Role Based Access Control (RBAC)

Nicolas Adrién | GeneXus Training

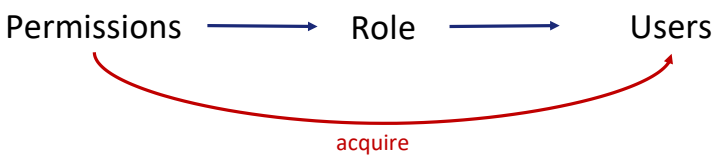
Role Based Access Control (RBAC)

Confidentiality	Integrity	Information Availability

O controle de acesso baseado em roles, ou RBAC como são suas siglas, é uma abordagem para restringir o acesso ao sistema a usuários não autorizados.

Tem como objetivo garantir a confidencialidade, integridade e disponibilidade da informação, pois baseando-se no princípio de privilégio mínimo, limita o acesso dos usuários e as ações que podem realizar.
Desta forma, é reduzido o risco de sofrer violações de segurança quando a conta de um usuário é comprometida.

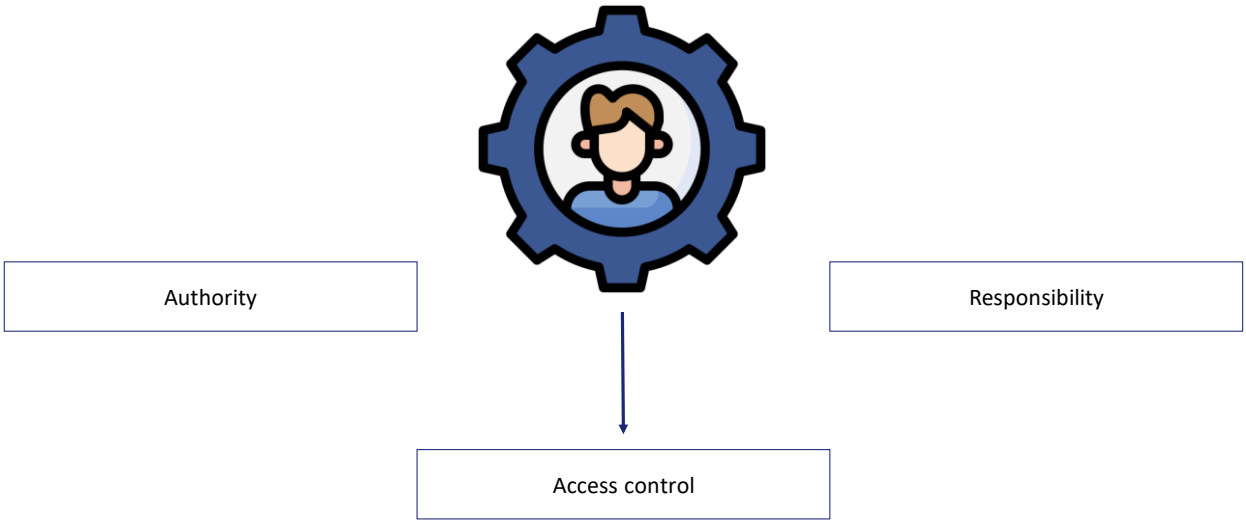
Role Based Access Control (RBAC)



Dentro de uma organização, são criadas **roles** para diversas funções.
As **permissões** para realizar certas operações são atribuídas a roles específicas.
Aos **usuários** são atribuídas roles particulares e, através dessas atribuições de roles, adquirem as permissões para realizar funções específicas do sistema de computador.

Dado que aos usuários não são atribuídas permissões diretamente, pois só as adquirem através de sua role (ou roles), a administração dos direitos de usuário individuais torna-se uma questão de simplesmente atribuir as roles apropriadas à conta do usuário. isto simplifica operações comuns, como adicionar um usuário ou alterar o departamento de um usuário.

Roles



Dentro dos elementos destacados do RBAC, em primeiro lugar temos as roles.

Uma role é uma função ou cargo dentro de uma organização com alguma semântica associada em relação à autoridade e responsabilidade conferida a um membro da função.

É visto corretamente como uma construção semântica em torno da qual é formulada a política de controle de acesso.
A coleção particular de usuários e permissões reunidas por uma role é transitória, mas a role é mais estável porque as atividades ou funções de uma organização tendem a mudar com menos frequência.

Com RBAC é possível predefinir as relações entre roles e permissões, o que simplifica a atribuição de usuários às roles predefinidas.

What is the difference between groups and roles?

