10 Guía Servidor LDAP (conexión servidores)

MATERIAL

- Los contenidos de la unidad y esta guía
- Máquinas Virtuales Ubuntu 22.04 Desktop (1 Servidor DNS + 1 Servidor Web y Tomcat).
- Virtualbox
- Ordenador con S.O. Windows 10.
- Navegador para comprobar la realización de la tarea.
- Procesador de textos para elaborar la documentación y los archivos de la tarea.
- Acceso a Internet.

10.1 Instalación Servidor LDAP

Antes de realizar los dos últimos apartados de la actividad vamos a asegurarnos de que están configurados nuestro directorio LDAP como se nos solicita. Para ello nos vamos apoyaremos en el phpLDAPadmin, por lo que vamos a instalarlo.

Para instalar phpLDAPadmin ejecutamos.

#sudo apt -y install apache2 php php-cgi libapache2-mod-php php-mbstring php-common php-pear

Habilitamos el módulo para integrar php en Apache.

#sudo a2enconf php7.4-cgi

#sudo systemctl reload apache2

Instalamos el phpLDAPadmin.

#sudo apt -y install phpldapadmin

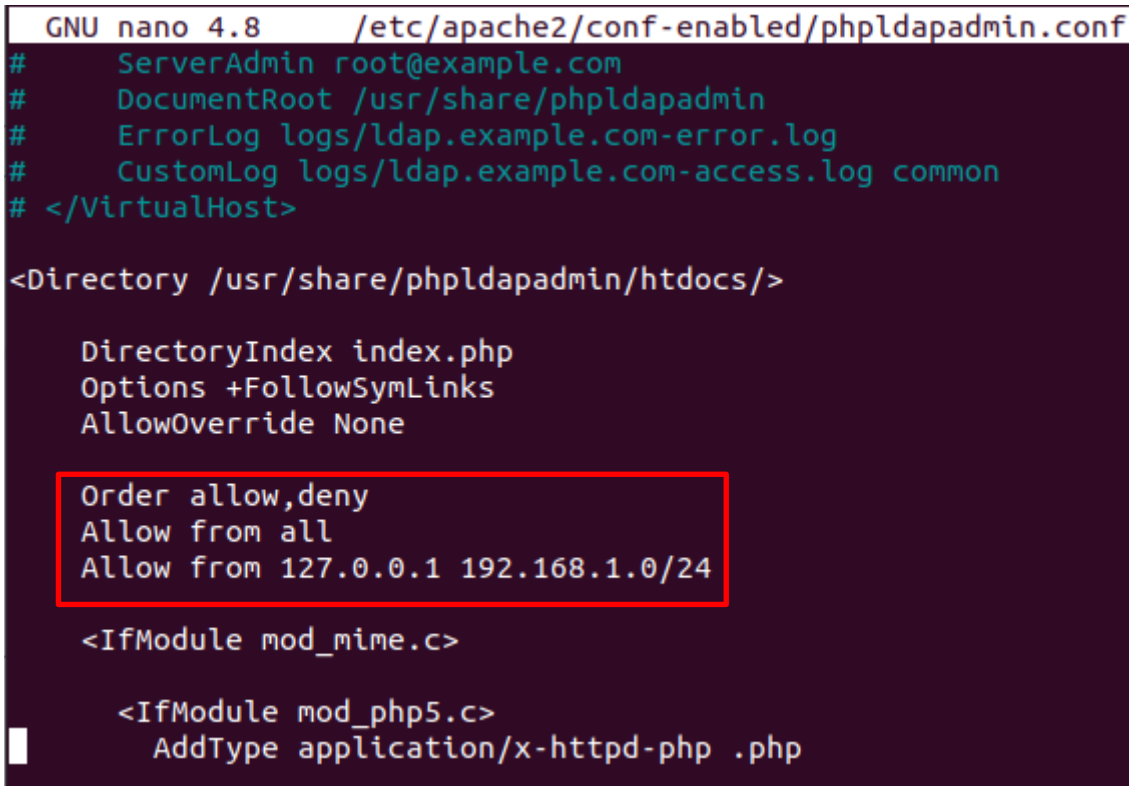
A continuación vamos a modificar los permisos para permitir el acceso sólo de las redes deseadas.

#sudo nano /etc/apache2/conf-enabled/phpldapadmin.conf

donde deberemos de poner lo siguiente o modificar si existe.

Order deny,allow Deny from all Allow from 127.0.0.1 IP_red_servidor_LDAP

Quedaría de esta manera.



Indicar que el servidor LDAP que estoy configurando para hacer esta guía se encuentra en la IP 192.168.1.137, de ahí que ponga en la configuración anterior la IP de red 192.168.1.0/24.

Para que nos cargue cualquier directorio LDAP del dominio que establezcamos en el login comentamos una línea que hace que por defecto vaya a un directorio, el dc=example,dc=net. Para conseguir que vaya a cualquiera de los indiquemos en la pantalla de login editamos config.php en la ruta /etc/phpldapadmin/ y le añadimos a la línea el carácter para comentarlo "#".

Editamos el archivo config.php.

