscrição, controlar a prioridade em casos de subscrições concorrentes, monitorar a requisição da informação desejada e comunicar com o módulo de controle de acesso e privacidade.

- Controle de Acesso e Privacidade: O módulo de Controle de Acesso e Privacidade atua como um filtro sobre o fluxo de dados entre a plataforma e as aplicações, com base em um conjunto de restrições envolvendo preferências de privacidade dos usuários e as políticas de privacidade das aplicações, permitindo estabelecer limites de visibilidade para os dados coletados pela plataforma. As descrições de preferências de privacidade são modeladas com o uso de ontologias. Essas descrições permitem aos usuários expressar suas preferências sobre a utilização dos seus dados pessoais por outras entidades, garantindo um controle sobre o acesso indiscriminado a seus dados de forma simples e não obstrutiva.
- Interpretador de Contexto: Responsável pela manipulação, derivação, refinamento e inferência de contextos a partir de informações contextuais primitivas, provenientes de diferentes fontes, com a finalidade de torná-las disponíveis para as aplicações de forma transparente. A complexidade do processo de tratamento das informações contextuais em ambientes ubíquos é abstraída para as aplicações.
- Acesso e Integração de Dados: Esse componente fornece aos demais módulos da plataforma e às aplicações uma interface homogênea e transparente de acesso aos dados de diferentes provedores de contexto. As informações contextuais são semanticamente descritas através de ontologias de domínio. Além disso, o uso das ontologias facilita a definição da própria interface de acesso aos dados, através da correspondência entre os esquemas ontológicos e os esquemas do módulo de acesso de integração aos dados. Além do contexto atual, este componente ainda provê acesso às bases de dados contendo o histórico das informações contextuais periodicamente coletadas. Finalmente, as preferências de cada usuário ou configurações de dispositivos são modeladas através de ontologias e armazenadas no repositório de perfil dos usuários. Essas preferências são utilizadas na personalização do acesso aos serviços para os usuários.
- Gerente de Serviços: Responsável pelas atividades de publicação, descoberta, seleção e composição de serviços. A abordagem utilizada na plataforma Infraware, baseia-se na descrição e seleção semântica de serviços, na descoberta dinâmica e na utilização de mecanismos de segurança. A seleção baseada em descrição semântica adotada na Infraware permite a descoberta de serviços que atendam às características descritas pelo usuário, em vez de limitar a seleção em parâmetros

pré-estabelecidos e restritos ao protocolo. A descoberta dinâmica garante que o serviço descoberto esteja disponível no momento da solicitação. Para garantir um nível aceitável de segurança dos dados na descoberta dos serviços, a autenticação, a confidencialidade e o não repúdio, são aspectos considerados pelo protocolo da Infraware.

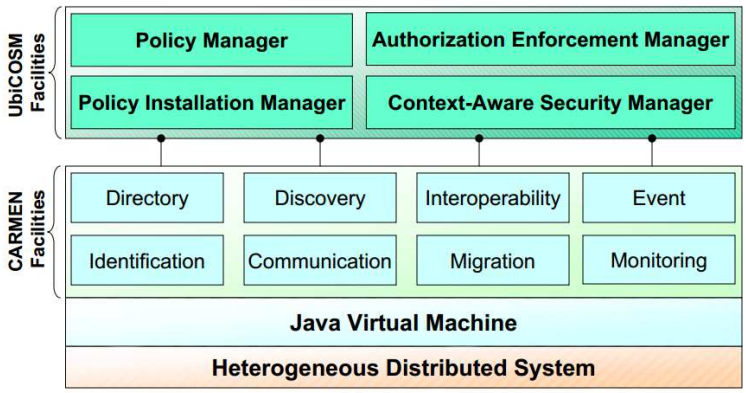
- Coordenador: Responsável pelo gerenciamento, geração e disparo de planos de ações executados por outros componentes do middleware. O Coordenador recebe os pedidos tratados pelo Gerente de Subscrição, já devidamente filtrados e verificados pelo componente de Controle de Acesso e Privacidade. Ele analisa as requisições e elabora um plano de execução de tarefas e atividades a serem realizadas. Em seguida, dispara ações específicas para o Interpretador de Contexto e para o Gerente de Serviços, de acordo com as características das requisições e dos perfis dos usuários envolvidos. Ao final do plano de execução, a resposta é enviada à aplicação solicitante do serviço. Além disso, o Coordenador também age como um componente que gerencia o monitoramento de atividades que exigem um controle contínuo das informações e do contexto geral do sistema e realiza a resolução de conflitos entre as tarefas e atividades da plataforma.

3.2 UbiCOSM

O UbiCOSM (*Ubiquitous Context-based Security Middleware*) é um *middleware* desenvolvido para controle de acesso sensível ao contexto. Além de permitir que administradores de segurança especifiquem as diretivas do sistema de controle de acesso para impedir o acesso ilegal aos serviços locais, ele também permite que os usuários especifiquem os requisitos de privacidade para a divulgação de informações pessoais, ao entrar em um novo contexto (CORRADI et al., 2004).

A Figura 5 apresenta a arquitetura do UbiCOSM. Nesta arquitetura estão presentes módulos responsáveis pelo controle do sistema, que fazem o refinamento do acesso. O sistema possui seus recursos codificados em JAVA (ORACLE, 2012) e módulos como o gerenciador de políticas, o gerenciador de instalação de políticas, o gerenciado de autorizações e o gerenciador de segurança sensível ao contexto.

Figura 5: Arquitetura do UbiCOSM.



Fonte: (CORRADI et al., 2004)

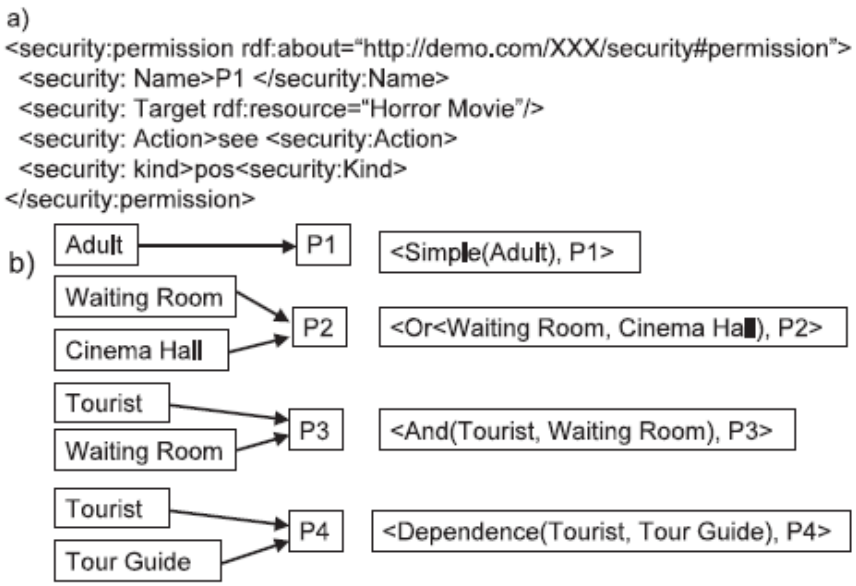
O UbiCOSM utiliza o contexto como base para a segurança da especificação de políticas de execução. Ao contrário dos modelos tradicionais de controle de acesso, as permissões neste modelo estão diretamente associadas com os contextos, ao invés da identidade de usuários e funções (CORRADI et al., 2004).

O trabalho do UbiCOSM, procede-se da seguinte forma: qualquer usuário portando um dispositivo móvel, adquire um conjunto de permissões por entrar em um contexto específico. O UbiCOSM utiliza um formato padrão baseado em RDF para expressar as permissões de controle de acesso. Como mostrado na Figura 6 (a), a definição de uma permissão inclui um nome que identifica a permissão, uma ação que especifica a operação desejada, um alvo que representa o recurso da ação a ser aplicada e um tipo que representa o significado positivo ou negativo da permissão. Figura 6 (a) mostra que a permissão P1 autoriza ao usuário ver filmes de terror.

A UbiCOSM também autoriza que as permissões possam ser associadas a um contexto individual (associação simples) ou a vários contextos compostos.

A semântica da política muda de acordo com o tipo de associação. De acordo com as Figuras 6(a) e 6(b), os adultos têm permissão para ver filmes de terror. Da mesma forma, a permissão P2 se aplica a usuários cujo contexto físico é: ou a sala de espera ou a sala de cinema. Já a permissão P3 está associada ao turista e ao contexto sala de espera e é ativado quando ambos estiverem válidos. A permissão P4 informa que turistas dependem de guias turísticos. Desta forma, as políticas de controle de acesso serão aplicadas a qualquer cliente móvel que esteja operando dentro desse contexto.

Figura 6: Políticas de Controle de acesso a contexto centralizado.

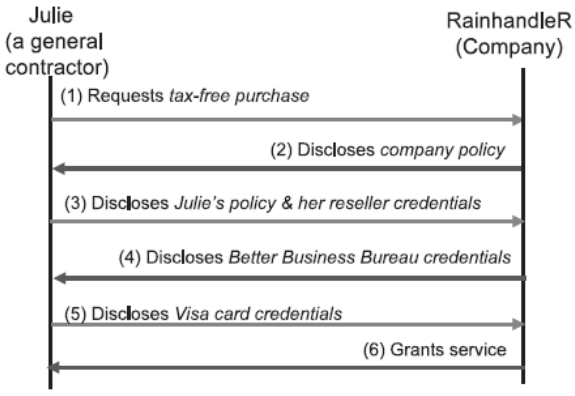


Fonte: (CORRADI et al., 2004)

O sistema também possui recursos de negociação de confiança, que é uma "expectativa generalizada de que a palavra, promessa, oral ou declaração por escrito de outro indivíduo ou grupo pode ser invocada". A negociação de confiança pode permitir que estranhos acessem dados e serviços sensíveis ao contexto, através da Internet. A confiança é a negociação de divulgação de credenciais e pedidos de credenciais entre duas partes, com o objetivo de estabelecer confiança suficiente para que as partes possam completar uma atividade.

A Figura 7 mostra um exemplo de negociação de confiança suportado pelo sistema de negociação para permitir o acesso a serviços.

Figura 7: Negociação de confiança.



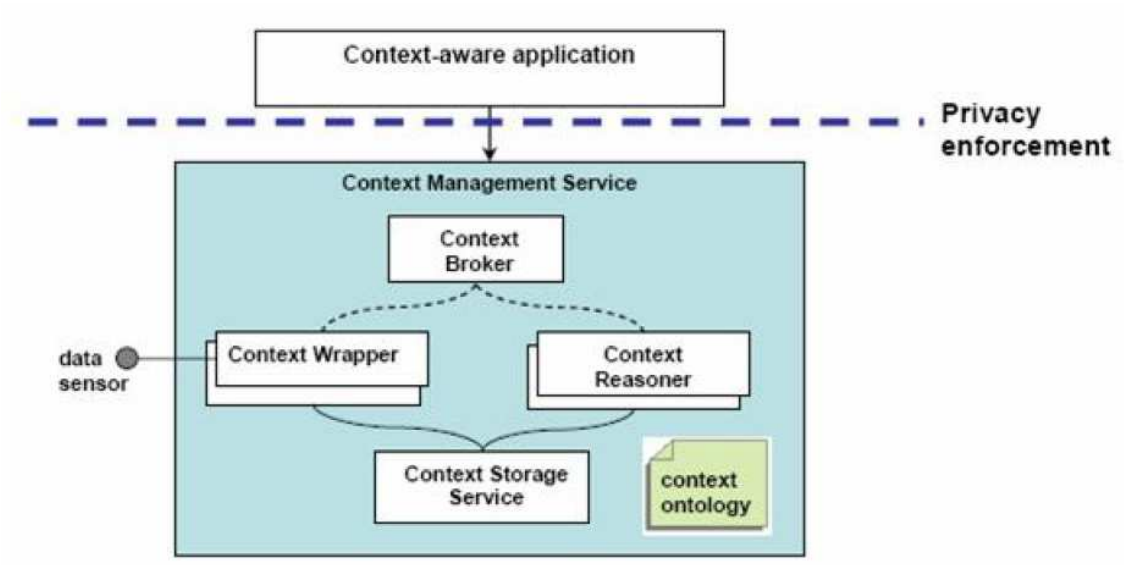
Fonte: (CORRADI et al., 2004)

Para aumento na eficiência, o UbiCOSM deve ser integrado com os mecanismos de mobilidade para antecipar a migração dos utilizadores móveis para novas localidades e para permitir a determinação antecipada de novas permissões.

3.3 AWARENESS

O projeto *AWARENESS (Context-AWARE NEtworks and ServiceS)* foi desenvolvido por instituições holandesas e tem como objetivo a criação de infraestrutura de suporte a serviços e aplicações sensíveis ao contexto com controle de acesso. Cenários médicos reais foram utilizados como áreas de domínio da aplicação (controle de acesso), para validar o desenvolvimento. Este sistema busca a integração entre os serviços amparados pela computação ubíqua, juntamente com o uso de técnicas de processamento de informações contextuais e de aplicações proativas, além do uso de ontologias para a descoberta de novos serviços. Sua estrutura foi projetada para dar suporte à mobilidade do usuário em ambientes sensíveis ao contexto, além de novos métodos de inferência aos contextos em domínios variados (PESSOA,2006).

Figura 8: *AWARENESS.*



Fonte: (WEGDAM, 2005)

Como mostrado na Figura 8, os serviços *Context Wra*pper, *Context Storage Service*, e *Context Reasoner* são os representantes das informações contextuais. O sistema gerenciador

tem o conhecimento de qual contexto está sendo tratado, através de ontologias. O processo *Context Broker* é o que executa a descoberta das fontes.

O sistema possui entidades responsáveis pela integração e controle dos dados com as aplicações e com as fontes. O sistema executa a entrega ativa dos dados segundo o padrão ECA. Este padrão separa três dos estágios vistos, dando assim, grande flexibilidade ao sistema. Estas ações, ao serem executadas, podem ser:

- chamada a *Web Services*;
- resposta para a aplicação ;
- execução de serviço interno.

A *ECA Controlling Service* é a estrutura encarregada por sua execução baseada em *middleware*, que utiliza a arquitetura *publish-subscribe* (SINDEREN, 2006). O *AWARENESS* possui expressividade para eventos compostos, através de lógica booleana, além de aceitar predicados sob demanda.

3.4 SOCAM

O middleware SOCAM (*Service-Oriented Context-Aware Middleware*) foi desenvolvido para um suporte sobre tarefas e serviços em locais que possuam controle de acesso sensível ao contexto em ambientes inteligentes (GU T., 2004).

O SOCAM utiliza a ontologia CONON (*Context Ontology*) (WANG et al., 2004), para que seja permitida a interoperabilidade entre as aplicações.

A CONON foi estruturada em duas partes: uma parte chamada ontologia de alto nível, contendo as definições sobre os contextos básicos como localização, usuário, controle e atividade realizada e outras ontologias específicas de domínio utilizadas como especializações da ontologia de alto nível. Estas ontologias irão definir os detalhes de cada subdomínio.

O reuso de conceitos genéricos fornece uma interface para a definição de conhecimento para a aplicação (WANG et al., 2004).

O SOCAM foi desenvolvido disponibilizando suporte às seguintes tarefas:

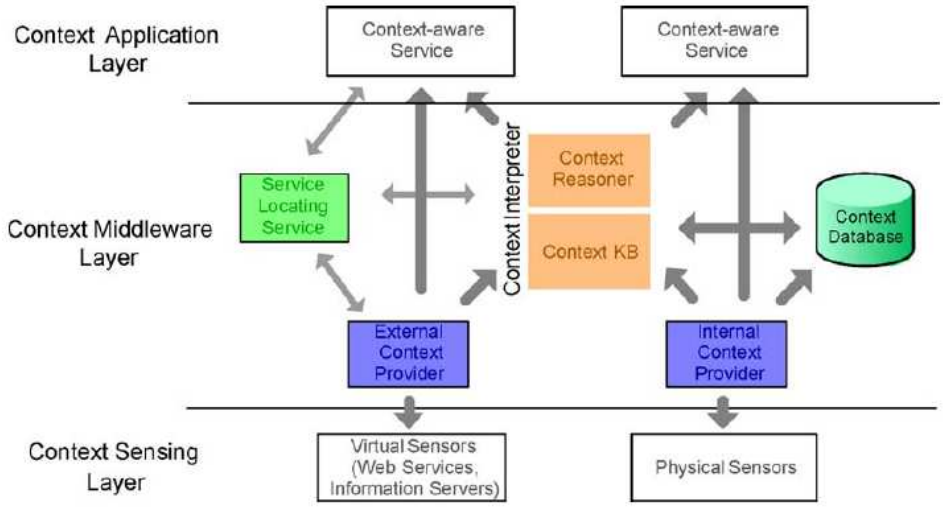
- **aquisição do contexto**: Esta tarefa é realizada pelos componentes *context providers*, que definem os contextos a partir de outras fontes (internas ou externas) e os converte para a ontologia CONON;
- **compartilhamento**: Ocorre através do uso da CONON, permitindo que contextos possam ser compartilhados por todos os componentes da CONON;
- **raciocínio sobre o contexto**: Nesta tarefa são realizadas inferências de contexto, segundo regras definidas, resolução de conflitos de contexto e a manutenção de consistência da base de conhecimento contextual;
- **armazenamento**: É realizado através do uso de uma base de dados de contextos (*context database*) que tem a função de persistir as ontologias de contexto. Esta base de conhecimento de contextos disponibiliza serviços que permitem que outros componentes também possam adicionar, consultar ou remover este conhecimento contextual;
- **disseminação do contexto**: Esta atividade é feita pelo serviço de localização do serviço (*service locating service*). Um mecanismo onde os interpretadores de contexto possam divulgar suas presenças e os usuários possam utilizar os serviços ofertados.

