# Controle de Acesso à Informação com base em Expressões Contextuais

Junior Marcos Bandeira
*Unicruz - Universidade Cruz Alta*
junior.bandeira@gmail.com

Raul Ceretta Nunes
*UFSM – Universidade Federal de Santa Maria*
ceretta@inf.ufsm.br

Maria A. Figueiredo
*UFSM – Universidade Federal de Santa Maria*
mariaangelicafo@gmail.com

RESUMO

*O controle de acesso à informação baseado em expressões contextuais pretende aperfeiçoar modelos de controle de acesso fundamentados em perfis, por considerar também variáveis ambientais que descrevem o contexto de uma requisição de acesso. Este artigo propõe um mecanismo de controle de acesso com base em expressões contextuais, o que melhor reflete a estrutura organizacional e a flexibilidade no acesso aos dados. A principal contribuição da pesquisa é a inserção de expressões contextuais na construção de regras de acesso. Como resultado, obtém-se melhor flexibilidade na manipulação de políticas de segurança.*

Palavras-chave: Segurança.Controle de acesso. Contexto.

## 1. INTRODUÇÃO

A área de Tecnologia da Informação (TI) atualmente passa por dois desafios importantes em relação ao aumento da competitividade e adequação das estratégias de negócios: heterogeneidade e mudança. A heterogeneidade se refere ao fato de muitas empresas possuírem diversos sistemas, aplicações a arquiteturas de diferentes épocas e tecnologias. As mudanças acontecem em função da alta competitividade no mercado e, conseqüentemente, alteram os requisitos dos clientes, principalmente em relação à interação com bancos de dados e sistemas de arquivos, tendo em vista que o acesso pode se dar de qualquer lugar devido às tecnologias de computação móvel. A heterogeneidade sugere o uso de arquitetura fracamente acoplada nas ferramentas de apoio à segurança da informação, como por exemplo, o uso de serviços de autorização, enquanto mudanças apontam para o uso de informações contextuais na composição das credenciais para controle de acesso.

A nova realidade necessita de uma complementação dos modelos de controle de acesso tradicionais, como o *Roled Basic Access Control* (RBAC) que prevê um mecanismo de acesso baseado em perfis (FERRAIOLO, 2001). Nesse modelo, cada usuário do sistema é associado a uma ou mais funções (perfis) que indicam as suas responsabilidades dentro da organização. Com isso, é possível especificar regras de acesso de acordo com cada perfil e, ao mesmo tempo, um mapeamento da hierarquia organizacional. Porém, não se consegue levar em conta informações que indiquem quais são as características do ambiente em que está acontecendo a interação, tais como onde se está, quem está presente e quais recursos estão próximos.

Sendo assim, este artigo mostra um mecanismo de Controle de Acesso baseado em Expressões Contextuais, um aperfeiçoamento do modelo de controle de acesso com base em perfis RBAC, que introduz as informações do contexto de onde ocorre a interação com o banco de dados para melhor mapear as políticas de controle de acesso.

O trabalho está organizado da seguinte forma: a seção 2 define o modelo RBAC e algumas de suas principais extensões. A seção 3 define o termo Expressões Contextuais e todos os elementos utilizados na sua representação. A seção 4 especifica o mecanismo de controle de acesso baseado em Expressões Contextuais. Por fim, a seção 5 conclui o trabalho.

## 2. MODELO RBAC

O modelo de controle de acesso baseado em perfis RBAC foi proposto por (FERRAIOLO, 2001) como um modelo de controle de acesso genérico. No RBAC perfis são funções dentro de uma organização. Autorizações então são dadas a funções, ao invés de serem atribuídas a usuários em particular. As Autorizações garantem que os perfis estão estritamente relacionados aos objetos de dados e recursos necessários para exercício de suas funcionalidades. Usuários simplesmente herdam as autorizações que o perfil possui.