#sudo nano /etc/phpldapadmin/config.php

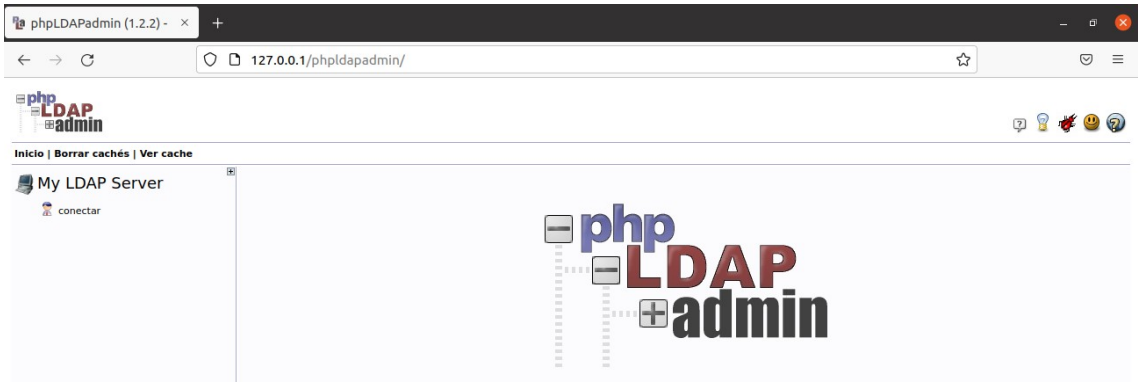
Ahora en el editor nano pulsamos CTRL+w y buscamos, dc=example. En la siguiente captura se refleja resaltada la línea que debemos comentar.

Guardamos, cerramos y reiniciamos el servicio de LDAP (slapd).

#sudo systemctl restart slapd

Con las configuraciones realizadas, ya estamos en disposición de acceder a phpLDAPadmim. Para ello abrimos el navegador y ponemos la IP de la máquina Servidor LDAP o 127.0.0.1, también nos podríamos conectar desde otra máquina de la red, ya que son las IPs que hemos autorizado.

En este caso me conectaré desde el navegador del Servidor LDAP.



Pulsamos en conectar que aparece a la izquierda y nos aparecerá la ventana de autenticación.

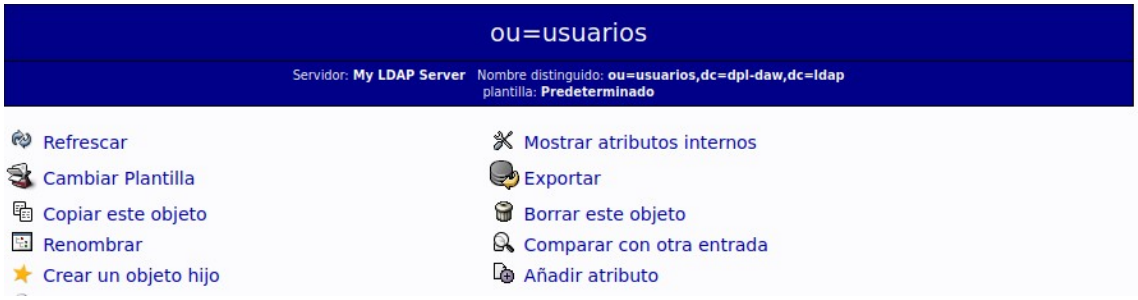
Los datos que tendremos que poner para que nos lleve al directorio que hemos creado son:

- Login: **cn=admin,dc=dpl-daw,dc=ldap**
- Contraseña: Ponemos la que hayamos creado

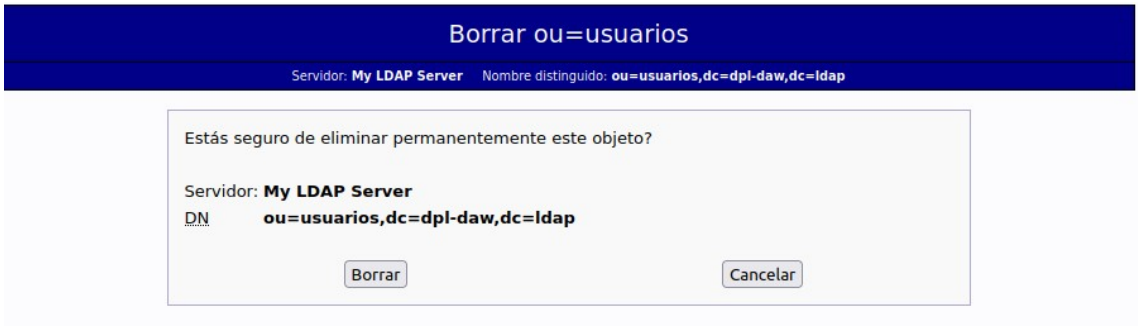
Pulsamos en identificarse y deberíamos de ver la estructura de nuestro directorio. Desplegando junto al + que aparece al lado del icono del mundo.

Como vemos sólo nos aparecen los unidades organizativas grupos y usuarios, por lo que vamos a realizar una operación para que nos quede como se pide.

Pulsamos en una de ellas y en la parte derecha nos saldrán varias opciones como las que se muestran.



