Role Based Access Control (RBAC)

Nicolas Adrién | GeneXus Training

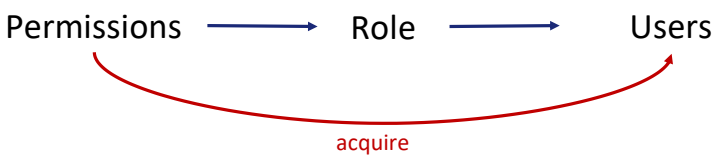
Role Based Access Control (RBAC)

Confidentiality	Integrity	Information Availability

El Control de acceso basado en roles, o RBAC como son sus siglas, es un enfoque para restringir el acceso al sistema a usuarios no autorizados.

Tiene como objetivo asegurar la confidencialidad, integridad y disponibilidad de la información, puesto que basándose en el principio de privilegio mínimo, limita el acceso de los usuarios y las acciones que pueden llevar a cabo.
De esta forma, se reduce el riesgo de sufrir violaciones de seguridad cuando la cuenta de un usuario se ve comprometida.

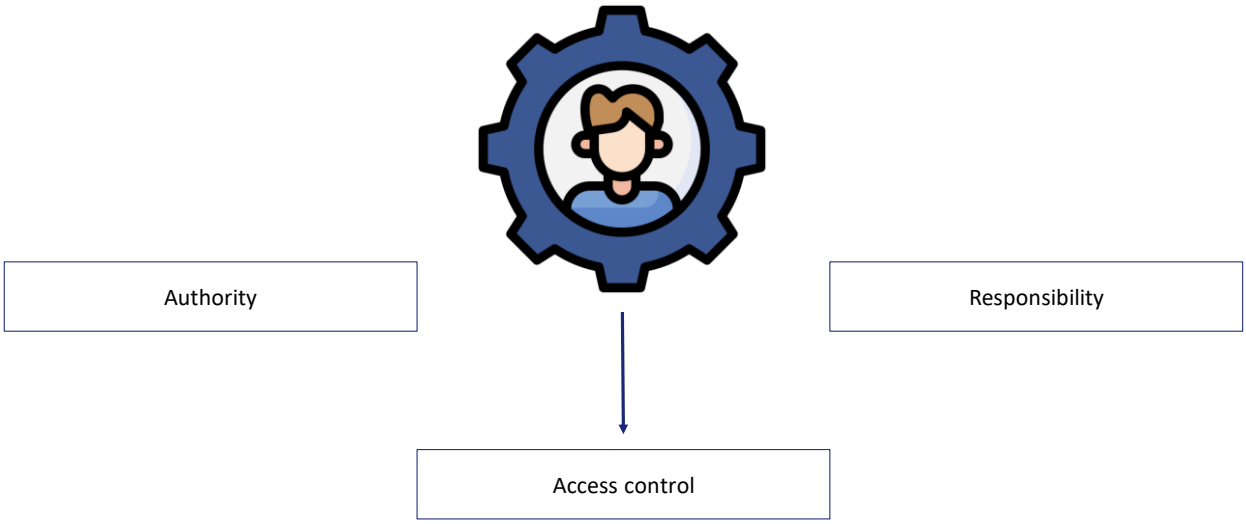
Role Based Access Control (RBAC)



Dentro de una organización, se crean **roles** para diversas funciones.
Los **permisos** para realizar ciertas operaciones se asignan a roles específicos.
A los **usuarios** se les asignan roles particulares y, a través de esas asignaciones de roles, adquieren los permisos para realizar funciones específicas del sistema informático.

Dado que a los usuarios no se les asignan permisos directamente, ya que solo los adquieren a través de su rol (o roles), la administración de los derechos de usuario individuales se convierte en una cuestión de simplemente asignar los roles apropiados a la cuenta del usuario; esto simplifica las operaciones comunes, como agregar un usuario o cambiar el departamento de un usuario.

Roles



Dentro de los elementos destacados del RBAC, en primer lugar tenemos los roles.

Un rol es una función o título de trabajo dentro de una organización con alguna semántica asociada con respecto a la autoridad y responsabilidad conferida a un miembro de la función.

Se ve correctamente como una construcción semántica en torno a la cual se formula la política de control de acceso.
La colección particular de usuarios y permisos reunidos por un rol es transitoria, pero el rol es más estable porque las actividades o funciones de una organización suelen cambiar con menos frecuencia.

Con RBAC es posible predefinir las relaciones entre roles y permisos, lo que simplifica la asignación de usuarios a los roles predefinidos.