Como perfis representam funções organizacionais, um modelo baseado em perfis pode suportar diretamente as políticas de segurança da organização. A administração das autorizações também fica mais simplificada, pois se um usuário troca de função dentro da organização não é necessário revogar suas autorizações, simplesmente ele será associado ao novo perfil e terá todas as autorizações pertinentes à nova função.

O modelo RBAC constitui-se basicamente de quatro componentes: um conjunto de u*suários,* um conjunto de p*erfis,* um conjunto de p*ermissões* e um conjunto de s*essões.* Um usuário é um humano ou um agente autônomo, um perfil é uma coleção de permissões necessárias para exercer determinado trabalho dentro da organização, a permissão é um modo de acesso que pode ser usufruído sobre determinado objeto ou recurso no sistema e a sessão relata o fato de que um usuário pode ter vários perfis. Quando o usuário acessa o sistema ele estabelece uma sessão; durante esse período ele pode solicitar a ativação de alguns perfis que está autorizado a exercer. Se a solicitação for atendida, o usuário passa a contar com todas as permissões que aquele perfil é associado. Em um conjunto de *usuários, perfis, permissões* e s*essões* várias funções são definidas. As diretivas *User Assignment (UA)* e *Permission Assignment (PA)* são funções do modelo que associam usuários a perfis e perfis a permissões respectivamente. Um usuário pode ser autorizado a exercer vários perfis e muitos usuários podem ser autorizados a exercer o mesmo perfil. Além disso, um perfil pode ter várias permissões e uma permissão pode ser atribuída a vários perfis. Os perfis são hierarquicamente definidos, denotados por ≥. Se $r_i \geq r_j$ e $r_{i,} r_j \in$ Perfis então o perfil $r_i$ herdou as permissões de $r_j$.

A definição do modelo RBAC consiste nos seguintes componentes:

- *Users, Roles, Permissions* e *Sessions* representam usuários, perfis, permissões e sessões respectivamente;
- *PA – Perfis -> Permissões* as permissões associadas à função que são necessárias para o exercício do trabalho;
- *UA – Users -> Roles* Usuários associados a perfis que indicam quais perfis determinado usuário possui;
- *User: sessions->users* associa cada sessão para um único usuário;
- *Role: sessions->*$2^{roles}$ associa cada sessão a diferentes perfis.

O controle de acesso nos sistemas computacionais normalmente segue as seguintes diretrizes (SANDHU, 1994): Modelo Discricionário (DAC); Modelo Obrigatório (MAC); ou Modelo Baseado em Perfis (RBAC). Além destes modelos de controle de acesso, que podem ser complementares entre si, também já foram propostos outros modelos relacionados especificamente à utilização de contextos, tais como o E-RBAC (COVINGTON *et al*., 2003), o CS-RBAC (KUMAR *et al*. 2002) e o GTRBAC (JOSHI *et al*., 2003).

O modelo E-RBAC (*Environment Role-Based Access Control*) foi desenvolvido para ser aplicado em uma arquitetura de sistema para "residências inteligentes". O E-RBAC estende o RBAC básico com a introdução de um perfil que não possui usuários associados, denominado perfil ambiental. Este perfil contém autorizações e uma regra que estabelece as condições de ativação/desativação automática do perfil, com base em um conjunto de informações ambientais (por exemplo: horário, temperatura, nível de ruído, localização do

usuário, etc). Estas informações são capturadas por sensores distribuídos pela residência. O acesso para a execução de uma operação é concedido a um usuário quando este possui um perfil tradicional e todos os perfis do conjunto de perfis ambientais estão ativos no momento da requisição de autorização. As regras contextuais são suportadas pelo modelo na definição das condições de ativação/desativação dos perfis ambientais.

O CS-RBAC (*Context-Sensitive Role-Based Access Control*) (KUMAR *et al*. 2002) estende o RBAC básico através da introdução dos conceitos de contextos de perfis e de filtros contextuais. O modelo foi especificado sem hierarquias de perfis e sem restrições de separação de responsabilidades e utiliza apenas dois contextos (de usuário e de objeto), sendo que eles estão fixos na linguagem usada para expressar os filtros contextuais.