Para borrar el objeto pulsamos sobre “Borrar este objeto”.



Nos sale la advertencia indicándonos el objeto que vamos a proceder a borrar. Pulsamos en borrar y lo borraríamos.

Procedemos igual con el otro objeto grupos.

Nos debe quedar la estructura de nuestro directorio como se muestra a continuación:

Para cargar la estructura vamos a modificar el archivo que creamos en la anterior guía content.ldif. En la parte inferior cuando hacemos las relaciones de los usuarios hacia los grupos a los que pertenecen tenemos que poner en vez de grupo, en plural grupos. En la siguiente imagen se muestra corregido.

```
  GNU nano 4.8                          content.ldif
sn: atenea
uid: atenea
userPassword: {SSHA}V53LYZvQFUadVkJ7qU+W+jF5uIFhabj+

dn: cn=romanos,ou=grupos,dc=dpl-daw,dc=ldap
objectClass: groupOfUniqueNames
objectClass: top
cn: romanos
uniqueMember: uid=minerva,ou=usuarios,dc=dpl-daw,dc=ldap
uniqueMember: uid=saturno,ou=usuarios,dc=dpl-daw,dc=ldap

dn: cn=griegos,ou=grupos,dc=dpl-daw,dc=ldap
objectClass: groupOfUniqueNames
objectClass: top
cn: romanos
uniqueMember: uid=atenea,ou=usuarios,dc=dpl-daw,dc=ldap
uniqueMember: uid=cronos,ou=usuarios,dc=dpl-daw,dc=ldap
```

Cargamos el fichero content.ldif al LDAP.

#sudo ldapadd -x -D cn=admin,dc=dpl-daw,dc=ldap -W -f content.ldif

```
informatica@informatica-VirtualBox:~$ sudo ldapadd -x -D cn=admin,dc=dpl-daw,dc=
ldap -W -f content.ldif
[sudo] contraseña para informatica:
Enter LDAP Password:
adding new entry "ou=usuarios,dc=dpl-daw,dc=ldap"

adding new entry "ou=grupos,dc=dpl-daw,dc=ldap"

adding new entry "uid=minerva,ou=usuarios,dc=dpl-daw,dc=ldap"

adding new entry "uid=saturno,ou=usuarios,dc=dpl-daw,dc=ldap"

adding new entry "uid=cronos,ou=usuarios,dc=dpl-daw,dc=ldap"

adding new entry "uid=atenea,ou=usuarios,dc=dpl-daw,dc=ldap"

adding new entry "cn=romanos,ou=grupos,dc=dpl-daw,dc=ldap"

adding new entry "cn=griegos,ou=grupos,dc=dpl-daw,dc=ldap"
```

Si refrescamos en el botón de “Refrescar” del LDAP veremos la estructura que se solicitaba.

4. **Integrar la autenticación del Servidor Web Apache y Tomcat con LDAP.**

   Integración LDAP con Servidor Web Apache

   Habilitamos el módulo authzn_ldap

   #sudo a2enmod authzn_ldap

   Reiniciamos Apache

   #sudo systemctl restart apache2

5. **Crear dos páginas con parte limitada controlada por la autenticación de LDAP, una página con acceso a los usuarios de cada grupo. Mostrar que pueden realizar el acceso los usuarios a la página de su grupo.**

   Configuraremos tres directorios (profesor, griego y romano) de manera que nos solicite las credenciales para acceder a los mismos. Permitirá el acceso al usuario que pertenezca al grupo que hayamos declarado. Para ello habilitaremos las directivas en el archivo de apache2.conf para que solicite la autenticación contra el directorio LDAP para que controle los accesos.

Configuración carpeta profesor

Creamos el directorio profesor en /var/www/html/

#sudo mkdir /var/www/html/profesor

Vamos a introducir un índice que nos indique la carpeta en la que nos encontramos si accedemos a ella.

#sudo nano /var/www/html/profesor/index.html

De igual modo procederemos con las otras dos carpetas romanos y griegos.

#sudo mkdir /var/www/html/romanos

#sudo mkdir /var/www/html/griegos

Editamos un índice en cada uno de ellos donde se diferencia que estamos en una u otra carpeta. Podemos usar como base el que hemos creado en la carpeta profesor y luego lo editamos cambiando tan sólo la identificación profesor por romanos o griegos según corresponda.