Quando há incertezas em relação à definição do contexto, a SOCAM utiliza redes bayesianas (GU T., 2004). Com este modelo, valores de probabilidades dos predicados são anexados a CONON.

A escolha pelas redes bayesianas é justificada pela eficiência em lidar com raciocínio probabilístico e por permitir representar relacionamentos causais entre vários contextos. A lógica *fuzzy* pode ser utilizada para representar e raciocinar sobre noções imprecisas de contexto, como "quente", "muito baixo" ou "confiança".

Como mostrado na Figura 9, o middleware SOCAM fornece suporte para as seguintes atividades na administração do contexto: aquisição, compartilhamento, raciocínio, armazenamento e disseminação. A aquisição do contexto é realizada pelos componentes *context providers*, que abstraem os contextos a partir de diferentes fontes externas ou internas, e convertem-nos na representação formal da ontologia CONON.

Figura 9: Arquitetura do SOCAM.



Fonte: (GU T., 2004)

3.5 Considerações sobre os trabalhos relacionados

A Tabela 1 mostra um comparativo entre os trabalhos apresentados neste capítulo:

Tabela 1: Comparação entre os trabalhos apresentados.

Trabalho	Sensível ao Contexto	Baseado em Trilhas	Mobilidade	Contextos Dinâmicos	Domínio	Confiança
Infraware	Sim	Parcial	Não	Sim	Genérico	Não
UbiCOSM	Sim	Não	Sim	Sim	Genérico	Sim
AWARENESS	Sim	Não	Sim	Sim	Médico	Não
SOCAM	Sim	Não	Sim	Sim	Genérico	Não

Os aspectos que foram levados em consideração para as comparações foram os seguintes:

- **sensível ao contexto**: Indica se a tecnologia trata a questão de se adaptar ao contexto ao qual está inserida;
- **baseado em trilhas**: Informa se o sistema utiliza alguma informação de trilha para controlar o acesso;

- **mobilidade**: Informa se é possível utilizar serviços em dispositivos móveis;
- **contextos dinâmicos**: Define se é possível adaptar-se a contexto, dinamicamente, ou apenas trata contextos estáticos (predefinidos);
- **domínio**: Informa se o sistema foi projetado para ser utilizado com domínios específicos;
- **confiança**: Indica se a tecnologia faz o uso do atributo de confiança para conceder privilégios.

O Infraware realiza a interpretação semântica do contexto e faz a integração dos dados contextuais para caracterizá-lo. São utilizadas ontologias para especificar modelos e serviços. Todo pedido feito ao Infraware é recebido por uma camada específica e é realizado o controle de acesso e privacidade.

O uso que o sistema faz de trilhas, diz respeito apenas à manutenção de um histórico de contextos já visualizados, ou seja, o sistema controla através de um histórico de acessos os contextos que já foram utilizados pelo usuário, não armazenando as atividades realizadas por este usuário, para uma futura validação ou uso.

O UbiCOSM usa a sensibilidade ao contexto para a especificação de políticas de trabalho (controle de acesso ligado ao contexto e não ao usuário). Ele não faz o uso de ontologias para a definição de contextos e também faz com que permissões sejam associadas a múltiplos contextos. O sistema também utiliza a negociação de confiança entre usuários para acesso aos dados.

O AWARENESS busca a integração entre os serviços disponíveis que sejam amparados pela computação ubíqua, sempre levando em consideração as informações contextuais (controle) e a colaboração de aplicações proativas. Caracteriza-se pelo uso da mobilidade em ambientes sensíveis ao contexto.

O SOCAM fornece uma infraestrutura para a modelagem e criação de serviços sensíveis ao contexto. Os contextos obtidos e trabalhados pelo modelo podem ser compartilhados e acessados através dos serviços disponibilizados. É formado por componentes independentes que se encarregam de controlar os dados periféricos para a consistência das respostas dadas pelo sistema.

Os trabalhos estudados obtêm as informações para controlar o acesso em ambientes contextualizados, baseados em informações registradas, ligadas ao usuário ou ao papel que o usuário possui no momento do acesso. Nenhum dos trabalhos relacionados usa o histórico de acessos para permitir acesso ao serviço do contexto. Este recurso torna o sistema versátil, a ponto de permitir acessos a entidades, que historicamente já utilizaram os serviços (com base em regras de controle de acesso).

O sistema a ser desenvolvido ainda utiliza alguns dos recursos que os trabalhos relacionados descrevem como a sensibilidade ao contexto, a mobilidade e a não vinculação a um domínio específico, além de alguns recursos especiais que serão explicados no capítulo quatro.

Neste capítulo foram abordados trabalhos que utilizam o acesso contextualizado. O próximo capítulo irá tratar sobre aspectos referentes ao modelo *EasyConn4All*, que terá como objetivo dar uma contribuição em quesitos não usados como o tratamento e controle de trilhas.

4 MODELO EASYCONN4ALL

Nesta seção será apresentado o modelo do gerenciador de controle de acesso denominado *EasyConn4All*. Neste capítulo, serão tratados os assuntos que envolvem a criação, o projeto e o desenvolvimento do modelo. Na primeira seção, será feita uma abordagem geral a respeito de sua arquitetura e suas características de trabalho. Na segunda seção, será explanada a arquitetura do modelo. Na terceira seção será tratado o recurso de confiança que é utilizado pelo modelo para delegar permissões a outras entidades. Na quarta seção, serão explanadas as rotinas de controle de acesso feitas no sistema.

4.1 Visão Geral

O sistema de controle de acesso *EasyConn4All* foi projetado visando as seguintes características:

- **Controle de Acesso**: Todo acesso a determinadas atividades dentro de um contexto possuem um nível de controle que mantém a integridade das permissões concedidas às entidades. Esta verificação é feita por módulos responsáveis pelo controle de acesso;
- **Sensibilidade ao Contexto**: O modelo utiliza contextos que são caracterizados por alguns quesitos (ver subseção 2.2) no momento da solicitação de acesso ao sistema. As entidades, somente terão acesso ao contexto, caso todas as informações que sejam coletadas durante a realização do controle de acesso venham a ser verificadas corretamente;
- **Gerenciamento de Trilhas**: O sistema realiza um controle dos acessos e das atividades que a entidade possui concessões e registra isto em uma trilha de acessos, que são posteriormente consultadas para controlar novamente o acesso as atividades;
- **Papéis de Entidades**: Cada entidade poderá ter associada a si um papel que será utilizado no sistema para agrupar atividades que possam ser utilizadas pela entidade. A mudança de papel torna mais fácil o controle da alteração de um grande número de atividades utilizadas pela entidade;

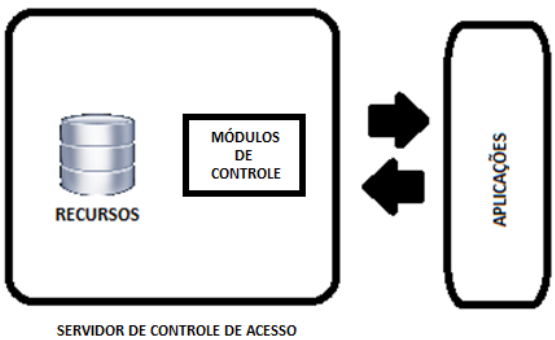
- **Inclusão / Remoção de Entidades**: Uma entidade previamente registrada em um contexto pode ser desligada a qualquer momento do seu uso pelo sistema. Esta atividade se torna pertinente quando se faz necessário que a entidade não receba mais os privilégios dados a ela, pelo papel que portava ou pela trilha que possuía no contexto;
- **Inclusão de Atividades**: O modelo possibilita a inclusão de novas atividades ao gerenciador, fazendo com os módulos controladores tenham mais informações e controles a serem feitos a respeito das atividades disponíveis no sistema;
- **Inativação de Entidades/Atividades**: É possível tornar inativas algumas entidades, que por ventura, venham e ser desligadas do sistema. Este recurso faz com que os componentes não possam coletar informações na trilha de entidades, que não estejam mais ativas no contexto em que elas estejam solicitando acesso;
- **Definição de Regras**: O modelo permite que o administrador do sistema possa definir regras para o comportamento de busca nas trilhas de acesso.

A implementação dos recursos descritos simplifica o processo de controle de acesso em um ambiente contextualizado, dada a grande quantidade de parâmetros disponíveis para refinamento dos critérios de concessão de acesso.

Com estes parâmetros, o sistema se torna flexível, sendo possível sua utilização, praticamente, em qualquer área.

A Figura 10 exemplifica uma visão geral do processo de controle de acesso feito pelo modelo.

Figura 10: Controle de Acesso do modelo.



Fonte: Elaborado pelo autor.

Na Figura 10 é mostrado que o modelo utiliza um servidor para realizar o controle de acesso. Este servidor será composto por módulos que terão a tarefa de coletar dados relacionados ao usuário e fazer um cruzamento destas informações para chegar ao nível de acesso que o usuário da aplicação terá. A forma de trabalho dos módulos e o detalhamento sobre o armazenamento das informações dos usuários serão abordados na sequência deste trabalho.

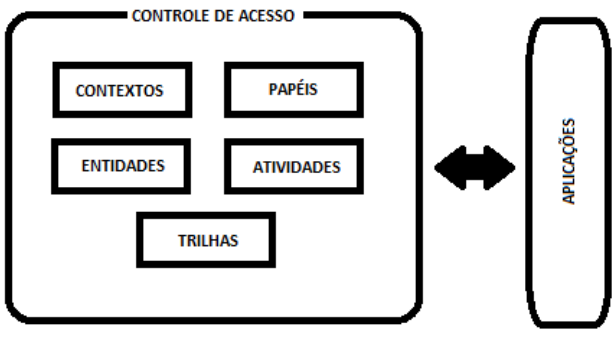
4.2 Arquitetura

O modelo *EasyConn4All* é organizado em cinco módulos que controlarão o armazenamento de informações que são cruzadas pelos módulos responsáveis pelo controle de acesso. Os módulos possuem recursos próprios para armazenamento, exclusão, alteração e inativação de registros para a correta análise por parte dos módulos. Os módulos que fazem parte do sistema são os seguintes:

- entidades;
- papéis;
- contextos;
- atividades;
- trilhas.

A Figura 11 mostra o modelo de arquitetura proposto para o *EasyConn4All*.

Figura 11: Arquitetura do *EasyConn4All*.



Fonte: do próprio autor.

O sistema é composto por dois aplicativos relacionados entre si, que irão trocar informações da entidade e de seus recursos e tomar as decisões para o controle do acesso. Os aplicativos são o ***EasyConn4AllServer*** e as estações clientes chamadas de ***EasyConn4AllClient***.

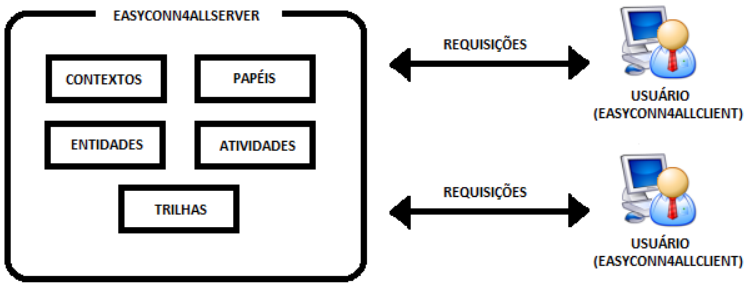
O *EasyConn4AllClient* recebe a identificação da entidade e do contexto ao qual a entidade estiver inserida e providencia o envio destas informações ao *EasyConn4AllServer*.

Após o processamento por parte do servidor, é remetido ao *EasyConn4AllClient* uma descrição das atividades com acesso permitido ou ainda um código de erro informando o motivo da permissão ser negada. Esta mensagem é repassada à entidade pelo aplicativo cliente.

O *EasyConn4AllServer* será o responsável por validar, consistir e remeter a permissão de acesso a atividades registradas, solicitadas pelo aplicativo cliente. No aplicativo servidor ficarão residentes os módulos de controle do modelo responsáveis pela validação das informações repassadas pelo *EasyConn4AllClient*.

Na Figura 12, é mostrada a forma de comunicação entre os sistemas clientes e o servidor de controle de acesso. Na figura é possível verificar que as requisições enviadas pelas estações clientes com o intuito de solicitar o controle de acesso são enviadas até o servidor de controle e este, após processamento da requisição, envia de volta o controle de acesso verificado.

Figura 12: *EasyConn4AllClient* e *EasyConn4AllServer*.



Fonte: do próprio autor.

As próximas subseções descrevem as principais funcionalidades de cada um dos módulos que compõem o modelo.

4.2.1 Entidades

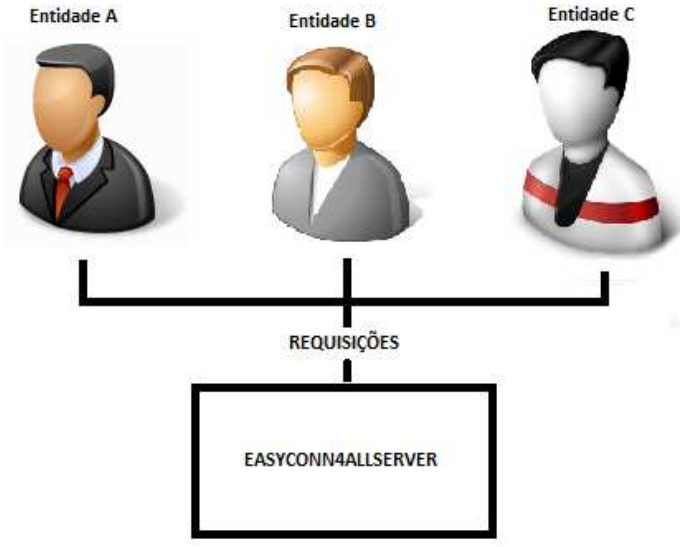
No *EasyConn4All*, entidade é qualquer pessoa ou objeto que venha a interagir com o ambiente, seja por utilizar as atividades disponíveis no contexto ou por fornecer atividades a outras entidades.

Toda entidade no modelo terá seu acesso controlado por módulos. Toda atividade utilizada por ela ou permissível a ela por outro meio (como por papéis ou por atributos de confiança) possuirá um registro do histórico de usos anteriores para serem verificados novamente, por parte de módulos.

Neste sistema, as entidades serão as responsáveis por executar as atividades disponíveis no contexto e por manter o registro de histórico de acessos para futura visualização por parte dos módulos. No modelo, as entidades serão os objetos de maior interação entre as atividades.

A Figura 13 mostra que o modelo controla o acesso de entidades diferentes. Dependendo da entidade, o controle de acesso pode conceder permissões em níveis diferentes, ou seja, entidades diferentes poderão ter permissões diferentes no mesmo contexto.

Figura 13: Entidades e suas atribuições no modelo *EasyConn4All* sendo controladas.



Fonte: Elaborado pelo autor.

4.2.2 Papéis

O uso dos papéis na determinação de atividades que podem ser executadas por algumas entidades foi a melhor forma de administração encontrada. O uso de papéis torna simples a tarefa de gerenciar e administrar quais atividades determinados grupos de entidades poderão executar no modelo. Se por ventura alguma alteração nas permissões for diagnosticada e necessite ser aplicada a um grupo de entidades do modelo, uma simples alteração ou criação de um papel irá satisfazer esta necessidade, visto que neste modelo o papel é responsável por agregar atividades distintas ou grupo de atividades vinculadas a um contexto.

No modelo proposto os papéis serão representados por atributos de uma entidade no banco de dados. Os papéis possuirão um repositório onde serão armazenados no modelo. Estes devem estar associados a entidades em um contexto para serem úteis. Ao realizar o cadastro de uma entidade no modelo basta associar a ela um papel para que o sistema saiba como controlar o acesso.

Para demonstrar o uso e utilidade do papel, diga-se que em um contexto estudantil seja possível ter dois papéis: o *professor* e o *aluno*.

O professor poderá, neste modelo, registrar a frequência de alunos, poderá registrar o resultado das avaliações de cada aluno, poderá postar materiais didáticos para os alunos e poderá receber (via *post*) algumas atividades que ele venha a propor para os estudantes.

Os estudantes, por sua vez, não poderão ter as mesmas atividades, porém os alunos poderão consultar suas avaliações, suas frequências, poderão postar atividades propostas pelo professor e poderão enviar mensagens ao professor (ou a qualquer integrante do grupo), a respeito de alguma atividade proposta, sendo estas mensagens sugestivas, mensagens de questões a respeito das atividades ou ainda mensagens comentando formas alternativas de resolução da atividade proposta.

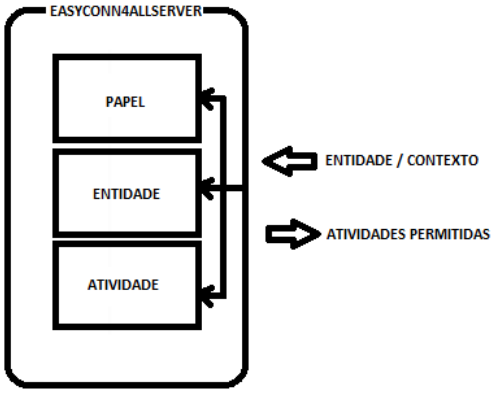
Com o uso dos papéis, esta forma de administração ficaria controlada, dado que seria possível criar um papel (professor, por exemplo) e associar a ele todas as atividades que um “professor” teria disponível sob seu domínio. Se por ventura alguma atividade precisar ser adicionada a um “professor”, todas as entidades que portarem este papel em um determinado contexto seriam contempladas com a mudança.