O GTRBAC (*Generalized Temporal Role-Based Access Control*) (JOSHI et al., 2003) generaliza o modelo TRBAC (*Temporal Role-Based Access Control*) (BERTINO et al., 2001), que estende o RBAC básico com a introdução de uma linguagem que especifica várias restrições temporais em perfis.

## 3. EXPRESSÕES CONTEXTUAIS

De acordo com Dey e Abowd (1999), contexto é qualquer informação relevante que possa ser utilizada para caracterizar a situação de uma entidade. Uma entidade pode ser uma pessoa, um lugar ou um objeto, relevantes para a interação entre o usuário e a aplicação. Uma aplicação que utiliza **expressões contextuais** é uma "aplicação ciente de contexto".

O primeiro trabalho a utilizar o termo "ciente de contexto" foi o de Schilit e Theimer (1994), os quais se referiam a contexto como localização, identidades de pessoas e objetos e mudanças desses objetos (DEY, 1999). Outras abordagens definem contexto como o ambiente ou situação em que uma determinada interação ocorre. Tanto a definição de Shilit (1994), em que os principais aspectos do contexto são onde você está, quem está com você, e quais recursos estão próximos - quanto à de Pascoe (1998), para o qual contexto é o subconjunto de estados físicos e conceituais de interesse de uma entidade particular - são muito específicas e indicam que as expressões contextuais definem uma situação relevante à aplicação e seu conjunto de usuários.

O acesso baseado em contexto fundamenta-se no princípio da construção de regras que usam expressões contextuais para mapear as políticas de segurança de uma dada organização. Por exemplo, em uma instituição de saúde onde uma autorização de acesso deve considerar regras ambientais (pacientes internados, local do acesso, etc) associado a operações (visualização de dados, prescrição de laudos, etc), bem como requisitos temporais (período de plantão, tempo internação, etc), para cada domínio pode ser considerada uma expressão contextual determinante para a autorização do acesso.

É importante ressaltar que a existência de regras de acesso definem políticas de segurança e determinam quais expressões contextuais são ou não necessárias para a elaboração de uma condição de contexto. Uma condição de contexto pode ser composta por uma ou mais expressões contextuais, que devem ser verificadas para permitir a concessão da autorização de acesso, conforme observado na figura 1.

Como se pode verificar, as expressões contextuais são formadas por propriedades do contexto que possuem determinados nomes, operadores e valores:

***Propriedade: P(N,V,O)***

- *N =Nome*
- *O = Operador*
- *V = Valor*

As propriedades simplesmente indicam o nome da característica pertinente ao ambiente e o valor assumido por ela naquele momento; no exemplo da figura 1 uma das propriedades chama-se *local*, o operador é = e o valor é *UTI.*

Cada propriedade é associada a um tipo de contexto específico. O tipo de contexto é o elemento através do qual se está tentando interagir com o sistema. No exemplo há dois elementos - u*suário* e o*bjeto.* Portanto, as propriedades terão de fazer parte de um deles; sendo assim, pode-se dizer que um tipo de contexto contém propriedades. Pode-se defini-lo da seguinte forma:

- ***Tipo de Contexto: CT***
  - *CT = Sujeito ou Objeto e CT (P1,...,PN).*

O conjunto de expressões contextuais descreve determinada situação que mapeia um contexto, as quais se referem aos tipos de contexto que podem ser identificados. Definiu-se para aplicação somente dois tipos de contexto - s*ujeito e objeto*, porém uma expressão contextual pode ter vários tipos de contexto associados a ela. Sendo assim, pode-se defini-la como:

- ***ExpressãoContextual: EC***
  - *EC = (CT1,CT2,...,CTN)*

No exemplo citado na figura 1 há dois tipos de contexto, logo:

- ***ExpressãoContextual: EC***
  - *EC = (CT1,CT2)*

Há uma expressão contextual no exemplo EC1 = (CT1(P1,P2),CT2 (P1)) que pode ser traduzida por EC1=(*sujeito-usuario*(função-*chefe da uti*, local-*uti),objeto-pep* (local-*uti)*), onde EC $\subset$ CTs.