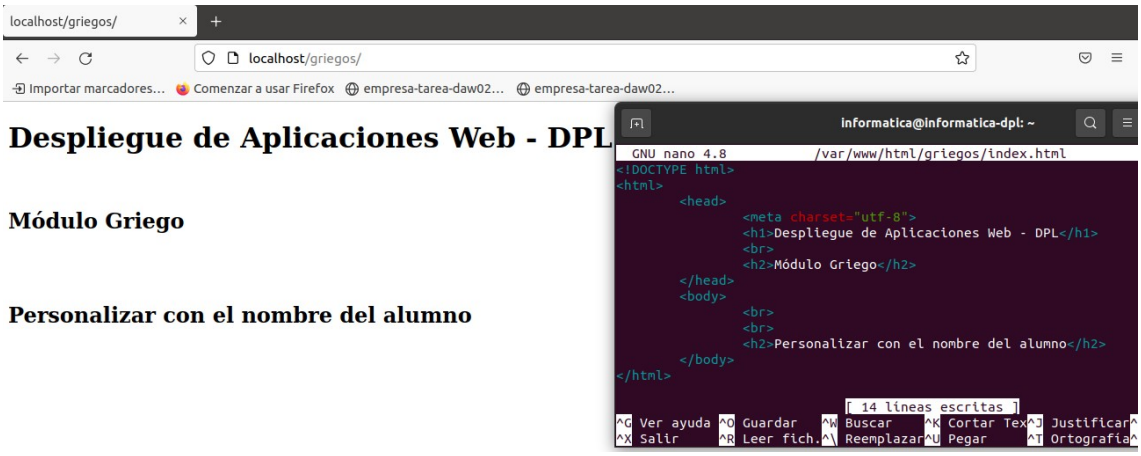
#sudo cp /var/www/html/profesor/index.html /var/www/html/griegos/

#sudo cp /var/www/html/profesor/index.html /var/www/html/romanos/

Ahora editamos los index.html para que cambiar la identificación por el nombre que corresponda según en la carpeta que se encuentre el índice.

Quedando de la siguiente manera:

#sudo nano /var /www/html/griegos/index.html



#sudo nano /var /www/html/romanos/index.html

Ahora mecanizamos las directivas para controlar los accesos a cada carpeta con el LDAP.

Editamos el archivo apache2.conf

#sudo nano /etc/apache2/apache2.conf

Añadimos las siguientes directivas.

Control acceso carpeta profesor:

```
  GNU nano 4.8                                /etc/apache2/apache2.conf
</Directory>
<Directory /var/wwww/todo-empresa-tarea-daw02/delimitado>
        Options Indexes FollowSymLinks Multiviews
        AllowOverride AuthConfig
</Directory>
<Directory /var/www/html/profesor/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        Allow from all
        #--Autenticación LDAP--
        AuthType Basic
        AuthBasicProvider ldap
        AuthName "Introduzca su usuario y password"
        AuthLDAPURL "ldap://192.168.1.137/dc=dpl-daw,dc=ldap?uid?sub?(objectClass=*)"
        AuthLDAPBindDN "cn=admin,dc=dpl-daw,dc=ldap"
        AuthLDAPBindPassword 1234
        Require ldap-group cn=profesores,ou=grupos,dc=dpl-daw,dc=ldap
</Directory>
```

Control acceso carpeta romanos:

```
  GNU nano 4.8                                /etc/apache2/apache2.conf
<Directory /var/www/html/romanos/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        Allow from all
        #--Autenticación LDAP--
        AuthType Basic
        AuthBasicProvider ldap
        AuthName "Introduzca su usuario y password"
        AuthLDAPURL "ldap://192.168.1.137/dc=dpl-daw,dc=ldap?uid?sub?(objectClass=*)"
        AuthLDAPBindDN "cn=admin,dc=dpl-daw,dc=ldap"
        AuthLDAPBindPassword 1234
        Require ldap-group cn=romanos,ou=grupos,dc=dpl-daw,dc=ldap
</Directory>
```

Control acceso carpeta griegos:

```
 GNU nano 4.8                                /etc/apache2/apache2.conf
        AuthLDAPBindPassword 1234
        Require ldap-group cn=romanos,ou=grupos,dc=dpl-daw,dc=ldap
</Directory>
<Directory /var/www/html/griegos/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        Allow from all
        #--Autenticación LDAP--
        AuthType Basic
        AuthBasicProvider ldap
        AuthName "Introduzca su usuario y password"
        AuthLDAPURL "ldap://192.168.1.137/dc=dpl-daw,dc=ldap?uid?sub?(objectClass=*)'
        AuthLDAPBindDN "cn=admin,dc=dpl-daw,dc=ldap"
        AuthLDAPBindPassword 1234
        Require ldap-group cn=griegos,ou=grupos,dc=dpl-daw,dc=ldap
</Directory>
```

Guardamos, cerramos y reiniciamos el servicio.