Groups

Roles

Muitas vezes costumam ser misturados os conceitos de Grupos e Roles.
Uma diferença importante entre a maioria das implementações de grupos e o conceito de roles é que os grupos normalmente são tratados como uma coleção de usuários e não como uma coleção de permissões.

Os grupos de usuários como unidade de controle de acesso são fornecidos comumente em muitos sistemas de controle de acesso.

Por outro lado, as roles são tanto uma coleção de usuários, por um lado, quanto uma coleção de permissões, por outro.

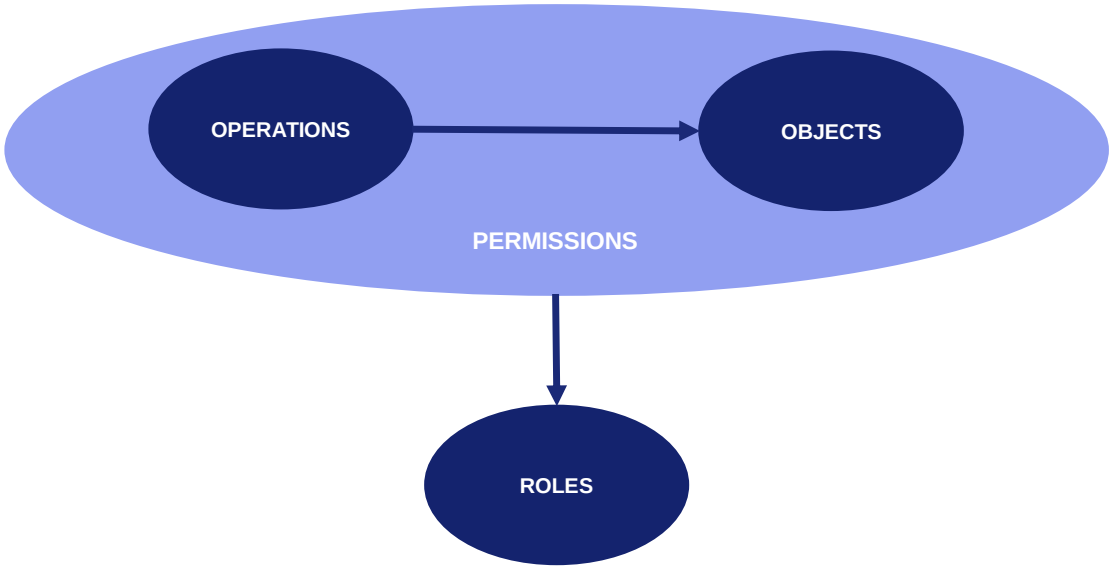
O papel serve como intermediário para unir estas duas coleções.

Users

Posteriormente temos os Usuários.
No modelo do RBAC, os usuários são seres humanos.

O conceito deles pode ser generalizado para incluir agentes autônomos inteligentes como robôs, computadores ou mesmo redes de computadores.

Permissions



Quanto às Permissões, estas descrevem a possibilidade de realizar uma operação sobre um objeto específico, sendo como uma aprovação de um modo particular de acesso a um ou mais objetos no sistema.
Podem estar associadas a uma ou mais roles, mas a coleção de permissões vinculadas por role é transitória.

Através das permissões associadas às roles ativas de cada sessão é que são realizadas as decisões de acesso.

Algumas literaturas de controle de acesso falam de "permissões negativas" que negam acessos, em vez de concedê-los. Em nosso contexto ministrado, a proibição de acesso modelamos como uma restrição em vez de uma permissão negativa.

Sessions

Finalmente e no último ponto estão as Sessões, que realizam o mapeamento entre os usuários e suas roles ativas.

Em relação aos usuários e as sessões, um pode ter mais de uma sessão estabelecida de forma simultânea. Cada usuário estabelece uma sessão durante a qual ativa algum subconjunto de roles para as quais está autorizado.