O significado de expressão contextual leva a uma definição de contexto como encadeamento de informações sobre um ambiente, ou o conjunto de idéias, situações, eventos e informações necessárias para o correto entendimento do ambiente em que as informações são mostradas na forma de propriedade. Uma "expressão contextual", então, pode ser reconhecida como um "conjunto de propriedades de um elemento", ela contém informações que caracterizam um elemento. Assim, define-se um contexto como **um encadeamento de propriedades de um elemento em um ambiente,** o que pode ser traduzido como um **Conjunto de Expressões Contextuais**. Observa-se que um "elemento" pode ser um usuário, um dispositivo ou um recurso, tornando assim o contexto uma definição aplicável a diferentes domínios, ou contendo diferentes tipos de contexto.

Já que contexto pode ser formado por uma ou mais expressões contextuais, ele pode ser definido como:

- ***Contexto: CTX***
  - *CTX = (EC1,....,ECN)*

Neste exemplo um contexto é composto por uma expressão contextual, portanto, CTX = (EC1).

## 4. Mecanismo de Controle de Acesso baseado em Expressões Contextuais

Um mecanismo que leve em conta expressões contextuais deve possibilitar nas autorizações de acesso a verificação dessas expressões. No RBAC uma autorização é expressa pela tupla ***<p,obj,tp,opr>***, onde ***p*** corresponde ao papel para o qual um privilégio é estabelecido, ***obj*** é o recurso ou objeto para qual o privilégio se aplica, ***tp*** especifica o tipo de privilégio, positivo quando concedido e negativo quando proibido; ***opr*** é o tipo de

privilégio estabelecido. Para integrar as expressões contextuais nesse modelo deve-se estender essa tupla permitindo que ***tp*** seja substituída por um contexto CTX.