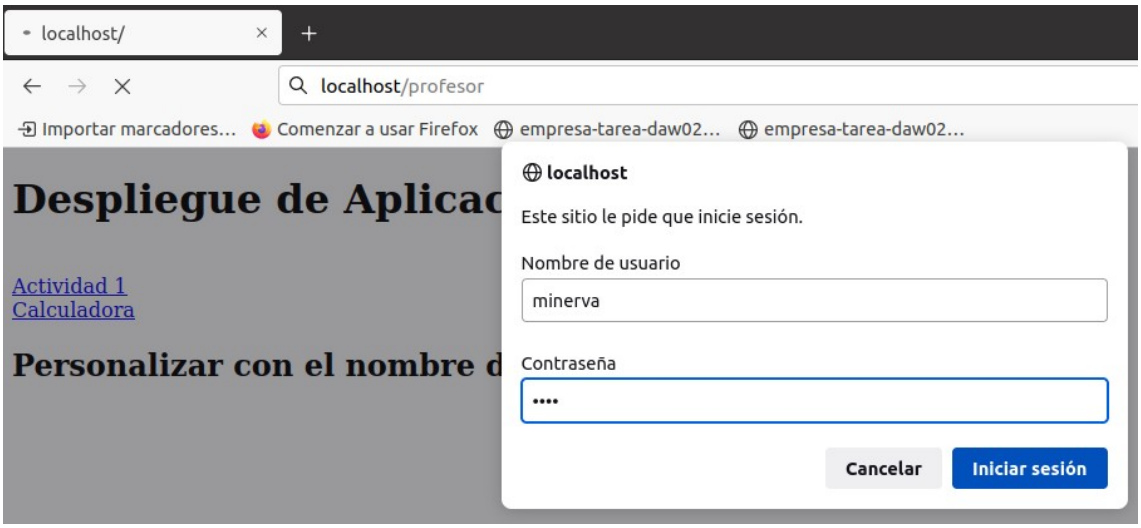
#sudo systemctl restart apache2

Cabe destacar que las directivas son las mismas para los tres casos, teniendo en cuenta la variaciones de las carpetas a controlar, así como el grupo al que queremos permitir que pueda acceder a las mismas.

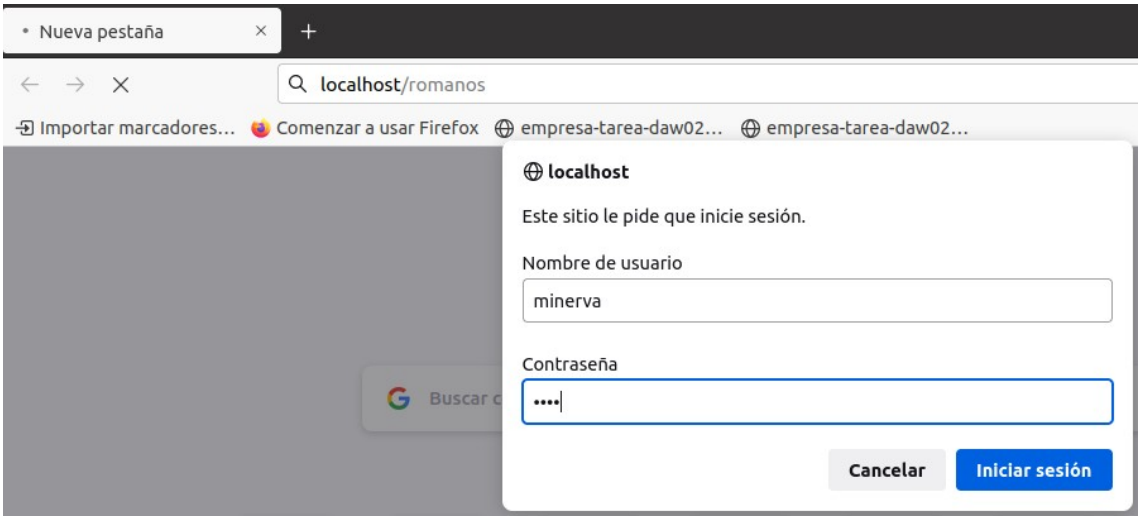
Indicar que en AuthLDAPBindPassword se ha puesto la contraseña de administrador de LDAP.

Comprobamos que nos permite autenticarnos con los usuarios en función de los grupos configurados para cada una de las páginas.

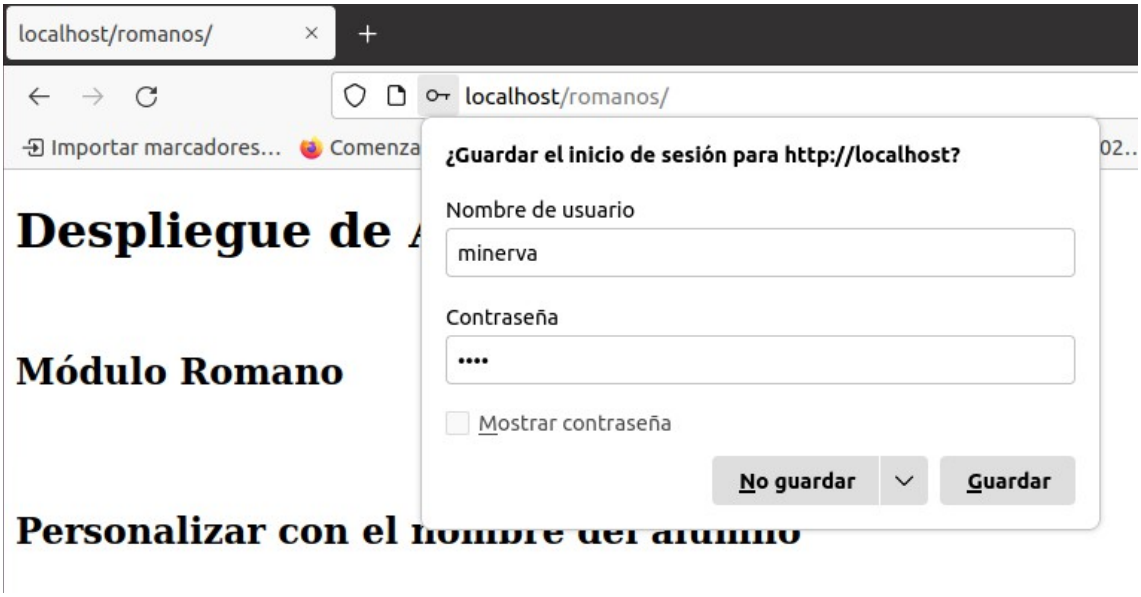
Con el usuario minerva si intentamos acceder a la carpeta profesor nos vuelve a pedir las credenciales.