Caso algum papel tenha que ser retirado do modelo por estar obsoleto ou por ter atividades distintas associadas ao novo modelo de papel, a inativação deste pode ser feita sem empecilhos à análise dos componentes responsáveis, no momento da verificação da trilha de acessos anteriores.

A inativação de um papel faz apenas com que novas entidades (ou ainda entidades já existentes no modelo) não venham a utilizar mais este papel. Este recurso permitirá que todas as entidades que o utilizavam possam ter seu papel alterado no contexto em questão e ter seus novos grupos de atividades associados, com apenas a associação do papel à entidade.

A Figura 14 exemplifica este modelo, em que papéis portam permissões para uma entidade. Na figura é possível verificar que o modelo associa a entidade um papel. Cada papel por sua vez pode ter associado a si atividades que podem ser desempenhadas.

Figura 14: Papéis portam concessões no modelo *EasyConn4All*.



Fonte: Elaborado pelo autor.

4.2.3 Contextos

No modelo *EasyConn4All*, os contextos são tratados genericamente como “espaços de atividades”, em que será possível agregar, neste espaço, entidades e atividades segundo regras definidas previamente pelo administrador do modelo.

O modelo *EasyConn4All* utiliza a sensibilidade ao contexto como forma de controlar o acesso a um determinado espaço, dados os seguintes critérios avaliadores:

- data (início e fim);

- hora (início e fim);
- local;
- entidades presentes;
- atividades disponíveis.

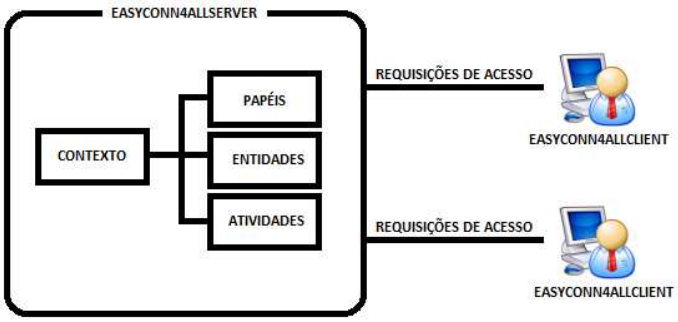
A sensibilidade ao contexto consiste na capacidade de um sistema computacional em estar ciente do contexto de atuação. Em uma situação normal, onde duas pessoas conversam, elas poderão utilizar explicitamente informações que acontecem ao seu redor para melhorar a interação que ocorre entre elas. Este modo de interação não ocorre entre humanos e computadores, uma vez que computadores não são habilitados a explorar o contexto durante a interação com os humanos. Desta forma, definir o contexto no âmbito de sistemas computacionais se torna uma tarefa delicada, pois este precisa ser tratado num nível de detalhamento que permita seu processamento interno, no sistema computacional que está sendo desenvolvido.

Levando estas características em questão, o modelo utiliza as informações descritas, anteriormente, como pontos básicos para a definição do contexto. Sabendo a data e hora de ocorrência do mesmo, o local onde ele está em atividade, os recursos (atividades) que estão disponíveis nele e quais entidades estão utilizando-o, são informações suficientes para a determinação do contexto.

A alteração de algumas das informações de definição do contexto durante o uso do mesmo acarretará em uma iniciativa dos componentes responsáveis de interromper o uso por “descaracterização”. Caso uma das informações que permitiram o uso do contexto venha se tornar inválida (como por exemplo, o término do tempo de existência do contexto) acarretará na desqualificação deste.

Na Figura 15 está detalhado o processo de busca das atividades de uma entidade que estão associadas a um papel em determinado contexto. Como mostrado na figura, uma entidade faz uma requisição de acesso e o modelo resgata a associação Entidade/Papel/Atividade que está ligada ao contexto em que a entidade estiver inserida no momento.

Figura 15: Relação Entidade / Papel / Atividades em um contexto.



Fonte: do próprio autor.

4.2.4 Atividades

No modelo proposto, atividades serão todos os recursos disponíveis em um determinado contexto que terão o objetivo de servir como “utilitários” para alguma entidade que estiver inserida no contexto.

Cada contexto registrado no modelo possuirá *N* atividades ligadas a ele e permissíveis de uso por determinadas entidades portadoras de um papel específico. Toda atividade no modelo poderá ter seu uso dirigido a entidades específicas ou a um grupo de entidades através da atribuição da mesma a papéis distintos.

Para demonstrar o uso de atividades no modelo, novamente será utilizado o exemplo de um ambiente de estudos. Em uma sala de aula, o “professor” poderia ter as seguintes atividades associadas a seu papel:

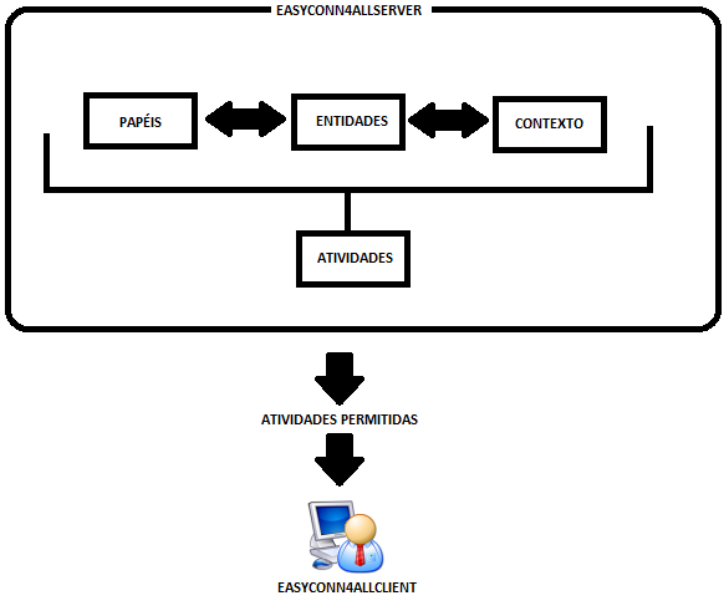
- Registrar Frequências;
- Registrar Avaliações;
- Postar Exercícios;
- Postar Avaliações;
- Enviar Mensagens a Alunos;
- Receber Atividades realizadas por alunos.

Estes são modelos de atividades que poderão ser descritos na base de dados de controle de acesso feito pelos componentes responsáveis. Todas estas atividades poderão se

tornar inativas por parte do administrador e fazer com que papéis que utilizem a atividade em seu modelo não mais disponibilizem o seu uso.

A Figura 16 mostra que as atividades suportadas pelo modelo são associadas a um papel que uma entidade porta em um determinado contexto. Com base nesta relação, o sistema busca quais são as atividades que possuem permissão de acesso por esta entidade nestas condições.

Figura 16: Atividades vinculadas a Entidade/Papel/Contexto.



Fonte: Elaborado pelo autor.

4.2.5 Trilhas

O modelo de trilhas do sistema *EasyConn4All* monitora e registra todos os acessos feitos no seu banco de dados, para uma futura averiguação de atividades executadas previamente.

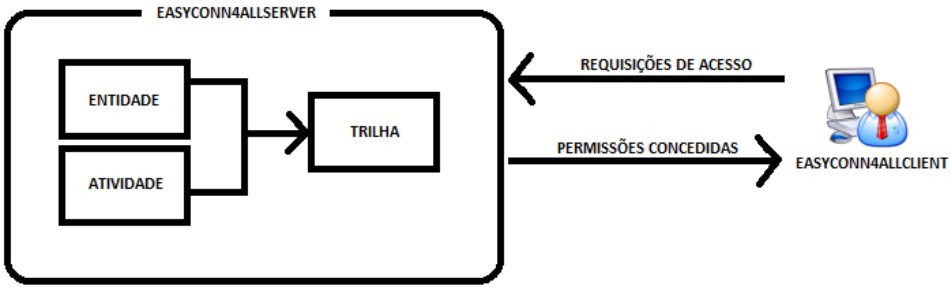
O uso de trilhas já tem sido utilizado por outros sistemas para torna-lo personalizado. O trabalho relacionado à educação descrito em (CAMBRUZZI, 2012) utiliza múltiplas trilhas com o objetivo de gerenciar os dados gerados pela entidade que estará sendo acompanhada, permitindo a disponibilização e aplicação em sistemas de apoio à educação.

No modelo proposto, a análise da trilha tem o objetivo de permitir que atividades previamente utilizadas possam servir de parâmetro para novas concessões.

As trilhas, no modelo, serão registradas em um formato sequencial, contendo informações que caracterizem qual atividade foi utilizada previamente, em que situações e por quem ela foi utilizada. No momento em que os componentes responsáveis forem verificar as trilhas de uma entidade e constatarem o acesso, e, ainda assim não encontrarem requisitos que bloqueiem ou impeçam a liberação do recurso para a entidade, esta atividade será novamente liberada para o uso.

Todo acesso feito ao modelo ocasionará um registro na trilha da entidade. Em outro acesso que a entidade venha a ter no modelo, mesmo que agora portando um novo papel dentro do contexto, o sistema fará o registro das atividades disponíveis em sua trilha de acessos.

Figura 17: As trilhas de atividades realizadas são controladas pelo sistema.



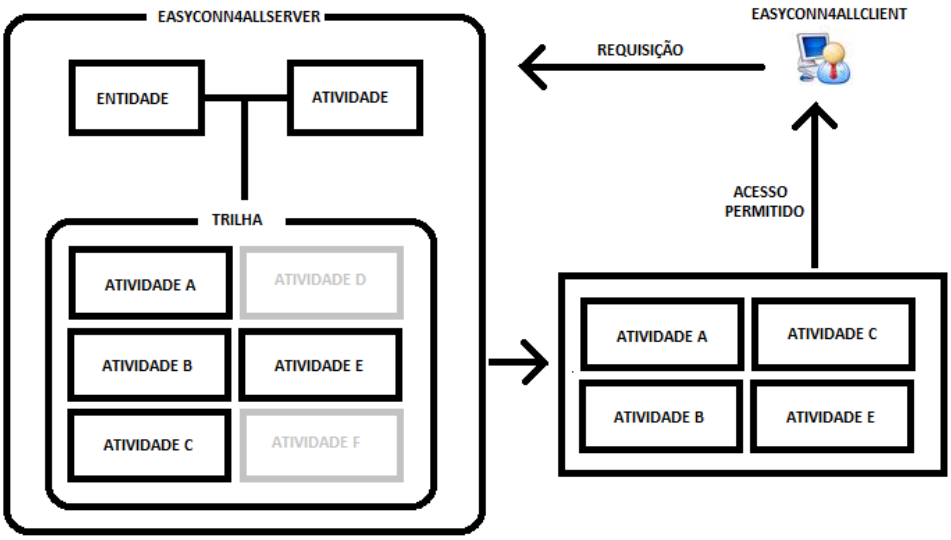
Fonte: Elaborado pelo autor.

A Figura 17 monstra o processo de registro de trilhas no modelo. Uma vez que uma entidade tenha executado uma atividade em um contexto, esta passa a fazer parte da trilha de atividades realizadas. No modelo, se o contexto disponibilizar a atividade para as entidades e a entidade tiver o registro do uso em sua trilha, será disponibilizada a ela também.

Como o modelo possui recursos de configuração de regras e inativação de atividades, será possível tornar inativa alguma atividade específica para o uso da entidade, sem que os componentes responsáveis venham a validar este acesso. Isto será possível, pois o modelo é capaz de transformar atividades em inativas e torná-las “invisíveis” para a ação dos componentes. Este recurso pode ser entendido, por exemplo, como se ele estivesse sendo utilizado por uma entidade e viesse a se danificar ou a ser retirado do contexto. Mesmo que a trilha de acessos registre um uso prévio deste recurso, os componentes responsáveis não concederão o acesso para a entidade que a requisita.

A Figura 18 mostra como as atividades inativas não são reconhecidas pelos componentes responsáveis e não são concedidas permissões para o uso posterior. Na figura é possível ver atividades ativas e algumas inativas (que compõem a trilha da entidade). Ao serem resgatadas pelos componentes, não serão vistas como “válidas” e não é concedido o acesso a elas novamente.

Figura 18: Atividades ativas e inativas no momento da busca.



Fonte: Elaborado pelo autor.

O sistema ao conceder o acesso não irá buscar as atividades que não estejam ativas para novos usos.

No modelo proposto, as informações persistidas a respeito dos recursos utilizados pela entidade são cruzadas e avaliadas pelos módulos do sistema, segundo as regras estabelecidas no contexto. Esta forma de análise feita no momento do controle de acesso por parte dos módulos caracteriza-se em uma das tarefas mais importantes na definição das permissões de atividades.

4.3 Confiança

O modelo cria o controle de acesso a seus recursos fazendo uma leitura das atividades associadas ao papel em que a entidade está vinculada ou ainda verificando a trilha deixada pela entidade em um determinado contexto.

Quando for necessário delegar o uso de uma atividade a uma entidade que não possua esta atividade ligada a seu papel, um dos recursos utilizados no modelo para contornar este problema é o uso da atividade pelo princípio da troca de confiança.

O modelo faz o uso da confiança entre as entidades para delegar permissões de acesso umas às outras, partindo do pressuposto que uma entidade com permissões aceitas, confia na índole e nas boas intenções de outra, delegando a ela o uso desta atividade também.

A concessão de permissões de acesso a uma entidade pode ser definida por tempo determinado ou por um prazo indeterminado. Por se tratar de permissão com prazo de validade, esta permissão não será associada à trilha de acessos da entidade e assim, não será verificada posteriormente por parte dos módulos.

No modelo a confiança é tratada como uma tabela de dados, onde é armazenada a identificação da entidade que detêm o uso de uma atividade, a identificação da atividade a ser compartilhada com outra entidade, a identificação da entidade que receberá o benefício e também o período que a atividade ficará disponível para a entidade de destino.

No modelo o uso da confiança é feito utilizando uma solicitação de compartilhamento de atividades, sendo necessário apenas informar a entidade de origem da atividade, a entidade de destino e o período de validade desta confiança.

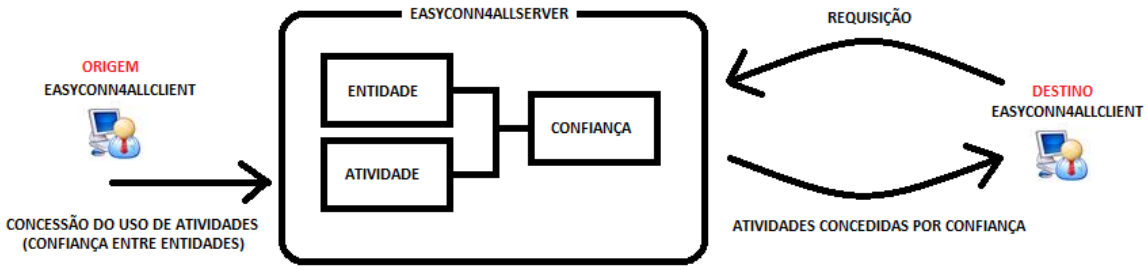
Uma vez atribuída a confiança à outra entidade, caberá aos módulos, no momento da verificação das atividades disponíveis, verificar se existem atividades concedidas pelo critério de confiança e ainda se elas permanecem ativas.

Com o uso da confiança é possível alterar atribuições de entidades que não possuam determinadas atividades associadas ao seu papel no contexto ou ainda permitir que uma entidade faça o uso de determinadas atividades, momentaneamente, em um contexto mesmo que ela não possua papéis que permitam este acesso.

A permissão da atividade de uma entidade para outra irá se valer da permissão atual que a entidade possui, ou seja, ela poderá delegar o acesso momentâneo por confiança somente naquelas atividades que ela própria possui acesso.

A Figura 19 mostra a delegação de atividades através do uso da confiança.

Figura 19: Confiança no modelo *EasyConn4All.*



Fonte: Elaborado pelo autor.

4.4 Regras

O modelo *EasyConn4All* possui recursos que permitem ao administrador definir regras para o comportamento de busca de informações.

A definição de regras se torna um recurso útil no modelo, pois é com a definição destas regras que será possível buscar as trilhas de uma entidade em diferentes contextos.

Para a definição das regras, o modelo utiliza uma tabela de dados que possui os seguintes registros para definir as normas de trabalhos dos módulos quando forem requisitados:

- **Somente Contexto em Uso**: Este registro da tabela de regras indica se o componente, ao fazer a busca pelas atividades feitas anteriormente, irá buscar as atividades feitas no contexto registrado em que a entidade tenha estado presente ou se os módulos podem buscar todas as atividades feitas pela entidade, não importando em qual contexto esta atividade tenha sido utilizada anteriormente;
- **Somente Serviços Ativos**: Informa aos componentes se ao resgatarem atividades previamente executadas eles poderão resgatar todas as atividades ativas (em uso) do modelo ou se eles deverão resgatar todos os registros de entidades independente de a atividade estar ativa ou inativa.

O recurso funciona ligado a um contexto específico, ou seja, as regras que serão definidas serão aplicadas no contexto em que a regra de ação dos módulos está sendo definida. O aumento da abrangência de ação dos componentes poderá ser feito neste recurso, tornando o escopo de ação dos componentes mais abrangentes.

As regras poderão sofrer alterações após serem criadas. Se for necessário que os componentes mudem a forma de fazer a busca das atividades, aumentando ou diminuindo o escopo de atuação, o sistema fornece um recurso para alteração destas regras e assim fazer com que os componentes mudem também os critérios de buscas futuras.

5 ASPECTOS DE IMPLEMENTAÇÃO

Com o objetivo de tornar o modelo proposto uma ferramenta de uso prático, um protótipo foi desenvolvido com as funcionalidades descritas na especificação do modelo.