What is the difference between groups and roles?

Groups

Roles

Muchas veces se suelen mezclar los conceptos de Grupos y Roles.
Una diferencia importante entre la mayoría de las implementaciones de grupos y el concepto de roles es que los grupos normalmente se tratan como una colección de usuarios y no como una colección de permisos.

Los grupos de usuarios como unidad de control de acceso se proporcionan comúnmente en muchos sistemas de control de acceso.

Por otro lado, los roles son tanto una colección de usuarios por un lado como una colección de permisos por el otro.

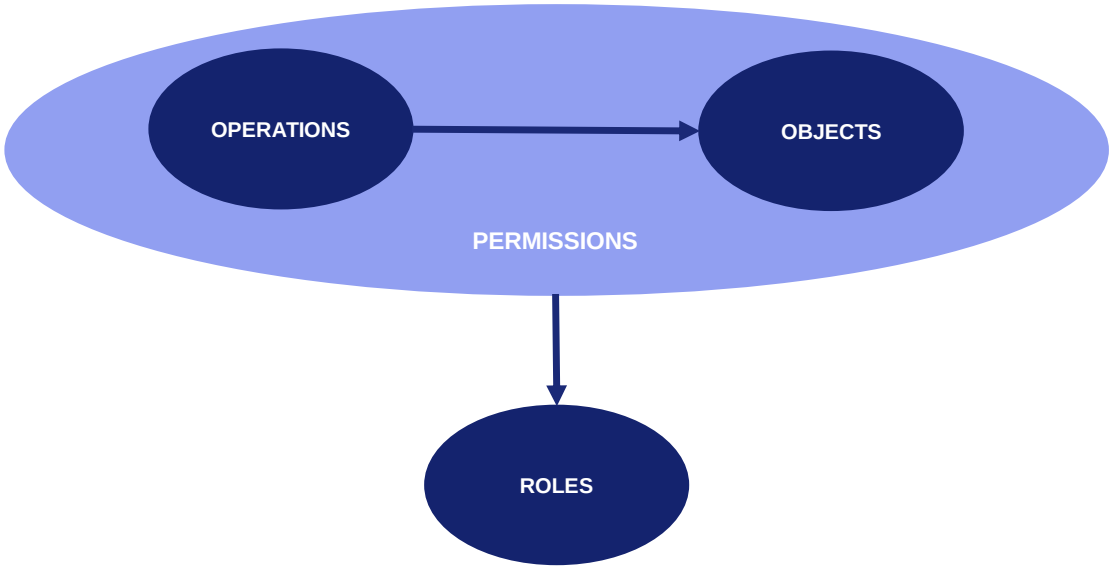
El papel sirve como intermediario para unir estas dos colecciones.

Users

Posteriormente tenemos a los Usuarios.
En el modelo del RBAC, los usuarios son seres humanos.

El concepto de los mismos se puede generalizar para incluir agentes autónomos inteligentes como robots, computadoras o incluso redes de computadoras.

Permissions



En cuanto a Permisos, estos describen la posibilidad de realizar una operación sobre un objeto determinado, siendo como una aprobación de un modo particular de acceso a uno o más objetos en el sistema.
Estos pueden estar asociados con uno o mas roles, pero la colección de permisos vinculados por rol es transitoria.

A través de los permisos asociados con los roles activos de cada sesión es que se realizan las decisiones de acceso.

Algunas literaturas de control de acceso hablan de "permisos negativos" que niegan accesos, en lugar de otorgarlos. En nuestro marco dictado, la denegación de acceso la modelamos como una restricción en lugar de un permiso negativo.

Sessions

Finalmente y en el ultimo punto se encuentran las Sesiones, que realizan el mapeo entre los usuarios y sus roles activos.

En cuanto a los usuarios y las sesiones, uno puede tener mas de una sesión establecida de manera simultanea. Cada usuario establece una sesión durante la cual activa algún subconjunto de roles para los que está autorizado.