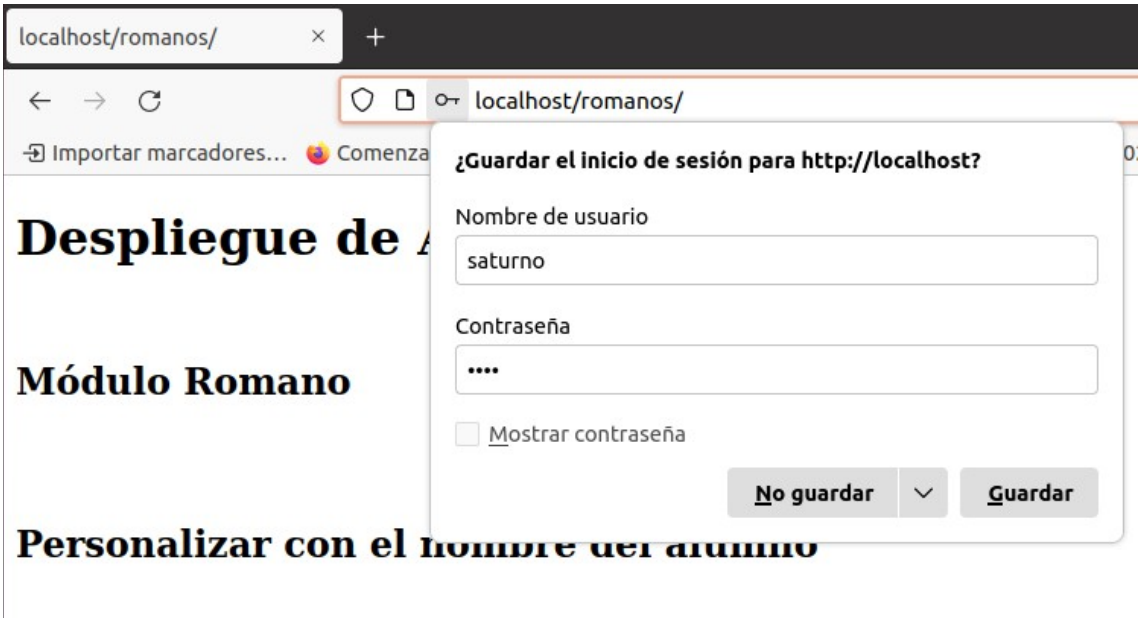
Ahora probamos con la carpeta romanos.



Accedemos, e incluso nos pide que si queremos guardar la contraseña. Hemos limpiado el historial, cerrado el navegador y entrado de nuevo.

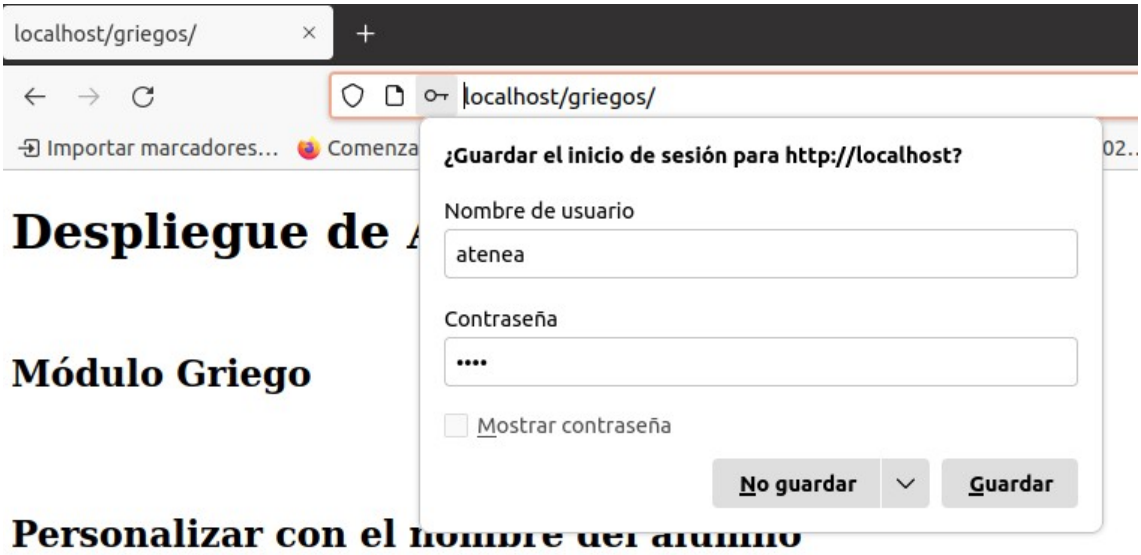


Probamos con el usuario saturno en la carpeta romanos. Tenemos que limpiar el historial, cerrar y entrar al navegador nuevamente. Aquí se muestra la captura donde nos pregunta si queremos guardar las credenciales y de fondo vemos la página de romanos.