Junto ao desenvolvimento do aplicativo, cenários de uso para o protótipo foram criados, em que foi possível avaliar sua operacionalidade, medir sua eficiência e apontar seus pontos vulneráveis.

Este capítulo está dividido de forma a demonstrar o processo de criação do protótipo, desde sua análise e concepção até a sua codificação e aplicação prática. Primeiramente, será explanado sobre o processo de concepção do aplicativo, onde foram projetados os módulos principais de controle de acesso, e, após serão explanadas as técnicas utilizadas para efetivar a funcionalidade do mesmo.

Este capítulo não tem o objetivo de demonstrar a fundo todo o processo de implementação utilizado, mas sim, de salientar os tópicos principais que tornaram possível o desenvolvimento deste projeto.

5.1 Implementação do *EasyConn4All*

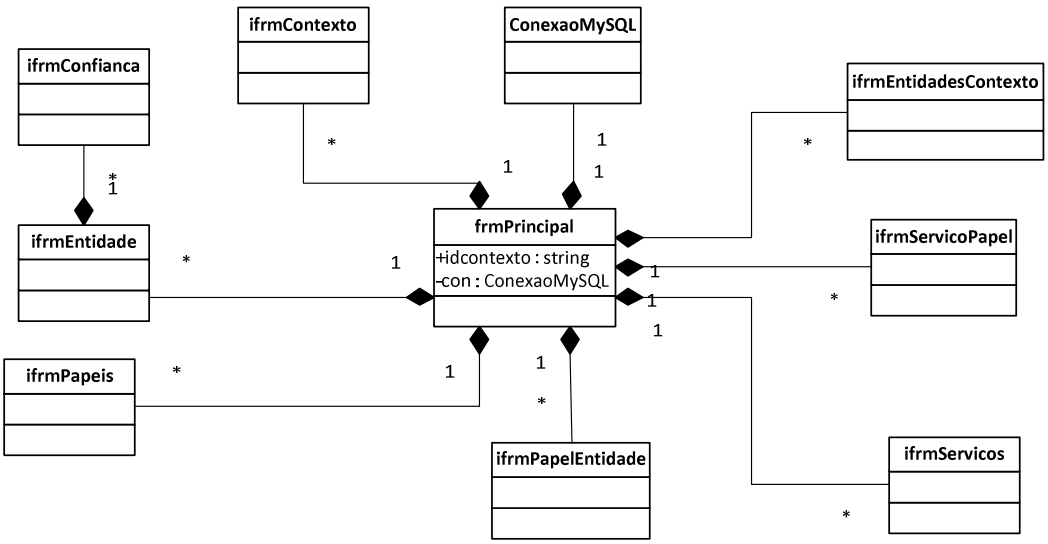
A implementação do aplicativo foi dividida em etapas de análise e projeto do protótipo. Logo em seguida foi realizada a etapa de codificação e testes, onde foi possível realizar a avaliação.

A etapa de análise foi o ponto de partida da criação do *EasyConn4All*, sendo necessário fazer coleta de material teórico sobre o tema de controle de acesso contextualizado, analisar a funcionalidade de projetos relacionados, e finalmente, propor o modelo conceitual que em seguida foi utilizado para a codificação.

No momento da modelagem das classes que comporiam o protótipo, foi utilizada a ferramenta UML (*Unified Modeling Language*). O UML (GROUP, 2012b) é um padrão internacionalmente reconhecido e difundido tanto no meio acadêmico quanto no mercado, para criação de diagramas estruturais e comportamentais que visam auxiliar o desenvolvimento de um sistema de informação.

A ferramenta utilizada para modelar as classes que irão compor o protótipo, foi a Microsoft Visio 2010. A Figura 20 monstra o diagrama de classes com o uso da composição entre classes utilizadas no protótipo.

Figura 20: Diagrama de classes do modelo *EasyConn4All*.



Fonte: do próprio autor.

O diagrama de classes mostra a estrutura de composição das classes e o fluxo de informação que deve trafegar dentro do protótipo. A interação entre os objetos das classes é necessária para a correta aplicação das técnicas propostas.

O *EasyConn4AllServer* opera seus módulos de cadastro e configuração através de interface MDI (*Multiple Document Interface*) onde existe na classe *frmPrincipal* um objeto *JDesktopPane* responsável pelo armazenamento das *JInternalFrames,* que exibirão os recursos disponíveis.

Cada uma das classes especificadas no modelo é responsável pelo gerenciamento das informações referentes ao assunto que a classe manipula. Serviços como a inclusão, a alteração (modificação dos registros), a vinculação ou até mesmo a inativação de recursos são atividades inerentes a cada uma das classes responsáveis.

Uma breve descrição será feita sobre as responsabilidades e incumbências de cada uma das classes do modelo.

A classe **ifrmEntidade** é a classe responsável pelo gerenciamento do controle das entidades do modelo. Entidade pode ser definida como uma pessoa ou um objeto relevante para a interação entre um usuário e uma aplicação. Esta classe controla o cadastramento de todas as entidades que poderão utilizar algum recurso (atividade) ou ceder serviços para o modelo. Ela possui atributos próprios que são característicos de uma entidade, como seu *login* e senha de acesso, que poderão ser utilizados para medir seu nível de permissões.

É possível, nesta classe, tornar uma entidade inativa através da alteração de seu *status* de atividade no modelo, impedindo assim o acesso futuro.

A classe **ifrmPapeis** controla a administração dos diferentes tipos de papéis gerenciados pelo modelo. Os papéis serão os responsáveis por agrupar atividades afins dentro de um contexto. Também é possível neste recurso tornar um papel inativo para futuros (e atuais) usos.

A **ifrmContexto** é a classe que administra o cadastramento das regras e das definições de um contexto. Para caracterizar um contexto, o sistema utiliza algumas informações que podem ser usadas para definir a situação de uma entidade. Neste sentido, o modelo armazena em sua base de dados, os seguintes atributos para qualificar um contexto:

- data e hora de ocorrência do contexto;
- localização;
- número mínimo de entidades presentes no contexto;
- definição da presença obrigatória de entidades no modelo;
- atividades pertencentes ao contexto.

Estas são as informações necessárias à definição e qualificação do contexto. Qualquer destas informações que venha a se tornar inválida transformará o contexto em inválido e o modelo tomará a iniciativa de torná-lo inacessível.

A classe **ifrmServicos** gerencia a inclusão de atividades que terão seu acesso controlado pelo servidor. É possível, na classe, tornar uma atividade inativa temporariamente ou definitivamente, através da modificação de atributos. Todas as tarefas referentes à administração dos serviços também estão disponíveis, como a inclusão de novos serviços e a atualização de serviços já cadastrados (uma análise de integridade referencial será feita antes destas operações).

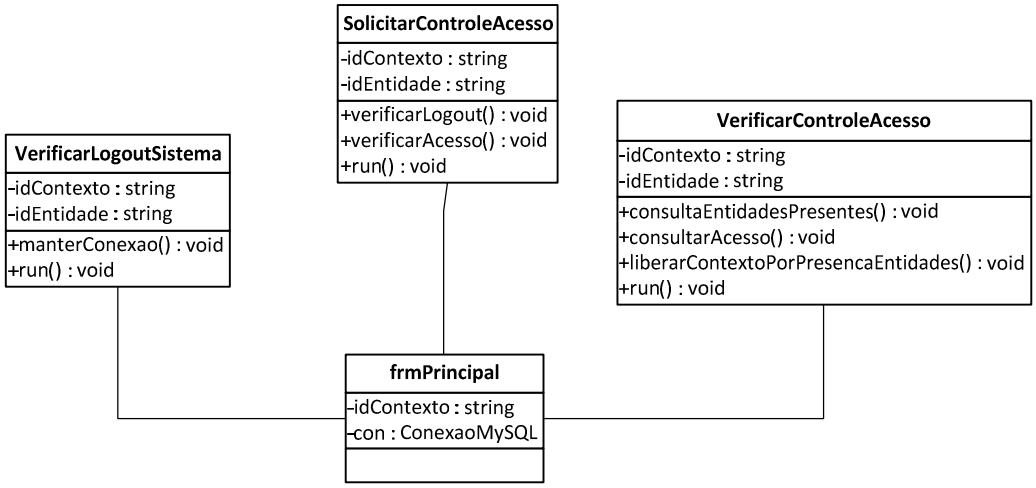
As classes **ifrmPapelEntidade**, **ifrmServicoPapel**, **ifrmEntidadesContextos** são utilizadas para fazer ligação entre os objetos instanciados em cada uma das classes abordadas anteriormente. Estas classes farão o vínculo necessário para a posterior comparação e análise por parte dos componentes, na autorização do uso destas atividades.

A classe **ifrmConfianca** é utilizada para poder conceder permissões entre entidades utilizando o atributo de confiança entre elas. Nesta classe estão disponíveis recursos que podem conceder temporariamente a permissão do uso de uma atividade que a entidade possua a concessão para outra entidade que esteja necessitando também utilizá-la. O período de concessão poderá variar desde o acesso por um único dia até por um período indeterminado.

Ainda estão presentes no modelo as classes **VerificarLogoutSistema**, **SolicitarControleAcesso** e **VerificarControleAcesso** que têm o objetivo de validar e controlar o acesso das entidades para o uso das atividades de um contexto. Estas classes possuem métodos que fazem a verificação de informações em tempo de uso, e, com isto consistem na permanência ou não, de uma entidade em um determinado contexto.

A Figura 21 mostra o diagrama de classes referente às classes responsáveis pela continuidade do acesso concedido.

Figura 21: Classes responsáveis pelo Controle de Acesso.



Fonte: do próprio autor.

Todo o processo de implementação do *EasyConn4AllServer*, que serve de responsável pelo controle de acesso para as estações clientes, foi implementado utilizando a linguagem JAVA (ORACLE, 2012) e foi utilizado como editor de código a ferramenta NetBeans

(NETBEANS, 2012), devido à facilidade de uso e também por possuir recursos que auxiliam o desenvolvimento, como a importação de pacotes de classes de terceiros e os próprios recursos de codificação do modelo, como o auto complemento de código e a depuração avançada de classes, em que é possível monitorar pontos críticos do sistema e assim controlar suas ações.

Para a persistência dos dados, o modelo utiliza o software MySQL (MYSQL, 2012). O sistema é um gerenciador de banco de dados, já difundido entre os utilizadores deste tipo de software, além de oferecer ferramentas extras de controle e funções proprietárias que facilitam a implementação por parte do programador. Por este motivo o *EasyConn4AllServer* faz todas as suas requisições de dados sobre uma base de dados MySQL.

O modelo desenvolvido é formado por duas partes distintas: uma que executa em cada uma das estações clientes e outra que fica residente na estação principal que irá controlar o acesso.

O aplicativo da estação principal (*EasyConn4AllServer*) será responsável por receber as requisições das estações clientes e realizar as verificações de controle de acesso. Ele fará a recepção da solicitação de verificação e efetuará as verificações propostas no modelo, realizando a conferência da entidade, sua relação com o contexto através do papel e delinear as atividades disponíveis através da verificação de suas permissões e da trilha de acessos anteriores. O resultado de sua pesquisa é enviado ao sistema cliente (*EasyConn4AllClient*) que será responsável por efetivar o controle de acesso nas estações clientes.

O *EasyConn4AllClient* possui uma versão produzida para ser executada sobre a plataforma Android [1] . O protótipo segue os mesmos padrões de operabilidade do *EasyConn4AllClient*, só que nesta versão se tem a possibilidade da execução do aplicativo de forma móvel, ou seja, o dispositivo que portará o programa é um dispositivo móvel e seu deslocamento dentro do ambiente deve ser levado em consideração.

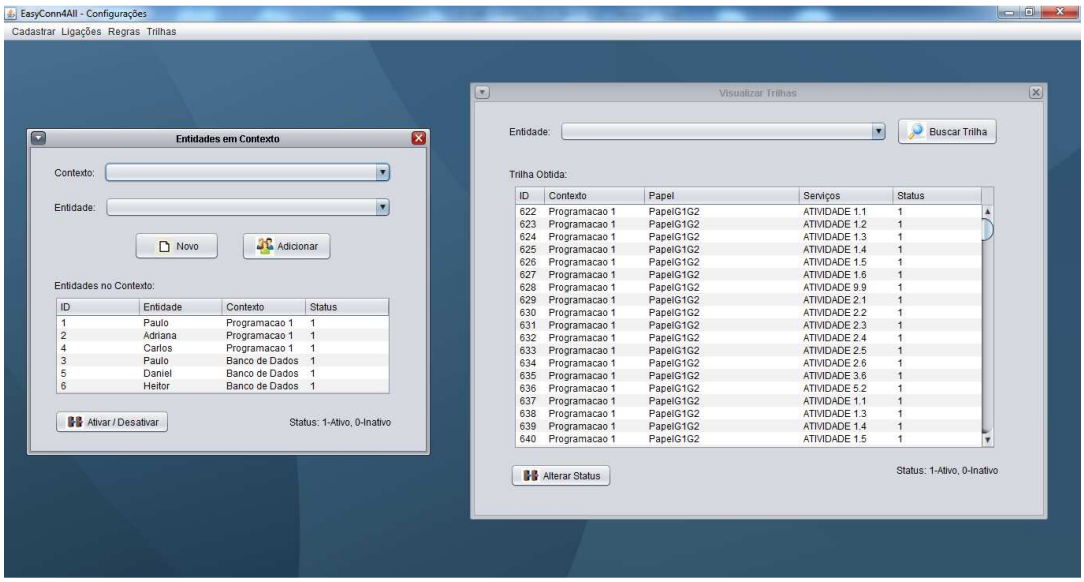
A Figura 22 mostra o aplicativo de controle de acesso *EasyConn4AllServer*. Na figura está em execução o módulo de controle de entidades pertencentes a um contexto (na esquerda) onde é possível fazer o cadastro e a manutenção das entidades que poderão estar presentes em um contexto no sistema. Do lado direito está o módulo de trilhas, onde é possível visualizar as trilhas de acessos de determinada entidade.

[1] http://www.android.com/

Operando em segundo plano (através de *Threads*) no sistema estão os componentes responsáveis pelo controle de acesso solicitado pelos sistemas clientes. Eles acompanham as solicitações feitas pelas estações e se responsabilizarão pela coleta e refinamento de dados relativos à permissão de acesso.

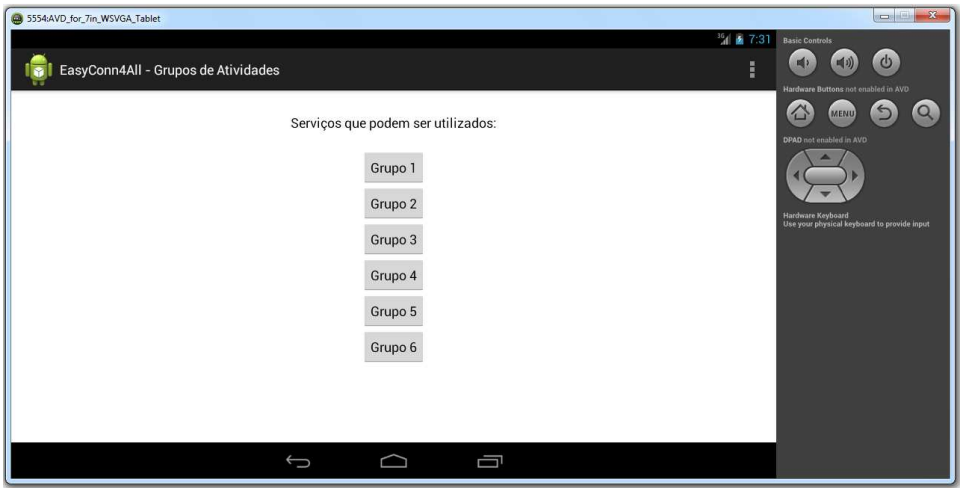
A Figura 23 mostra o sistema *EasyConn4AllClient* executando, em um simulador para dispositivo móvel, pronto para enviar requisições ao servidor de controle de acesso.

Figura 22: *EasyConn4AllServer* em execução.



Fonte: do próprio autor.

Figura 23: *EasyConn4AllClient* executando em um simulador de dispositivo móvel.



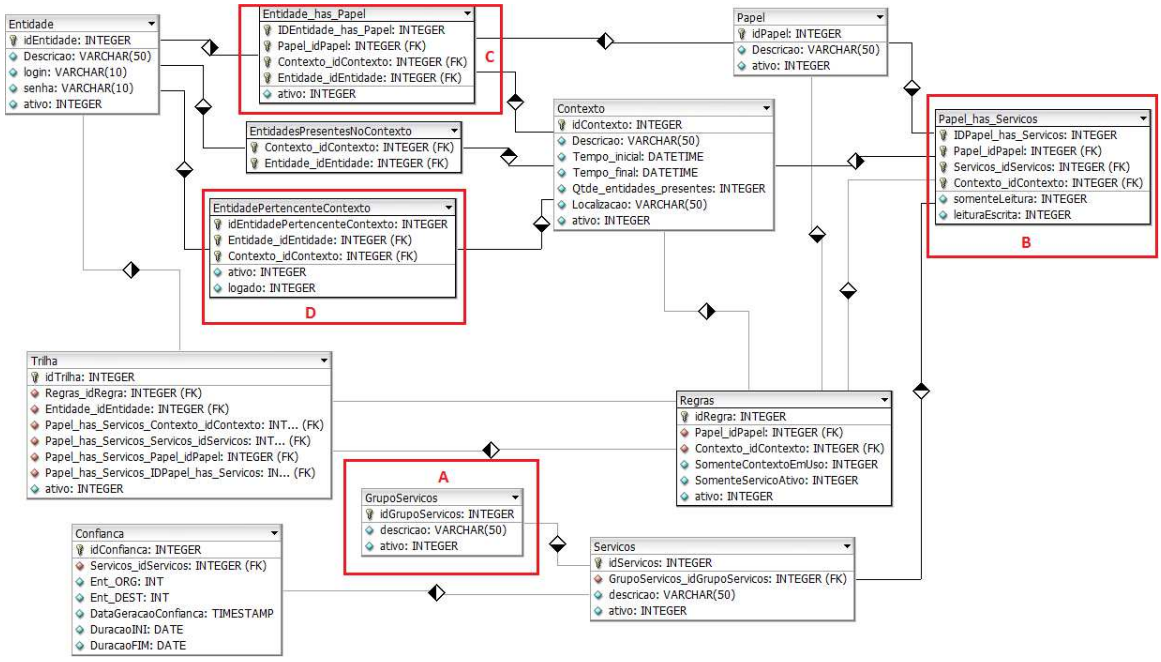
Fonte: do próprio autor.

