Autenticación y autorización
Introducción

- Autenticación
  - Proceso para verificar que alguien es realmente quien dice ser.
- Autorización
  - Proceso por el que se permite a alguien hacer o acceder a algo que quiere.

Autenticación y autorización
Tipos de autenticación

- *Basic*

```
<login-config>
    <auth-method>BASIC</auth-method>
    <realm-name>Acceso al curso</realm-name>
</login-config>
```

- *Digest.*
- Formularios .

```
<login-config>
    <auth-method>FORM</auth-method>
    <realm-name>Compras</realm-name>
    <form-login-config>
        <form-login-page>/WEB-INF/seguro/login.jsp</form-login-page>
        <form-error-page>/WEB-INF/seguro/login-error.jsp</form-error-page>
    </form-login-config>
</login-config>
```

- Certificados digitales.

Autenticación y autorización
Usuarios y roles

- Los usuarios de las aplicaciones web en *Tomcat* se identifican con **usuario/password**.
- Los usuario son asignados a roles.
- *Tomcat* concede acceso a las aplicaciones a los roles en lugar de a usuarios individuales.

Autenticación y autorización

Realms

- Un **Realm** es un archivo, base de datos o servicio de directorio que contiene una colección de usuarios/passwords y roles.
- Utilizar **Realms**
  - Interfaz
    - `org.apache.catalina.Realm`
  - Varias implementaciones (tipos de *Realms*).
- Webs
  - http://tomcat.apache.org/tomcat-7.0-doc/realm-howto.html
  - http://tomcat.apache.org/tomcat-7.0-doc/config/realm.html

Autenticación y autorización

Realms

- Elemento **<Realm>** para configurarlos:
  - Dentro de **<Engine>**
    - Será compartido por todas las aplicaciones de todos los virtual host a menos que sea sobrescrito en **<Host>** o **<Context>** subordinado.
  - Dentro de **<Host>**
    - Será compartido por todas las aplicaciones de ese virtual host a menos que sea sobrescrito en un **<Context>** subordinado
  - Dentro de un **<Context>**
    - Solo para esa aplicación.

Autenticación y autorización

Realms

- IMPORTANTE
  - Solo puede existir un **Realm** activo para una aplicación en un momento dado
  - En la configuración inicia del *Tomcat* está definido un **Realm** a nivel de **<Engine>.**

```
<!-- You should set jvmRoute to support load-balancing via AJP ie :
<Engine name="Catalina" defaultHost="localhost" jvmRoute="jvm1">
-->
<Engine name="Catalina" defaultHost="localhost">

  <!--For clustering, please take a look at documentation at:
      /docs/cluster-howto.html  (simple how to)
      /docs/config/cluster.html (reference documentation) -->
  <!--
  <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
  -->

  <!-- Use the LockOutRealm to prevent attempts to guess user passwords
       via a brute-force attack -->
  <Realm className="org.apache.catalina.realm.LockOutRealm">
    <!-- This Realm uses the UserDatabase configured in the global JNDI
         resources under the key "UserDatabase".  Any edits
         that are performed against this UserDatabase are immediately
         available for use by the Realm.  -->
         <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
          resourceName="UserDatabase"/>
  </Realm>
```

Autenticación y autorización
Tipos de Realms

- **MemoryRealm**
  - Acceso a información almacenada en un fichero (normalmente **tomcat-users.xml**).
- **UserDatabaseRealm**
  - Acceso a información almacenada en un fichero (normalmente **tomcat-users.xml**) vía JNDI.
- **JDBCRealm**
  - Acceso a la información de autenticación almacenada en una base de datos relacional a través de un controlador JDBC.

Autenticación y autorización
Tipos de `Realms`

- **`DataSourceRealm`**
  - Acceso a la información de autenticación almacenada en una base de datos relacional a través de JNDI.
- **`JNDIRealm`**
  - Acceso a la información almacenada en un servicio de directorio (LDAP) a través de JNDI.
- **`JaasRealm`**
  - Acceso a un servidor JAAS (*Security authentication using Java Authentication*).

Autenticación y autorización
Tipos de `Realms`

- **`CombinedRealm`**
  - Permite el uso de múltiples *Realms* simultáneamente.
- **`LockOutRealm`**
  - Extiende CombinedRealm para bloquear usuario con varios intentos de *login* fallidos.