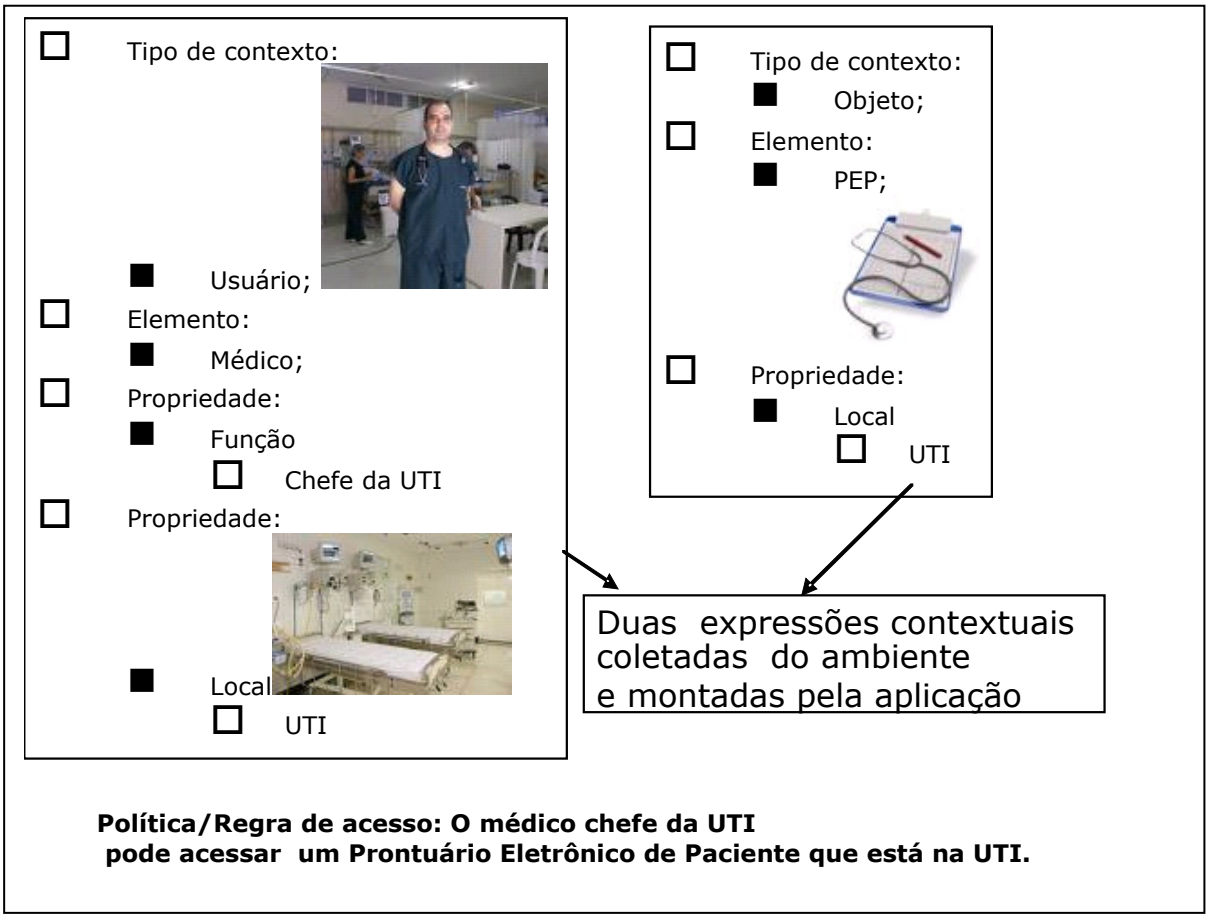


Figura 1. Exemplo da extração de expressões contextuais do ambiente confrontado com uma política de controle de acesso.

O contexto ***ctx*** terá quatro estados possíveis: n*egado, permitido, indefinido e indeterminado* e será o resultado da avaliação de um contexto. A tupla anterior pode então ser reescrita da seguinte forma:

Autorização: ***<p,obj,ctx,opr>***

Onde ***ctx*** pode ser mapeado por uma política de autorização de acesso através das expressões de contexto na forma de propriedades de contexto. Quando ocorre a interação do usuário com o sistema, o mesmo deverá mapear o contexto existente. Esse contexto será comparado com os vários contextos que são cadastrados previamente em forma de Políticas de Controle de Acesso (PCAs); se um corresponder com pelo menos uma delas o acesso será concedido.

Uma PCA pode ser representada como um contexto: PCA=(CTX), sendo que CTX $\subset$ (ECs), então basta verificar se as expressões de contexto retiradas do ambiente se refletem em alguma PCA existente. Como as PCAs podem ser representadas através da linguagem XACML *eXtensible Access Control Markup Language*, conforme pode ser observado na figura 2, o mapeamento regras de acesso dentro das definições de contexto se torna possível.

Na figura 2 são mostradas duas regras de acesso. A primeira EC1=(CT1(P1,P2)) pode ser traduzida como EC1=(*Sujeito((tempo,=,10:00),(função,=,enfermeira))*. A segunda EC2=(CT2(P1,P2)) pode ser traduzida por EC2=(*objeto((contador,=,20),(local,=,UTI))*. O contexto informado agrega duas expressões de contexto e dentre elas uma precisa ser correspondida para que o acesso seja concedido. Assim, há flexibilidade em relação às políticas de acesso, pois quando se verifica mais uma possibilidade de acesso ela pode ser mapeada em forma de mais uma expressão contextual, independente de quantos tipos de contexto estejam envolvidos.

```
<Context>
 <Express_Context>
   <Context_Type="Sujeito">
    <Property Name="TEMPO">
      <Operador OP=">"/>
       <Value V="10:00"/>
    </Property>
    <Property Name ="Funcao">
      <Operador="="/>
       <Value="Enfermeira"/>
    </Property>
   </Context_Type>
 </Express_Context>
 <Express_Context>
  <Context_Type="Objeto">
    <Property Name="contador">
      <Operador OP="="/>
      <Value V="20:00"/>
    </Property>
    <Property Name ="local">
      <Operador="="/>
      <Value="UTI"/>
    </Property>
  </Context_Type>
 </Express_Context>
</context>
```