5.2 Relacionamentos entre Recursos

O *EasyConn4All* armazena os registros de cada um dos recursos indispensáveis à eficácia do modelo de forma que possam formar uma “rede de interligações” em um modelo relacional. Com este relacionamento será possível determinar, por exemplo, qual entidade utilizou qual atividade inserida em um determinado contexto.

A Figura 24 mostra o modelo ER do sistema e suas relações.

Figura 24: Modelo ER do Sistema de Controle de Acesso *EasyConn4All.*



Fonte: do próprio autor.

Serão descritas na sequência as relações mais relevantes no controle de acesso:

- **Grupos de Serviços**: Os serviços ou atividades, vistos aqui como sinônimos, podem possuir finalidades relacionadas, como por exemplo, atividades de uso de um estudante. Por se tratar de serviços que poderão ter fins semelhantes e objetivos relacionados, o modelo permite que os serviços sejam agrupados em atividades com finalidades em comum. Com este fim, o protótipo implementa uma relação de grupos de atividades, destacado na Figura 24a, em que é possível agrupar atividades e controlar melhor a permissão de acesso a várias atividades simultaneamente.

- **Serviços de Papéis**: Os serviços do protótipo, destacado na Figura 24b, poderão ser associados a um papel e assim disseminar a permissão do seu uso para um

grande número de entidades com a associação do serviço a um papel que pertença ao contexto. Desassociando a atividade do papel, o controle de acesso atingiria todas as entidades diretamente ligadas ao papel que necessitou ser alterado.

- **Papel de Entidades**: Cada uma das entidades pertencentes ao modelo portará consigo um papel que definirá qual ou quais atividades esta entidade poderá usufruir no modelo, como destacado na Figura 24c. Uma entidade pode ter papéis diferentes em contextos diferentes.
- **Entidades pertencentes a um contexto**: Como descrito anteriormente, cada uma das entidades do modelo poderá ter associado a si um papel que possuirá atividades que podem ser utilizadas por ela, como demonstra a Figura 24d. Cada entidade poderá estar inserida em vários contextos com permissões diferentes (em tempos diferentes). Sendo assim o modelo prevê que as entidades, para terem seus acessos controlados, deverão estar ligadas ao contexto que pretendem utilizar com um determinado papel associado a si. Cada entidade, alterando de contexto, terá seu nível de acesso alterado também pelo sistema.

5.3 Detalhes da Implementação

O sistema *EasyConn4AllServer* tem aproximadamente 5000 linhas de código contemplando cadastros básicos, recursos de manutenção (alteração e inativação de dados), recursos para a consistência, incluindo as ligações que são possíveis de ser definidas entre os recursos, e telas, para realizar consultas de trilhas de acessos.

O sistema utiliza o conceito de MDI (*Multiple Document Interface*) para exibir as telas dos segmentos utilizados. Este recurso permite que o utilizador possa abrir várias telas para consultas relacionadas ou para ativação/inativação de registros em tempo de execução.

Toda vez que um novo registro é criado, seja ele originário de qualquer um dos recursos que compõem o protótipo, o sistema irá validar a sua criação através de consistência nos campos informados pelo utilizador. Qualquer discrepância na informação cedida ocasionará um pedido de verificação ao usuário por parte do sistema.

Estando corretos os dados informados, o sistema se encarrega de encapsular a informação e a envia para o SGBD fazer seu registro. Assim que o banco de dados retornar o

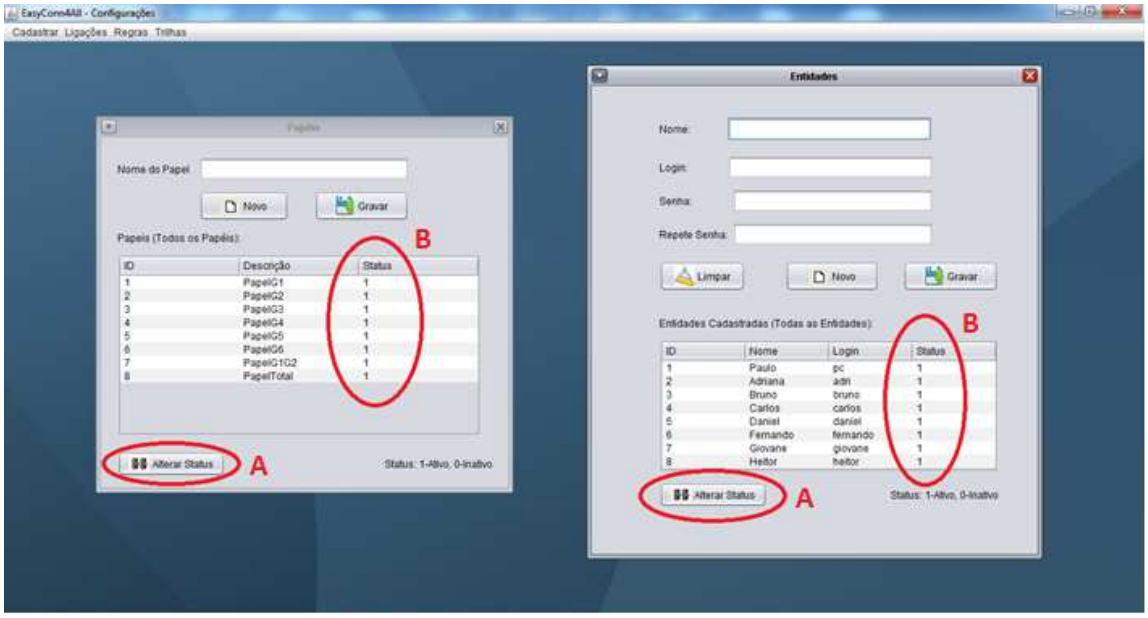
sucesso na inserção do registro, será atualizada uma *JTable* que terá as informações recém-inseridas na base de dados para a verificação do utilizador.

As informações previamente inseridas em cada uma das bases de dados, eventualmente, poderão sofrer alguma alteração. O sistema permite que o usuário possa realizar algumas alterações simples. Levando em conta, neste quesito, que informações de ligação entre os recursos dificilmente poderão ser alteradas por motivos de integridade entre as tabelas do banco de dados.

Como o sistema não oferece a oportunidade de exclusão, o que é disponibilizado no modelo é um recurso para desativar o registro. Isto tornará a informação “oculta” e fará com que o protótipo possa tornar inativas informações que não mais sejam válidas ou que por algum motivo tenham sido desativadas. Esta tarefa pode ser desfeita a qualquer momento pelo utilizador do *EasyConn4AllServer*.

Como mostrado na Figura 25, os módulos possuem um botão de ação (A) que tem o objetivo de tornar o registro inativo (ou ativo) para os componentes ou para registros futuros. Na figura da esquerda está demonstrado o módulo de controle de papéis do sistema. Por definição os papéis são criados como “ativos”. Se for necessário fazer com que nenhuma entidade mais possa utilizar este papel, basta torná-lo inativo para que nenhuma entidade mais possa utilizá-lo. Ainda na figura, a letra B mostra o estado atual de um papel no sistema. Como é possível ver na figura, o sistema atribui “1” para papéis ativos e “0” para inativos.

Figura 25: Como Ativar/Desativar um recurso no modelo *EasyConn4All*.



Fonte: do próprio autor.

Analogamente, na direita da Figura 25 são mostrados os mesmos recursos e finalidades, só que agora com o módulo de Entidades.

O processo de autenticação e validação das atividades que estão disponíveis a uma determinada entidade segue a seguinte sequência de passos:

1. a entidade, inserida em um determinado contexto já registrado, solicita *login* no mesmo, através de um dispositivo computacional (computador, *notebook*, *smartphone* ou *tablet*) que possua o *EasyConn4All*;
2. o dispositivo envia até o servidor *EasyConn4AllServer* uma descrição da entidade e em qual contexto está solicitando inserção;
3. ao receber a solicitação, o servidor inicia o processo de validação da entidade, verificando se esta entidade está registrada no sistema;
4. se a verificação der positiva, inicia-se o processo de validação e qualificação do contexto informado, para se certificar que o contexto está ativo no momento (através de informações de localização, tempo e composição de integrantes do contexto);
5. validado o contexto, o próximo passo é verificar o papel da entidade que solicita acesso e buscar todas as permissões de atividades ligadas a este papel;
6. em seguida, o sistema irá fazer uma varredura na trilha deixada pela entidade no contexto. Todas as atividades que a entidade já tenha realizado e que estejam ativas e válidas serão novamente concedidas à entidade, analisando antes as regras (ver seção 4.4) definidas quanto a quais atividades podem ser resgatadas;
7. o modelo busca uma lista de atividades que tenham sido ofertadas à entidade por meio de confiabilidade. Estas atividades, se existirem, serão disponibilizadas momentaneamente e não serão utilizadas na montagem da trilha de atividades previamente executadas;
8. após a liberação das atividades, o modelo irá registrar um histórico de acessos permitidos (a trilha) que ficará armazenado na base de consulta do *EasyConn4AllServer* para futura liberação de serviços;
9. feita a coleta de informações referentes a quais atividades são permissíveis à entidade em questão, o modelo envia uma mensagem ao *EasyConn4AllClient*, que

solicitou o controle de acesso, informando quais atividades serão disponibilizadas para a entidade em questão.

De um modo geral, este é o processo que o sistema realiza para controlar o acesso e conceder a permissão de uso de atividades.

Atividades, entidades, contextos ou qualquer informação referente à definição de qualquer recurso integrante do sistema poderá se tornar inativo. Sendo assim, a qualquer momento que uma entidade solicitar o controle de acesso e algum dos recursos analisados estiver indisponível, este não será avaliado e sua permissão para a entidade solicitante não será permitida.

5.4 Teste de Funcionalidade

O processo de teste de funcionalidade do *EasyConn4All* tem o objetivo de gerar dados e informações de usuários do sistema de controle de acesso com o intuito de verificar o comportamento do sistema e medir suas eventuais falhas ou limitações que a situação, o ambiente ou os participantes poderiam impor.

Para fazer este teste foram criados dados de uma turma de estudantes de uma disciplina escolar, e, nesta população foram inseridas entidades com papéis distintos entre si. Algumas das entidades poderiam realizar algum tipo de atividade, como poder ler conteúdos didáticos postados pelo professor, ou ainda poder registrar frequências ou avaliações no sistema, atividades estas, pertinentes apenas ao professor da disciplina.

Neste teste foi criado todo ambiente propício para a sua implementação, como a criação de papéis distintos para as entidades participantes da turma, a criação de um contexto didático com regras para a sua concretização (como a participação de determinadas entidades para qualificar o contexto), atividades distintas para cada um dos papéis do modelo.

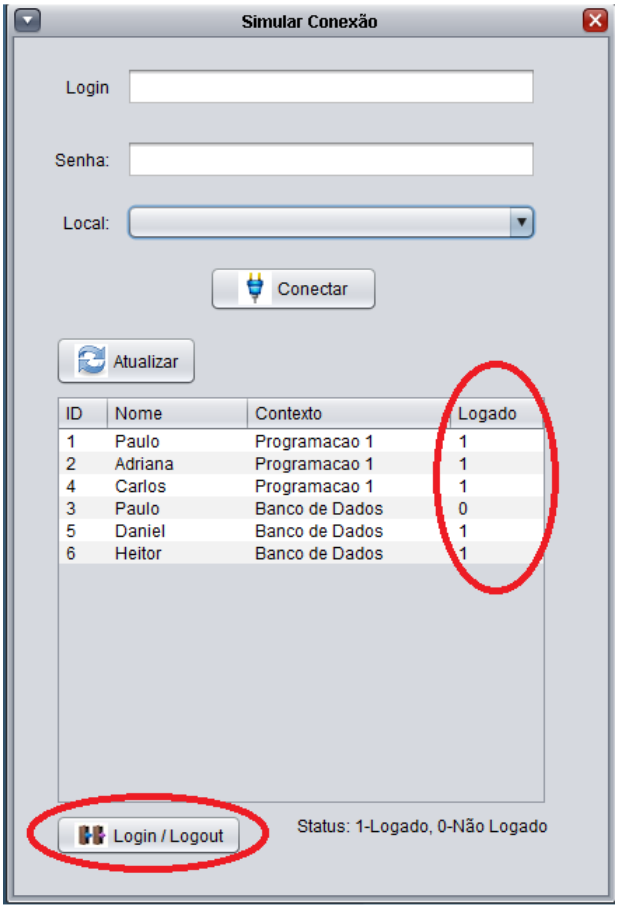
Como primeiro passo para a criação da simulação foram adicionadas ao controlador de acesso as entidades que iriam ser as responsáveis por manipular as atividades do modelo. Para este fim, o modelo solicita apenas que as entidades sejam registradas com um *login* de acesso e uma senha para futura verificação.

Caso as entidades venham a ser objetos que devam estar no contexto, para estas entidades, o modelo prevê uma tela de alteração do status de presença da entidade no

contexto, onde é necessário apenas o utilizador alterar a presença da entidade, para que o modelo perceba esta mudança.

A Figura 26 mostra a tela de simulação de presença de entidades em um contexto.

Figura 26: Simulação de Presença de Entidades em um Contexto.



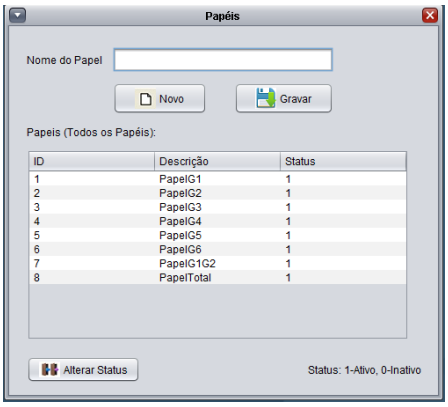
Fonte: do próprio autor.

Depois de adicionadas as entidades ao modelo, o passo seguinte é a definição dos papéis que estas entidades portarão no contexto. Para esta tarefa, é necessário apenas definir a descrição que o papel terá no modelo para posterior uso.

O sistema não utiliza o nome como critério de distinção entre os registros, sendo assim, poderia ser usado como alternativa adicionar junto com o nome do papel uma breve descrição do contexto ao qual este papel estiver sendo usado (como por exemplo, “aluno de programação 1”).

A Figura 27 mostra a adição de papéis no modelo.

Figura 27: Definição de papéis no modelo.



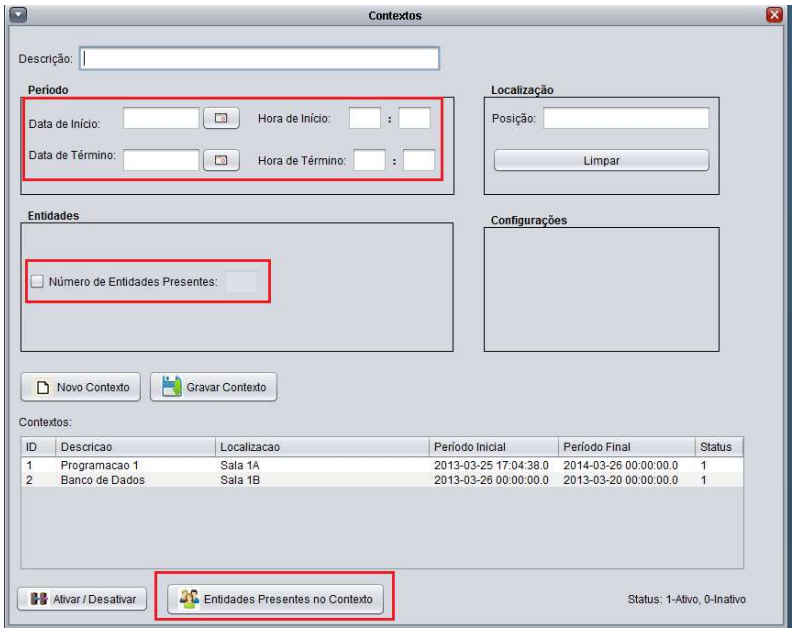
Fonte: do próprio autor.

Criados os papéis das entidades, o próximo passo é a definição do contexto de uso juntamente com as regras que irão permitir a sua aplicação. Nesta definição, é necessário que se estabeleça o local, a hora e data de início, a hora e data de término, o número mínimo de entidades que devem utilizar o contexto e quais entidades devem, obrigatoriamente, estar no contexto para que ele seja qualificado.

Definidos os parâmetros de configuração do contexto, ele poderá ser adicionado ao modelo a imediatamente ser controlado.

A Figura 28 mostra a tela de adição de contextos ao modelo.

Figura 28: Definição dos atributos de um contexto.



Fonte: do próprio autor.

