

Estándares  
de desarrollo de aplicaciones del GOIB  
**Guía de configuración del entorno de  
desarrollo**



G CONSELLERIA  
O ADMINISTRACIONS  
I PÚBLIQUES I  
B MODERNITZACIÓ  
/ DIRECCIÓ GENERAL  
MODERNITZACIÓ I  
ADMINISTRACIÓ DIGITAL

Palma, septiembre de 2022

## Índice

<b>HISTORIAL DE VERSIONES.....</b>	<b>3</b>
<b>1. INTRODUCCIÓN.....</b>	<b>4</b>
<b>2. OPENJDK 11.....</b>	<b>5</b>
2.1. Instalación.....	5
<b>3. APACHE MAVEN.....</b>	<b>6</b>
<b>4. JBOSS EAP 7.2.....</b>	<b>8</b>
4.1. Instalación.....	8
4.2. Configuración de los datasources.....	10
4.3. Cambios importantes respecto a la versión EAP 5.2.....	13
<b>5. KEYCLOAK 6.0.1.....</b>	<b>14</b>
5.1. Instalación.....	14
5.2. Configuración.....	15
<b>6. CONEXIÓN JBOSS - KEYCLOAK.....</b>	<b>22</b>
6.1. Instalación del conector.....	22
6.2. Configuración.....	22
<b>7. ERRORES COMUNES.....</b>	<b>28</b>
7.1. Versión incorrecta de JDK.....	28
7.2. Error de autenticación.....	28
7.3. Activación de servicios Jakarta EE adicionales.....	28
7.4. Contexto de solo lectura.....	29

## Historial de versiones

Fecha	Versión	Descripción	Autor
15/01/20	9.0	Primera versión	DGMAD
28/02/20	9.1	Corrección de errores.	DGMAD
03/12/20	9.5	Traducción al castellano	DGMAD
08/09/22	9.6	Corrección error datasource	DGMAD



# 1. INTRODUCCIÓN

Los actuales estándares de desarrollo del *Gobierno de las Illes Balears* (GOIB) se basan en cuatro componentes básicos:

- **OpenJDK 11** como plataforma de desarrollo (en sustitución de Java SE 7).
- **Maven 3.6** como herramienta para la gestión y construcción de proyectos Jakarta EE.
- **JBoss EAP 7.2** como servidor de aplicaciones (en sustitución de JBoss EAP 5.2).
- **Keycloak 6.0.1** como sistema de administración de identidades (en sustitución de Seycon).

La finalidad de este documento es proporcionar una guía de configuración del entorno de desarrollo necesario para usar los estándares de desarrollo del GOIB, especialmente para configurar estos componentes.

Este guía es una ayuda para el desarrollador, su uso no es obligatorio.

**Para el correcto seguimiento de esta guía es necesario tener permisos de administrador local sobre la máquina.** Además, es recomendable crear un directorio de trabajo en el disco local donde almacenar el software, las librerías externas, y los proyectos de desarrollo.

Los ejemplos y capturas de pantalla descritos corresponden a un sistema operativo Windows 10 de 64 bits. Si se quiere, se puede obtener una versión para Unix / Linux del todo el software mediante el gestor de paquetes de la distribución.

La guía muestra el proceso de instalación y configuración desde el terminal. Una vez instalados y configurados los servicios descritos, si se quiere, se puede configurar el entorno de desarrollo integrado (Eclipse, IntelliJ, Netbeans,..) para facilitar el despliegue de aplicaciones.

## Nota:

**Este documento es una traducción del documento original en catalán. En caso de conflicto prevalecerá la versión catalana del mismo.**

## 2. OpenJDK 11

OpenJDK 11 es la versión libre de la plataforma de desarrollo Java SE Development Kit 11 (JDK 11). No debe confundirse con la plataforma de ejecución Java Runtime Environment (JRE). En esta guía supondremos que ya se tiene instalado el entorno de ejecución JRE.

### 2.1. Instalación

El proceso de instalación es el siguiente:

1. Acceder a la dirección <https://jdk.java.net/java-se-ri/11><sup>1</sup> y escoger entre versión Linux/x64 o Windows/x64 (en esta guía utilizaremos Windows 10).
2. Descargar el fichero **openjdk-11+28\_windows-x64\_bin.zip**.
3. Descomprimir el fichero en el directorio **C:\Program Files\Java**.
4. Establecer la variable de entorno de sistema **JAVA\_HOME** con el valor **C:\Program Files\Java\jdk-11**.
5. Añadir el valor **%JAVA\_HOME%\bin** a la variable de entorno **PATH**.
6. Abrir un terminal y verificar que está configurada la versión 11 de JDK con el comando **java -version**.

```
C:\Desarrollo>java -version
openjdk version "11" 2018-09-25
OpenJDK Runtime Environment 18.9 (build 11+28)
OpenJDK 64-Bit Server VM 18.9 (build 11+28, mixed mode)
```

---

<sup>1</sup> Esta es una implementación de referencia de OpenJDK 11. Las versiones actualizadas se pueden descargar desde la web de Oracle (requiere licencia) o bien desde otras entidades que mantienen builds gratuitas y actualizadas de OpenJDK 11 (como por ejemplo las mantenidas por AdoptOpenJDK).

### 3. Apache Maven

Apache MAVEN es una herramienta de software para la gestión y construcción de proyectos Java. Es similar en funcionalidad a Apache Ant pero tiene un modelo de configuración de construcción más simple basado en un formato XML. Según los estándares de desarrollo del GOIB todos los nuevos proyectos de desarrollo tienen que utilizar MAVEN.

El proceso de instalación y configuración es el siguiente:

1. Descargar la versión 3.6.X desde la dirección <http://maven.apache.org/download.html>.
2. Descomprimir el fichero **apache-maven-3.6.x-bin.zip** en el directorio de trabajo.
3. Establecer la variable de entorno de sistema<sup>2</sup> **M2\_HOME** con el valor **C:\Desarrollo\apache-maven-3.6.X**, substituyendo X por el número de revisión.
4. Añadir el valor **%M2\_HOME%\bin** a la variable de entorno de sistema **Path**.
5. Abrir un terminal y verificar que está configurada la versión 3.6.x de MAVEN con el comando **mvn -version**.

```
C:\Desarrollo>mvn -version
Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)
Maven home: C:\Desarrollo\apache-maven-3.6.3\bin\..
Java version: 11, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-11
Default locale: es_ES, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"
```

6. Crear, dentro del directorio de trabajo, el directorio donde se almacenaran las dependencias MAVEN de todos los proyectos (en nuestro caso, **C:\Desarrollo\repo-maven\**).
7. Crear el fichero de propiedades MAVEN **settings.xml** del usuario en el directorio home de éste (en nuestro caso, **C:\Users\XXX\.m2\settings.xml**).
8. Añadir el siguiente contenido al fichero de propiedades **settings.xml** creado, indicando el repositorio local (**localRepository**) que acabamos de

---

<sup>2</sup> En Windows, como administrador local de la máquina, hay que ir a la opción «Panel de control > Sistema y seguridad > Sistema > Configuración avanzada del sistema».



crear, y el usuario uXXX y contraseña YYY del **proxy** si se está dentro de la red del GOIB.

```
<settings>
<localRepository>C:\Desarrollo\repo-maven</localRepository>

<proxies>
  <proxy>
    <id>caibproxy</id>
    <active>>true</active>
    <protocol>http</protocol>
    <username>uXXX</username>
    <password>YYY</password>
    <host>rproxy1.caib.es</host>
    <port>3128</port>
    <nonProxyHosts>localhost</nonProxyHosts>
  </proxy>
</proxies>
</settings>
```

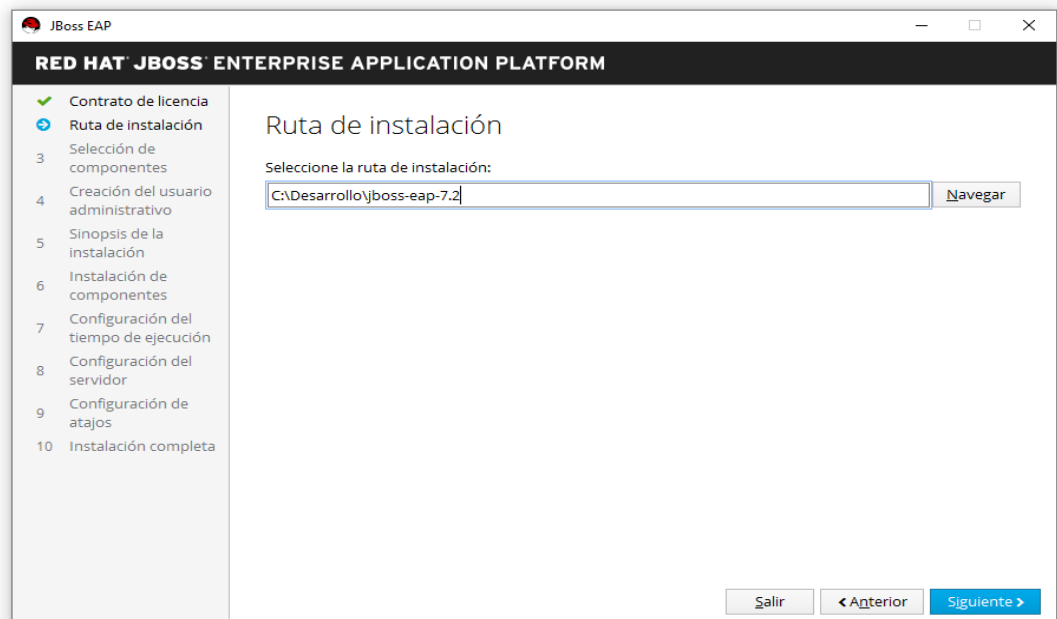
## 4. JBoss EAP 7.2

Red Hat JBoss Enterprise Application Platform 7.2 (JBoss EAP 7.2) es una implementación certificada de las especificaciones completas y del perfil web de Jakarta Enterprise Edition 8 (Jakarta EE 8, antes Java EE). Aunque existe una versión libre de esta implementación llamada WildFly, en la DGMAD se utiliza JBoss EAP por motivos de soporte y rendimiento.

### 4.1. Instalación

El proceso de instalación es el siguiente:

1. Acceder a la sección de descargas de la página oficial de JBoss <https://developers.redhat.com/products/eap/download/>.
2. Descargar y ejecutar el instalador de la versión 7.2.0<sup>34</sup> (fichero **jboss-eap-7.2.0-installer.jar**).
3. Especificar el directorio de instalación del JBoss (en nuestro caso, **C:\Desarrollo\jboss-eap-7.2**).



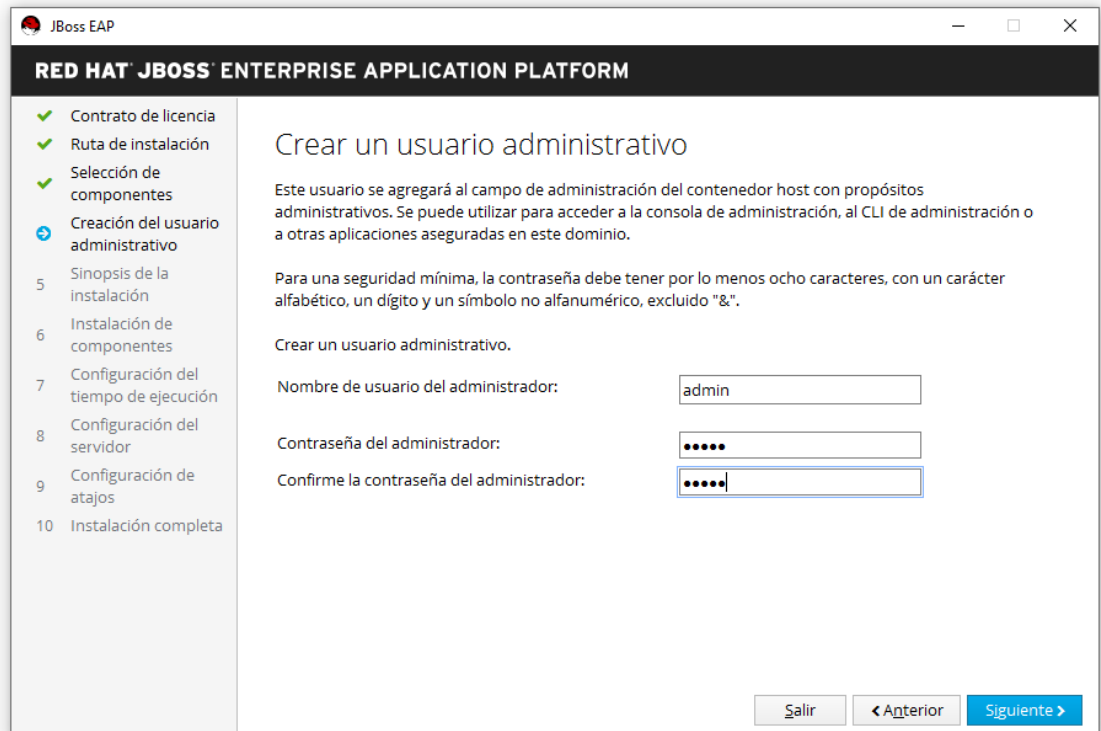
3 Es necesario registrarse previamente en la página de RedHat con una cuenta gratuita.

4 Las releases notes con los bugs solucionados en la versión 7.2 están publicados en la dirección

[https://access.redhat.com/documentation/en-us/red\\_hat\\_jboss\\_enterprise\\_application\\_platform/7.2/](https://access.redhat.com/documentation/en-us/red_hat_jboss_enterprise_application_platform/7.2/). Para evitar errores, se recomienda aplicar el último parche publicado.



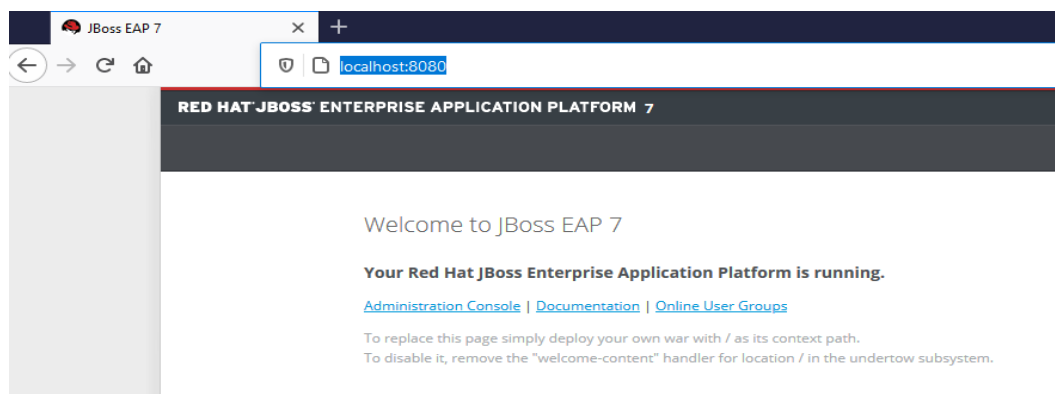
4. Dar de alta el usuario administrador del JBoss<sup>5</sup> (por ejemplo: **admin**).



5. Establecer la variable de entorno de sistema **JBOSS\_HOME** con el valor del directorio de instalación del JBoss (en nuestro caso, **C:\Desarrollo\jboss-eap-7.2**).

6. Iniciar el servidor ejecutando el script **JBOSS\_HOME\bin\standalone.bat**.

7. Acceder a la página principal de JBoss desde la dirección <http://localhost:8080/> para comprobar que funciona correctamente.



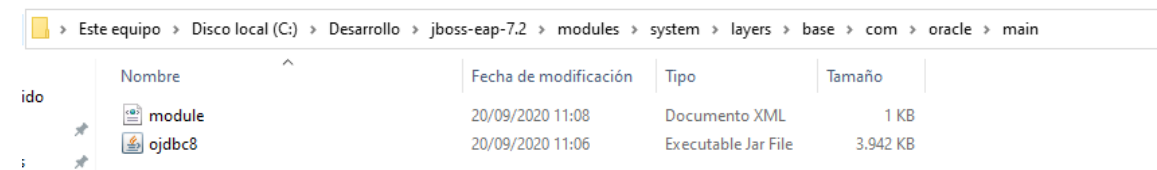
5 Alternativamente, este usuario se puede crear posteriormente con el script **JBOSS\_HOME\bin\add-user**.

## 4.2. Configuración de los datasources

A continuación se describe el proceso de configuración de datasources para sistemas gestores de base de datos Oracle y PostgreSQL según los estándares de base de datos del GOIB (para más información ver documento «Estándares de base de datos»).

### Oracle:

1. Crear el directorio **JBOSS\_HOME\modules\system\layers\base\com\oracle\main**.



Nombre	Fecha de modificación	Tipo	Tamaño
module	20/09/2020 11:08	Documento XML	1 KB
ojdbc8	20/09/2020 11:06	Executable Jar File	3.942 KB

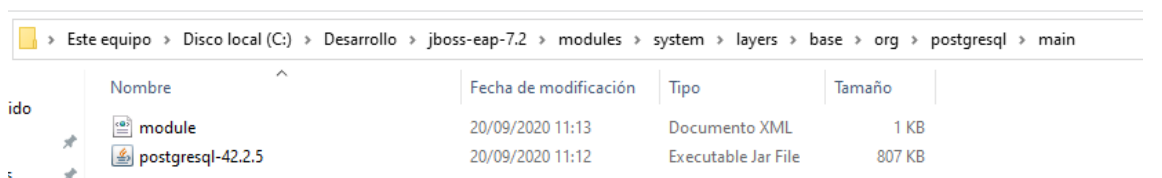
2. Descargar el fichero **ojdbc8.jar** desde la dirección web <https://www.oracle.com/technetwork/database/features/jdbc/jdbc-ucp-122-3110062.html><sup>6</sup> y copiarlo dentro del directorio anterior.

3. Añadir el fichero **module.xml** al directorio anterior con el siguiente contenido:

```
<module xmlns="urn:jboss:module:1.0" name="com.oracle">
  <resources>
    <resource-root path="ojdbc8.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

### PostgreSQL:

1. Crear el directorio **JBOSS\_HOME\modules\system\layers\base\org\postgresql\main**.



Nombre	Fecha de modificación	Tipo	Tamaño
module	20/09/2020 11:13	Documento XML	1 KB
postgresql-42.2.5	20/09/2020 11:12	Executable Jar File	807 KB

<sup>6</sup> Se han de aceptar los términos y condiciones y tener una cuenta de Oracle.

2. Descargar el fichero **postgresql-42.2.5.jar** en el directorio anterior desde la dirección <https://jdbc.postgresql.org/download.html><sup>7</sup> y copiarlo dentro del directorio anterior.
3. Añadir el fichero **module.xml** en el directorio anterior con el siguiente contenido:

```
<module xmlns="urn:jboss:module:1.0" name="org.postgresql">
  <resources>
    <resource-root path="postgresql-42.2.5.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

### Oracle y PostgreSQL:

1. Añadir la siguiente configuración de conectores (*drivers*) en el fichero **JBOSS\_HOME\standalone\configuration\standalone.xml**.

```
<datasources>
  ...
  <drivers>
    <driver name="h2" module="com.h2database.h2">
      <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>
    </driver>
    <!-- GOIB drivers -->
    <driver name="oracle" module="com.oracle">
      <xa-datasource-class> oracle.jdbc.xa.client.OracleXADataSource
    </xa-datasource-class>
    </driver>
    <driver name="postgresql" module="org.postgresql">
      <xa-datasource-class>org.postgresql.xa.PGXADatasource
    </xa-datasource-class>
    </driver>
  </drivers>
  ...
</datasources>
```

2. Reiniciar el JBoss (si se encontrase en marcha) y añadir los datasources de las aplicaciones a desplegar. En este punto tenemos tres **opciones**:

<sup>7</sup> Se recomienda la descarga de la versión 42.2.5 que se encuentra en la tabla «Other versions», en la columna «JDBC 4.2».



- a) (**Opción recomendada** según los estándares de base de datos) Crear un fichero XML independiente (llamado **nombreAplicacion-ds.xml**) para cada aplicación dentro del directorio **JBOSS\_HOME\standalone\deployments** y añadir el siguiente contenido:

Para un *datasource* de tipo **Oracle** el contenido sería el siguiente:

```
<datasource jndi-name="java:jboss/datasources/codiAppDS" pool-name="codiAppDS"
enabled="true" use-java-context="true">
  <connection-url>jdbc:oracle:thin:@host:puerto/sid</connection-url>
  <driver>oracle</driver>
  <security>
    <user-name>userapp</user-name>
    <password>pass</password>
  </security>

  <new-connection-sql>
  BEGIN
  EXECUTE IMMEDIATE 'ALTER SESSION SET CURRENT_SCHEMA = NOMBREBD';
  END;
  </new-connection-sql>
</datasource>
```

Para un *datasource* de tipo **PostgreSQL** el contenido sería el siguiente:

```
<datasource jndi-name="java:jboss/datasources/codiAppDS" pool-name="codiAppDS"
enabled="true" use-java-context="true">
  <connection-url>jdbc:postgresql://host:puerto/nombrebd</connection-url>
  <driver>postgresql</driver>
  <security>
    <user-name>userapp</user-name>
    <password>pass</password>
  </security>

  <new-connection-sql>
  BEGIN
  EXECUTE IMMEDIATE 'ALTER SESSION SET CURRENT_SCHEMA = NOMBREBD';
  END;
  </new-connection-sql>
</datasource>
```

En la sección «4.3 Acceso a bases de datos» del documento «Estándares Jakarta EE» aparece una descripción más detallada de los parámetros a configurar.

- b) Añadir los parámetros de los *datasources* directamente dentro de la etiqueta `<datasources>` del fichero **JBOSS\_HOME\standalone\configuration\standalone.xml**.
- c) Utilizar la **consola web de administración** del JBoss (<http://localhost:9990/console/index.html>). Hay que tener en cuenta que los valores por defecto que incluye no son exactamente los mismos.

### **4.3. Cambios importantes respecto a la versión EAP 5.2**

1. Para iniciar el JBoss se ha de ejecutar el script **JBOSS\_HOME\bin\standalone.bat** en entornos Windows y el script **JBOSS\_HOME\bin\standalone.sh** en entornos Unix/Linux.
2. Para desplegar aplicaciones se ha de copiar el fichero EAR dentro del directorio **JBOSS\_HOME\standalone\deployments**.
3. Aunque se puede desplegar ficheros ear «en caliente» (una vez iniciado el JBoss), no es recomendable hacerlo más de dos o tres veces ya que el rendimiento baja notablemente.
4. El fichero de configuración principal se encuentra en **JBOSS\_HOME\standalone\configuration\standalone.xml**.

## 5. Keycloak 6.0.1

Keycloak es un producto de software de código abierto que permite el inicio de sesión único (IdP) con Identity Management y Access Management. En el GOIB usaremos, por un lado, Keycloak como servidor esperando peticiones de autenticación, y por otro, JBoss 7.2 EAP con un adaptador para poder conectarlo con Keycloak.

El servidor de Keycloak se puede instalar y configurar de manera manual o, alternativamente, usar una imagen Docker adaptada en el entorno de desarrollo del GOIB. En esta guía describiremos el proceso de instalación y configuración manual (la utilización de imágenes Docker se describe en el documento «Primeros pasos con Docker»).

### 5.1. Instalación

El proceso de instalación es el siguiente:

1. Acceder a la dirección <https://www.keycloak.org/downloads.html>.
2. Descargar la versión **Standalone Server Distribution 6.0.1**.
3. Descomprimir el fichero **keycloak-6.0.1.zip** dentro del directorio de trabajo (en nuestro caso, **C:\Desarrollo\keycloak-6.0.1**).
4. Establecer la variable de entorno **KEYCLOAK\_HOME** con el valor del directorio de instalación (en nuestro caso, **C:\Desarrollo\keycloak-6.0.1**).
5. Keycloak es un JBoss modificado. Para que no haya conflictos entre el JBoss EAP 7.2 y el JBoss del Keycloak, se tienen que modificar los puertos en uno de los dos servidores (en este manual cambiaremos los puertos del Keycloak).

Por defecto, el juego de puertos del JBoss es el siguiente:

- 8080/8443 para acceso HTTP/HTTPS
- 9990/9993 para configuración HTTP/HTTPS
- 8009 para AJP

Para evitar conflictos, sumaremos 100 a los puertos de Keycloak:

- 8180/8543 para acceso HTTP/HTTPS
- 10090/10093 para configuración HTTP/HTTPS

- 8109 para AJP

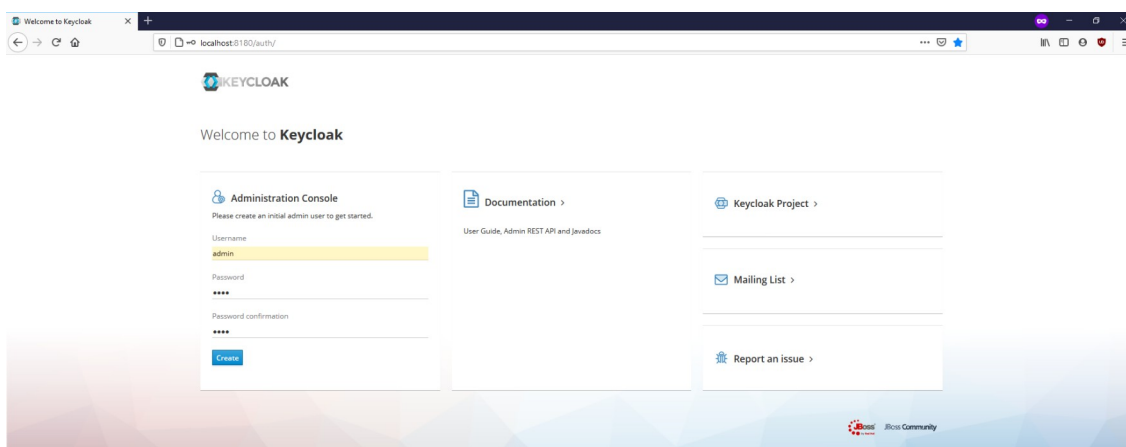
El cambio de los puertos se realiza substituyendo el valor de la propiedad **port-offset**. Para esto tenemos dos opciones :

- (**Recomendada**) Modificar el parámetro port-offset de la propiedad socket-binding-group del fichero **KEYCLOAK\_HOME\standalone\configuration\standalone.xml**

```
<socket-binding-group name="standard-sockets" default-interface="public" port-  
offset="{jboss.socket.binding.port-offset:0}">  
<socket-binding-group name="standard-sockets" default-interface="public" port-  
offset="{jboss.socket.binding.port-offset:100}">
```

- Iniciar Keycloak con el comando **KEYCLOAK\_HOME\bin\standalone.bat -Djboss.socket.binding.port-offset=100**.

- Reemplazar todas las ocurrencias de la variable JBOSS\_HOME para KEYCLOAK\_HOME en el script **KEYCLOAK\_HOME\bin\standalone.bat**.
- Iniciar el servidor de Keycloak ejecutando el script **KEYCLOAK\_HOME\bin\standalone.bat**.<sup>8</sup>
- Acceder a la consola de administración del Keycloak desde la dirección <http://localhost:8180/auth><sup>9,10</sup> para comprobar que funciona correctamente.



## 5.2. Configuración

En Keycloak se tienen que configurar tres elementos básicos: el dominio de actuación (realm), los clientes (client), y los roles de los usuarios (role).

<sup>8</sup> Hay que añadir la regla al Firewall si sale un mensaje de confirmación la primera vez.

<sup>9</sup> Si no tenemos ningún usuario administrador, hay que añadirlo por primera vez mediante la propia consola de administración.

<sup>10</sup> También se puede añadir ejecutando el script KEYCLOAK\_HOME\bin\add-user-keycloak

En los servidores de la DGMAD se ha definido el dominio de actuación GOIB y tres tipos de clientes:

- goib-default: permite al usuario autenticarse con certificado digital (cualquiera de los admitidos por @firma) o, si no dispone de certificado, mediante su usuario y contraseña del GOIB.
- goib-ws: se trata del mecanismo de autenticación que tiene que aplicarse para proteger módulos que contengan servicios REST. Permite autenticación BASIC sin redirigir la petición en la web centralizada de autenticación.
- goib-cert: solo habilita la autenticación con certificado, con lo cual, solo se podrá acceder a los contenidos protegidos del módulo, mediante el uso de un certificado válido.

**Los roles son los mismos que los definidos en Seycon puesto que los servidores de la DGMAD se conectan al LDAP del GOIB. Por lo tanto, si se quiere definir un rol nuevo de aplicación, el proceso de creación y asignación se mantiene igual.**

**Aunque se puede definir los nombres de cliente y roles que se quiera durante el desarrollo, se recomienda mantener la misma nomenclatura para evitar errores durante los despliegues a preproducción y producción.**

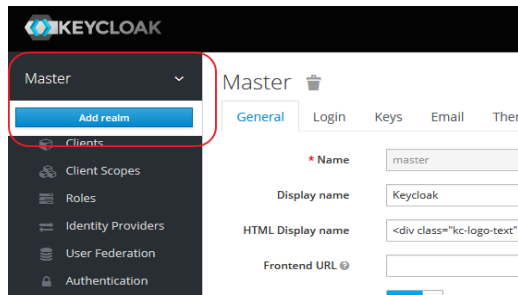
**En el entorno de desarrollo local no es posible configurar la conexión con @firma; por lo tanto, se tendrá que utilizar acceso por usuario y contraseña.**

A continuación se muestra un ejemplo de configuración para controlar el acceso a una aplicación llamada **goibusuari**. Esta aplicación muestra información del usuario autenticado y proporciona varios métodos para una API REST.

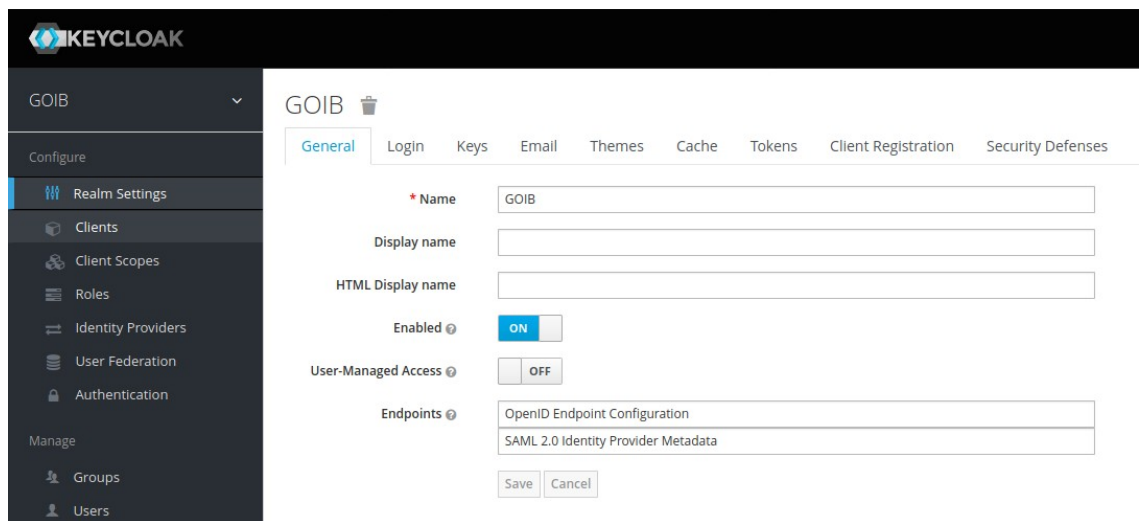
El proceso de configuración del ejemplo es el siguiente:

1. Acceder a la consola de administración local de Keycloak desde la dirección <http://localhost:8180/auth>.
2. Pulsar sobre el desplegable del menú y seleccionar «**Add realm**»

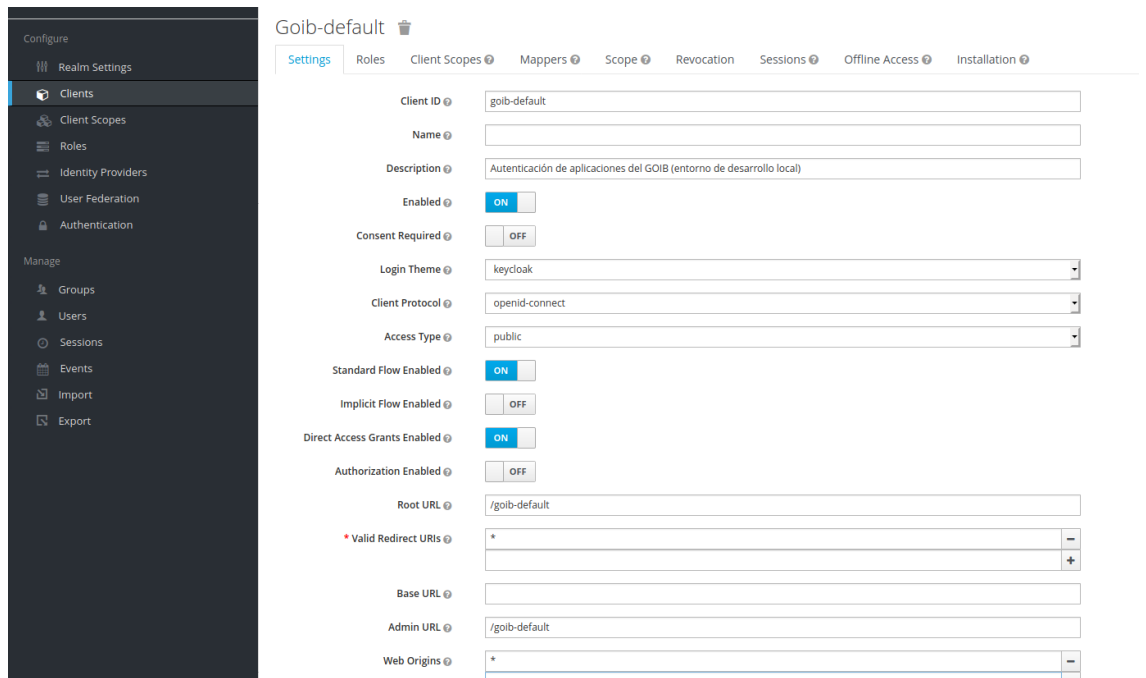




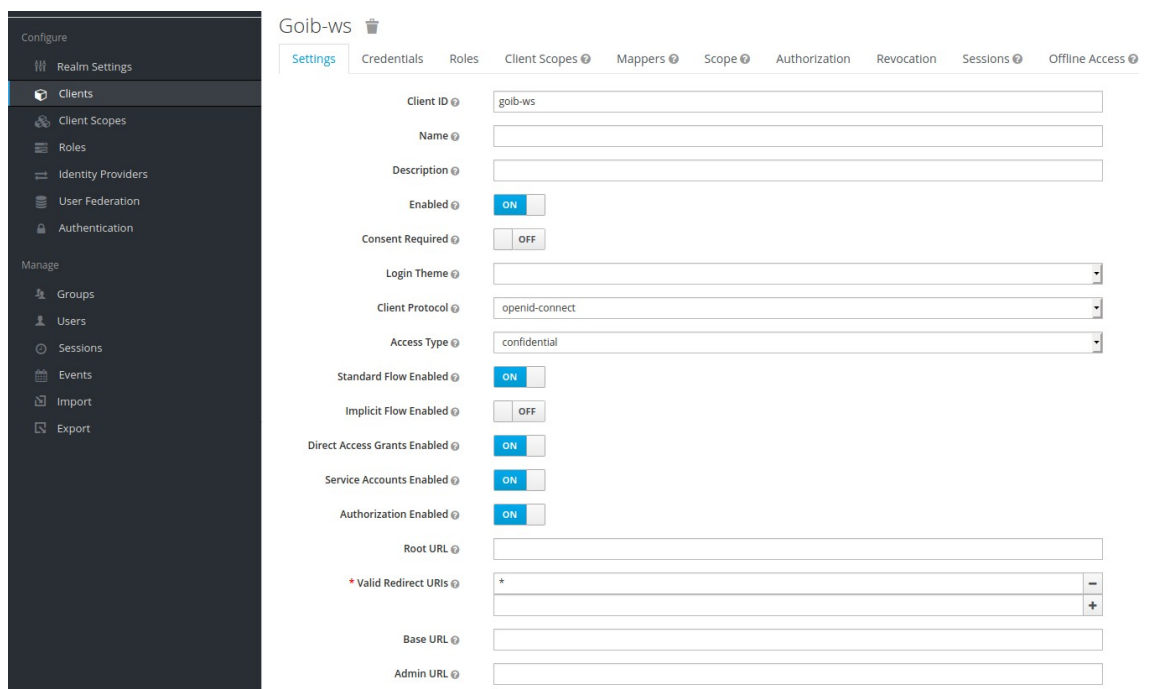
3. Establecer el nombre **GOIB** en el realm y dejar el resto de campos con los valores por defecto.



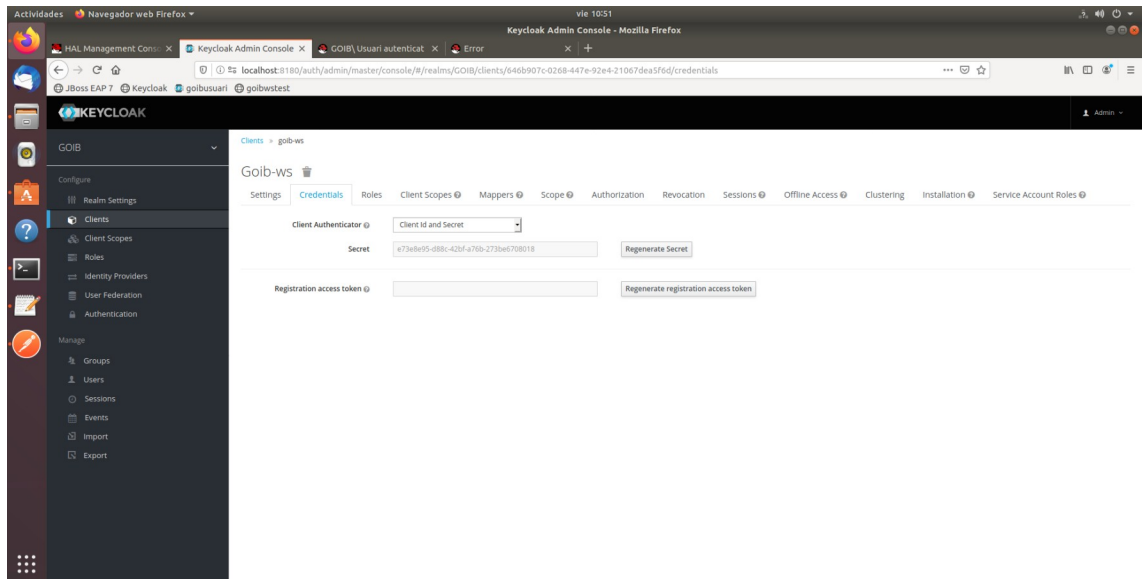
4. Añadir el cliente **goib-default** (destinado al backoffice) con los siguientes valores: **/goib-default** en el campo **ROOT URL** , y posteriormente **\*** en el campo **Valid Redirect URIs**; el resto de campos hay que dejarlos con el valor por defecto.



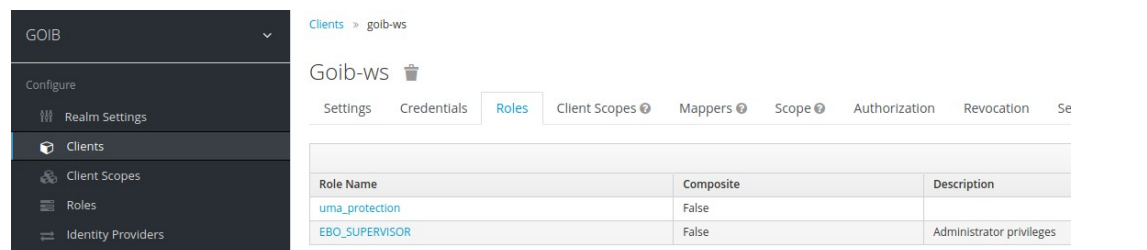
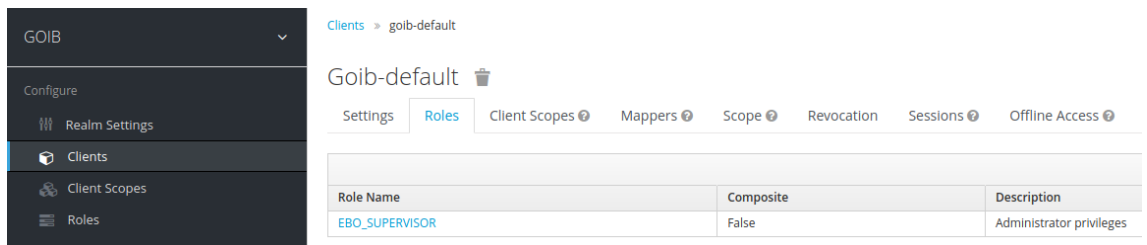
- Añadir el cliente **goib-ws** (destinado a los servicios REST) con los siguientes valores: **confidential** en el campo **Access Type**, **\*** en el campo **Valid Redirect URIs**, y las opciones **Authorization Enabled** y **Service Accounts Enabled** activadas; el resto de campos hay que dejarlos con el valor por defecto.



- Copiar la contraseña («credentials») generada automáticamente ya que se necesitará más adelante para configurar el JBoss.



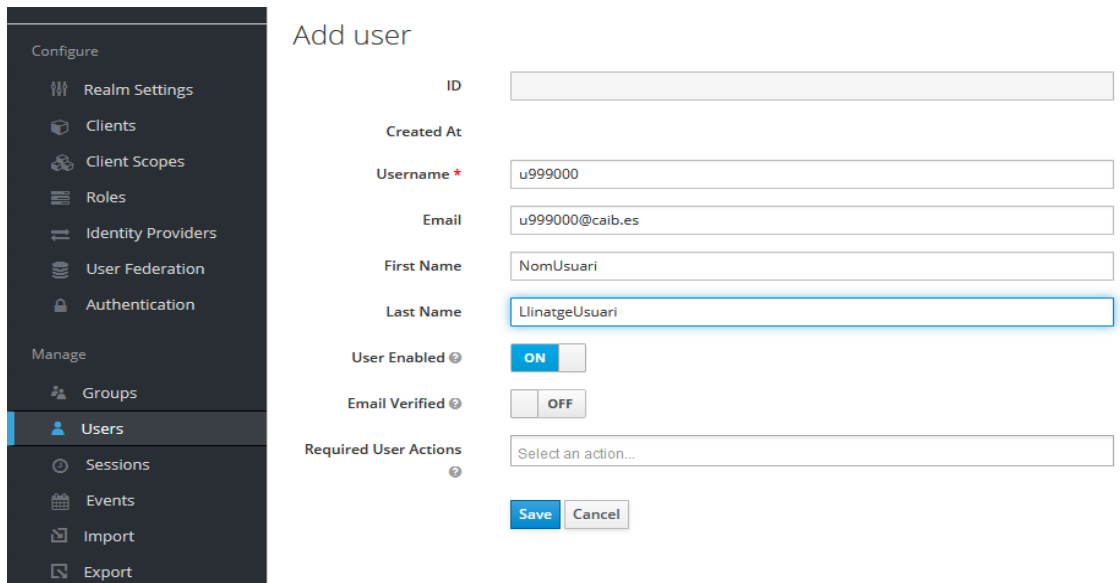
7. Añadir el rol **EBO\_SUPERVISOR** dentro de cada cliente<sup>1112</sup>.



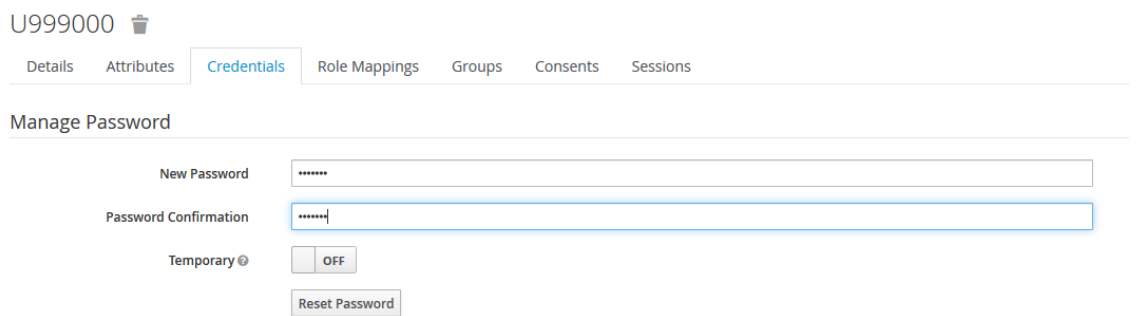
8. Añadir el usuario **u999000** rellenado los campos: Username, Email, First Name y Last Name. El campo UserEnabled ha de estar activado. Este usuario permitirá autenticarse en el backoffice.

11 Se puede configurar roles a nivel de dominio (realm) para dar acceso a todos los clientes o a nivel de cliente para que los usuarios tengan acceso sólo a un cliente en particular.

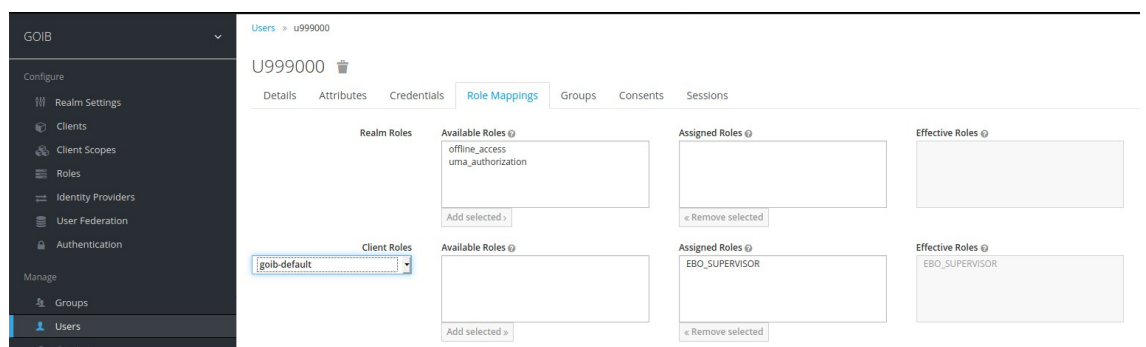
12 En el capítulo 6 veremos cómo configurar el conector de JBoss con Keycloak para establecer el nivel de autenticación utilizando el parámetro «use-resource-role-mappings»



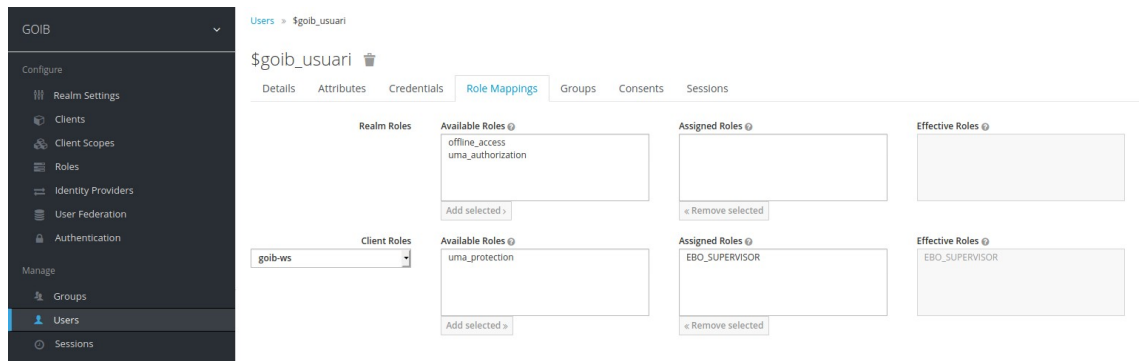
9. Establecer la misma contraseña dentro del apartado «credentials» (si no se establece ninguna contraseña no se podrá autenticar!!!).



10. Asignar el rol **EBO\_SUPERVISOR** al usuario **u999000** para el cliente **goib-default** (apartado «Role Mappings, Client Roles» dentro de la misma gestión del usuario).



11. Añadir el usuario **\$goib\_usuari** y asignarle la misma contraseña dentro del apartado «**credentials**». Este usuario lo utilizaremos para autenticar los servicios REST.
12. Asignar el rol **EBO\_SUPERVISOR** al usuario **\$goib\_usuari** para al cliente **goib-ws** (apartado «**Role Mappings, Client Roles**» dentro de la misma gestión del usuario).



## 6. Conexión JBoss - Keycloak

Para conectar JBoss EAP 7.2 con Keycloak se tiene que instalar un adaptador.

### 6.1. Instalación del conector

El proceso de instalación es el siguiente:

1. Acceder a la dirección <https://www.keycloak.org/downloads.html>.
2. Descargar el **Client Adapter** de Keycloak (OPENID CONNECT) para JBoss 7 EAP dentro del directorio JBOSS\_HOME.
3. Descomprimir el fichero **keycloak-wildfly-adapter-dist-6.0.1.zip**.
4. En el directorio **JBOSS\_HOME\bin** se añadirán los siguientes ejecutables:
  - adapter-install-offline.cli
  - adapter-install.cli
  - adapter-elytron-install-offline.cli
  - adapter-elytron-install.cli<sup>13</sup>

La diferencia entre las versiones «normal» y las «offline» es que el su éxito depende de si el JBoss está en marcha o no, respectivamente.

5. Con el JBoss parado, ejecutar el comando **jboss-cli.bat --file=adapter-install-offline.cli**.

```
C:\DesarrolloSimo\jboss-eap-7.2\bin>jboss-cli.bat --file=adapter-install-offline.cli
OpenJDK 64-Bit Server VM warning: Ignoring option PermSize; support was removed in 8.0
OpenJDK 64-Bit Server VM warning: Ignoring option MaxPermSize; support was removed in 8.0
{"outcome" => "success"}
{"outcome" => "success"}
{"outcome" => "success"}
{"outcome" => "success"}
Presione una tecla para continuar . . .
```

### 6.2. Configuración

Tal como se describe en el documento «Estándares Jakarta EE» los parámetros principales de configuración del subsistema de autenticación son los siguientes:

- **real-name** i **realm**: Nombre del dominio de confianza definido en Keycloak (normalmente pondremos «GOIB»).

<sup>13</sup> Actualmente, la versión con ELYTRON tiene UN BUG y da problemas con los EJBs. Por tanto, se desaconseja su uso.

- **auth-server-url:** Dirección del servidor de Keycloak. Generalmente será una dirección de un servidor local (en nuestro caso, <http://localhost:8181/auth>; el servidor de desarrollo de la DGMAD está destinado por las aplicaciones corporativas).
- **ssl\_required:** Para configurar si aceptamos conexiones SSL. Los posibles valores son: «ALL» (por todas las peticiones), «EXTERNAL» (para peticiones externas), o «NONE» (ninguna).
- **secure-deployment:** nombre del WAR.
- **resource:** Nombre del cliente o módulo a que se hace referencia dentro del Keycloak. Como he visto anteriormente, en los servidores de la DGMAD hay disponibles tres recursos: goib-default, goib-ws, y goib-cert.
- **use-resource-role-mappings:**
  - TRUE: evalúa el rol a nivel de CLIENT.
  - FALSE: evalúa el rol a nivel de REALM.
- Para el caso de un war de tipo **backoffice** se tienen que añadir las propiedades public-client y verify-token-audience con valor true.
- Para el caso de un war de tipo API REST se tienen que añadir las propiedades bearer-only, enable-basic-auth, y establecer el credential "secret" con el valor de la contraseña del cliente REST.

A continuación se muestra un ejemplo para conectar la aplicación **goibusuari** con el servidor Keycloak local descrito en el apartado «5.2. Configuración» del Keycloak; es decir, usaremos el realm **GOIB** y los clientes **goib-default** y **goib-ws**. El fichero ear de la aplicación contiene dos ficheros war: userinfo.war (backoffice) y rest.war (Api REST).

1. Suponiendo que queremos acceder solo a nivel de cliente, la configuración del fichero **JBOSS\_HOME\standalone\configuration\standalone.xml** sería la siguiente:



```
<subsystem xmlns="urn:jboss:domain:keycloak:1.1">
  <realm name="GOIB">
    <auth-server-url>http://localhost:8180/auth</auth-server-url>
    <ssl-required>EXTERNAL</ssl-required>
  </realm>

  <secure-deployment name="userinfo.war">
    <realm>GOIB</realm>
    <resource>goib-default</resource>
    <use-resource-role-mappings>true</use-resource-role-mappings>
    <public-client>true</public-client>
    <verify-token-audience>true</verify-token-audience>
    <principal-attribute>preferred_username</principal-attribute>
  </secure-deployment>

  <secure-deployment name="rest.war">
    <realm>GOIB</realm>
    <resource>goib-ws</resource>
    <use-resource-role-mappings>true</use-resource-role-mappings>
    <bearer-only>true</bearer-only>
    <enable-basic-auth>true</enable-basic-auth>
    <principal-attribute>preferred_username</principal-attribute>
    <credential name="secret">e73e8e95-d88c-42bf-a76b-273be6708018</credential>
  </secure-deployment>

</subsystem>
```

\* El valor del parámetro «Secret» será diferente de una instalación a otra.

2. La aplicación **goibusuari** ya tiene configurado el rol **EBO\_SUPERVISOR** para los ficheros **userinfo.war** y **rest.war**; en concreto, en el fichero **src\main\webapp\WEB-INF\web.xml**:

```
<security-constraint>

  <web-resource-collection>
    <web-resource-name>UserInfo</web-resource-name>
    <url-pattern>/*</url-pattern>
    <http-method>POST</http-method>
    <http-method>GET</http-method>
  </web-resource-collection>

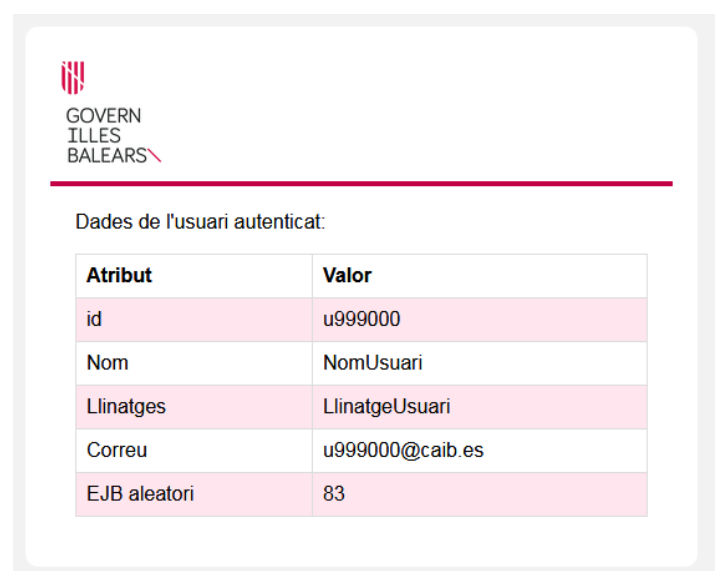
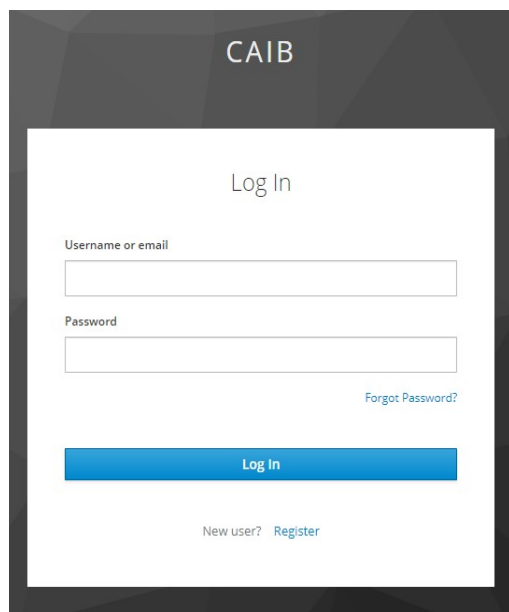
  <auth-constraint>
    <role-name>EBO_SUPERVISOR</role-name>
  </auth-constraint>
</security-constraint>
```



```
<login-config>
  <auth-method>KEYCLOAK</auth-method>
  <realm-name>Autenticacio</realm-name>
</login-config>

<security-role>
  <role-name>EBO_SUPERVISOR</role-name>
</security-role>
```

3. Desplegar el fichero **goibusuari.ear**<sup>14</sup> dentro del directorio **JBOSS\_HOME\standalone\deployments** del JBoss EAP 7.2.
4. Probar el acceso autenticado al backoffice accediendo a la dirección <http://localhost:8080/goibusuari/>. Aparecerá una ventana donde se tienen que insertar las credenciales del usuario (en este caso, el usuario **u999000** creado a la sección 4.2). Si los datos son correctos, aparecerá una ventana con los datos del usuario autenticado.



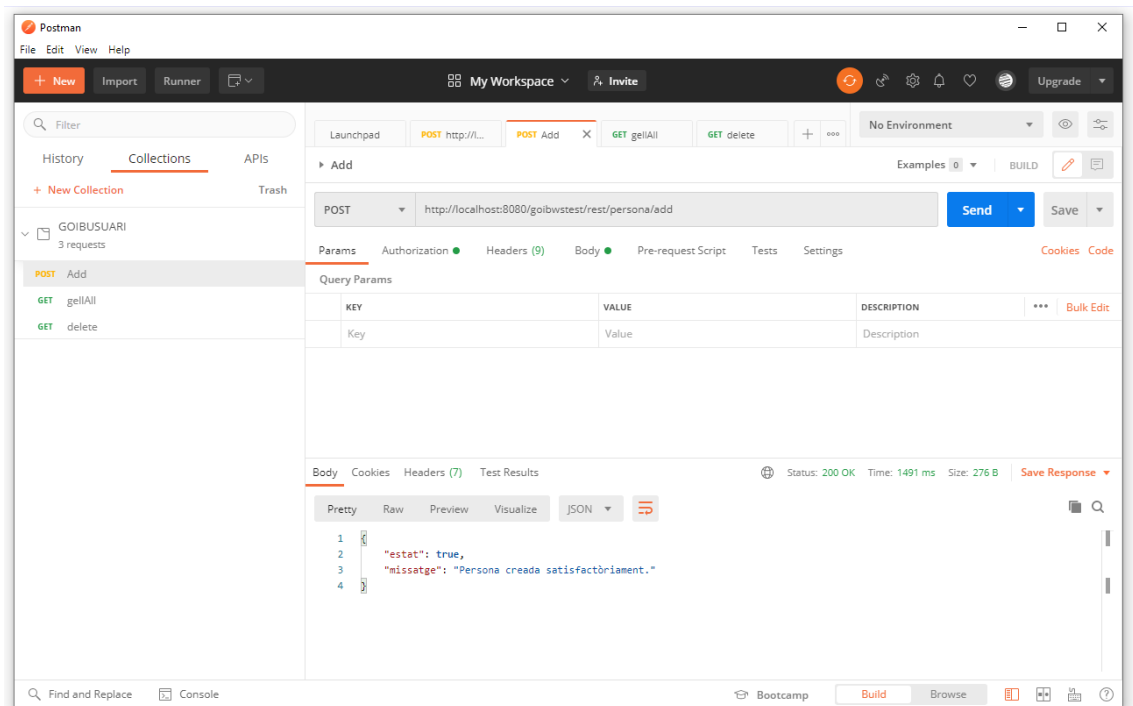
5. Para probar el acceso autenticado al API REST, se ha de utilizar un cliente específico (como Postman o SoapUI) que permita realizar llamadas a los métodos proporcionados:
  - POST: <http://localhost:8080/goibwstest/rest/persona/add>. Añade una persona en formato JSON, por ejemplo:

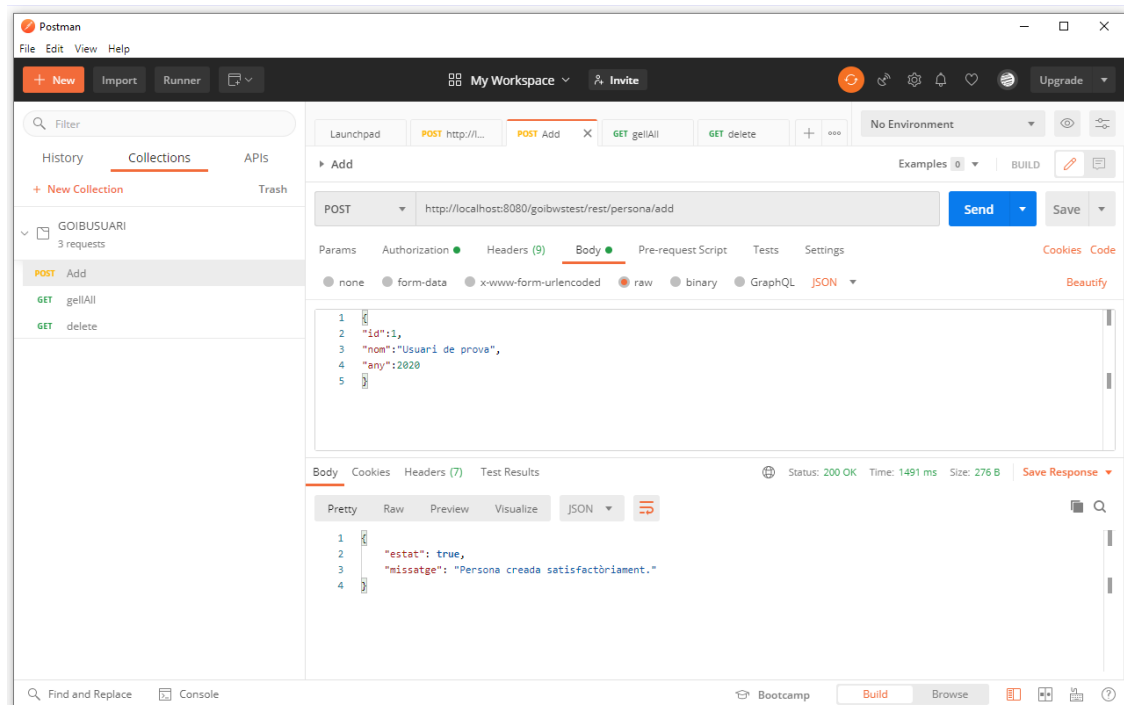
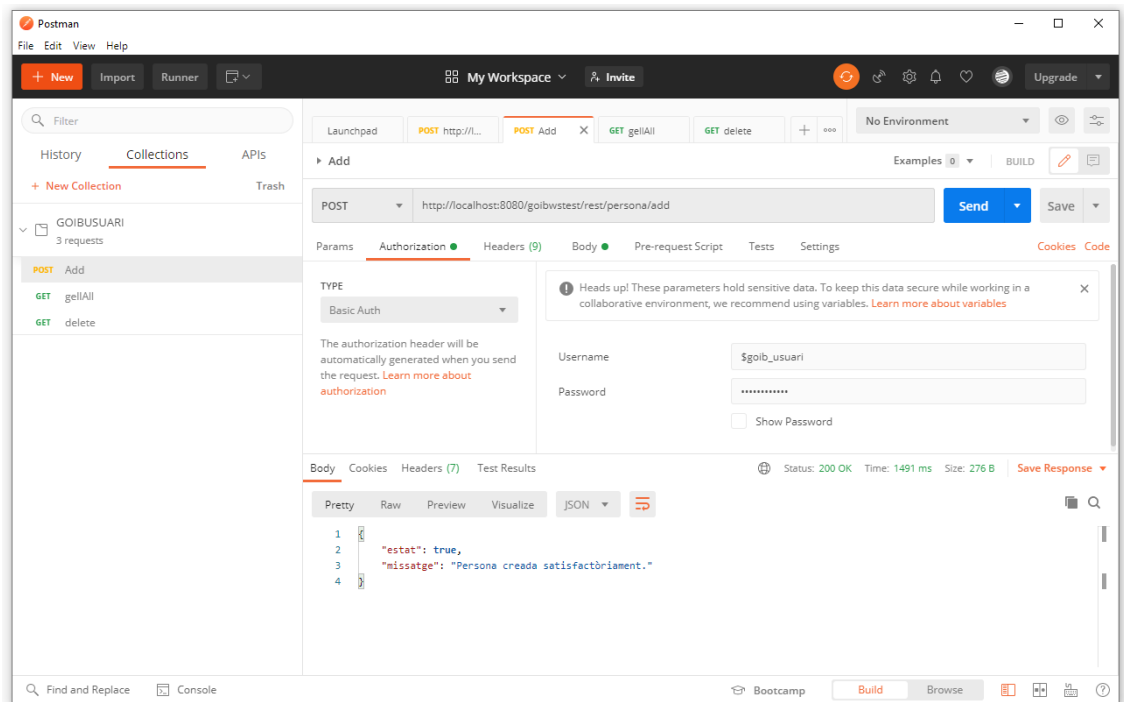
<sup>14</sup> Este EAR se encuentra dentro del directorio doc del ProjecteBase <https://github.com/GovernIB/projectebase/tree/projectebase-1.0/doc>.

```
{
  "id":1,
  "nom":"Usuari de prova",
  "any":2020
}
```

- GET: <http://localhost:8080/goibwstest/rest/persona/{id}/delete>. Elimina la persona con el identificador indicado.
- GET: <http://localhost:8080/goibwstest/rest/persona/getAll>. Devuelve todas las personas añadidas previamente.

En el cliente se tienen que configurar los datos de la conexión de tipo «Basic». Por ejemplo, dentro de Postman las llamadas al método de tipo POST <http://localhost:8080/goibwstest/rest/persona/add> se configuran de la manera siguiente:





## 7. Errores comunes

### 7.1. Versión incorrecta de JDK

Si durante el inicio del JBoss o del Keycloak apareciese el error:

```
Unrecognized VM option 'MetaspaceSize=96M'
Error: Could not create the Java Virtual Machine.
Error: A fatal exception has occurred. Program will exit.
```

Se ha de revisar que se ha configurado correctamente la variable JAVA\_HOME con el comando **java -version**.

```
C:\Desarrollo>java -version
openjdk version "11" 2018-09-25
OpenJDK Runtime Environment 18.9 (build 11+28)
OpenJDK 64-Bit Server VM 18.9 (build 11+28, mixed mode)
```

### 7.2. Error de autenticación

Si apareciese un error de credenciales al acceder al backoffice o al API REST de la aplicación, se ha de revisar:

1. Que el usuario esté creado en el Keycloak.
2. **Que tenga asignado una contraseña.**
3. Que tenga asignado el rol adecuado dentro del clienet correspondiente.
4. Que el contenido del fichero JBOSS\_HOME\standalone\configuration\standalone.xml sea correcto (ver ejemplo en la sección 5.1).
5. Que el fichero web.xml del módulo web y la securización de los EJBs estén mal configurados (para más información, ver el capítulo «5. Seguridad» del documento «Estándares de Aplicaciones Jakarta EE»).

### 7.3. Activación de servicios Jakarta EE adicionales

Hay que tener en cuenta que el fichero JBOSS\_HOME\standalone\configuration\standalone.xml inicia el JBoss con una configuración básica que no incluye todos los servicios disponibles de Jakarta EE; lo más destacable es que no incluye JMS/MDB.



La configuración que incluye estos servicios se encuentra al fichero **JBOSS\_HOME\standalone\configuration\standalone-full.xml**. Para iniciar con esta configuración hace falta añadiendo la opción "-c" con el nombre de la configuración (por ejemplo: JBOSS\_HOME\bin\standalone.bat/.sh -c standalone-full.xml).

#### **7.4. Contexto de solo lectura**

Si durante el inicio del JBoss apareciese el error «Contexto de solo lectura» al WeldStartService, se tiene que añadir el parámetro **require-bean-descriptor="true"** en el subsistema Weld del fichero JBOSS\_HOME\standalone\configuration\standalone.xml.

```
<subsystem xmlns="urn:jboss:domain:weld:4.0" require-bean-descriptor="true"/>
```