# DAW Práctica 3.5: Mecanismos de seguridad en Apache Tomcat

Captura de pantalla de computadora

Descripción generada automáticamente

**Utilidad del archivo tomcat-users.xml**

* Este archivo contiene la configuración de usuarios y roles para acceder a distintas funciones de Tomcat, como la administración a través de la interfaz gráfica (manager-gui).

Texto

Descripción generada automáticamente

**Roles en tomcat-users.xml:**

* manager-gui: Permite acceder a la interfaz gráfica de administración.
* manager-script: Habilita la gestión de aplicaciones mediante scripts.
* manager-jmx: Permite acceso a JMX para la monitorización.
* manager-status: Ofrece acceso al estado del servidor.

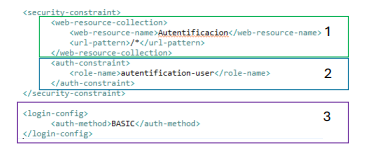
Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Texto

Descripción generada automáticamente

Implementaremos la necesidad de autentificación para una de nuestras aplicaciones. Para ello, modificaremos su archivo web.xml desde Eclipse añadiendo las siguientes instrucciones:



Una vez configurada la aplicación nos quedaría configurar los usuarios dentro del servidor. Para ello deberás modificar el archivo tomcat-users.xml añadiendo el siguiente contenido.



Captura de pantalla de computadora

Descripción generada automáticamente

Forma

Descripción generada automáticamente

Una captura de pantalla de una computadora

Descripción generada automáticamente

Forma

Descripción generada automáticamente con confianza baja

Sí, durante las últimas prácticas, he utilizado algunos de los componentes principales de Apache Tomcat, tales como:

1. **Manager App**:
   * Configuraste el acceso al **Manager App** de Tomcat. Esto implica el uso de roles y usuarios configurados en el archivo tomcat-users.xml. Este componente fue utilizado para gestionar las aplicaciones desplegadas en Tomcat y verificar el estado del servidor.
2. **Catalina** y **Coyote**:
   * Estos componentes están involucrados en el proceso de gestión de solicitudes HTTP/HTTPS. Estás configurando Tomcat para que utilice HTTPS, lo que implica que **Coyote** se encarga de manejar las solicitudes HTTPS.
3. **Realm**:
   * Usaste el componente **Realm** para la autenticación de usuarios y roles en el archivo tomcat-users.xml. Esto fue esencial para proteger la aplicación de **Manager** y garantizar que solo los usuarios autorizados pudieran acceder.
4. **JSP y Servlets**:
   * Aunque no se menciona explícitamente en las instrucciones que hayas desarrollado una aplicación basada en **JSP** o **Servlets**, el proceso de configuración de seguridad de las aplicaciones web (en web.xml) y la implementación de autenticación sobre una aplicación específica implica que el contenedor de **Servlets** y **JSP** de Tomcat está involucrado en el manejo de la lógica de la aplicación.