Autenticación y autorización

`MemoryRealm`

- La información de usuarios y roles almacenada en un fichero que se carga en memoria al iniciar *Tomcat*.
- Por defecto el fichero es **tomcat-users.xml**.

Autenticación y autorización
MemoryRealm

- Configuración
  - 1) Definir el fichero (por defecto **tomcat-users.xml**) con los usuario y roles.

```
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<role rolename="curso"/>
<user username="tomcat" password="despliegue" roles="manager-gui"/>
<user username="tomcatscript" password="despliegue" roles="manager-script"/>
<user username="alumno" password="alumno" roles="curso"/>
<user username="profesor" password="profesor" roles="curso"/>
```

  - 2) Configurar el `Realm` en el ámbito que se considere más adecuado (`<Engine>`, `<Host>`, `<Context>`, ...)

```
<Context>
    <Realm className="org.apache.catalina.realm.MemoryRealm" />
</Context>
```

Autenticación y autorización
MemoryRealm

- 3) Proteger el recurso (en el descriptor de despliegue **web.xml** de la aplicación).

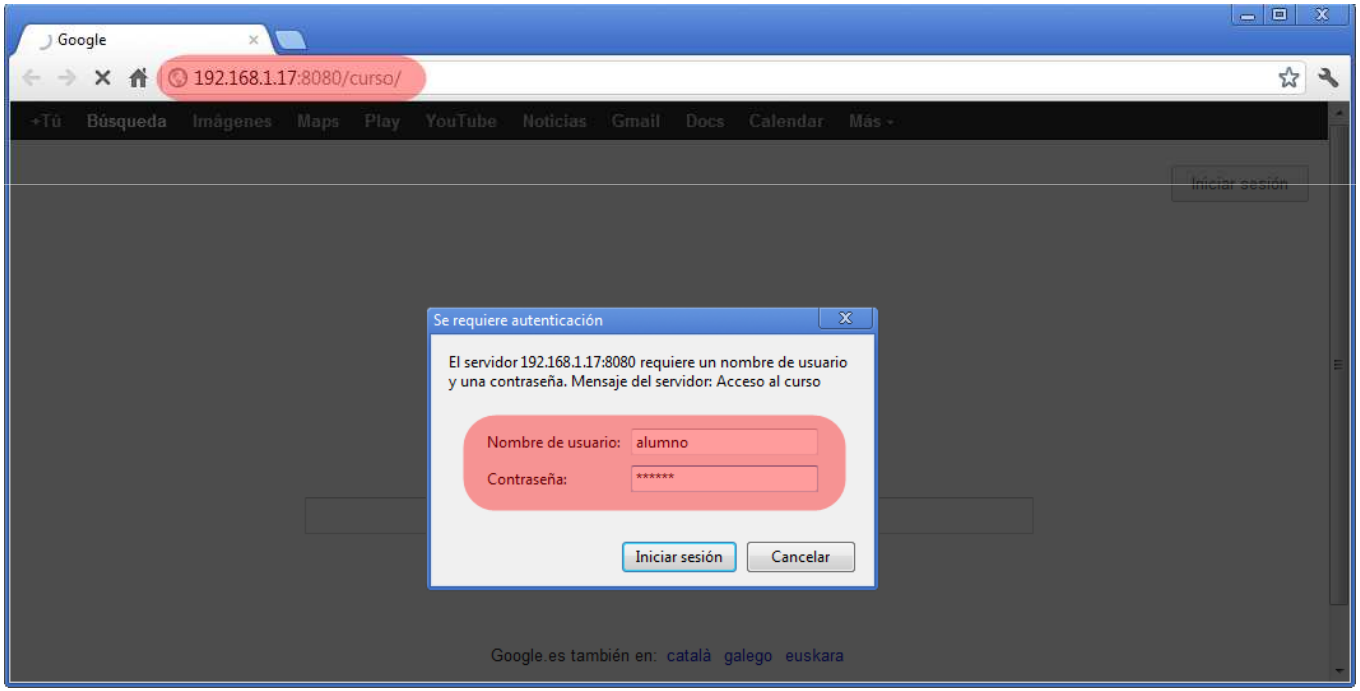
```
<security-constraint>
    <web-resource-collection>
        <web-resource-name>MemoryRealm</web-resource-name>
        <url-pattern>/*</url-pattern>
    </web-resource-collection>
    <auth-constraint>
        <role-name>curso</role-name>
    </auth-constraint>
</security-constraint>
```