Intentamos acceder a la carpeta griegos con uno de los usuarios anteriores. Repetimos los pasos de limpiar el historial y reiniciar el navegador. En este caso nos vuelve a pedir las credenciales.

Probamos con los usuarios del grupo griegos en su carpeta. Para atenea:



En el caso de cronos al no disponer de contraseña no nos deja acceder.

Probamos con el usuario atenea a las otras dos carpetas y no nos deja, nos vuelve a solicitar las credenciales.

Integración con Tomcat

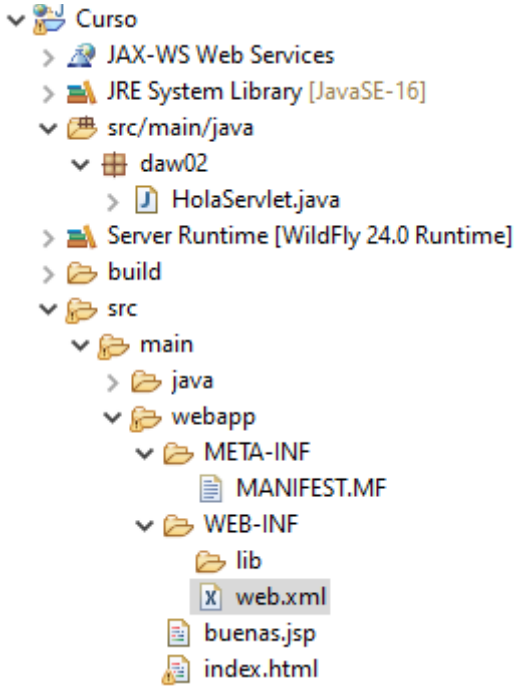
Este punto es opcional hacerlo, ya que hay que investigar un poco. Existen varias formas de implementar la integración de Tomcat. Como control de la parte de administración del servicio o como control de acceso a algunas aplicaciones web.

Para la integración de LDAP en Tomcat vamos a crear una aplicación web en Eclipse que llamaremos Curso. Crearemos los siguientes componentes:

- index.html
- HolaServlet.java
- buenas.jsp
- Descriptor de despliegue web.xml

Recuerda activar el tick para la creación del descriptor de despliegue antes de pulsar en Finish.

Nos debe quedar la siguiente estructura para nuestra aplicación web.



El contenido de los archivos es el siguiente:

index.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transactional//EN" "http://www.w3.org/TR/html4/
loose.dtd"
<html>
<head>
<meta http-equiv="Content-Type" content="text/html"; charset="ISO-8859-1">
<title>DPL</title>
</head>
<body>
        <h1>Curso Despliegue de Aplicaciones WEB - DPL</h1>
        <ul>
        <li><h2>
                <a href="HolaServlet">Hola (Servlet)</a>
                </h2>
        <li><h2>
                <a href="buenas.jsp">Buenas (JSP)</a>
                </h2>
        </ul>
</body>
</html>
```

HolaServlet.java

```
package daw;

import java.io.*;
import javax.servlet.ServletException;
```

```
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/HolaServlet")
public class HolaServlet extends HttpServlet {
	private static final long serialVersionUID = 1L;

    public HolaServlet() {
        super();
    }

	protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

		response.setContentType("text/html");
		PrintWriter out = response.getWriter();
		try {
			out.println("<html>");
			out.println("<head><title>Hola, alumno</title></head>");
			out.println("<body>");
			out.println("<h1>Hola, alumno!</h1>");
			out.println("<p>URL: " + request.getRequestURI() +
"</p>");
			out.println("<p>Protocolo: " + request.getProtocol() +
"</p>");
			out.println("<p>IP del cliente: " +
request.getRemoteAddr() + "</p>");
			out.println("<body></html>");
		}finally {
			out.close();
		}

	}
}
```

buenas.jsp

```
<%@ page language="java" contentType="text/html; charset=utf-8"
    pageEncoding="utf-8"%>
<%@
	page import="java.util.Calendar"%>
<!DOCTYPE html
	PUBLIC ".//W3C//DTD HTML 4.01 Trasitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
```

```
<head>
<meta http-equiv="Content-Type" content="text/html"; charset="utf-8">
<title>Bienvenida</title>
</head>
<body>
        <div class="content">
                <b>Bienvenido al curso</b>
                <p>
                        Hoy es
                        <%=Calendar.getInstance().getTime()%>
                </p>
                <%
                        String saludo;
                        int horaDelDia = Calendar.getInstance().get(Calendar.HOUR_OF_DAY);
                        if (horaDelDia<12){
                                saludo = "Buenos días!";

                        }else if (horaDelDia >= 12 && horaDelDia < 21){
                                saludo= "Buenas tardes!";
                        }else{
                                saludo= "Buenas noches!";
                        }
                %>
                <p><%=saludo%></p>
        </div>
</body>
</html>
```

web.xml (descriptor de despliegue)

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/
xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/
xml/ns/javaee/web-app_4_0.xsd" id="WebApp_ID" version="4.0">
  <display-name>Cursos</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.jsp</welcome-file>
    <welcome-file>default.htm</welcome-file>
  </welcome-file-list>
</web-app>
```

Una vez tenemos la aplicación implementada vamos a realizar los pasos para que nos permita tener el control de acceso por medio de LDAP.