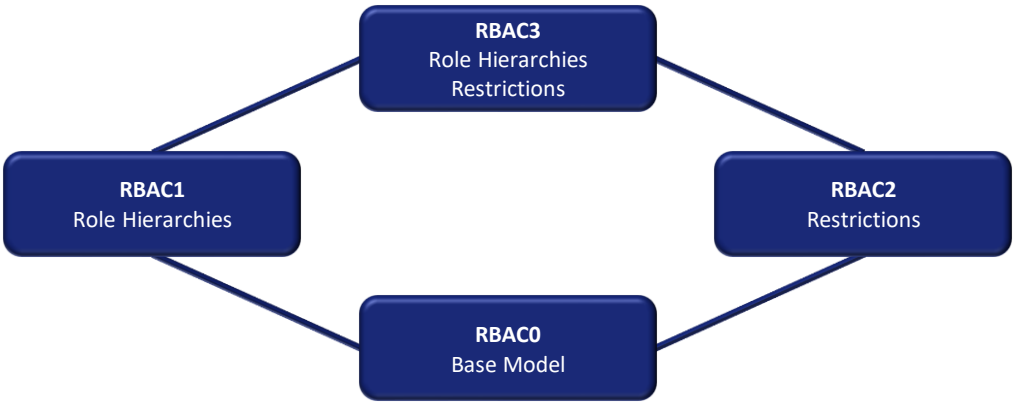
Sessions

Set of valid session identifiers of type Session

Session Mapping with User and active roles

El estado de las sesiones del sistema esta dividido en dos partes:
La primera es un conjunto de identificadores validos de sesión de tipo Session.
La segunda parte es una función que mapea cada sesión con el usuario dueño de la misma y el conjunto de sus roles activos.

Levels



Para comprender las diversas dimensiones de RBAC, en este se define una familia de cuatro modelos conceptuales.

En primer lugar se encuentra RBAC0, el modelo base ubicado en la parte inferior, que indica que es el requisito mínimo para cualquier sistema que pretenda admitir RBAC.

Un escalón mas arriba, se encuentran RBAC1 y RBAC2 que incluyen el RBAC0, pero agregan características al mismo:

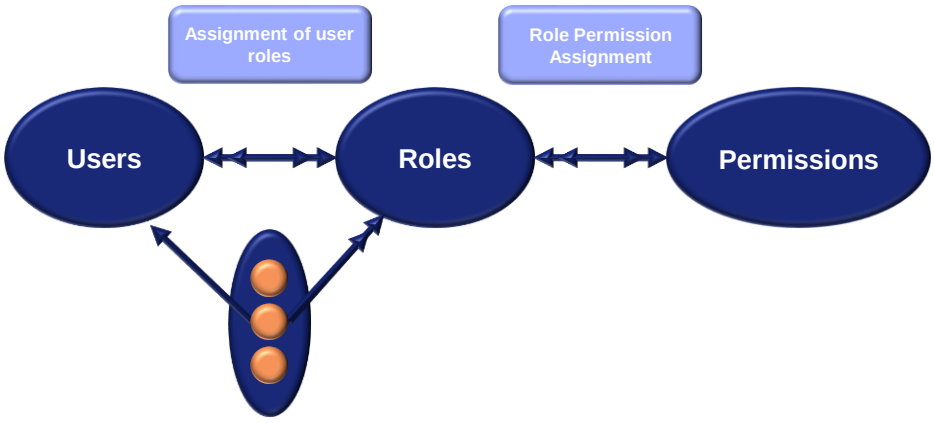
- RBAC1 agrega el concepto de jerarquías de roles, que son situaciones en las que los roles pueden heredar permisos de otros roles.
- RBAC2 agrega restricciones, que imponen restricciones a las configuraciones aceptables de los diferentes componentes de RBAC.

RBAC1 y RBAC2 son incomparables entre sí.

Finalmente en el ultimo escalón tenemos a RBAC3, que incluye RBAC1 y RBAC2, y por transitividad RBAC0.

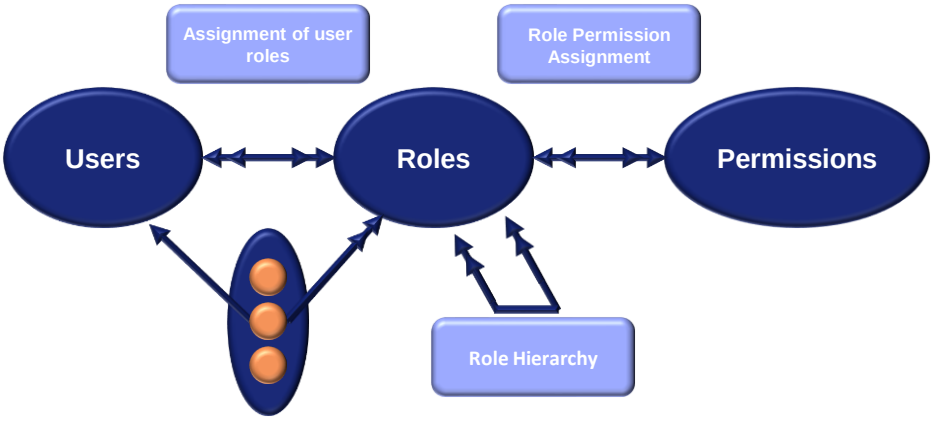
Estos modelos pretenden ser puntos de referencia para la comparación con sistemas y modelos utilizados por otros investigadores y desarrolladores. También pueden servir como guía para el desarrollo de productos y su evaluación por parte de posibles clientes.