- 4) Configurar el tipo autenticación (en el descriptor de despliegue **web.xml** de la aplicación).

```
<login-config>
    <auth-method>BASIC</auth-method>
    <realm-name>Acceso al curso</realm-name>
</login-config>
```

Práctica

- Práctica 7.10
  - Autenticación y autorización (`MemoryRealm`).



Autenticación y autorización

`UserDatabaseRealm`

- Versión avanzadas de **`MemoryRealm`**.
- Es configurable usando JNDI (*Java Naming Directory Interface*)
- La información de usuario y roles almacenada en un fichero que se carga en memoria al iniciar *Tomcat*.
- Por defecto el fichero es **`tomcat-users.xml`.**

Autenticación y autorización
UserDatabaseRealm

- Configuración
  - 1) Definir el fichero (por defecto **tomcat-users.xml**) con los usuario y roles.
  - 2) Configurar un recurso JNDI

```
<!-- Global JNDI resources
     Documentation at /docs/jndi-resources-howto.html
-->
<GlobalNamingResources>
  <!-- Editable user database that can also be used by
       UserDatabaseRealm to authenticate users
  -->
  <Resource name="UserDatabase" auth="Container"
            type="org.apache.catalina.UserDatabase"

            description="User database that can be updated and saved"
            factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
            pathname="conf/tomcat-users.xml" />
</GlobalNamingResources>
```

Autenticación y autorización

UserDatabaseRealm

- 3) Configurar el `Realm` en el ámbito que se considere más adecuado (`<Engine>`, `<Host>`, `<Context>`, ...)

```
<!-- This Realm uses the UserDatabase configured in the global JNDI
     resources under the key "UserDatabase".  Any edits
     that are performed against this UserDatabase are immediately
     available for use by the Realm.  -->
      <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
       resourceName="UserDatabase"/>
```

Autenticación y autorización
UserDatabaseRealm

- 4) Proteger el recurso (en el descriptor de despliegue **`web.xml`** de la aplicación).

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>HTML Manager interface (for humans)</web-resource-name>
    <url-pattern>/html/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
     <role-name>manager-gui</role-name>
  </auth-constraint>
</security-constraint>
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Text Manager interface (for scripts)</web-resource-nam$
    <url-pattern>/text/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
     <role-name>manager-script</role-name>
  </auth-constraint>
</security-constraint>
<security-constraint>
```

Autenticación y autorización

UserDatabaseRealm

- 5) Configurar el tipo autenticación (en el descriptor de despliegue **web.xml** de la aplicación).

```
<!-- Define the Login Configuration for this Application -->
<login-config>
  <auth-method>BASIC</auth-method>
  <realm-name>Tomcat Manager Application</realm-name>
</login-config>
```

Práctica

- Práctica 7.11
  - Autenticación y autorización (`UserDatabaseRealm`).

```
<!-- Global JNDI resources
     Documentation at /docs/jndi-resources-howto.html
-->
<GlobalNamingResources>
  <!-- Editable user database that can also be used by
       UserDatabaseRealm to authenticate users
  -->
  <Resource name="UserDatabase" auth="Container"
            type="org.apache.catalina.UserDatabase"

            description="User database that can be updated and saved"
            factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
            pathname="conf/tomcat-users.xml" />
</GlobalNamingResources>
```

```
<!-- This Realm uses the UserDatabase configured in the global JNDI
     resources under the key "UserDatabase".  Any edits
     that are performed against this UserDatabase are immediately
     available for use by the Realm.  -->
      <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
       resourceName="UserDatabase"/>
```

Autenticación y autorización

`JDBCRealm`

- La información de usuarios y roles es almacenada en una base de datos.
- El acceso a la base de datos se hace utilizando JDBC.

Autenticación y autorización
JDBCRealm

- Configuración
  - 1) Crear la base datos para almacenar los usuarios y roles.