Figura 2. Mapeamento de regras de acesso dentro das definições de contexto.

Dentro dessa definição é possível compor regras de acesso que correspondem a qualquer domínio, pois o mapeamento das mesmas não depende da estrutura do modelo, mas das expressões contextuais que variam de acordo com cada ambiente. Abaixo mais um exemplo, desta vez de uma PCA mapeada para um Administrador de Redes de Computadores obter acesso a um objeto através de uma rede Wireless, como mostra a figura 3:

```
<Context>
 <Express_Context>
  <Context_Type="Sujeito">
   <Property Name="Função">
    <Operador OP="="/>
    <Value V="Administrador da Rede"/>
   </Property>
  </Context_Type>
  <Context_Type="Objeto">
   <Property Name="local">
    <Operador OP="="/>
    <Value V="Rede Movel"/>
   </Property>
  </Context_Type>
 </Express_Context>
</context>
```

Figura 3. Esta figura mostra a expressão contextual para um acesso a um objeto pelo administrador de rede.

A PCA da figura 3 admite o acesso a determinado objeto da rede que pode ser um arquivo tipo httpd.conf através da rede sem fio, desde que o usuário do sistema esteja na função de administrador da rede. A tupla inicial de solicitação de acesso seria a seguinte: ***<p,obj,tp,opr>*** traduzida para ***<Gerente de Informática, httpd.conf, CTX, leitura>*** Onde CTX trata-se do contexto extraído do ambiente, que poderia ser o seguinte:

CTX=(EC1) e

EC1=(CT1(P1), CT2(P1)) onde:

EC1 = ***(sujeito(função=gerente de informática), (Objeto(local=Rede Móvel))***

Esse contexto é encaminhado para que possa ser avaliado em relação à PCA correspondente. O resultado será disponibilizado dentro das quatro opções de resposta (*negado, permitido, indefinido e indeterminado).* Nesse caso será *negado* tendo em vista que a política somente permite o acesso através da rede móvel se o papel ou perfil do sujeito é Administrador de Rede.

Como pode ser percebido, é necessário que se estabeleça um algoritmo que faça a avaliação do contexto extraído do ambiente com o contexto mapeado pela política, para identificar se os dois são correspondentes ou não. Para tanto, sugere-se o algoritmo da figura 4.

```
Receber requisição de acesso
   Selecionar todos os contextos que satisfaçam a requisição de acesso:<P,O,OPR>
   Selecionar Expressões de Contexto
   Determinar flage = false
   Enquanto houver Expressões de Contexto ou flage == false
   inicio
      flagT = True
      Enquanto encontrar Tipo de Contexto ou FlagT == True
      inicio
                        Para cada Propriedade
                        Se Propriedade == falsa
                       flagT = false
      fim
      Se flagT == true
       flage = true
 fim
```

Figura 4. Algoritmo que faz a avaliação do contexto extraído do ambiente em relação às regras de controle de acesso.

O algoritmo proposto recebe a requisição de acesso na forma de tupla e recupera todas as regras que se referem ao objeto, perfil e modo de acesso. De posse das expressões de contexto que se referem ao objeto, tipo de acesso e operação, é necessário que as mesmas sejam avaliadas para conferir se existe alguma que corresponde ao contexto extraído do ambiente. Determina-se *flage = FALSE* indicando que se **nenhuma** expressão verdadeira foi encontrada, esse valor indicará a resposta *negado* como resultado da avaliação.