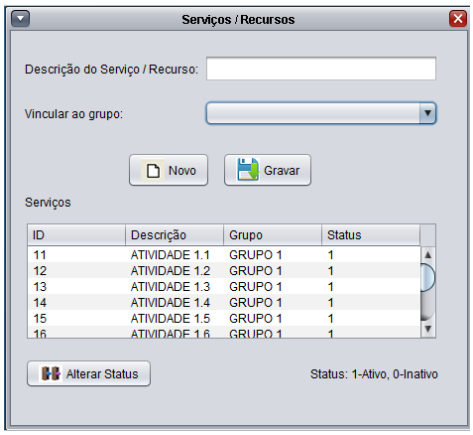
Definido o contexto, o passo seguinte é o de registrar todas as atividades (recursos) que poderão ser utilizadas pelas entidades registradas. Nesta etapa será necessário informar quais atividades terão seu acesso controlado pelo sistema, do mesmo modo que a definição de alguns aspectos técnicos como, por exemplo, se a atividade permite a gravação de novos registros por parte do usuário (*read/write*) ou apenas sua visualização (*read only*).

O sistema permite também que atividades se tornem inativas, não sendo possível seu acesso futuro mesmo que a trilha de acessos possua referências a estas atividades anteriormente. Este recurso torna o sistema flexível, pois permite ao gerenciador de controle de acesso que determinadas atividades não sejam mais utilizadas futuramente, mesmo havendo referência em trilhas.

Este recurso poderá ser desfeito a qualquer momento, permitindo assim que o sistema faça a leitura e a consequente permissão de acesso à atividade.

A Figura 29 mostra a tela de registro de atividades disponíveis para serem utilizadas pelos papéis.

Figura 29: Cadastro de atividades e vinculação a grupos.

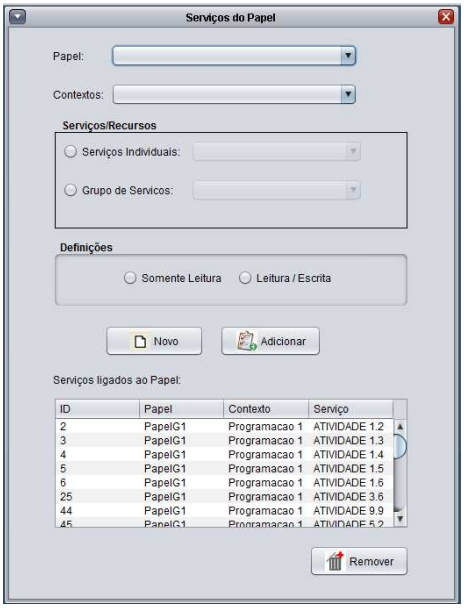


Fonte: Elaborado pelo autor.

Criadas as atividades, no próximo passo é necessário criar as ligações de todos os recursos. Primeiramente, é possível associar quais atividades estarão vinculadas aos papéis do sistema. Esta tarefa se faz necessária para que se possa controlar melhor e de uma maneira mais fácil as várias atividades que serão atribuídas a uma entidade.

A Figura 30 mostra o recurso de vinculação de atividades em um determinado papel, que deve estar presente em um determinado contexto.

Figura 30: Associação de atividades com papéis no sistema.



Fonte: do próprio autor.

O sistema permite que as entidades que serão controladas sejam cadastradas e vinculadas a um contexto. Esta tarefa é necessária, pois as entidades não terão as permissões concedidas para qualquer contexto que elas estejam inseridas. Seguindo o princípio que as entidades terão um papel associadas a si, em um determinado contexto, é necessário informar ao sistema quais entidades estarão ligadas ao contexto.

A Figura 31 mostra a tela de vinculação de entidades e um determinado contexto.

Figura 31: Entidades sendo associadas e um contexto.

Fonte: do próprio autor.

Definidas as atividades que serão utilizadas pelos papéis, no passo seguinte é possível associar quais entidades portarão os papéis e em qual contexto. Esta atividade se faz necessária devido a mesma entidade poder utilizar contextos diferentes com papéis diferentes. Um exemplo desta situação seria uma entidade ser “aluno” em um contexto “X” e ser um “professor” em um contexto “Y”.

A Figura 32 mostra a tela de ligação entre a entidade, o papel e o contexto ao qual estarão vinculados.

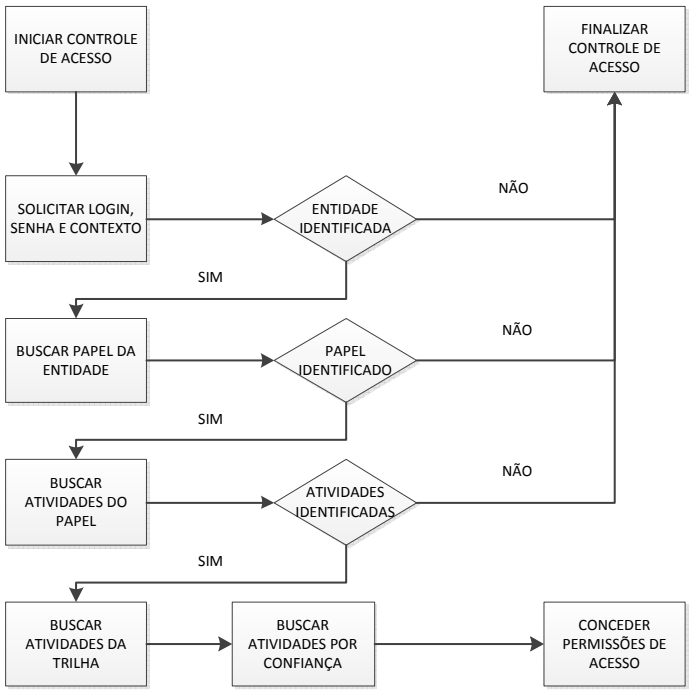
Figura 32**:** Associação Entidade X Papel X Contexto.

Fonte: do próprio autor.

Definidos os parâmetros de uso, é possível empregar o controle de acesso utilizando o software *EasyConn4AllClient*. Com o aplicativo cliente em uso e operando em modo de comunicação com o *EasyConn4AllServer*, o sistema identificará a entidade que está solicitando o acesso em um contexto e após a análise de sua situação atual e da trilha de acessos anteriores, concederá as permissões cabíveis.

A Figura 33 mostra como o sistema realiza o procedimento de controle. Nela, é possível ver o encaminhamento das informações solicitadas e sua destinação para que seja possível controlar o acesso a recursos.

Figura 33: Fluxograma do controle de acesso ao sistema.



Fonte: do próprio autor.

Mediante tal teste é possível destacar as seguintes considerações:

- cada contexto contempla um caso de uso específico, independente da situação em ele esteja inserido;
- as alterações de status de todos os recursos do sistema podem ser alteradas em tempo de execução, tornando o sistema flexível;
- o controle de acesso é executado por rotinas que monitoram o sistema constantemente. Qualquer informação que seja alterada e venha a desqualificar o contexto causará uma ação que provocará o encerramento do contexto atual.
- o servidor do sistema gerencia simultaneamente vários contextos. Com isto um número grande de entidades pode ser administrado.
- a inclusão/inativação de qualquer um dos recursos utilizados na simulação pode ser feito durante a mesma, sem que o sistema valide informações desatualizadas.

6 ASPECTOS DE AVALIAÇÃO

Este capítulo aborda os aspectos de avaliação do modelo, juntamente com os resultados obtidos. Segundo Edwards et al. (2003) a avaliação de uma infraestrutura é considerada problemática, uma vez que a mesma não é visível para o usuário final. Dessa forma, somente é possível avaliar as funcionalidades construindo aplicações que as utilizem e então avaliá-las, obtendo-se assim, uma avaliação indireta do modelo. Além disso, a comunidade científica vem usando em conjunto com os aplicativos, a criação de cenários para a validação de sistemas sensíveis ao contexto (DEY; ABOWD; SALBER, 2001).

Desta forma, foram criados dois cenários de testes para avaliação do sistema. O primeiro cenário foi implementado sob um modelo universitário hipotético, onde é possível verificar a presença de professores e alunos com atividades disponíveis distintas e com controles diferenciados. Este modelo mede a eficiência do controle de acesso, distribuído entre entidades diferentes que estiverem com um mesmo contexto.

O segundo cenário foi desenvolvido utilizando um ambiente hospitalar hipotético, onde também é possível verificar a presença de uma entidade (médico), tendo todas as suas atividades que constavam na trilha sendo resgatadas pelo sistema para novos usos.

Para testar os cenários, foram utilizados um notebook Core i3 com sistema operacional *Windows 7 Ultimate* como servidor de controle de acesso, executando uma instância do aplicativo *EasyConn4AllServer*. O sistema gerenciador de banco de dados MySQL também foi instalado nesta máquina. Para as estações de trabalho (estações clientes) foram utilizados um notebook Core i3 com sistema operacional *Windows 7 Ultimate* com uma instância do sistema *EasyConn4AllClient* e também um Tablet Samsung Galaxy Tab 2 P3110, igualmente executando a versão para *Android* do *EasyConn4AllClient*.

6.1 Avaliação 1 – Ambiente de Ensino

Neste cenário de avaliação foram utilizados recursos pertinentes a um ambiente universitário em que atividades possam ser atribuídas a um professor, como a postagem de documentos didáticos.

Para efetivar este cenário, foi criado no módulo ifrmContexto, um contexto específico para uma disciplina. Neste contexto, foi definido o seu início (data e hora) e também foi definido seu término (data e hora). Foi estabelecida a localização da ocorrência deste contexto (identificação da sala de aula de ocorrência) e também as entidades que teriam algum tipo de serviço.

O seguinte cenário foi modelado: "*Maria é uma aluna da disciplina de Programação 1 do curso de computação da UNISINOS. Ao chegar na sala de aula, Maria consulta seu tablet em busca de materiais que o professor tenha disponibilizado para os alunos. Como Maria é uma aluna matriculada, o software do tablet notifica a ela que existe um trabalho avaliativo para que ela responda e poste em até 24 horas. Maria acessa o trabalho em seu tablet e começa a resolver as questões. Ela verifica também que o professor postou apontamentos para salientar aos alunos pontos específicos de algumas questões do trabalho. Durante a realização das atividades, ela percebe que necessitaria consultar o material sobre virtualização, com o objetivo de obter informações referentes a questão. Ela verifica que em outra disciplina já cursada (Sistemas Operacionais), fora disponibilizado para os alunos este material, e, sabendo disto, ela tenta acessar o conteúdo disponível por aquela disciplina e consegue acesso ao material, inclusive ao material sobre virtualização. Retornando ao seu trabalho, ela resolve as questões que estavam pendentes e posta as respostas para o professor da disciplina, aguardando, agora, o resultado de seu trabalho*". A Tabela 2 resume o cenário proposto, destacando os atores participantes.

O modelo proposto começa sua ação solicitando ao usuário para que se identifique (*login* e senha) e informe em qual contexto deseja se conectar.

O sistema realiza as validações básicas de integridade do usuário e do contexto e em caso positivo, verifica se o usuário possui alguma ligação restritiva com o contexto solicitado.

Depois de verificado o papel que a entidade possui no contexto, o passo seguinte é buscar, seguindo suas propriedades, quais são as atividades que sejam possíveis realizar no contexto.

Em seguida, os componentes passam a verificar a trilha de acessos e verificar quais atividades a entidade já executou anteriormente, independente do papel que ela portava. O modelo recolherá estas atividades e juntará com as atividades ligadas ao papel que já haviam sido agrupadas. Este trabalho tem a influência de parâmetros estabelecidos pelo administrador

no *EasyConn4All*, local em que é possível, por exemplo, definir se as atividades da trilha serão apenas as ligadas ao contexto atual ou a qualquer contexto sem distinção.

Feitas as coletas destas atividades, no passo seguinte o sistema irá verificar se existem atividades concedidas à entidade pelo atributo de confiança. Este quesito não está vinculado à trilha da entidade, por isso o sistema de busca procurará esta informação em uma base separada e ela não será anexada à trilha de atividades executadas pela entidade.

Uma vez que o servidor de controle de acesso tenha feito a coleta das atividades permitidas à entidade em questão, este entra em contato com a estação cliente e envia uma sequência de atividades disponíveis para a entidade no contexto.

A estação cliente ao receber a mensagem, libera as atividades pertinentes à entidade e ela poderá controlar o acesso como foi definido pelas regras.

A Tabela 2 mostra a sequência dos passos que são executados pelos aplicativos *EasyConn4AllServer* e *EasyConn4AllClient* para controlar o acesso deste usuário.

Tabela 2: Dinâmica do controle de acesso no primeiro cenário

Ator	**Ação**
Professor	Executa o *login* de acesso ao sistema, informando sua identificação, senha de acesso e o contexto (sua disciplina)
EasyConn4AllClient	Encapsula a requisição de acesso e a remete ao servidor de controle de acesso para refinamento da solicitação.
EasyConn4AllServer	Através da informação recebida do cliente, realiza o cruzamento de dados e busca as atividades disponíveis para o professor (entidade).
EasyConn4AllClient	Recebe a resposta a sua solicitação por parte do servidor e libera as atividades listadas na resposta ao controle de acesso.
Professor	Posta o material a ser feito pelos alunos da disciplina ministrada por ele.
Maria	Faz *login* no sistema buscando por trabalhos e atividades a serem feitas.
EasyConn4AllClient	Encapsula a requisição de acesso e a remete ao servidor de controle de acesso.
EasyConn4AllServer	Busca as atividades disponíveis para a solicitante.
EasyConn4AllClient	Recebe a solicitação e libera TODAS as atividades que Maria tem disponível, incluindo atividades que constam em sua trilha.
Maria	Com as atividades disponíveis, realiza as tarefas com materiais distintos e os envia ao professor.

Fonte: Elaborado pelo autor.

O sistema mostrou-se pertinente neste cenário, pois foi possível verificar a busca dos componentes pelas atividades concernentes a esta entidade. Os componentes responsáveis foram capazes de buscar todas as permissões relativas ao papel que a entidade portava e as permissões contidas na trilha de acessos.

Também foi possível verificar que este modelo não possui restrições quanto a tipos de aplicações de destino, podendo ser implementado em, praticamente, qualquer aplicação.

As figuras a seguir mostram o processo de controle desde a requisição de acesso por parte da entidade “Maria”, passando pelos passos de coleta de informações a respeito da entidade até verificação de suas permissões de acesso.

Na Figura 34 é apresentado o momento em que a aluna “Maria” faz seu *login* no sistema e informa neste momento sua identificação, sua senha de acesso e a localização (contexto) no qual pretende se conectar.

Figura 34: *Login* no Sistema *EasyConn4AllClient*.

Fonte: Elaborado pelo autor.

No momento da conexão, o sistema irá realizar os passos para a autenticação da entidade solicitante, do contexto e das atividades disponíveis. A Figura 35 mostra que a entidade “Maria” está registrada no modelo e é uma entidade válida.

No momento que for verificado o registro da entidade, o sistema irá buscar informações sobre o contexto informado e sobre as atividades que estão disponíveis a ele.

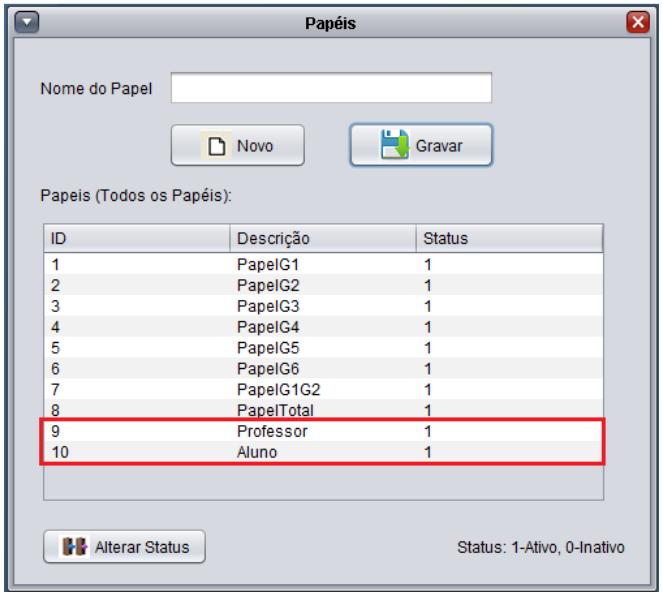
As atividades estarão associadas a uma entidade através de papéis. Sendo assim o modelo verificará qual papel a entidade está portando no contexto em questão.

Figura 35: Registro da Entidade solicitante na base de dados.

Fonte: Elaborado pelo autor.