Level 0 – Base Model



En el modelo base están los cuatro conjuntos de entidades mencionados anteriormente: usuarios, roles, permisos y sesiones. Como se ve en el diagrama, un usuario puede ser miembro de muchos roles y un rol puede tener muchos usuarios. De manera similar, un rol puede tener muchos permisos y el mismo permiso se puede asignar a muchos roles.
La clave de RBAC radica en estas dos relaciones.
Finalmente, una sesión puede estar compuesta por un usuario y muchos roles.

Level 1 – Hierarchy of Roles



En el modelo RBAC1 se introduce la jerarquías de roles, donde la podemos definir como un medio natural para estructurar los roles, con el fin de reflejar las líneas de autoridad y responsabilidad de una organización.

Las jerarquías de roles se incluyen casi inevitablemente siempre que se analizan los roles, y también se implementan comúnmente en sistemas que proporcionan roles.

Veamos un ejemplo…

Level 1 – Hierarchy of Roles - Example

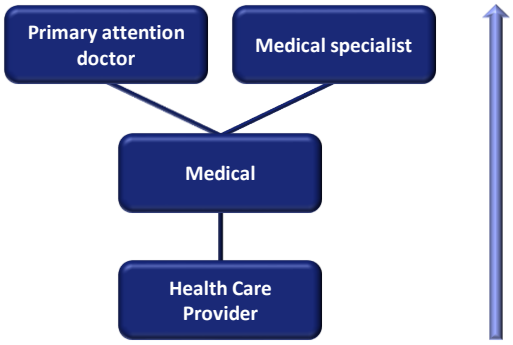


Figure A

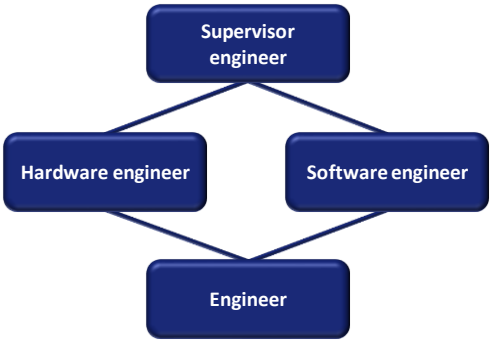


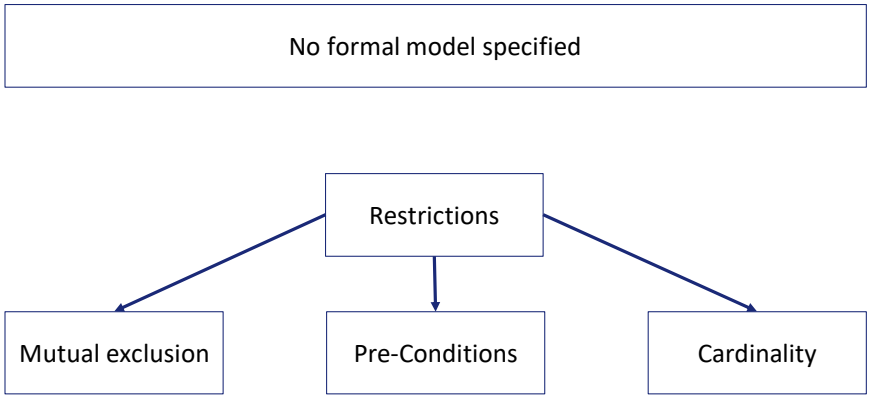
Figure B

Por convención, los roles más poderosos (o senior) se muestran en la parte superior de estos diagramas y los roles menos poderosos (o junior) en la parte inferior.

En la Figura A, el rol más subalterno es el de proveedor de atención médica. El papel del médico es superior al del proveedor de atención médica y, por lo tanto, hereda todos los permisos de él.
La función de médico puede tener permisos por si sola, además de los heredados de la función de proveedor de atención médica.
Como la herencia de permisos es transitiva, el rol de médico de atención primaria hereda los permisos de los roles de médico y proveedor de atención médica.
Tanto el médico de atención primaria como el médico especialista, heredan los permisos del rol médico, pero cada uno de ellos tendrá asignados permisos diferentes directamente.