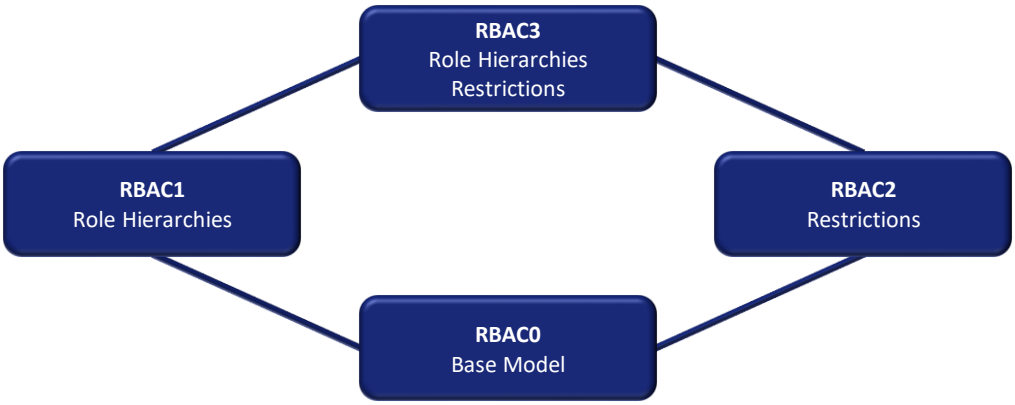
Sessions

Set of valid session identifiers of type Session

Session Mapping with User and active roles

O estado das sessões do sistema está dividido em duas partes:
A primeira é um conjunto de identificadores válidos do tipo Session.
A segunda parte é uma função que mapeia cada sessão com o usuário dono dela e o conjunto de suas roles ativas.

Levels



Para compreender as diversas dimensões de RBAC, neste é definida uma família de quatro modelos conceituais.

Em primeiro lugar está RBAC0, o modelo base localizado na parte inferior, que indica que é o requisito mínimo para qualquer sistema que pretenda suportar RBAC.

Um patamar acima, estão RBAC1 e RBAC2 que incluem o RBAC0, mas adicionam características a ele:

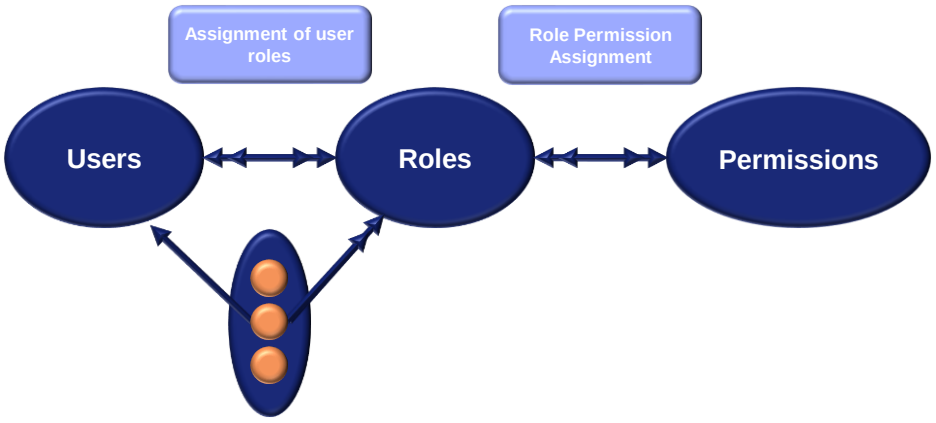
- RBAC1 adiciona o conceito de hierarquias de roles, que são situações em que as roles podem herdar permissões de outras roles.
- RBAC2 adiciona restrições, que impõem restrições para as configurações aceitáveis dos diferentes componentes de RBAC.

RBAC1 e RBAC2 são incomparáveis entre si.

Finalmente no último patamar temos RBAC3, que inclui RBAC1 e RBAC2, e por transitividade RBAC0.

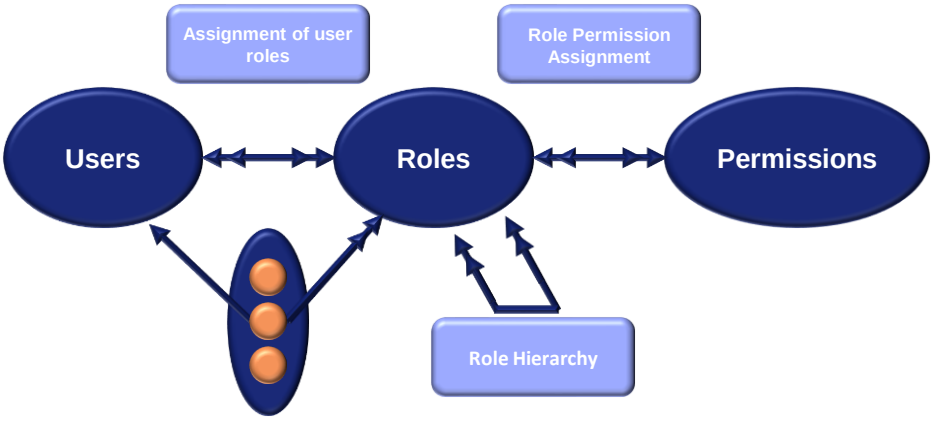
Estes modelos buscam ser pontos de referência para a comparação com sistemas e modelos utilizados por outros pesquisadores e desenvolvedores. Também podem servir como guia para o desenvolvimento de produtos e sua avaliação por parte de possíveis clientes.