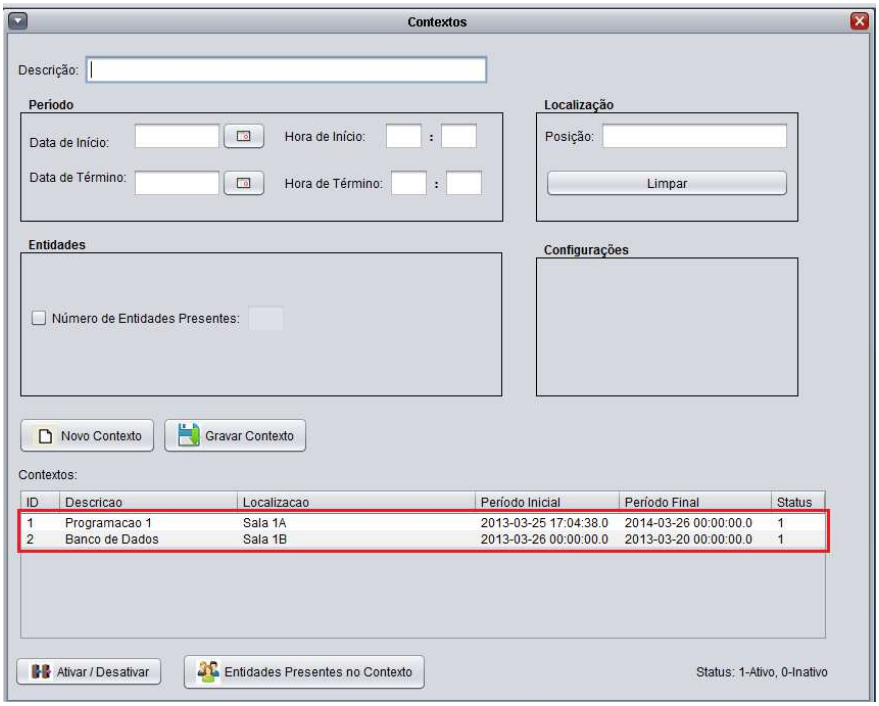
A Figura 36 mostra que existe um papel chamado “Aluno” e outro chamado “Professor”, que poderá ser usado pelas entidades. Na Figura 37 é mostrado que o sistema possui o registro de dois contextos, entre eles o contexto de “Programação 1” solicitado pela entidade.

Figura 36: Registro dos papéis utilizados nos testes.



Fonte: Elaborado pelo autor.

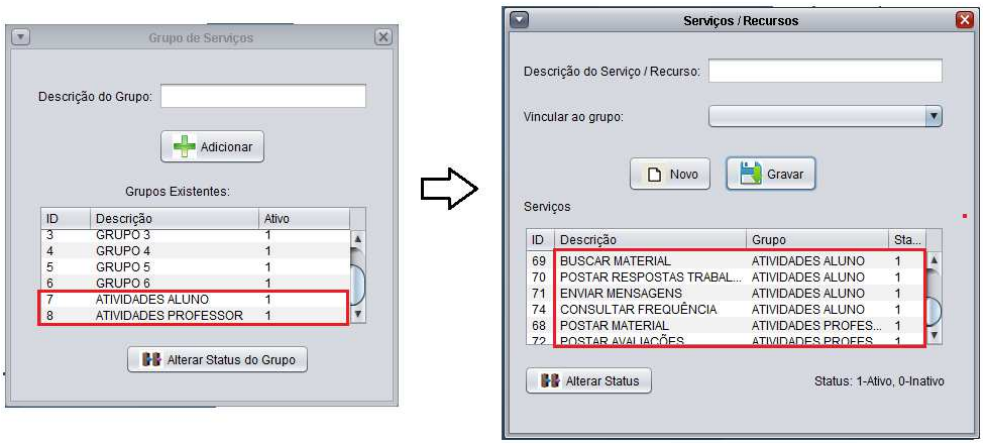
Figura 37: Registro dos contextos utilizados nos testes.



Fonte: Elaborado pelo autor.

Para que o gerenciamento das atividades fique controlado no sistema, foi criado um grupo de atividades chamado “Atividades Aluno” que terão atividades associadas a ele. Assim, para informar ao sistema quais atividades serão utilizadas por um aluno, basta que seja associado a um grupo de serviços todas as atividades que estarão disponíveis para ele. A Figura 38 mostra o grupo “Atividades Aluno” sendo associada a atividades previamente registradas no sistema.

Figura 38: Serviços disponíveis nos teste e seus agrupamentos.

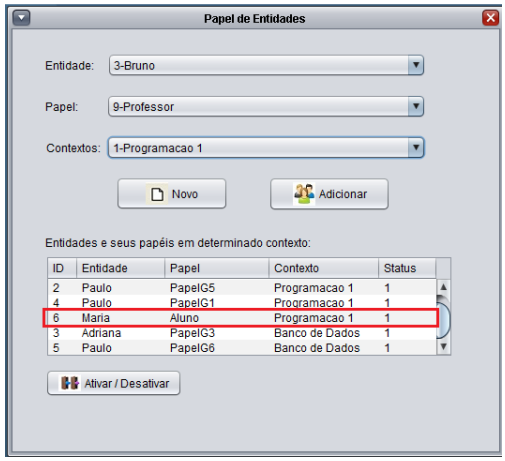


Fonte: Elaborado pelo autor.

Uma vez que sejam identificadas as atividades que estão associadas a um papel, o sistema deve relacionar o papel com o contexto no qual será utilizado. A Figura 39 mostra que o modelo possui a informação de que a entidade “Maria” possui o papel de “Aluno” no contexto “Programação 1”.

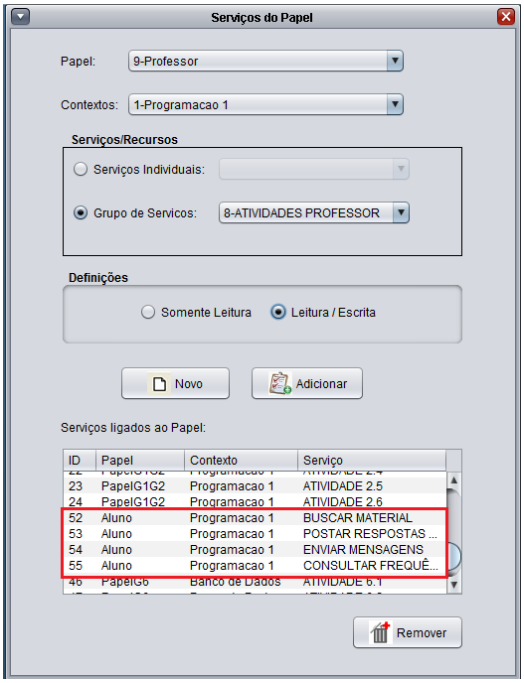
Sabendo desta informação, a Figura 40 mostra que o papel de “Aluno”, que é o papel utilizado por “Maria”, possui algumas atividades associadas a ele, e estas atividades estarão ligadas ao contexto “Programação 1”.

Figura 39: Anexando o papel a uma determinada entidade.



Fonte: Elaborado pelo autor.

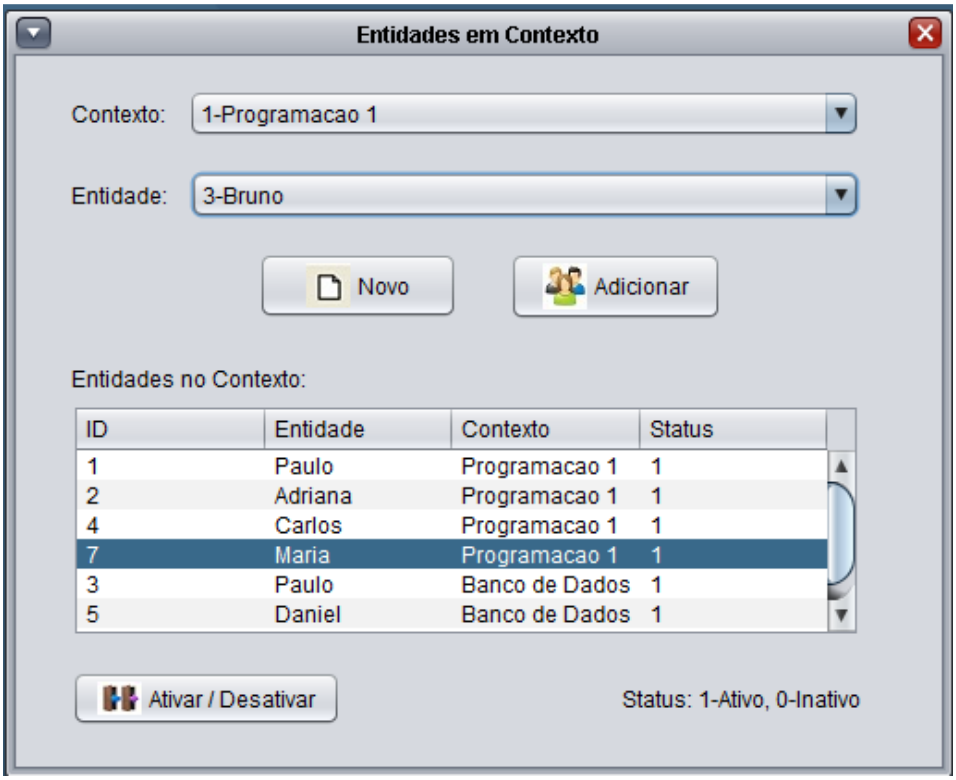
Figura 40: Vinculação dos serviços com seus papéis.



Fonte: Elaborado pelo autor.

Todas as entidades devem estar registradas com sua associação aos contextos dos quais necessitariam receber controle de acesso por parte do sistema. A Figura 41 mostra algumas entidades relacionadas a determinados contextos. Se por ventura alguma entidade necessitar ser desvinculada de um contexto, no próprio sistema, como mostrado na figura, o sistema possui o recurso para tornar a entidade inativa no contexto. O sistema não controlaria o acesso desta entidade.

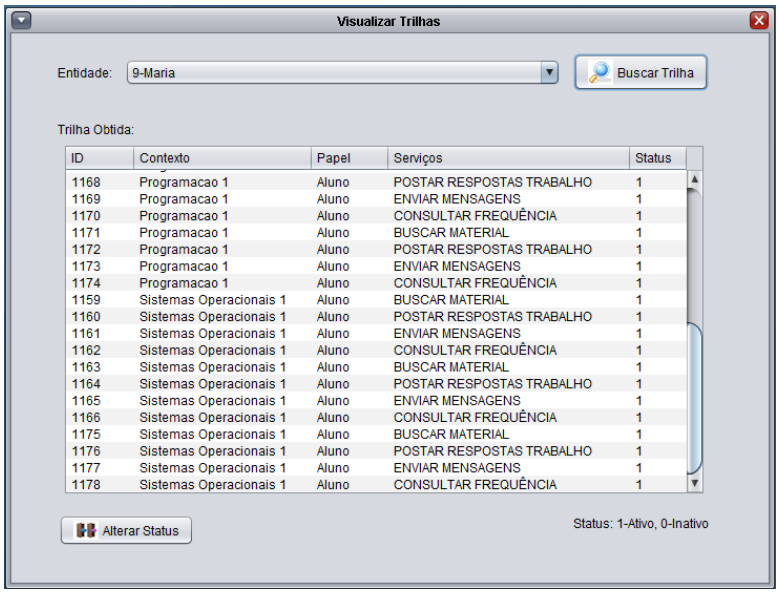
Figura 41: Ligação de Entidades com Contextos.



Fonte: Elaborado pelo autor.

Após verificado que a entidade possui um papel no contexto e este papel possui atividades que ela pode utilizar, o sistema também realiza a leitura de sua trilha de acessos anteriores para validar os novos acessos. A Figura 42 mostra a trilha de atividades que a entidade “Maria” já utilizou em acessos anteriores. Estes acessos, se não possuírem impedimentos, serão novamente autorizados.

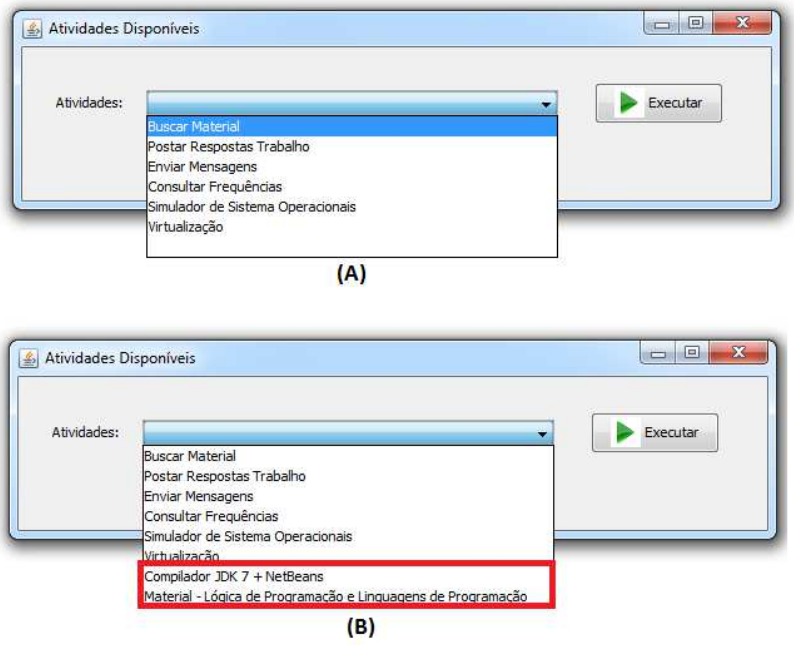
Figura 42: Acessos da entidade sendo registrados em sua trilha.



Fonte: Elaborado pelo autor.

A Figura 43 (a) e (b) mostra os resultados sendo recebidos pelo sistema cliente. Na Figura 43(a) são mostradas as atividades que estavam disponíveis nos primeiros acessos da entidade. Em outros contextos (onde as atividades estejam também disponíveis), é possível verificar na Figura 43(b) que o sistema, com base na leitura da trilha da entidade, habilitou as atividades atuais e também as que estavam disponíveis para ela em outros momentos.

Figura 43: Acesso verificado pelo servidor de controle.



Fonte: do próprio autor.

Com a realização deste cenário foi testado o acesso de uma mesma entidade que possui atividades em contextos diferentes e o sistema foi capaz de recolher todas as atividades com permissão para o uso sem que a entidade informasse ao sistema a localização e a descrição da tarefa desejada. O sistema se valendo da trilha de acessos buscou todas as atividades previamente executadas e as disponibilizou novamente.

6.2 Avaliação 2 – Ambiente Hospitalar

Neste cenário de avaliação foi utilizada uma situação corriqueira na vida de trabalho em um hospital.

Para concretizar este ambiente o seguinte cenário foi modelado: *Alberto é um médico que possui habilitação para o tratamento de doenças oncológicas. Alberto utiliza o sistema EasyConn4All no hospital em que trabalha. Como Alberto é um oncologista, ao fazer o login no sistema é identificado que este usuário possui acesso as ferramentas relativas à oncologia, visto que isto foi verificado ao analisar seu papel e sua trilha de acessos. Alberto realiza uma atividade relativa ao tratamento de um de seus pacientes (contexto) e em seguida sai do sistema. Minutos depois, Alberto volta ao sistema para realizar algumas atividades referentes a outro paciente (outro contexto). O sistema novamente verifica as atividades que Alberto poderá desempenhar no sistema verificando seu papel (médico oncologista) e sua trilha de acessos.*

Alberto se aperfeiçoou e fez cursos para o tratamento cardíaco. No sistema EasyConn4All, foram registradas as novas atividades que o médico pode desempenhar no hospital, pois agora Alberto tem um novo papel (cardiologista). Quando Alberto utilizar o sistema novamente, no momento que ele realizar o login, o sistema identificará que Alberto é um cardiologista e disponibilizará para ele as atividades do novo cargo. Porém, Alberto também é um oncologista, e o sistema verifica isto ao analisar a trilha de acessos. Agora, Alberto possui acesso às atividades relativas à cardiologia (papel) e também as atividades ligadas à oncologia (trilha), podendo trabalhar com ambas onde forem necessárias.

Para este cenário, primeiramente, foi necessário que a entidade envolvida no contexto analisado (hospital), fosse cadastrada no modelo *EasyConn4All*.

Todas as operações que serão envolvidas, neste panorama, devem ser cadastradas no modelo, para posteriormente poderem ser associadas às entidades que as executarão. A Tabela 3 mostra um exemplo de atividades e entidades que poderiam desempenhar.

Tabela 3: Dinâmica do controle de acesso no segundo cenário.

Ator	**Ação**
Médico	Executa o *login* de acesso ao sistema, informando sua identificação, senha de acesso e o contexto.
EasyConn4AllClient	Encapsula a requisição de acesso e a remete ao servidor de controle de acesso para refinamento da solicitação.
EasyConn4AllServer	Através da informação recebida da estação cliente, realiza o cruzamento de dados e busca as atividades disponíveis para o médico (entidade) através da análise de seu papel (oncologista) e de sua trilha.
EasyConn4AllClient	Recebe a resposta a sua solicitação por parte do servidor e libera as atividades listadas na resposta ao controle de acesso.
Médico	Executa atividades ligadas a oncologia no contexto e realiza o *logout* no sistema
Médico	Faz um *login* no sistema para executar algumas tarefas referentes a um paciente, informando sua identificação e o contexto (que envolve o paciente). Este *login* é feito após o registro de suas novas aptidões – cardiologista.
EasyConn4AllClient	Envia a solicitação de controle de acesso ao servidor de controle.
EasyConn4AllServer	Identifica a entidade, o contexto e envia para a estação cliente as atividades permitidas. A análise foi feita sobre o papel (cardiologista) e sobre a trilha (oncologista) de acessos da entidade.
EasyConn4AllClient	Recebe a identificação das atividades disponíveis para a entidade e libera as atividades (atividades da cardiologia e oncologia)
Médico	Consegue ter acesso a todas as atividades pertinentes a sua ocupação, seja ela do ramo da oncologia como do ramo da cardiologia.

Fonte: do próprio autor.

Neste cenário, o médico possui permissão para executar atividades que estejam disponíveis em contextos específicos (área da oncologia). Ao ser registrado no sistema a alteração da especialidade do médico, novas atividades serão disponibilizadas para que possa realizar sua nova atividade. Como o médico possui conhecimentos em outro ramo, ao analisar

a trilha de acessos é verificado que existem outras atividades que podem ser utilizadas, sendo também concedidas ao médico pelo sistema.