Por otro lado, en la Figura B se ilustra también la herencia, pero esta vez haciendo referencia a la herencia múltiple de permisos, donde el rol de Ingeniero Supervisor hereda tanto del rol de ingeniero de hardware como del de software.

Level 2 – RBAC0 + Restrictions



El modelo RBAC2 introduce el concepto de restricciones. No cambia con respecto a RBAC0, excepto que requiere que haya un conjunto de restricciones que determinen si los valores de varios componentes de RBAC0 son aceptables o no. Sólo se permitirán valores aceptables.

Las restricciones son un aspecto importante de RBAC y a veces, se argumenta que son la principal motivación para el control de acceso ya que son un mecanismo poderoso para diseñar políticas organizacionales de alto nivel.

Una vez que ciertos roles se declaran mutuamente excluyentes, no es necesario preocuparse tanto por la asignación de roles a usuarios individuales. Esta última actividad puede delegarse y descentralizarse sin temor a comprometer los objetivos generales de la política de la organización.

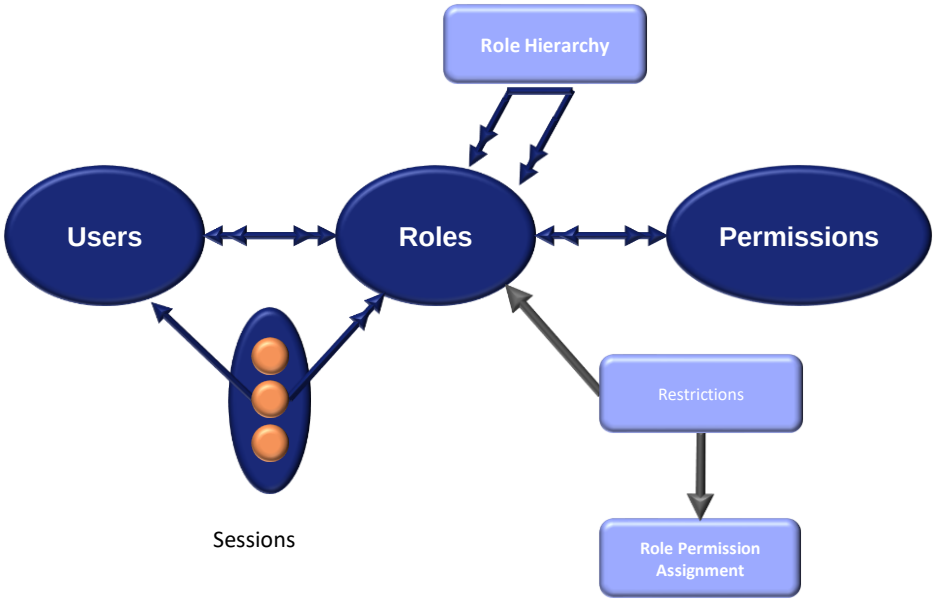
Podemos mencionar algunos tipos de ejemplos de Restricciones.
Uno podría ser la Exclusión mutua, donde una restricción podría ser que se pueda asignar o activar solo un rol del conjunto. Este tipo de restricciones se puede utilizar para hacer cumplir la separación de funciones.

Otro tipo puede ser las Pre-Condiciones, donde una restricción de ejemplo podría ser que se pueda asignar un rol a un usuario solo si éste posee algún otro rol de condición previa. En este caso, éstas se pueden utilizar para hacer cumplir el principio de privilegio mínimo.

Finalmente tenemos Cardinalidad, donde se pueden tener restricciones de ejemplo como:

- Máximo de usuarios a los que se les puede asignar un rol
- Funciones máximas que cualquier usuario puede poseer (posiblemente, en una sesión)
- Máximo de roles que tienen un cierto permiso
- Entre otras

Level 3 – Consolidated Model



Finalmente tenemos el modelo RBAC3, que combina RBAC1 y RBAC2 para proporcionar jerarquías de roles y restricciones en simultaneo. Por esto hay varias cuestiones que surgen al unir estos dos conceptos:
Las restricciones pueden aplicarse a la propia jerarquía de roles, y pueden limitar el número de roles senior (o junior) que puede tener un rol determinado.
También se pueden restringir dos o más roles para que no tengan un rol senior (o junior) en común.

Este tipo de restricciones son útiles en situaciones en las que se ha descentralizado la autoridad para cambiar la jerarquía de roles, pero el administrador de seguridad desea restringir la forma general en que se pueden realizar dichos cambios.

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