Para que uma expressão contextual seja considerada válida e, conseqüentemente, o valor do *flage = true* resultando em uma resposta *permitido,* é necessário que todos os seus tipos de contexto tenham propriedades verdadeiras. Inicialmente supõe-se que todas as propriedades são verdadeiras, mas o algoritmo irá conferir uma a uma. Se alguma não for similar ao contexto extraído do ambiente, toda expressão contextual será desconsiderada e o algoritmo verificará se ainda existem expressões a serem avaliadas. Em caso positivo, os tipos de contexto das mesmas também serão submetidos à análise. Caso o algoritmo percorra todas as expressões e não encontre nenhuma válida, o *flage* será *falso* e o acesso não será concedido, pois a resposta será *negada.* Do contrário, se pelo menos uma expressão contextual for válida, será atribuído ao *flage* a condição de verdadeiro e a busca acaba.

## 5. CONCLUSÃO

Neste artigo foi apresentado um mecanismo de controle de acesso baseado em expressões contextuais que pode ser agregado ao modelo de controle de acesso tradicional RBAC. Percebe-se que o mecanismo corresponde às exigências da realidade atual, que necessita de um mapeamento fiel do ambiente de onde se quer estabelecer o acesso; isso é conseguido através das expressões contextuais. Estas permitem controle com granularidade fina, isto é, pode-se descrever em detalhes de qual ambiente se está tentando interagir com um sistema. Assim, pode-se construir políticas de controle de acesso mais abrangentes e que aumentam a segurança dos dados.

A utilização de expressões contextuais no controle de acesso ainda depende de sua implementação e interação com um Sistema Gerenciador de Políticas de Controle de Acesso. Isso facilitará o processo de criação das políticas de segurança que definirão as condições de contexto nas quais os objetos, aplicações ou recursos poderão ser acessados e manipulados. O mecanismo proposto nesse artigo está sendo implementado, levando em conta expressões contextuais no controle de acesso com a tecnlogogia *Web Service.* Isto vai possibilitar a independência dos sistemas legados, pois *Web Services* se comunicam através da troca de mensagens XML, um padrão que possui grande aceitabilidade. Portanto, sua implementação atenderá à demanda atual, que exige ferramentas não dependentes em relação à heterogeneidade presente em ambientes computacionais.

## 6. REFERÊNCIAS

FERRAIOLO, F. AND SANDHU, S. (2001) Standard for Role-Based Access Control, In: *Advances in Computer Science* , pages:224–274. Information and System Security.

WILIKENS, M. AND FERITI, S.(2002) A context-related authorization and access control method based on RBAC: a case study from the health care domain. In: Seventh ACM Symposium on Access Control Models and Technologies, Proceedings p. 117-124.

DEY, K. AND ABOWD, D. (1999 ) Towards a Better Understanding of Context and Contextawareness. In: Gvu technical report GIT-GVU-99-22, College of Computing, Georgia Institute of Technology.

BACON, J AND MOODY, K (2002) A Model of OASIS Role-Based Access Control and its Support for Active Security. In: ACM Transaction on Information and System Security, v.5, p.492-540.

BERTINO, E AND GHAFOOR, A. (2008) Context-Aware Adaptation of Access-Control Policies. In IEEE Computer Society, pag 51-54 February.

SANDHU, R, FERRAIOLO, D. AND KUHN, R. (2000) The NIST Model for Role-Based Access Control: Towards a Unified Standard. In Proceedings. In: 5th ACM Workshop on Role-Based Access Control, Berlin, Germany, July.

MOTTA, B, FURUIE, S. (2002) Um modelo de autorização contextual para o controle de acesso baseado em papéis. In: II Workshop em Segurança de Sistemas Computacionais (WSeg2002), páginas 137–144, Porto Alegre-RS, Brasil. SBC.

Web Site do RBAC – NIST: http://csrc.nist.gov/rbac