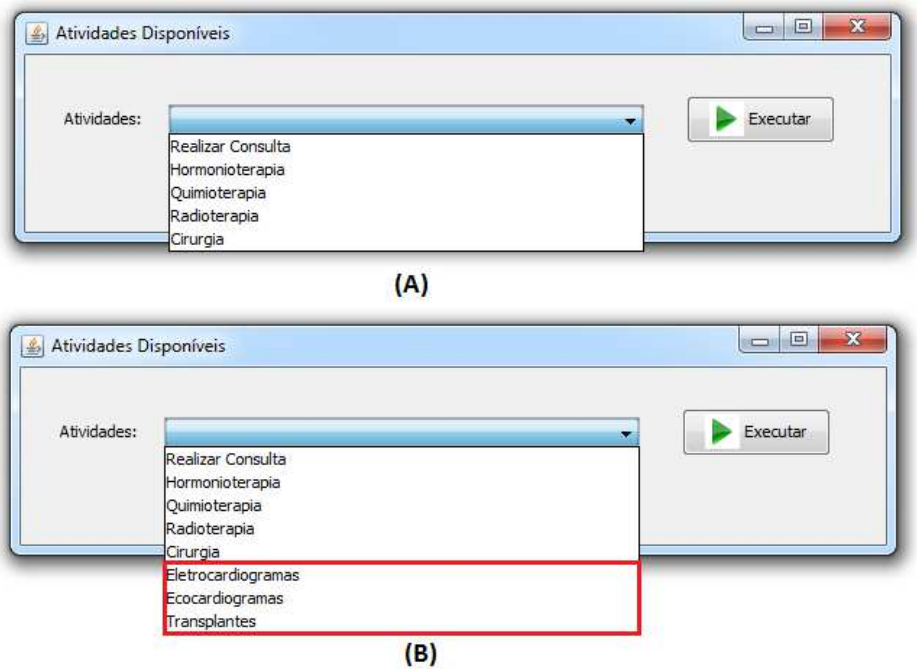
Caso o médico possua em sua trilha de acessos o registro de alguma atividade que por algum motivo (ser um caso específico de outro profissional, por exemplo) não possa ser disponibilizada novamente, o sistema possui recursos que tornam a atividade inacessível para a entidade, mesmo que conste em sua trilha o acesso.

Todas as atividades que serão realizadas, neste cenário, necessárias ao tratamento da saúde do paciente, serão anexadas à trilha da entidade, como atividades previamente realizadas. Se por ventura outro paciente for internado com os sintomas semelhantes ao paciente deste cenário, o médico terá condições de auxiliar no tratamento deste, pois anteriormente está registrado que tarefas similares já foram realizadas pelo médico em questão.

O processo de autenticação e busca das informações relativas a entidade/contexto/atividades são similares as Figuras 34 a 42 do cenário anterior, adaptadas agora para este cenário.

As Figuras 44(a) e 44(b) mostram a entidade médico, utilizando atividades anteriormente e após, sendo reaproveitadas na nova situação de trabalho.

Figura 44: Aplicação prática do *EasyConn4All*.



Fonte: do próprio autor.

Nas Figuras 44(a) e 44(b) é possível verificar o médico tendo os seus trabalhos ligados à oncologia habilitados para o seu uso. Depois de feitos cursos de especialização e aperfeiçoamento, o médico se especializa em doenças cardíacas (e atividades relacionadas), mas ele ainda consegue trabalhar com procedimentos oncológicos. Sendo assim, o *EasyConn4All* disponibiliza a ele novamente as atividades que ele já havia realizado anteriormente por considerar que a entidade tem condições de manipular estas atividades.

Assim como no cenário anterior, conclui-se que o sistema, baseado na trilha de acessos da entidade, resgatou todas as atividades que o usuário possuía acesso e as disponibilizou novamente.

É importante salientar neste ponto, a facilidade que o modelo apresenta ao usuário em resgatar suas atividades predecessoras e anexá-las juntamente com as atividades atuais, sem qualquer necessidade de solicitação por parte do usuário. Partindo do princípio que "*atividade já realizada é atividade conhecida*," o sistema se encarrega de atribuir estas atividades à entidade.

7 CONSIDERAÇÕES FINAIS

Este capítulo aborda as considerações finais da dissertação. Nele serão apresentadas as conclusões, como também as principais contribuições e os trabalhos futuros.

7.1 Conclusões

Inicialmente este trabalho abordou no capítulo um os aspectos elencados com a decisão de elaboração deste projeto, salientando os aspectos de motivação, problemática, questões de pesquisa e objetivos para o desenvolvimento do sistema.

No capítulo dois foi abordada a conceituação de alguns temas relacionados com o trabalho. Os aspectos abordados foram o controle de acesso, a sensibilidade ao contexto e o armazenamento e gerenciamento de trilhas. No capítulo três foram elencados alguns trabalhos relacionados ao tema de dissertação, os quais contribuíram no desenvolvimento deste estudo.

No capítulo quatro foi apresentado o modelo proposto, fazendo um apanhado sobre os módulos de controle e componentes utilizados. Já no capítulo cinco foram feitas as explanações técnicas sobre a implementação do *EasyConn4All*.

No sexto capítulo, foram demonstrados dois cenários de uso prático. Nele, foram salientados pontos de destaque que tornam o modelo conveniente para o uso em situações similares às apresentadas.

Vistos os cenários propostos e verificados alguns pontos do modelo, as principais conclusões alcançadas foram:

- o modelo não possui restrições quanto a sua forma de trabalho, sendo possível sua operação nos mais diferentes ramos de negócio;
- o modelo controla apenas as atividades registradas. Nenhuma atividade externa ao modelo ou não registrada possuirá regras para controlar seu uso;
- o modelo realiza o controle de acesso se valendo de papéis e trilha de acesso;
- as trilhas são atualizadas sempre que uma entidade realiza o acesso ao sistema.
- um acesso a uma atividade do modelo que não mais estiver ativa será ignorado pelos componentes e não será permitido seu uso;

- a criação de testes a respeito do controle de acesso permitiu verificar que o acesso a atividades realizadas anteriormente (registradas nas trilhas) é possível ser utilizado como um acréscimo à forma tradicional de controle de acesso;
- o primeiro cenário criado apresentou o uso de recursos educacionais em um ambiente escolar e seu uso, mostrou-se bastante útil no filtro de atividades disponíveis ao usuário;
- o segundo cenário apresentado mostrou que um especialista em mais de uma área da saúde pode ter todas as atividades relacionadas a seu campo de trabalho novamente permitidas sem a verificação de acessos extras.

7.2 Contribuições

A principal contribuição deste trabalho é a criação de uma alternativa ao critério de controle de acesso baseado em papéis ao fazer o uso da trilha de acessos anteriores a uma determinada atividade, como critério para estabelecer a permissão de acesso.

O modelo também introduz a aplicação de confiança para delegar permissões entre entidades. Se uma entidade confia em outra e gostaria que ela também pudesse executar uma atividade que ambas possuem qualificação e competência para realizá-la, a delegação de confiança no modelo permitirá que controle de acesso seja permitido, com base neste critério.

A Tabela 4 é similar a Tabela 1 do capítulo 3, sendo acrescentada uma coluna que permite a comparação dos trabalhos relacionados com o modelo *EasyConn4All*. A seguir é descrito como o *EasyConn4All* aborda cada um dos quesitos de comparação:

- **sensível ao contexto**: O modelo *EasyConn4All* possui cadastramento e identificação de contextos para o uso pelas entidades.
- **utiliza trilhas**: Um dos critérios de controle de acesso do modelo faz o uso do histórico de acessos às atividades permitidas em acessos anteriores.
- **mobilidade**: O modelo foi projetado para ser executado em dispositivos fixos e móveis, permitindo assim, que a entidade esteja em constante movimento e ainda assim o sistema valide sua colocação no contexto;

- **contextos dinâmicos**: Cada entidade poderá pertencer (estar cadastrada) em vários contextos. O modelo permite o gerenciamento de vários contextos simultaneamente com o tratamento de suas características em tempo de execução;
- **domínio**: O sistema foi projetado sem a limitação de domínios específicos, podendo ser utilizado desde um ambiente hospitalar até um ambiente escolar.
- **confiança**: Além da trilha, outro critério utilizado no controle de acesso e a troca de confiança entre as entidades.

Tabela 4: Comparação entre os trabalhos relacionados e o *EasyConn4All*

Trabalho	Sensível ao Contexto	Baseado em Trilhas	Mobilidade	Contextos Dinâmicos	Domínio	Confiança
Infraware	Sim	Parcial	Não	Sim	Genérico	Não
UbiCOSM	Sim	Não	Sim	Sim	Genérico	Sim
AWARENESS	Sim	Não	Sim	Sim	Médico	Não
SOCAM	Sim	Não	Sim	Sim	Genérico	Não
EasyConn4All	**Sim**	**Sim**	**Sim**	**Sim**	**Genérico**	**Sim**

7.3 Trabalhos Futuros

O modelo *EasyConn4All* é uma proposta para a diversificação da forma atual de controle de acesso. Com a codificação e testes do protótipo, foram identificados alguns pontos que poderiam receber melhorias.

A seguir serão listados alguns dos pontos verificados com maior relevância para alterações futuras:

- a inclusão de novos recursos, como a detecção automática de recursos e atividades disponíveis no contexto, para tornar o modelo flexível quanto à identificação e qualificação de atividades disponíveis.
- tornar o sistema acessível por navegadores de internet bastante comuns no mercado, deixando assim o sistema ainda mais acessível;
- realizar novos testes de funcionalidade com cenários diversificados e com planos de uso distintos, para verificar a eficiência do modelo nos mais diferentes campos de atuação;

- realizar testes com um número maior de cenários sendo gerenciados simultaneamente pelo gerenciador e cada cenário teria um número maior de contextos, entidades, etc.
- implantar o modelo para uso experimental em uma rede de ensino (pública ou privada) e verificar seu modo de ação, seus pontos fortes e fracos;
- incluir recursos que tornem a identificação da entidade no modelo uma tarefa segura, sem chances de fraude. Talvez a inclusão de certificação digital auxiliasse, neste sentido;

8 REFERÊNCIAS

ANDERSON, R.; Security Engineering - a guide to building dependable distributed systems. Wiley Computer Publishing, 2001.

BALASUBRAMANIAN, M., BHATNAGAR, A., CHATURVEDI, N., CHOWDHURY, A. D., GANESH, A.: A framework for decentralized access control, ASIACCS '07 Proceedings of the 2nd ACM symposium on Information, computer and communications security, p. 93-104, 2007.

CAMBRUZZI, W. L. ; RIGO, S. J. ; BARBOSA, J. L. V. . A Proposal for Managing Multiple Trails in Educational Environments. In: XVIII Brazilian Symposium on Multimedia and the Web (WebMedia), 2012, São Paulo. Proceedings of the WebMedia 2012. New York. p. 25-28, 2012.

CAO J, WANG J, LAW K, ZHANG S, LI M. An interactive service customization model. Journal of Information and Software Technology.; v.48, n.4, p.280–296. 2006.

CORRADI A., MONTANARI R., TIBALDI D. Context-based access control for ubiquitous service provisioning. COMPSAC. IEEE, Los Alamitos, v.1, p 444–451, 2004.

COSTA, C. A.; YAMIN, A. C.; GEYER, C. F. R. Toward a General Software Infrastructure for Ubiquitous Computing. IEEE Pervasive Computing, [S.l.], v.7, p.64–73, 2008.

COSTA, P. D., Towards a Services Platform for Context-Aware Applications, Tese de Mestrado, University of Twente, Enschede, The Netherlands. 2003

DEY, A. Understanding and Using Context. Personal and Ubiquitous Computing, [S.l.], v. 6, n. 1, p. 4–7, 2001.

DEY, A.; ABOWD, G.; SALBER, D. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. [S.l.]: Human-Computer Interaction, v. 16, n.2, p.97–166. 2001.

DRIVER, C.; CLARKE, S. Hermes: a software framework for mobile, context-aware trails. In: Workshop On Computer Support For Human Tasks And Activities At Pervasive 2004, 2004. Anais. . . [S.l.: s.n.], 2004.

EDWARDS, W. K.; BELLOTTI, V.; DEY, A. K.; NEWMAN, M. W. The challenges of user-centered design and evaluation for infrastructure. In: Sigchi Conference On Human Factors In Computing Systems, New York, NY, USA. p. 297–304, 2003.

FRAINER, G., DA, L., GEYER, C., AUGUSTIN, I., YAMIN, A.: Mecanismos Adaptativos Para o Acesso a Arquivos em Ambientes Pervasivos. Dissertação de Mestrado. Instituto de Informática, Universidade Federal do Rio Grande do Sul (UFRGS), 2006.

GIANG P. D., HUNG, L. X., LEE S., LEE, Y. K., LEE, H.: A Flexible Trust-Based Access Control Mechanism for Security and Privacy Enhancement in Ubiquitous Systems. p. 698-703, 2007.

GU, T., WANG, X.H., PUNG, H.K., ZHANG, D.Q. An Ontology-based Context Model in Intelligent Environments. [S.l.]: In: Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference, p.270-275, 2004.

GU, T., et al.: A Middleware for Building Context-Aware Mobile Services, In Proceedings of IEEE Vehicular Technology Conference (VTC-Spring 2004), Milan, Italy, v.5, p. 2656 - 2660. 2004.

HECKMANN, D. Ubiquitous User Modeling. 2005. Tese (Doutorado em Ciência da Computação) — Department of Computer Science, Saarland University, Germany, 2005.

HISAZUMI, K., et al.: CAMPUS: A Context-Aware Middleware, The 2nd CREST Workshop on Advanced Computing and Communicating Techniques for Wearable Information Playing, Nara Institute of Science Technology, Nara, Japan, 2003.

HULSEBOSCH, R. J., SALDEN, A. H., BARGH, M. S., EBBEN, P. W. G., REITSMA, J. Context sensitive access control. In SACMAT '05. New York, NY, USA. ACM Press: p.111-119, 2005.

JAJODIA, S., SAMARATI, P., SAPINO, M. L., SUBRAHMANIAN, V. S. Flexible support for multiple access control policies. ACM Trans. Database Syst., v.26(2), p.214-260, 2001.

LI, L.; CAO, T. Context-Role Based Access Control Model for Ubiquitous Computing Enviroment, Asian Journal of Information Technology, v.7, p. 74-78, 2008.

MIRANDA, D. C.; VALENTE, M. T. O.. Um Middleware para Desenvolvimento de Aplicações Sensíveis ao Contexto. REIC. Revista eletrônica de iniciação científica, Soc. Brasileira de Computação, v. 3, p. 1-10, 2006.

MYSQL. Mysql Enterprise Edition. http://www.mysql.com/products/enterprise/. Acessado em 20/9/2012.

NABHEN, R. C. RBPIM: Um modelo de políticas de segurança baseado em papéis – Tese de Mestrado, Pontifícia Universidade Católica do Paraná, 2003.

NETBEANS. NetBeans Docs & Suport. https://netbeans.org/kb/index.html. Acessado em 10/08/2012.

ORACLE. Oracle Technology Network for Java Developers. http://www.oracle.com/technetwork/java/index.html. Acessado em 10/08/2012.

PEREIRA FILHO, J. G.; PESSOA, R. M.; CALVI, C. Z; OLIVEIRA, N. Q.; ET AL. Infraware: um Middleware de Suporte a Aplicações Móveis Sensíveis ao Contexto. [S.l.]: In: SBRC – 24º Simpósio Brasileiro de Redes de Computadores, (SBRC 2006). Curitiba-PR, 2006.

PESSOA, R. M. ; CALVI, C. Z. ; PEREIRA FILHO, J. G. ; ANDREAO, R. V. . Aplicação de um Middleware Sensível ao Contexto em um Sistema de Telemonitoramento de Pacientes Cardíacos. In: SEMISH - Seminário Integrado e Software e Hardware, 2006, Campo Grande/MS. SBC 2006, v. 1, p. 32-46, 2006.

SANDHU , R. S., COYNE, E. J., FEINSTEIN , H. L., YOUMAN , C. E. Role-based access control models. IEEE Computer, v.29, n.2, p.38–47, 1996.

SATYANARAYANAN, M. Pervasive Computing: vision and challenges. IEEE Personal Communications, [S.l.], v. 8, p. 10–17, 2001.

SATYANARAYANAN, M. Mobile computing: the next decade, Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services, San Francisco, CA, July 2010.

SILVA, J. M., ROSA, J. H., BARBOSA, J. L. V., BARBOSA, D. N. F., PALAZZO, L. A. M., "Distribution of Content in Trail-Aware Environments," WebMedia - Brazilian Symposium on Multimedia Systems and Web, 2009.

SILVA, J. M.; ROSA, J. H.; BARBOSA, J. L. V.; BARBOSA, D. N. F. ; PALAZZO, L. A. M. Content Distribution in Trail-aware Environments. Journal of the Brazilian Computer Society (Impresso), v. 16, p. 163-176, 2010.

SINDEREN, M. e. a. Overall architecture of the AWARENESS infrastructure. Disponível em: http://www.freeband.nl/FreebandKC/documents?keyword_id=2432. Acesso em 12/08/2012.

WEGDAM, M. AWARENESS: a project on Context-AWARE NEtworks and ServiceS. [S.l.]: Proceedings of the 14th Mobile & Wireless Communications Summit 2005, p. 19-23, 2005.

WEISER, M. The Computer for the 21st Century. Scientific America, [S.l.], p. 94–104, 1991.

WEISER, M.; BROWN, J. S. Designing Calm Technology. Xerox PARC, 1995. Disponível em http://www.ubiq.com/hypertext/weiser/calmtech/calmtech.htm.

YAMADA, S., KAMIOKA, E.: Access Control for Security and Privacy in Ubiquitous Computing Environments. IEICE Trans on Comm. v. E88-B, n.3, p. 846-856, 2005.