Level 0 – Base Model



No modelo base estão os quatro conjuntos de entidades mencionados anteriormente: usuários, roles, permissões e sessões. Conforme visto no diagrama, um usuário pode ser membro de muitas roles e uma role pode ter muitos usuários. De forma similar, uma role pode ter muitas permissões e a mesma permissão pode ser atribuída a muitas roles.
A chave de RBAC reside nestas duas relações.
Finalmente, uma sessão pode ser composta por um usuário e muitas roles.

Level 1 – Hierarchy of Roles



No modelo RBAC1, são introduzidas as hierarquias de roles, onde podemos defini-la como um meio natural de estruturar as roles, com a finalidade de refletir as linhas de autoridade e responsabilidade de uma organização.

As hierarquias de roles são incluídas quase inevitavelmente sempre que são analisadas as roles e também são implementadas comumente em sistemas que proporcionam roles.

Vejamos um exemplo...

Level 1 – Hierarchy of Roles - Example

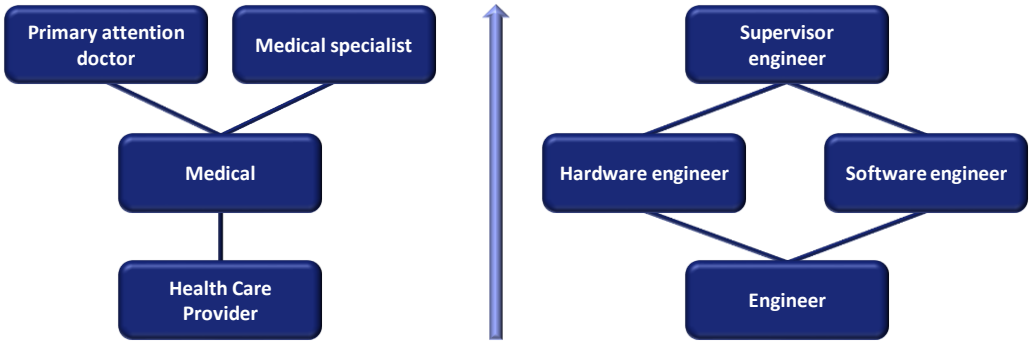


Figure A

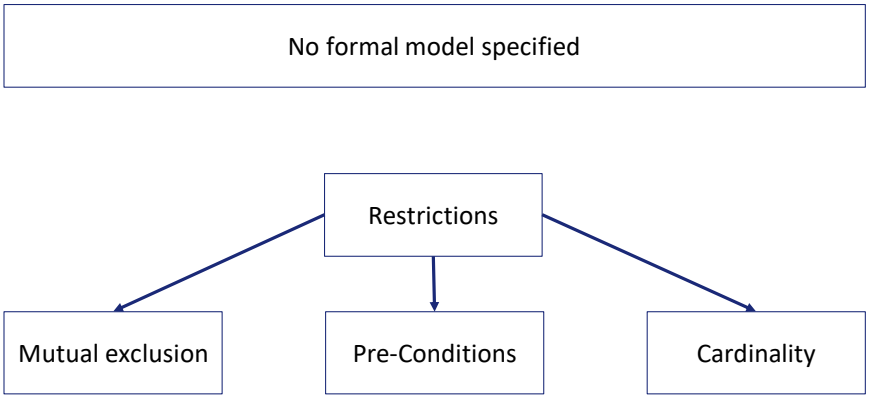
Figure B

Por convenção, as roles mais poderosas (ou senior) são mostradas na parte superior destes diagramas e as roles menos poderosas (ou junior) na parte inferior.

Na Figura A, a role mais subordinada é a de prestador de cuidados de saúde. O papel do médico é superior ao do prestador de cuidados de saúde e, portanto, herda todas as permissões dele.
A função de médico pode ter permissões próprias, além daquelas herdadas da função de prestador de cuidados de saúde.
Como a herança de permissões é transitiva, a role de médico de atenção primária herda as permissões das roles de médico e prestador de cuidados de saúde.
Tanto o médico de atenção primária como o médico especialista, herdam as permissões da role médico, mas cada um deles terá atribuídas permissões diferentes diretamente.

