

# Servidores de aplicaciones

# Sesión 3: Gestión de seguridad



## Índice

- Gestión de seguridad en el sistema
- Gestión de usuarios y grupos
- Seguridad en una aplicación



## Seguridad en WebLogic

- WebLogic dispone de un mecanismo de gestión de seguridad por defecto
- Mecanismo de autentificación (el usuario o aplicación son quienes dicen ser?) y autorización (el usuario en cuestión tiene permiso para realizar una determinada acción?)
- Podemos usar el mecanismo por defecto, pero normalmente se suelen usar sistemas externos (LDAP, el propio sistema operativo, etc.)
- Estos sistemas se denominan providers



# Proceso para proporcionar seguridad a una aplicación

- La recomendación para introducir seguridad en una aplicación es seguir estos pasos:
  - Crear restricciones de seguridad en la aplicación (mediante el descriptor de despliegue), independientemente del servidor donde vayamos a desplegar. Independencia del servidor
  - El servidor dispondrá de usuarios, grupos, roles (principales)
  - Mapear restricciones de seguridad con principales en el servidor. En WebLogic se consigue mediante un descriptor adicional (weblogic.xml)



## Gestión de seguridad en WebLogic

- WebLogic define realms para la seguridad (conjunto de usuarios, grupos, roles y políticas de seguridad)
- Los recursos a los que se puede imponer políticas de seguridad son: aplicaciones (web, EJB, jar, etc.), JDBC, JNDI, EIS, JMS
- Pinchamos en Security Realms y nos aparecerá un realm por defecto



#### Realms

#### **Summary of Security Realms**

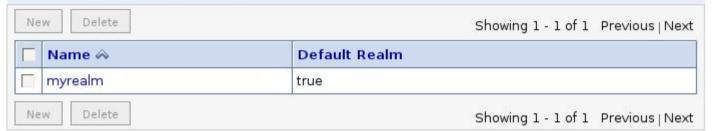
A security realm is a container for the mechanisms--including users, groups, security roles, security policies, and security providers--that are used to protect WebLogic resources. You can have multiple security realms in a WebLogic Server domain, but only one can be set as the default (active) realm.

This Security Realms page lists each security realm that has been configured in this WebLogic Server domain. Click the name of the realm to explore and configure that realm.

#### Customize this table

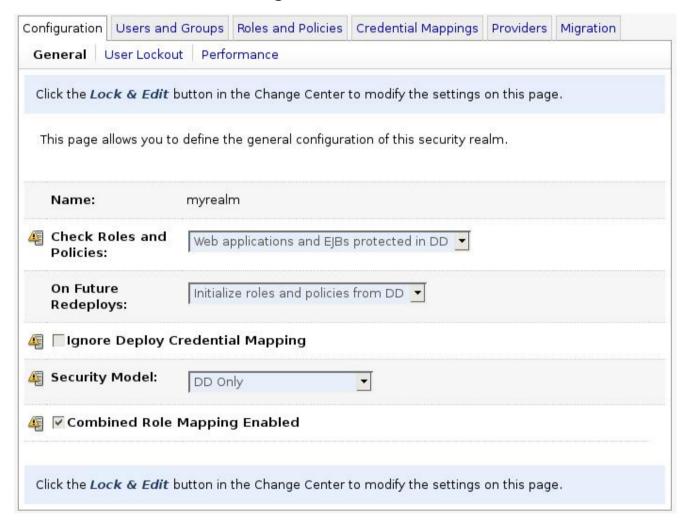
#### Realms

Click the Lock & Edit button in the Change Center to activate all the buttons on this page.



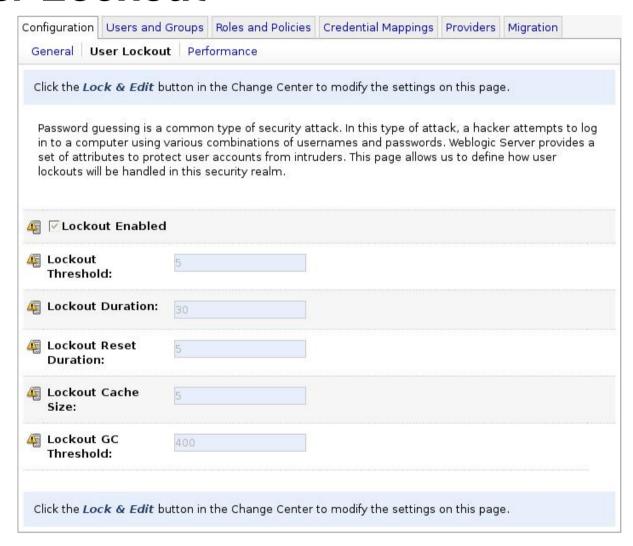


#### Pinchamos en myrealm





#### **User Lockout**





#### **Usuarios**

- Personas o programas
- Único en el sistema y se identifica mediante contraseña
- No existe guest (se puede definir, pero se desaconseja)
- Pinchamos en la solapa Users and Groups

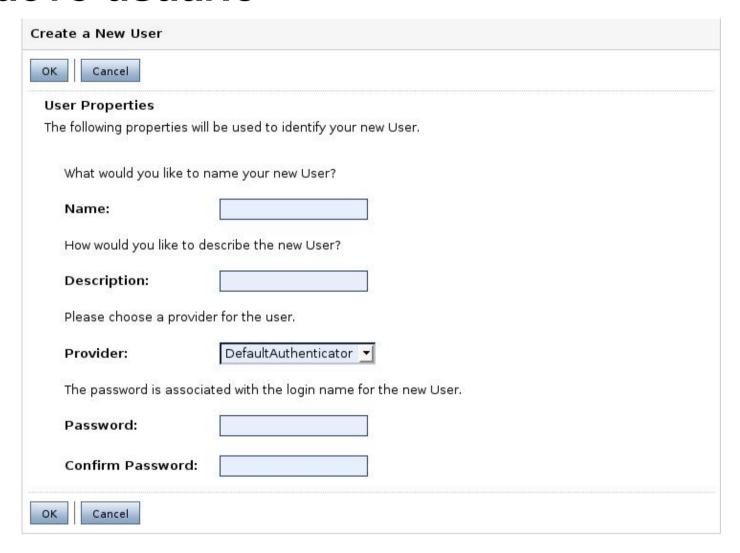


#### **Usuarios**





#### **Nuevo** usuario





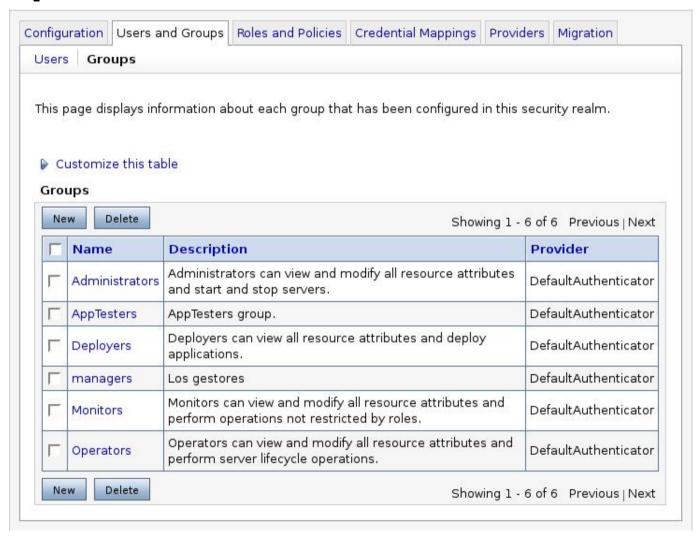
#### **Grupos**

- Agrupación de usuarios con alguna característica común
- Un usuario puede pertenecer a más de un grupo
- Al asignar una política o rol a un grupo, se le asigna a todos los miembros del grupo
- Existen 6 grupos por defecto
  - Administrators: administración del dominio
  - AppTesters: prueban aplicaciones
  - Deployers: despliegue de aplicaciones
  - Monitors: monitorización del sistema
  - Operators: operación con los servidores



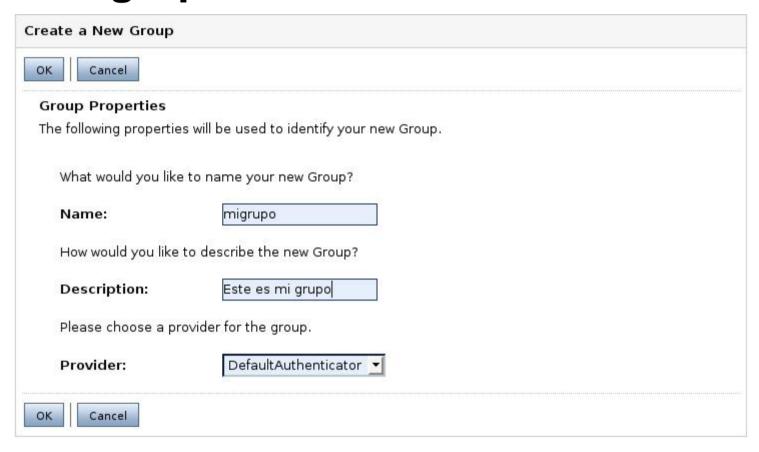


#### Grupos



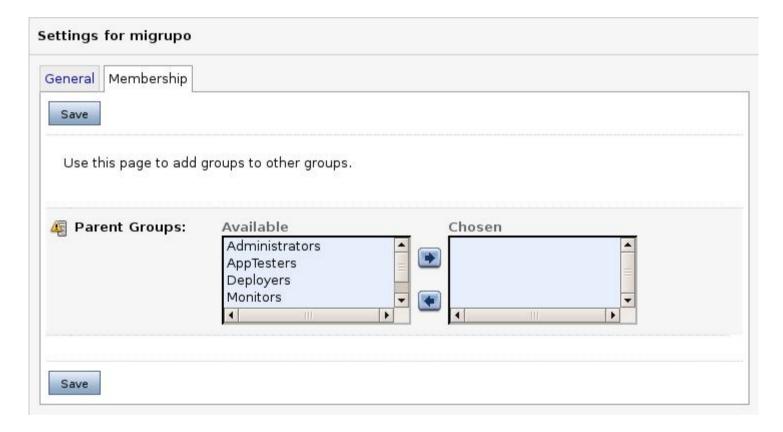


## Nuevo grupo





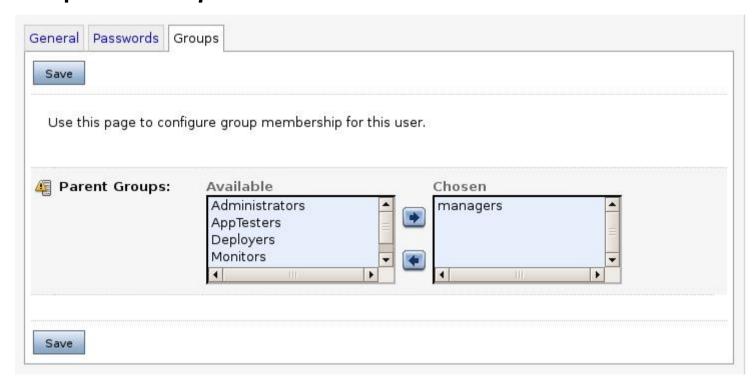
## Herencia del grupo





#### Asignar usuario a grupo

Lista de usuarios -> seleccionamos usuario -> solapa Groups





#### Gestión de roles

- Un rol es una asociación entre usuarios/grupos de manera dinámica: depende de cómo se esté accediendo se tendrá acceso o no (tiempo y forma de acceso, por ejemplo)
- Disponemos de dos tipos de roles: los locales (scoped) que se asignan a un determinado recurso (JDBC, JMS, etc.) y los globales que se suelen asignar a un grupo
- Asociado a un rol nos encontramos con la política de seguridad (policy)



## Gestión de roles: políticas de seguridad

- Una política de seguridad es una asignación con un determinado recurso
- Puede ser simplemente una asignación de un grupo a un recurso (el grupo administrator tiene permiso para acceder a cualquier recurso JDBC) o restricciones mucho más elaboradas (un determinado usuario tiene permiso para acceder al recurso miDS del árbol JNDI desde las 9:00 hasta las 17:00)



#### Descripción de la seguridad en el descriptor

- Disponemos de dos ficheros para definir la seguridad: web.xml y weblogic.xml
- Para restringir el acceso a un recurso
  - Declarar un patrón URL en el fichero web.xml
  - Declarar un rol
  - Asociar el patrón con el rol
  - Mapear el rol con usuarios o grupos en weblogic.xml



#### Web.xml

```
<?xml version="1.0" ?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN" "http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
    <servlet>
            <servlet-name>AbsenceReport</servlet-name>
            <servlet-class>com.servlets.absenceReport</servlet-class>
    </servlet>
    <security-constraint>
            <web-resource-collection>
                         <web-resource-name>timeoff</web-resource-name>
                         <url-pattern>/managers/*</url-pattern>
            </web-resource-collection>
            <auth-constraint>
                         <role-name>mirol</role-name>
            </auth-constraint>
    </security-constraint>
    <login-config>
            <auth-method>BASIC</auth-method>
            <realm-name>default</realm-name>
    <security-role>
            <role-name>mirol</role-name>
    </security-role>
</web-app>
```



## Weblogic.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE weblogic-web-app
 PUBLIC "-//BEA Systems, Inc.//DTD Web Application 8.1//EN"
 "http://www.bea.com/servers/wls810/dtd/weblogic810-web-
  jar.dtd">
<weblogic-web-app>
<security-role-assignment>
  <role-name>mirol</role-name>
  <principal-name>migrupo</principal-name>
</security-role-assignment>
</weblogic-web-app>
```