```
use tomcatusers;
create table users
(
    user_name varchar(12) not null primary key,
    user_pass varchar(12) not null
);

create table users_roles
(
    user_name varchar(12) not null,
    role_name varchar(12) not null,
    primary key(user_name, role_name)
);
```

Autenticación y autorización
JDBCRealm

- 2) Insertar los usuarios y roles.

```
insert into users values("mortadelo", "mortadelo");
insert into users values("filemon", "filemont");
insert into user_roles values("mortadelo", "compras");
insert into user_roles values("filemon", "compras");
```

3) Configurar el Realm en el ámbito que se considere más adecuado (<Engine>, <Host>, <Context>, ...)

```
<Context>
    <Realm  className="org.apache.catalina.realm.JDBCRealm"
            driverName="com.mysql.jdbc.Driver"
            connectionURL="jdbc:mysql://localhost/tomcatusers?user=tomcatusers&amp;password=tomcatusers"
            userTable="users"
            userNameCol="user_name" userCredCol="user_pass"
            userRoleTable="users_roles"
            roleNameCol="role_name"
    />
</Context>
```

Autenticación y autorización
JDBCRealm

- 4) Proteger el recurso (en el descriptor de despliegue **`web.xml`** de la aplicación).

```
<security-constraint>
    <web-resource-collection>
        <web-resource-name>JDBCRealm</web-resource-name>
        <url-pattern>/*</url-pattern>
    </web-resource-collection>
    <auth-constraint>
        <role-name>compras</role-name>
    </auth-constraint>
</security-constraint>
```

Autenticación y autorización
JDBCRealm

- 5) Configurar el tipo autenticación (en el descriptor de despliegue **web.xml** de la aplicación) (1)

```
<login-config>
    <auth-method>FORM</auth-method>
    <realm-name>Compras</realm-name>
    <form-login-config>
        <form-login-page>/WEB-INF/seguro/login.jsp</form-login-page>
        <form-error-page>/WEB-INF/seguro/login-error.jsp</form-error-page>
    </form-login-config>
</login-config>
```

Autenticación y autorización
JDBCRealm

- 5) Configurar el tipo autenticación (en el descriptor de despliegue **web.xml** de la aplicación) (2)
  - login.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

    <form method="POST"
        action='<%=response.encodeURL("j_security_check")%>'>
        Usuario:<input type="text" name="j_username">
        Password:<input type="password" name="j_password">
        <input type="submit" value="Login">
    </form>

</body>
</html>
```

Autenticación y autorización
JDBCRealm

- 5) Configurar el tipo autenticación (en el descriptor de despliegue **web.xml** de la aplicación) (3)
  - login-error.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Página de error de login</title>
</head>
<body>

Usuario o password incorrectos, prueba <a href="<%= response.encodeURL("index.jsp") %>"> de nuevo</a>.

</body>
</html>
```

Práctica

- Práctica 7.12
  - Autenticación y autorización (JDBCRealm).

```
use tomcatusers;
create table users
(
    user_name varchar(12) not null primary key,
    user_pass varchar(12) not null
);

create table users_roles
(
    user_name varchar(12) not null,
    role_name varchar(12) not null,
    primary key(user_name, role_name)
);
```

```
insert into users values("mortadelo", "mortadelo");
insert into users values("filemon", "filemont");
insert into user_roles values("mortadelo", "compras");
insert into user_roles values("filemon", "compras");
```

```
<Context>
    <Realm  className="org.apache.catalina.realm.JDBCRealm"
            driverName="com.mysql.jdbc.Driver"
            connectionURL="jdbc:mysql://localhost/tomcatusers?user=tomcatusers&amp;password=tomcatusers"
            userTable="users"
            userNameCol="user_name" userCredCol="user_pass"
            userRoleTable="users_roles"
            roleNameCol="role_name"
    />
</Context>
```

Autenticación y autorización
Acceso a usuarios autenticados

- Cuando un usuario se autentificado es posible acceder a él desde las aplicaciones porque su información se almacena en el objeto HttpServletRequest .

```
<html>
<head>
    <title>Ejemplo</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
    <div class="content">
        <b>Bienvenido, <%= request.getRemoteUser() %></b>
    </div>
</body>
</html>
```