Por outro lado, na Figura B é ilustrada também a herança, mas desta vez fazendo referência à herança múltipla de permissões, onde a role de Engenheiro Supervisor herda tanto da role de engenheiro de hardware como da de software.

Level 2 – RBAC0 + Restrictions



O modelo RBAC2 introduz o conceito de restrições. Não muda em relação a RBAC0, exceto que exige que haja um conjunto de restrições que determinem se os valores de vários componentes de RBAC0 são aceitáveis ou não. Somente serão permitidos valores aceitáveis.

As restrições são um aspecto importante de RBAC e às vezes se argumenta que são a principal motivação para o controle de acesso, pois são um mecanismo poderoso para desenhar políticas organizacionais de alto nível.

Uma vez que certas roles são declaradas mutuamente excludentes, não é necessário se preocupar tanto pela atribuição de roles a usuários individuais. Esta última atividade pode ser delegada e descentralizada sem medo de comprometer os objetivos gerais da política da organização.

Podemos mencionar alguns tipos de exemplos de Restrições.
Um poderia ser a Exclusão mútua, onde uma restrição poderia ser que seja possível atribuir ou ativar apenas uma role do conjunto. Este tipo de restrições pode ser utilizado para fazer cumprir a separação de funções.

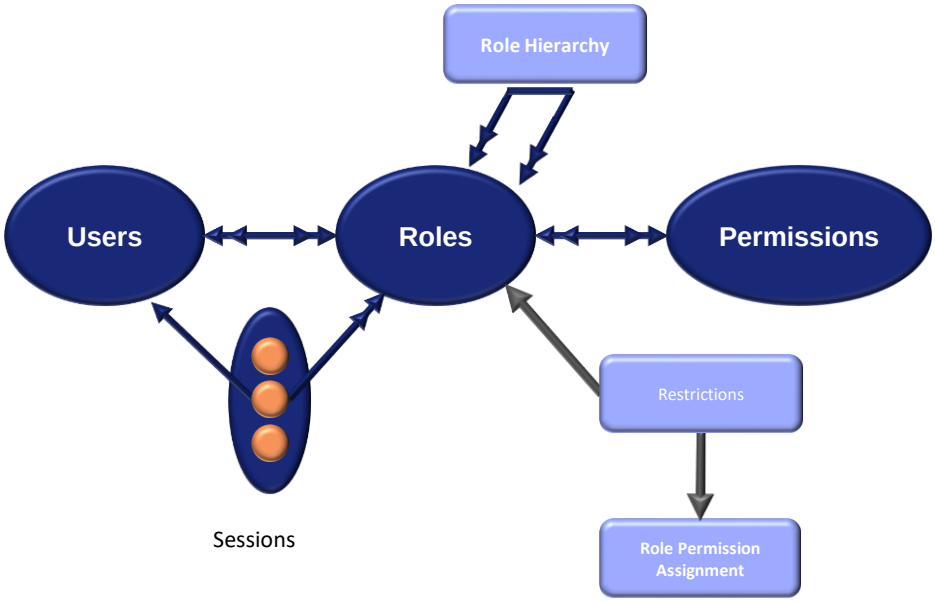
Outro tipo pode ser as Pré-condições, onde uma restrição de exemplo poderia ser que seja possível atribuir uma role a um usuário somente se ele tiver alguma outra role de condição prévia. Neste caso, elas podem ser utilizadas para fazer cumprir o princípio de privilégio mínimo.

Finalmente, temos Cardinalidade, onde é possível ter restrições de exemplo como:

- Máximo de usuários aos quais podem ser atribuídas uma role

- Funções máximas que qualquer usuário pode possuir (possivelmente, em uma sessão)
- Máximo de roles que têm uma determinada permissão
- Entre outras

Level 3 – Consolidated Model



Finalmente temos o modelo RBAC3, que combina RBAC1 e RBAC2 para proporcionar hierarquias de roles e restrições simultaneamente. Sobre isto há várias questões que surgem ao unir estes dois conceitos:
As restrições podem ser aplicadas à própria hierarquia de roles e podem limitar o número de roles senior (ou junior) que pode ter uma role determinada.
Também podem ser restringidas duas ou mais roles para que não tenham uma role senior (ou junior) em comum.

Este tipo de restrições são úteis em situações em que foi descentralizada a autoridade para alterar a hierarquia de roles, mas o administrador de segurança deseja restringir a maneira geral em que podem ser realizadas essas alterações.

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications