

Estàndards

Desenvolupament d'aplicacions del GOIB

**Guia de configuració de l'entorn de  
desenvolupament**



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

Palma, setembre de 2022

# Índex

<b>HISTORIAL DE VERSIONS.....</b>	<b>3</b>
<b>1. INTRODUCCIÓ.....</b>	<b>4</b>
<b>2. OPENJDK 11.....</b>	<b>5</b>
2.1. Instal·lació.....	5
<b>3. APACHE MAVEN.....</b>	<b>6</b>
<b>4. JBOSS EAP 7.2.....</b>	<b>8</b>
4.1. Instal·lació.....	8
4.2. Configuració dels datasources.....	11
4.3. Canvis importants respecte a la versió EAP 5.2.....	14
<b>5. KEYCLOAK 6.0.1.....</b>	<b>15</b>
5.1. Instal·lació.....	15
5.2. Configuració.....	16
<b>6. CONNEXIÓ JBOSS - KEYCLOAK.....</b>	<b>23</b>
6.1. Instal·lació del connector.....	23
6.2. Configuració.....	23
<b>7. ERRORS COMUNS.....</b>	<b>29</b>
7.1. Versió incorrecta de JDK.....	29
7.2. Error d'autenticació.....	29
7.3. Activació de serveis Jakarta EE addicionals.....	29
7.4. Context de només lectura.....	30

## Historial de versions

Data	Versió	Descripció	Autor
15/01/20	9.0	Primera versió	DGMAD
28/02/20	9.1	Correcció d'errors als exemples Nova imatge docker de keycloak	DGMAD
28/05/20	9.2	Afegir informació de proxy per imatges Docker	DGMAD
03/12/20	9.5	Aclariments	DGMAD
08/09/22	9.6	Correcció error datasource	DGMAD

## 1. INTRODUCCIÓ

Els actuals estàndards de desenvolupament del *Govern de les Illes Balears* (GOIB) es basen en quatre components bàsics:

- **OpenJDK 11** com a plataforma de desenvolupament (en substitució de Java SE 7).
- **Maven 3.6** com a eina per a la gestió i construcció de projectes Jakarta EE.
- **JBoss EAP 7.2** com a servidor d'aplicacions (en substitució de JBoss EAP 5.2).
- **Keycloak 6.0.1** com a sistema d'administració d'identitats (en substitució de Seycon).

La finalitat d'aquest document és proporcionar una guia de configuració de l'entorn de desenvolupament necessari per fer servir els estàndards de desenvolupament del GOIB, especialment per configurar aquests components.

Aquest guia és una ajuda per al desenvolupador, el seu ús no és obligatori.

**Pel correcte seguiment d'aquesta guia és necessari tenir permisos d'administrador local sobre la màquina.** A més, és recomanable crear un directori de treball en el disc local<sup>1</sup> on emmagatzemar el programari, les llibreries externes, i els projectes de desenvolupament.

Els exemples i captures de pantalla descrits corresponen a un sistema operatiu Windows 10 de 64 bits. Si es vol, es pot obtenir una versió per a Unix / Linux de tot el programari mitjançant el gestor de paquets de la distribució.

La guia mostra el procés d'instal·lació i configuració des del terminal. Una vegada instal·lats i configurats els serveis descrits, si es vol, es pot configurar l'entorn de desenvolupament integrat (Eclipse, IntelliJ, Netbeans,..) per facilitar el desplegament d'aplicacions.

---

<sup>1</sup> En el nostre cas utilitzarem el directori de treball «C:\Desarrollo».

## 2. OpenJDK 11

OpenJDK 11 és la versió lliure de la plataforma de desenvolupament *Java SE Development Kit 11* (JDK 11). No s'ha de confondre amb la plataforma d'execució *Java Runtime Environment* (JRE). A aquesta guia suposarem que ja es té instal·lat l'entorn d'execució JRE.

### 2.1. Instal·lació

El procés d'instal·lació és el següent:

1. Accedir a l'adreça <https://jdk.java.net/java-se-ri/11><sup>2</sup> i escollir entre versió Linux/x64 o Windows/x64 (en aquest manual farem servir Windows 10).
2. Descarregar el fitxer **openjdk-11+28\_windows-x64\_bin.zip**.
3. Extreure el fitxer en el directori **C:\Program Files\Java**.
4. Establir la variable d'entorn de sistema **JAVA\_HOME** amb el valor **C:\Program Files\Java\jdk-11**.
5. Afegir el valor **%JAVA\_HOME%\bin** a la variable d'entorn **PATH**.
6. Obrir un terminal i verificar que està configurada la versió 11 de JDK amb la comanda d'execució **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>2</sup> Aquesta és una implementació de referència d'OpenJDK 11. Les versions actualitzades es poden descarregar des de la web d'Oracle (requereix llicència) o bé des d'altres entitats que mantenen builds gratuïtes i actualitzades de OpenJDK 11 (com per exemple les mantingudes per AdoptOpenJDK).

### 3. Apache Maven

Apache MAVEN és una eina de programari per a la gestió i construcció de projectes Java. És similar en funcionalitat a Apache Ant però té un model de configuració de construcció més simple basat en un format XML. Segons els estàndards de desenvolupament del GOIB tots els nous projectes de desenvolupament han d'utilitzar MAVEN.

El procés d'instal·lació i configuració és el següent:

1. Descarregar la versió 3.6.X des de l'adreça <http://maven.apache.org/download.html>.
2. Descomprimir el fitxer **apache-maven-3.6.x-bin.zip** en el directori de treball.
3. Establir la variable d'entorn de sistema<sup>3</sup> **M2\_HOME** amb el valor **C:\Desarrollo\apache-maven-3.6.X**, substituint X pel número de revisió.
4. Afegir el valor **%M2\_HOME%\bin** a la variable d'entorn de sistema **Path**.
5. Obrir un terminal i verificar que està configurada la versió 3.6.x de MAVEN amb la comanda d'execució **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, dins del directori de treball, el directori on s'emmagatzemaran les dependències MAVEN de tots els projectes (al nostre cas, **C:\Desarrollo\repo-maven**).
7. Crear el fitxer de propietats MAVEN **settings.xml** de l'usuari en el directori home d'aquest (al nostre cas, **C:\Users\XXX\m2\settings.xml**).
8. Afegir el següent contingut al fitxer de propietats **settings.xml** creat, indicant el repositori local (**localRepository**) que acabem de crear, i l'usuari uXXX i contrasenya YYY del **proxy** si s'està dins la xarxa del GOIB.

<sup>3</sup> En Windows, com a administrador local de la màquina, cal anar a l'opció «Panel de control > Sistema y seguridad > Sistema > Configuración avanzada del sistema».

```
<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) és una implementació certificada de les especificacions completes i del perfil web de *Jakarta Enterprise Edition 8* (Jakarta EE 8, abans Java EE). Encara que existeix una versió comunitària d'aquesta implementació anomenada WildFly, a la DGMAD s'utilitza JBoss EAP per motius de suport i rendiment.

### 4.1. Instal·lació

El procés d'instal·lació és el següent:

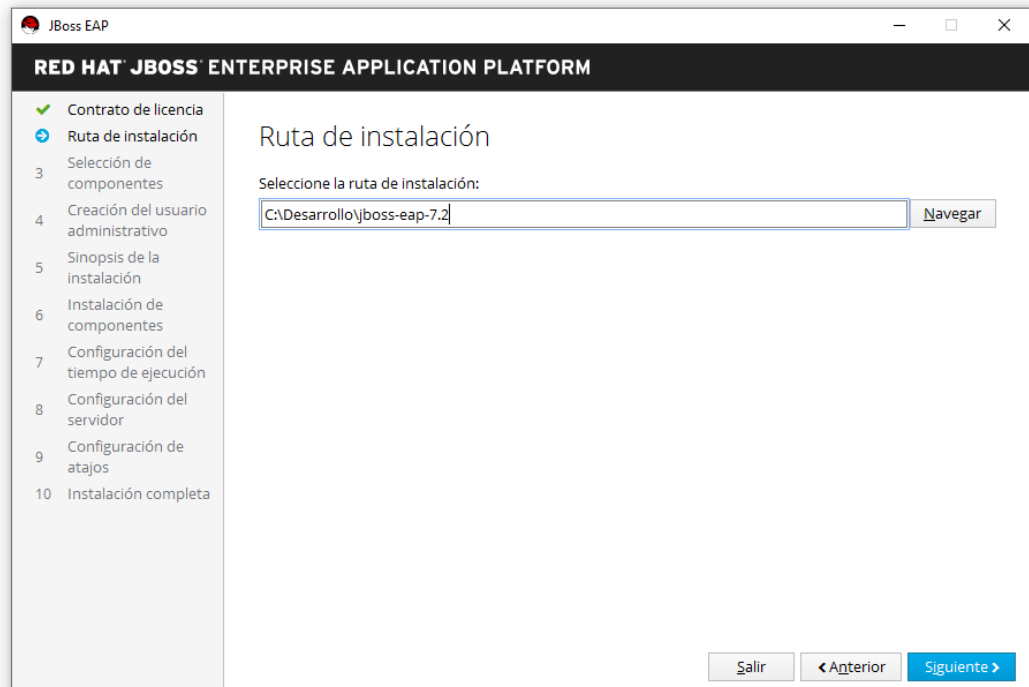
1. Accedir a la secció de descàrregues de la pàgina oficial de JBoss <https://developers.redhat.com/products/eap/download/>.
2. Descarregar i executar l'instal·lador de la versió 7.2.0<sup>45</sup> (fitxer **jboss-eap-7.2.0-installer.jar**).
3. Especificar el directori d'instal·lació del JBoss (al nostre cas, **C:\Desarrollo\jboss-eap-7.2**).

---

<sup>4</sup> És necessari registrar-se prèviament a la pàgina de RedHat amb un compte gratuït.

<sup>5</sup> Les *releases notes* amb els bugs solucionats a la versió 7.2 són a l'adreça [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/). Per evitar errors, es recomana aplicar el darrer patch publicat.

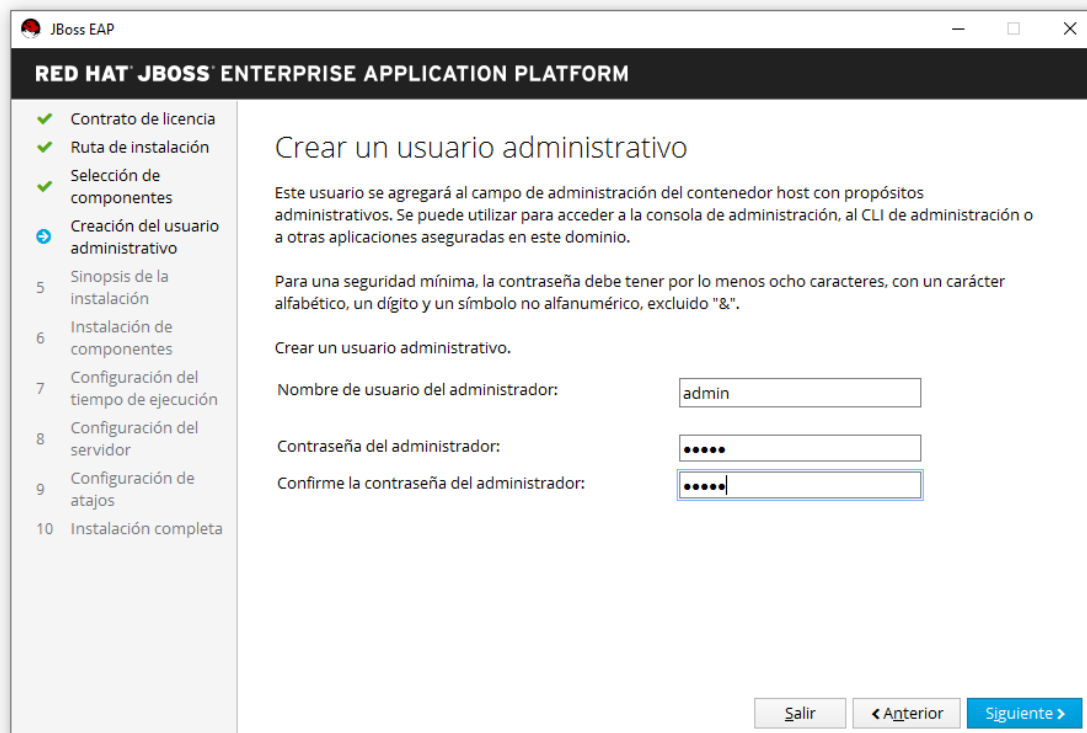




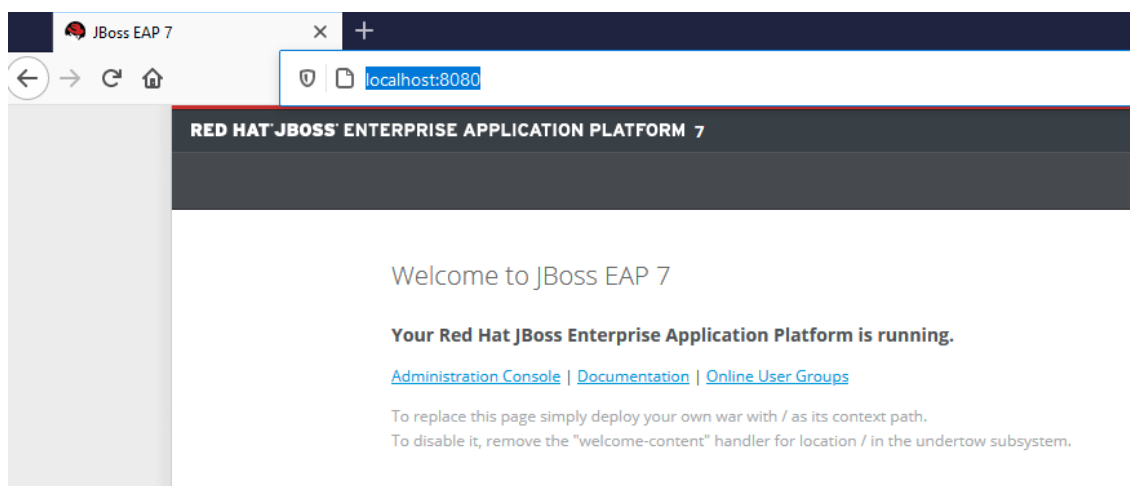
4. Donar d'alta l'usuari administrador del JBoss<sup>6</sup> (per exemple: **admin**).

---

<sup>6</sup> Alternativament, aquest usuari es pot crear posteriorment amb l'script JBOSS\_HOME\bin\add-user.



5. Establir la variable d'entorn de sistema **JBOSS\_HOME** amb el valor del directori d'instal·lació del JBoss (al nostre cas, **C:\Desarrollo\jboss-eap-7.2**).
6. Iniciar el servidor executant l'script **JBOSS\_HOME\bin\standalone.bat**.
7. Accedir a la pàgina principal de JBoss des de l'adreça <http://localhost:8080/> per comprovar que funciona correctament.

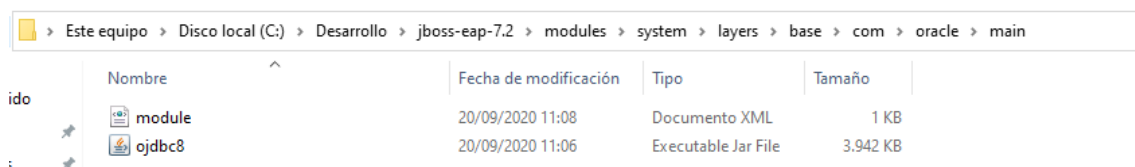


## 4.2. Configuració dels datasources

A continuació es descriu el procés de configuració de datasources per sistemes gestors de base de dades Oracle i PostgreSQL segons els estàndards de base de dades del GOIB (per més informació veure document «Estàndards de base de dades»).

### Oracle:

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



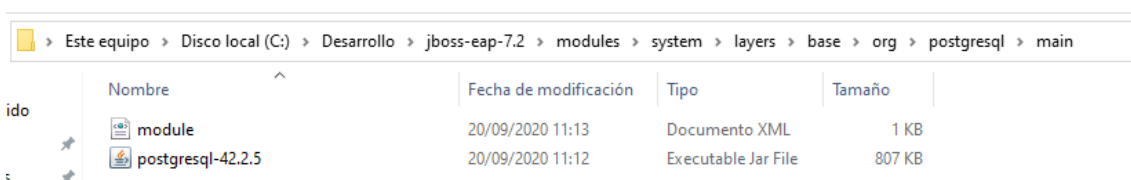
ido	Nombre	Fecha de modificación	Tipo	Tamaño
	module	20/09/2020 11:08	Documento XML	1 KB
	ojdbc8	20/09/2020 11:06	Executable Jar File	3.942 KB

2. Descarregar el fitxer **ojdbc8.jar** des de l'adreça <https://www.oracle.com/technetwork/database/features/jdbc/jdbc-ucp-122-3110062.html><sup>7</sup> i copiar-lo dins del directori anterior.
3. Afegir el fitxer **module.xml** en el directori anterior amb el següent contingut:

```
<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 directori **JBOSS\_HOME\modules\system\layers\base\org\postgresql\main**.



ido	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>7</sup> S'han d'acceptar els termes i condicions i tenir un compte d'Oracle.

2. Descarregar el fitxer **postgresql-42.2.5.jar** en el directori anterior des de l'adreça <https://jdbc.postgresql.org/download.html><sup>8</sup> i copiar-lo dins del directori anterior.
3. Afegir el fitxer **module.xml** en el directori anterior amb el següent contingut:

```
<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 i PostgreSQL:

1. Afegir la següent configuració de connectors (*drivers*) al fitxer **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 es trobàs en marxa) i afegir els datasources de les aplicacions a desplegar. A aquest punt tenim tres **opcions**:
  - a) (**Opció recomanada** segons els estàndards de base de dades) Crear un fitxer XML independent (anomenat **nomAplicacio-ds.xml**) per a cada

<sup>8</sup> Es recomana la descàrrega de la versió 42.2.5 que es troba en la taula «Other versions», en la columna «JDBC 4.2».

aplicació dins del directori **JBOSS\_HOME\standalone\deployments** i afegir-hi el següent contingut:

Per un *datasource* de tipus **Oracle** el contingut seria el següent:

```
<datasource jndi-name="java:jboss/datasources/codiAppDS" pool-name="codiAppDS"
enabled="true" use-java-context="true">
  <connection-url>jdbc:oracle:thin:@host:port/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 = NOMBD';
    END;
  </new-connection-sql>
</datasource>
```

Per un *datasource* de tipus **PostgreSQL** el contingut seria el següent:

```
<datasource jndi-name="java:jboss/datasources/codiAppDS" pool-name="codiAppDS"
enabled="true" use-java-context="true">
  <connection-url>jdbc:postgresql:@host:port/nombbdd</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 = NOMBD';
    END;
  </new-connection-sql>
</datasource>
```

A la secció «4.3 Accés a bases de dades» del document «Estàndards Jakarta EE» apareix una descripció més detallada dels paràmetres a configurar.

b) Afegir els paràmetres dels *datasources* directament dins l'etiqueta `<datasources>` del fitxer **JBOSS\_HOME\standalone\configuration\standalone.xml**.

c) Fer servir la **consola d'administració** del JBoss (<http://localhost:9990/console/index.html>). Cal tenir en compte que els valors per defecte que inclou no són exactament els mateixos.

### **4.3. Canvis importants respecte a la versió EAP 5.2**

1. Per iniciar el JBoss s'ha d'executar l'script **JBOSS\_HOME\bin\standalone.bat** a entorns Windows i l'script **JBOSS\_HOME\bin\standalone.sh** a entorns Unix/Linux.
2. Per desplegar aplicacions s'ha de copiar el fitxer EAR dins del directori **JBOSS\_HOME\standalone\deployments**.
3. Encara que es pot desplegar fitxers ear «en calent» (una vegada ja iniciat el JBoss), no és recomanable fer-ho més de dues o tres vegades ja que el rendiment baixa notablement.
4. El fitxer de configuració principal es troba dins del directori **JBOSS\_HOME\standalone\configuration\standalone.xml**.

## 5. Keycloak 6.0.1

Keycloak és un producte de programari de codi obert que permet l'inici de sessió únic (IdP) amb *Identity Management i Access Management*. En el GOIB farem servir, per un costat, Keycloak com a servidor esperant peticions d'autenticació, i per altre, JBoss 7.2 EAP amb un adaptador per poder connectar-lo amb Keycloak.

El servidor de Keycloak es pot instal·lar i configurar de manera manual o, alternativament, fer servir una imatge Docker adaptada a l'entorn de desenvolupament del GOIB. En aquesta guia descriurem el procés d'instal·lació i configuració manual (la utilització d'imatges Docker es descriu en el document «Primeres passes amb Docker»).

### 5.1. Instal·lació

El procés d'instal·lació és el següent:

1. Accedir a l'adreça <https://www.keycloak.org/downloads.html>.
2. Descarregar la versió **Standalone Server Distribution 6.0.1**.
3. Extreure el fitxer **keycloak-6.0.1.zip** dins del directori de treball (al nostre cas, **C:\Desarrollo\keycloak-6.0.1**).
4. Establir la variable d'entorn **KEYCLOAK\_HOME** amb el valor del directori d'instal·lació (al nostre cas, **C:\Desarrollo\keycloak-6.0.1**).
5. Keycloak és un JBoss modificat. Perquè no hi hagi conflictes entre el JBoss EAP 7.2 i el JBoss del Keycloak, s'han de modificar els ports a un dels dos servidors (a aquest manual canviarem els ports del Keycloak).

Per defecte, el joc de ports del JBoss és el següent:

- 8080/8443 per accés HTTP/HTTPS
- 9990/9993 per configuració HTTP/HTTPS
- 8009 per AJP

Per evitar conflictes, sumarem 100 als ports de Keycloak:

- 8180/8543 per accés HTTP/HTTPS
- 10090/10093 per configuració HTTP/HTTPS
- 8109 per AJP

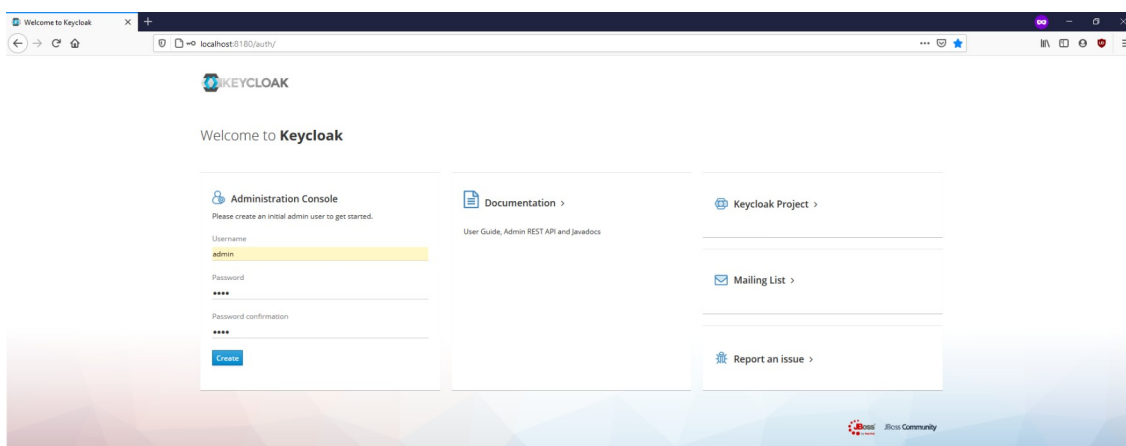
El canvi dels ports es fa substituint el valor de la propietat **port-offset**. Per això tenim dues opcions :

- a) **(Recomanada)** Modificar el paràmetre port-offset de la propietat socket-binding-group del fitxer **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}">
```

- b) Iniciar Keycloak amb la comanda d'execució **KEYCLOAK\_HOME\bin\standalone.bat -Djboss.socket.binding.port-offset=100**.

6. Reemplaçar totes les ocurrences de la variable JBOSS\_HOME per KEYCLOAK\_HOME a l'script **KEYCLOAK\_HOME\bin\standalone.bat**.
7. Iniciar el servidor de Keycloak executant l'script **KEYCLOAK\_HOME\bin\standalone.bat**.<sup>9</sup>
8. Accedir a la consola d'administració del Keycloak des de l'adreça <http://localhost:8180/auth><sup>1011</sup> per comprovar que funciona correctament.



## 5.2. Configuració

En Keycloak s'han de configurar tres elements bàsics: el domini d'actuació (realm), els clients (client), i els rols dels usuaris (role).

<sup>9</sup> Cal afegir la regla en el Firewall si surt un missatge de confirmació la primera vegada.

<sup>10</sup> Si no tenim cap usuari administrador, cal afegir-lo per primera vegada mitjançant la pròpia consola d'administració.

<sup>11</sup> També es pot afegir executant l'script KEYCLOAK\_HOME\bin\add-user-keycloak



Als servidors de la DGMAD s'ha definit el domini d'actuació GOIB i tres tipus de clients:

- goib-default: permet a l'usuari autenticar-se amb certificat digital (qualsevol dels admesos per *@firma*) o, si no disposa de certificat, mitjançant el seu usuari i contrasenya del GOIB.
- goib-ws: es tracta del mecanisme d'autenticació que ha d'aplicar-se per protegir mòduls que continguin *serveis REST*. Permet autenticació *BASIC* sense redirigir la petició a la web centralitzada d'autenticació.
- goib-cert: només habilita l'autenticació amb certificat, amb la qual cosa, només es podrà accedir als continguts protegits del mòdul, mitjançant l'ús d'un certificat vàlid.

**Els rols són els mateixos que els definits en el Seycon ja que els servidors de la DGDAM es connecten amb LDAP del GOIB. Per tant, si es vol definir un rol nou d'aplicació, el procés de creació i assignació es manté igual.**

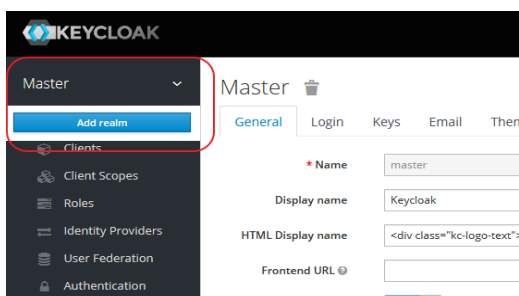
**Encara que es pot definir els noms de client i rols que es vulgui durant el desenvolupament, es recomana mantenir la mateixa nomenclatura per evitar errors durant els desplegaments a preproducció i producció.**

**A l'entorn de desenvolupament local no és possible configurar la connexió amb *@firma*; per tant, s'haurà d'utilitzar accés per usuari i contrasenya.**

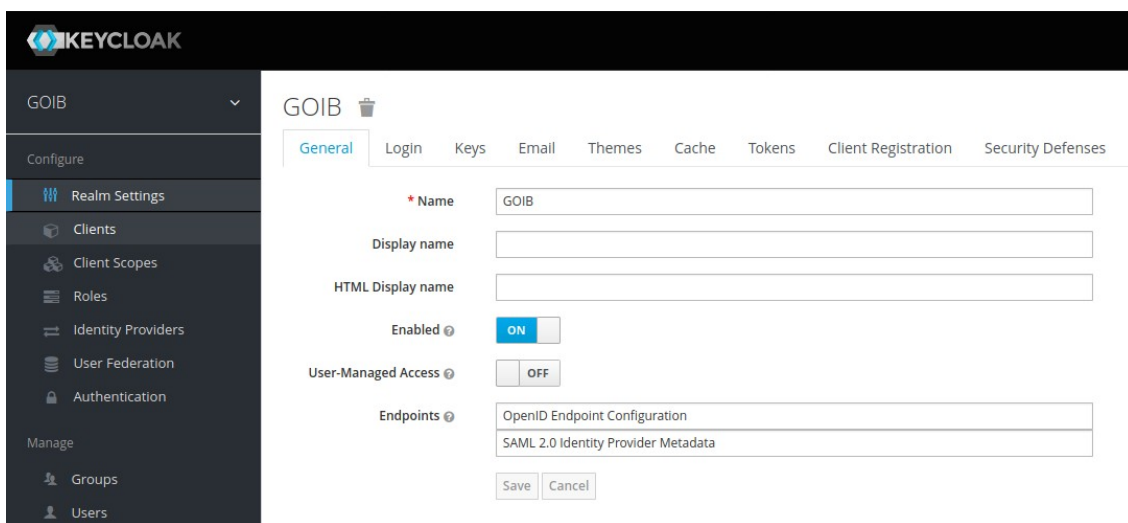
A continuació es mostra un exemple de configuració per controlar l'accés a una aplicació anomenada **goibusuari**. Aquesta aplicació mostra informació de l'usuari autenticat i proporciona diversos mètodes per a una API REST.

El procés de configuració de l'exemple és el següent:

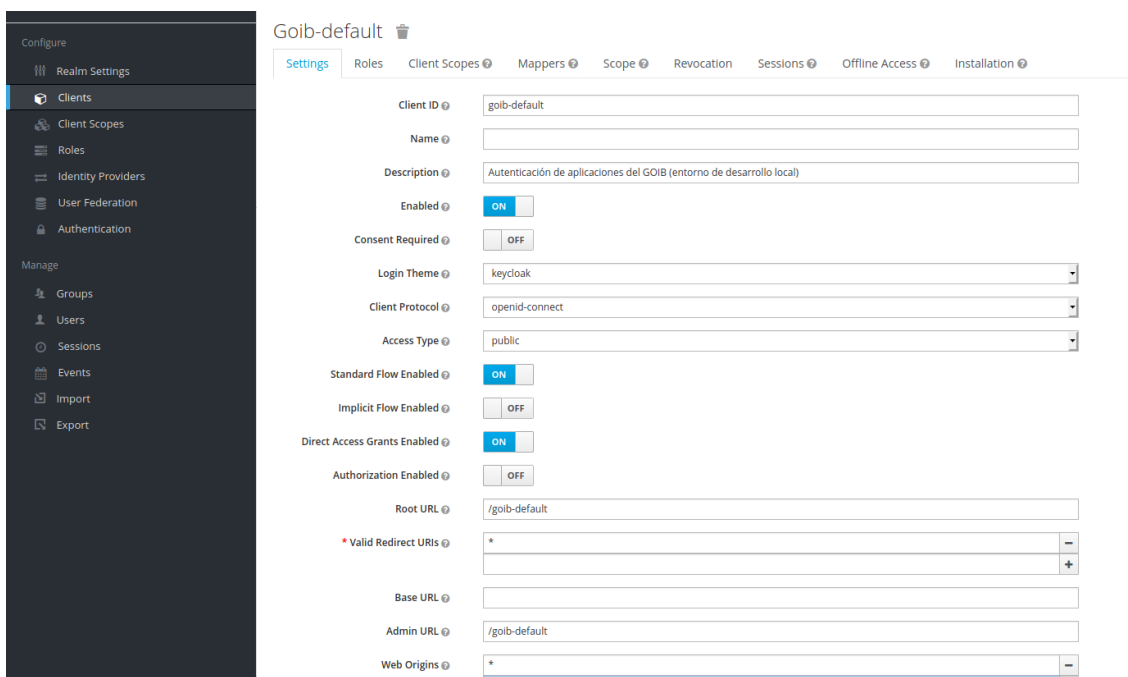
1. Accedir a la consola d'administració local de Keycloak <http://localhost:8180/auth>.
2. Pitjar sobre el desplegable del menú i seleccionar «**Add realm**»



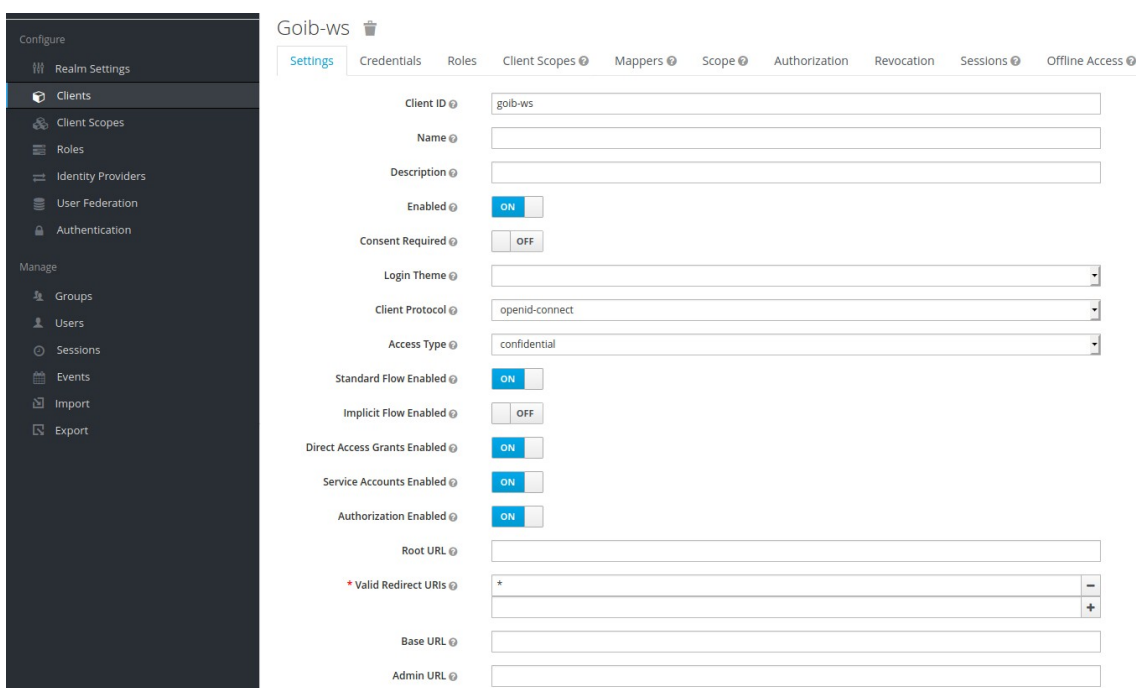
3. Establir el nom **GOIB** en el realm i deixar la resta de camps amb els valors per defecte.



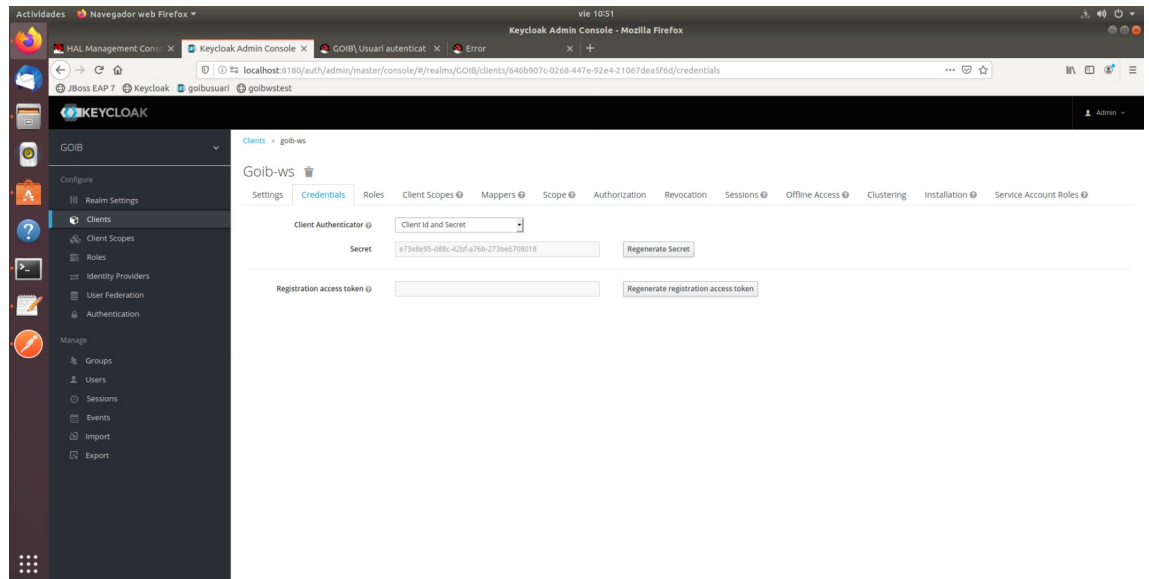
4. Afegir el client **goib-default** (destinat al backoffice) amb els següents valors: **/goib-default** en el camp **ROOT URL** , i posteriorment **\*** en el camp **Valid Redirect URIs**; la resta de camps deixar el valor per defecte.



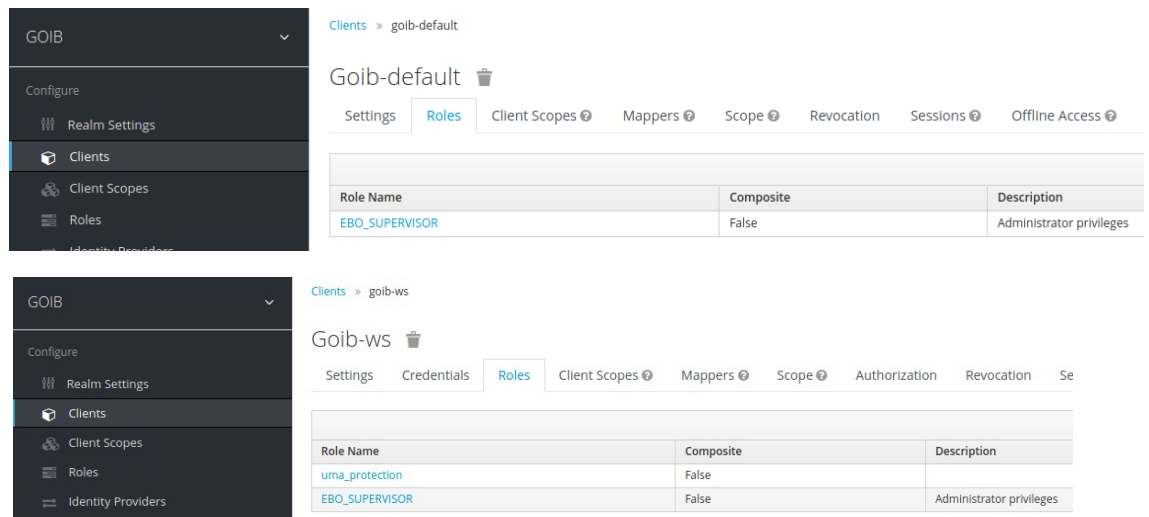
- Afegir el client **goib-ws** (destinat als serveis REST) amb els següents valors: **confidential** en el camp **Access Type**, **\*** en el camp **Valid Redirect URIs**, i les opcions **Authorization Enabled** i **Service Accounts Enabled** activades; la resta de camps deixar el valor per defecte.



- Copiar la contrasenya («credentials») generada automàticament ja que es necessitarà més endavant per configurar el JBoss.



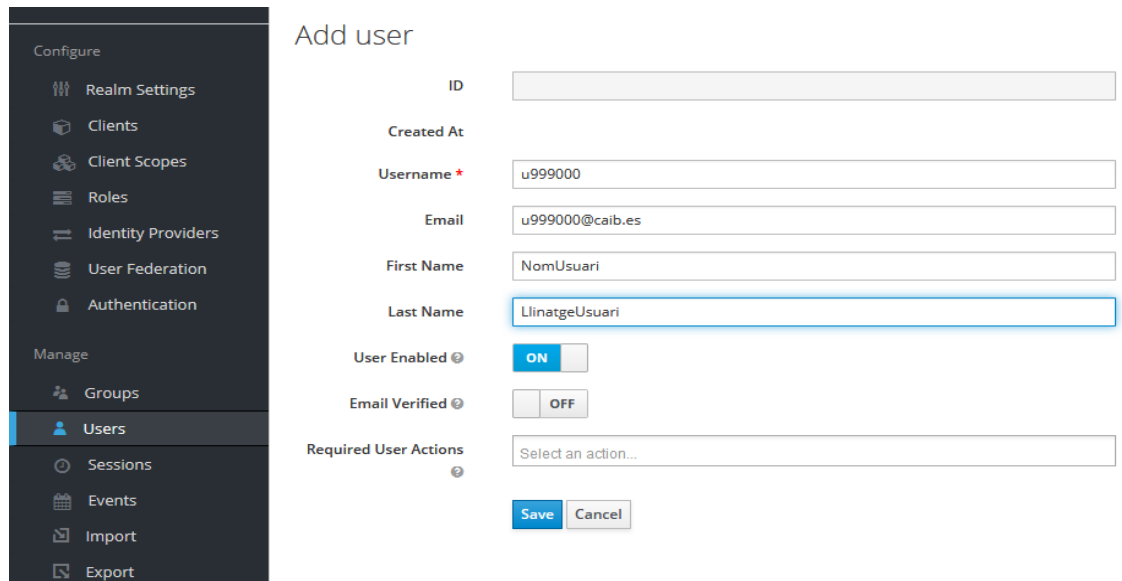
7. Afegir el rol **EBO\_SUPERVISOR** dins de cada client<sup>1213</sup>.



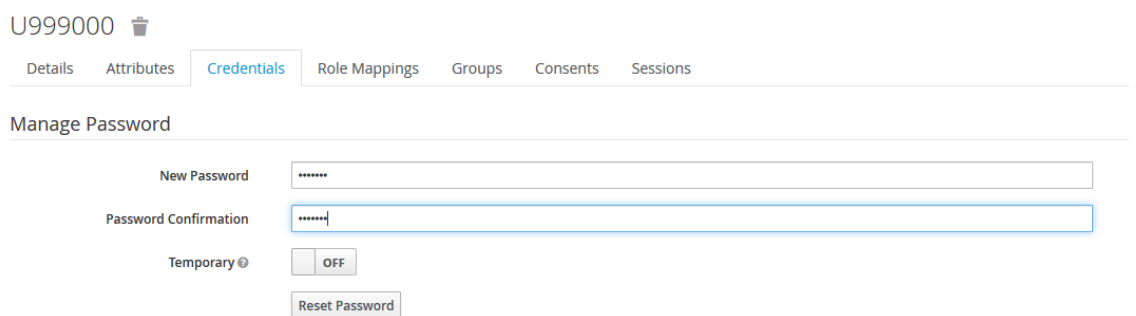
8. Afegir l'usuari **u999000** omplint els camps: Username, Email, First Name i Last Name. El camp UserEnabled ha d'estar activat. Aquest usuari permetrà autenticar-se dins del backoffice.

<sup>12</sup> Es pot configurar rols a nivell de domini (realm) per donar accés a tots els clients o a nivell de client perquè els usuaris tinguin accés només a un client en particular.

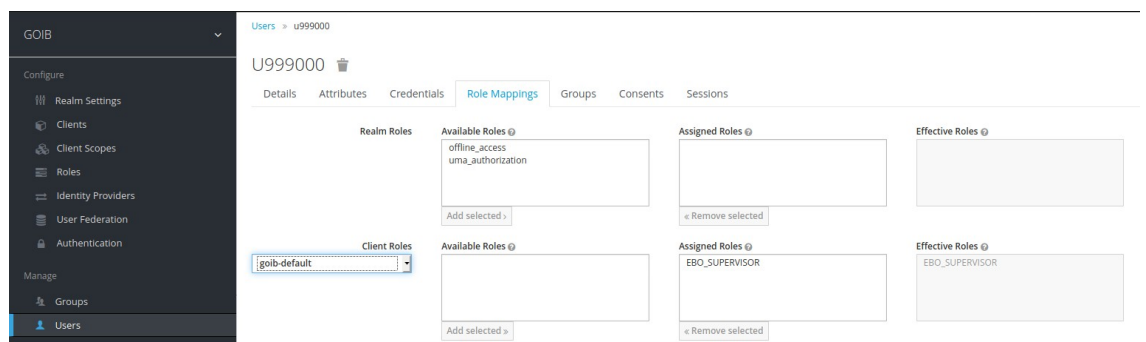
<sup>13</sup> En el capítol 6 veurem com configurar el connector de JBoss amb Keycloak per establir el nivell d'autenticació fent servir el paràmetre «use-resource-role-mappings»



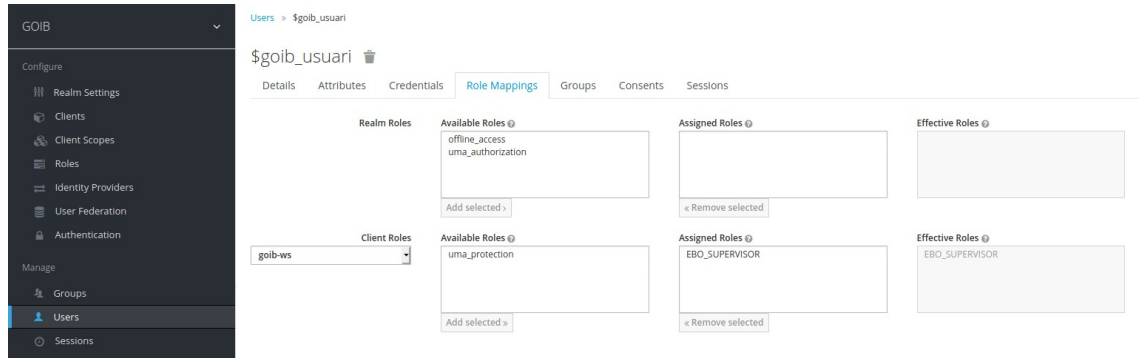
9. Establir-li la mateixa contrasenya dins l'apartat «**credentials**» (si no s'estableix cap contrasenya no es podrà autenticar!!!).



10. Assignar el rol **EBO\_SUPERVISOR** a l'usuari **u999000** per al client **goib-default** (apartat «Role Mappings, Client Roles» dins la mateixa gestió de l'usuari).



11. Afegir l'usuari **\$goib\_usuari** i li assigna-li la mateixa contrasenya dins l'apartat «**credentials**». Aquest usuari el farem servir per autenticar els serveis REST.
12. Assignar el rol **EBO\_SUPERVISOR** a l'usuari **\$goib\_usuari** per al client **goib-ws** (apartat «Role Mappings, Client Roles» dins la mateixa gestió de l'usuari).



The screenshot shows the Keycloak user management interface. On the left is a sidebar with the 'GOIB' logo and a navigation menu. The main area displays the configuration for the user '\$goib\_usuari'. The 'Role Mappings' tab is selected, showing two sections: 'Realm Roles' and 'Client Roles'. Under 'Client Roles', the client 'goib-ws' is selected. The 'Available Roles' list for 'goib-ws' includes 'uma\_protection'. The 'Assigned Roles' list shows 'EBO\_SUPERVISOR' has been assigned. The 'Effective Roles' list also shows 'EBO\_SUPERVISOR'.

## 6. Connexió JBoss - Keycloak

Per connectar JBoss EAP 7.2 amb Keycloak s'ha d'instal·lar un adaptador.

### 6.1. Instal·lació del connector

El procés d'instal·lació és el següent:

1. Accedir a l'adreça <https://www.keycloak.org/downloads.html>.
2. Descarregar el **Client Adapter** de Keycloak (OPENID CONNECT) per a JBoss 7 EAP dins del directori JBOSS\_HOME.
3. Extreure el fitxer **keycloak-wildfly-adapter-dist-6.0.1.zip**.
4. Al directori **JBOSS\_HOME\bin** s'afegiran els següents executables:
  - adapter-install-offline.cli
  - adapter-install.cli
  - adapter-elytron-install-offline.cli
  - adapter-elytron-install.cli<sup>14</sup>

La diferència entre les versions «normal» i les «offline» és que el seu èxit depèn de si el JBoss està en marxa o no, respectivament.

5. Amb el JBoss aturat, executar la comanda **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. Configuració

Tal com es descriu e el document «Estàndards Jakarta EE» els paràmetres principals de **configuració** del subsistema d'autenticació són els següents:

- **real-name** i **realm**: Nom del domini de confiança definit a Keycloak (normalment posarem «GOIB»).

<sup>14</sup> Actualment, la versió amb ELYTRON té un BUG i dona problemes amb els EJBs. Per tant, es desaconsella fer-lo servir.

- **auth-server-url**: Adreça del servidor de Keycloak. Generalment serà una adreça d'un servidor local (al nostre cas, <http://localhost:8181/auth>; el servidor de desenvolupament de la DGMAD està destinat per les aplicacions corporatives.
- **ssl\_required**: Per configurar si acceptam connexions SSL. Els possible valors són: «ALL» (per totes les peticions), «EXTERNAL» (per peticions externes), o «NONE».
- **secure-deployment**: nom del WAR.
- **resource**: Nom del client o mòdul a que es fa referència dins el Keycloak. Com he vist anteriorment, en els servidors de la DGMAD hi ha disponibles tres resources: goib-default, goib-ws, i goib-cert.
- **use-resource-role-mappings**:
  - TRUE: avalua el rol a nivell de CLIENT.
  - FALSE: avalua el rol a nivell de REALM.
- Per al cas d'un war de tipus **backoffice** s'han d'afegir les propietats public-client i verify-token-audience amb valor true.
- Per al cas d'un war de tipus **API REST** s'han d'afegir les propietats bearer-only, enable-basic-auth, i establir el credential "secret" amb el valor de la contrasenya del client API REST.

A continuació es mostra un exemple per connectar l'aplicació **goibusuari** amb el servidor Keycloak local descrit a l'apartat «5.2. Configuració» del Keycloak; es a dir, farem servir el realm **GOIB** i els clients **goib-default** i **goib-ws**. El fitxer ear de l'aplicació conté dos fitxers war: userinfo.war (backoffice) i rest.war (API REST).

1. Suposant que volem accedir només a nivell de client, la configuració del fitxer **JBOSS\_HOME\standalone\configuration\standalone.xml** seria la següent:



```
<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 paràmetre «Secret» serà diferent d'una instal·lació a d'altre. S'ha de configurar el valor apropiat.

2. L'aplicació **goibusuari** ja té configurat el rol **EBO\_SUPERVISOR** dins dels fitxers **userinfo.war** i **rest.war**; en concret, dins del fitxer **src\main\webapp\WEB-INF\web.xml**:

```
<security-constraint>

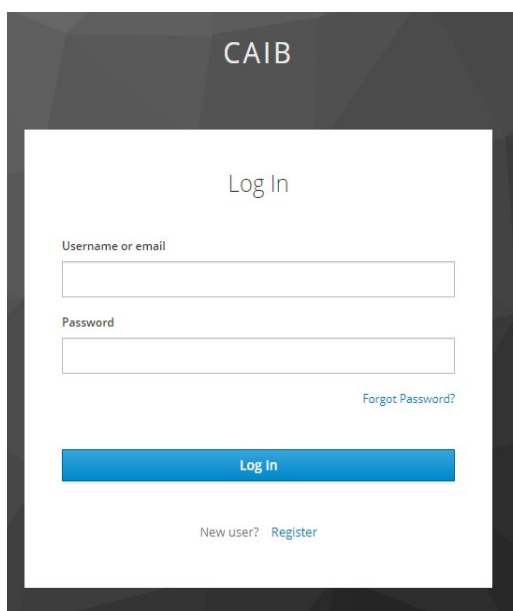
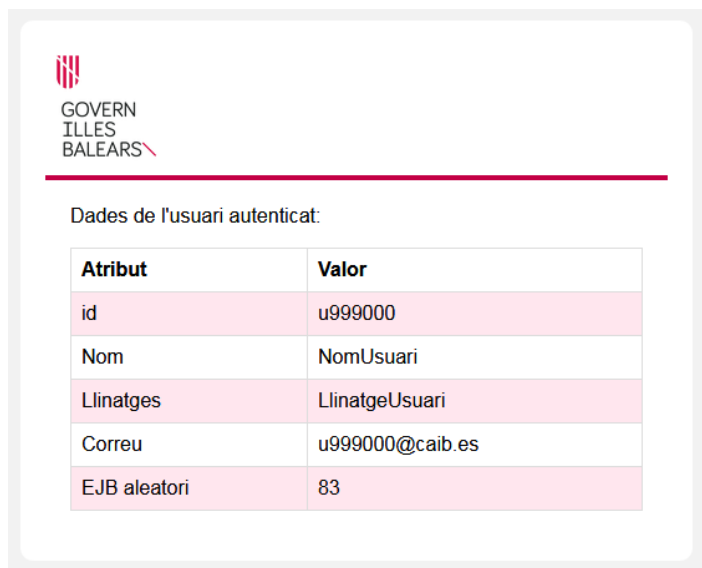
  <web-resource-collection>
    <web-resource-name>UserInfo</web-resource-name>
    <url-pattern>/*</url-pattern>
    <http-method>POST</http-method>
    <http-method>GET</http-method>
  </web-resource-collection>

  <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 fitxer **goibusuari.ear**<sup>15</sup> dins del directori **JBOSS\_HOME\standalone\deployments** del JBoss EAP 7.2.
4. Provar l'accés autenticat al backoffice accedint a l'adreça <http://localhost:8080/goibusuari/>. Apareixerà una finestra on s'han d'inserir les credencials de l'usuari (en aquest cas, l'usuari **u999000** creat a la secció 4.2). Si les dades són correctes, apareixerà una finestra amb les dades de l'usuari autenticat.

Atribut	Valor
id	u999000
Nom	NomUsuari
Llinatges	LlinatgeUsuari
Correu	u999000@caib.es
EJB aleatori	83

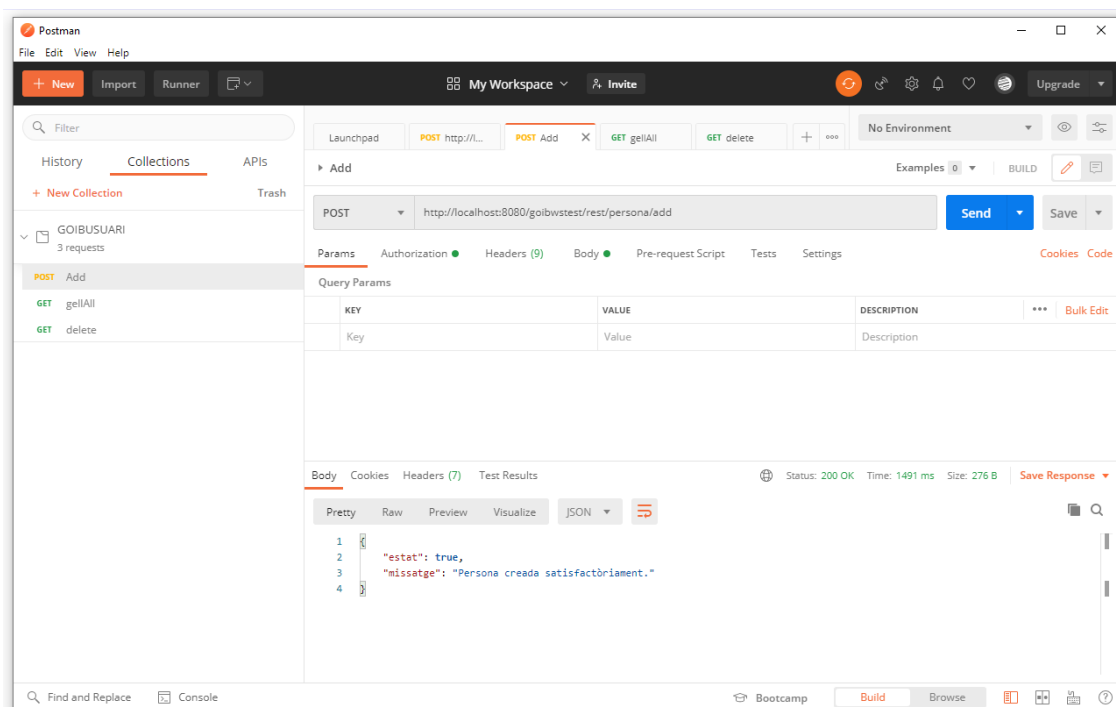
5. Per provar l'accés autenticat a l'API REST, s'ha de fer servir un client específic (com Postman o SoapUI) que permeti fer cridades als mètodes proporcionats:
  - POST: <http://localhost:8080/goibwstest/rest/persona/add>. Afegeix una persona en format JSON, per exemple:

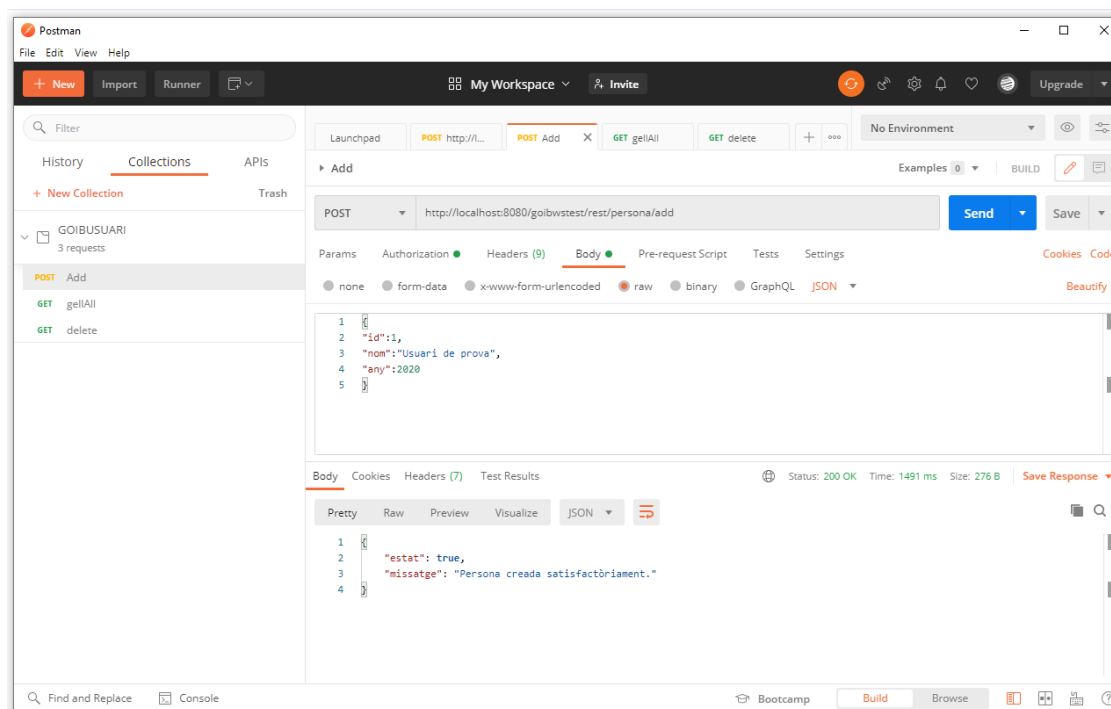
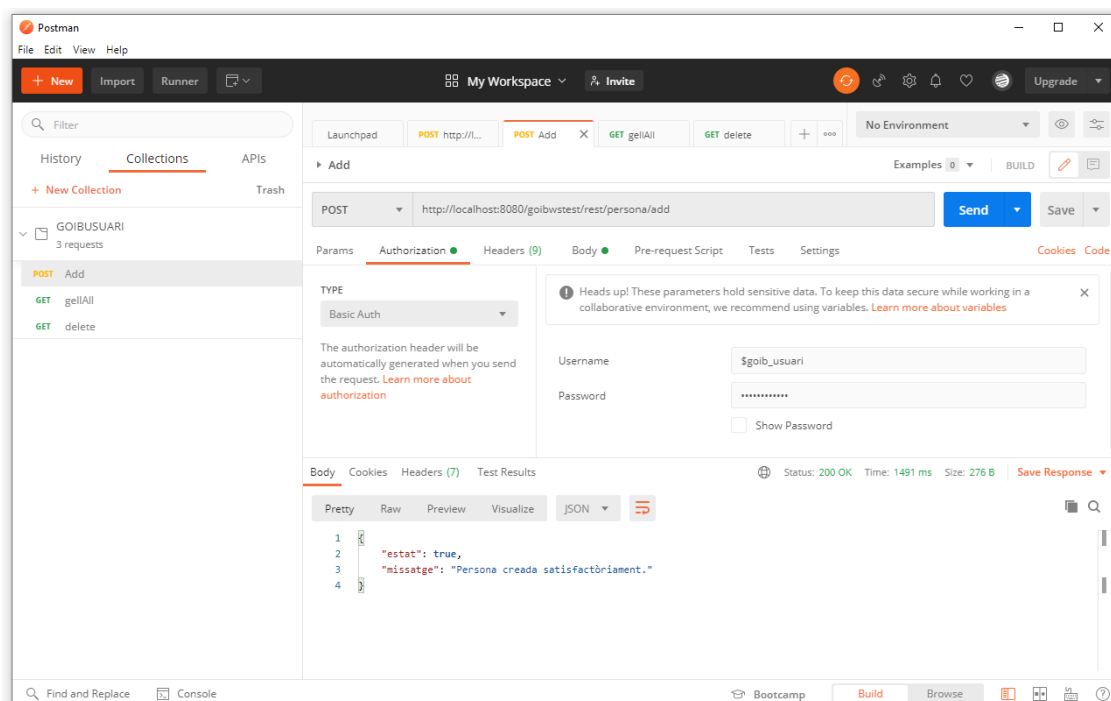
<sup>15</sup>Aquest EAR el podeu trobar dins del directori doc del ProjecteBase <https://github.com/GovernIB/projectebase/tree/projectebase-1.0/doc>.

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

- GET: <http://localhost:8080/goibwstest/rest/persona/{id}/delete>. Elimina la persona amb l'identificador indicat a {i}.
- GET: <http://localhost:8080/goibwstest/rest/persona/getAll>. Retorna totes les persones afegides prèviament.

Al client s'han de configurar les dades de la connexió de tipus «Basic». Per exemple, dins Postman les cridades al mètode de tipus POST <http://localhost:8080/goibwstest/rest/persona/add> es configuren de la manera següent:





## 7. Errors COMUNS

### 7.1. Versió incorrecta de JDK

Si durant l'inici del JBoss o del Keycloak aparegués l'error:

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

S'ha de revisar que es té configurat correctament la variable JAVA\_HOME amb la comanda d'execució **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 d'autenticació

Si aparegués un error de credencials al accedir al backoffice o a l'API REST de l'aplicació, s'ha de revisar:

1. Que l'usuari estigui creat dins del Keycloak.
2. **Que tengui assignat una contrasenya.**
3. Que tengui assignat el rol adequat dins del client corresponent.
4. Que el contingut del fitxer JBOSS\_HOME\standalone\configuration\standalone.xml sigui correcte (veure exemple a la secció 5.1).
5. Que el fitxer *web.xml* del mòdul web i la securització dels EJBs estiguin ben configurats (per més informació, veure el capítol «5. Seguretat» del document «Estàndards d'Aplicacions Jakarta EE»).

### 7.3. Activació de serveis Jakarta EE addicionals

Cal tenir en compte que el fitxer JBOSS\_HOME\standalone\configuration\standalone.xml inicia el JBoss amb una configuració bàsica que no inclou tots els serveis disponibles de Jakarta EE; per exemple, el més destacable és que no inclou JMS/MDB.

La configuració que inclou aquests serveis es troba dins del fitxer **JBOSS\_HOME\standalone\configuration\standalone-full.xml**. Per iniciar amb aquesta configuració cal afegint l'opció "-c" amb el nom de la configuració (per exemple: **JBOSS\_HOME\bin\standalone.bat/.sh -c standalone-full.xml**).

#### **7.4. Context de només lectura**

Si durant l'inici del JBoss aparegués l'error «Context de només lectura» en el WeldStartService, s'ha de afegir el paràmetre **require-bean-descriptor="true"** en el subsistema Weld del fitxer JBOSS\_HOME\standalone\configuration\standalone.xml.

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