Debemos de crear en la carpeta META-INF de la aplicación Curso el archivo **context.xml**. Y añadir lo siguiente.

context.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<context>
	<Realm className="org.apache.catalina.realm.JNDIRealm"
	connectionURL="ldap://ip_servidor_LDAP:389"
	userPattern="uid={0},ou=usuarios,dc=dpl-daw,dc=ldap"
	roleBase="ou=grupos,dc=dpl-daw,dc=ldap"
	roleName="cn"
	roleSearch="(uniqueMember={0})"
	connectionPassword="1234"
	/>
</context>
```

Personalizar en connectionURL con la IP del servidor LDAP.

Editamos el descriptor de despliegue de la aplicación y le añadimos el siguiente fragmento de código. Lo intercalamos entre las etiquetas del web.xml de la aplicación:

</welcome-file-list>

//introducir aquí el código

</web-app>

Código para introducir en web.xml de la aplicación.

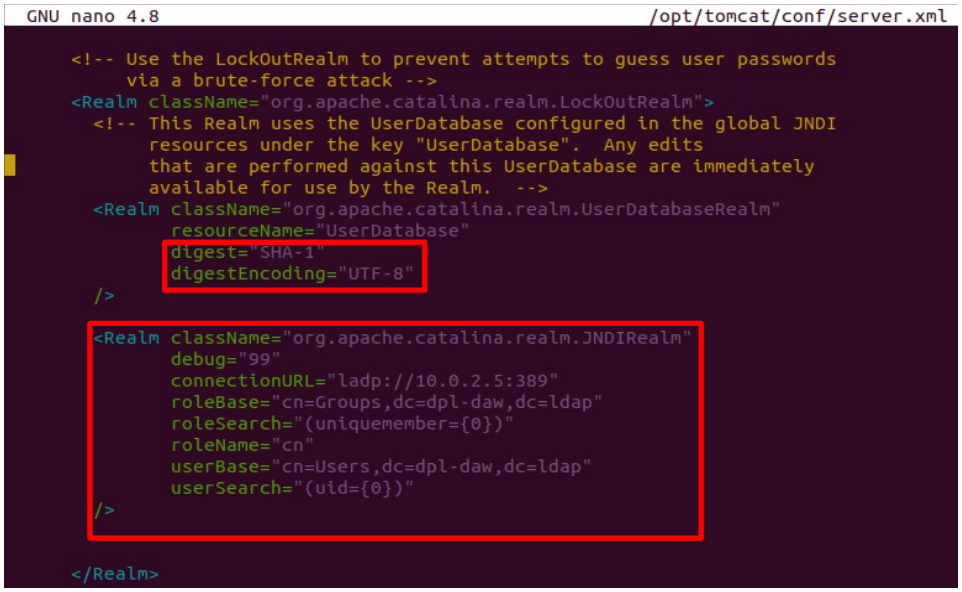
```
<security-constraint>
	<web-resource-collection>
		<web-resource-name>JNDIRealm</web-resource-name>
		<url-pattern>/*</url-pattern>
	</web-resource-collection>
	<auth-constraint>
		<role-name>griegos</role-name>
	</auth-constraint>
	<user-data-constraint>
		<transport-guarantee>CONFIDENTIAL</transport-guarantee>
	</user-data-constraint>
  <servlet>
	<servlet-name>HolaServlet</servlet-name>
	<servlet-class>daw</servlet-class>
  </servlet>
  </security-constraint>
```

Después de configurar el código procedemos a exportar la aplicación a web archive (curso.war) y la desplegamos en el Tomcat.

Tenemos que habilitar en el Tomcat la autenticación con el Realm JNDI. Para conseguirlo editamos el archivo de configuración server.xml.

#sudo nano /opt/tomcat/conf/server.xml

Quedando como se muestra:



Si sacamos el segundo recuadro fuera de la etiqueta </Realm>, ya no funcionará la autenticación que teníamos registrada en el archivo tomcat-users.xml, lo que no nos reconoce las credenciales del LDAP. Esta parte habría que estudiarla para corregirla. Habría que ver como asignar roles en el directorio LDAP para que esté en sintonía con Tomcat el LDAP.

En los siguientes enlaces se recogen una serie de aplicaciones que permiten realizar la autenticación contra LDAP:

- Aplicación para WildFly-Jboss y Tomcat.

  https://github.com/First8/jboss-ldap-authentication

- Aplicación para Tomcat.

  https://geronimo.apache.org/GMOxDOC20/ldap-sample-application.html

Y en este último enlace se explica el funcionamiento de

https://tomcat.apache.org/tomcat-7.0-doc/realm-howto.html