

DESPLIEGUE DE SOFTWARE EN ENTORNO DE DESARROLLO	
1. INFORMACIÓN GENERAL	
Requerimiento:	00177-2025
Denominación del Software:	SISTEMA DE GESTION DE ACCESOS
Fecha y Hora de Despliegue:	09/06/2025 20.30
2. DOCUMENTOS ASOCIADOS	
✓ TDR Servicio de Implementación de Solución de Seguridad SENACE.pdf.	
3. RECURSOS	
3.1 Hardware	
Tipo de Servidor	SERVIDOR LINUX
Funciones	<ul style="list-style-type: none"> Entorno de Desarrollo
Hardware	<ul style="list-style-type: none"> 16 GB RAM
Software	<ul style="list-style-type: none"> Oracle Linux Server
3.2 Software	
RECURSO	DESCRIPCIÓN
Oracle Linux	Sistema Operativo del servidor
Docker Engine	Motor de contenedores
Docker Compose	Herramienta de orquestación para contenedores
VPN	Acceso seguro al entorno de desarrollo
IP ADDRESS	172.16.75.231
4. INSTALACIONES Y CONFIGURACIONES	

4.1 Instalación de Docker en Linux

Instalar Docker siguiendo los pasos adecuados para Oracle Linux:

- Instalar los paquetes de configuración de repositorios oficiales de Oracle Linux 8.
sudo yum install -y oracle-linux-release-el8
- Instalar el EPEL (Extra Packages for Enterprise Linux) para Oracle Linux 8.)
sudo yum install -y oracle-epel-release-el8
- Activar el repositorio ol8_addons, que contiene herramientas complementarias del sistema operativo Oracle Linux 8.
sudo yum-config-manager --enable ol8_addons
- Agregar el repositorio oficial de Docker para CentOS compatible con Oracle Linux.
sudo dnf config-manager --add-repo=https://download.docker.com/linux/centos/docker-ce.repo
- Instalar Docker Community Edition (CE), la interfaz de línea de comandos de Docker (CLI) y containerd que es el runtime de contenedores
sudo dnf install docker-ce docker-ce-cli containerd.io
- Configurar Docker para que se inicie automáticamente al prender el servidor
sudo systemctl enable docker
- Iniciar el servicio de Docker de inmediato
sudo systemctl start docker
- Agregar al usuario actual al grupo docker
sudo usermod -aG docker \$USER

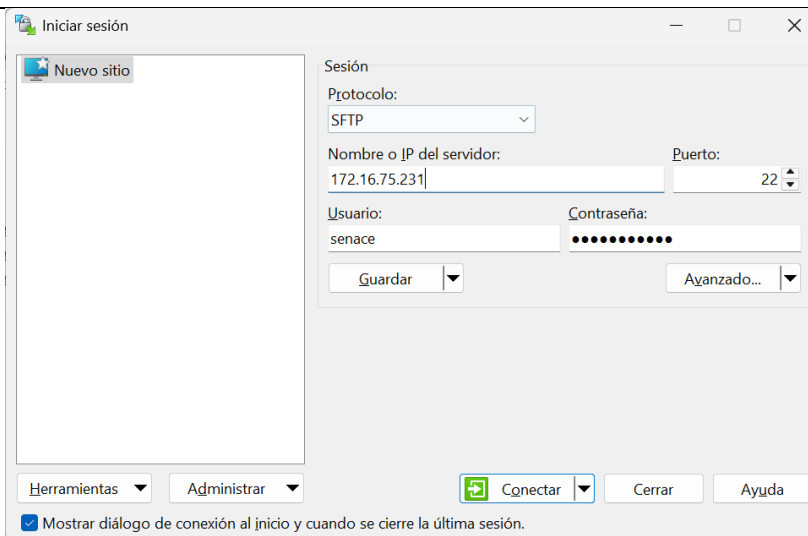
Con fines de probar que la instalación fue satisfactoria ejecutamos el comando **docker --version** debiendo tener la versión adecuada como se ve en la imagen

```
[senace@snc-ss0-des0 ~]$ docker --version
Docker version 26.1.3, build b72abbb
```

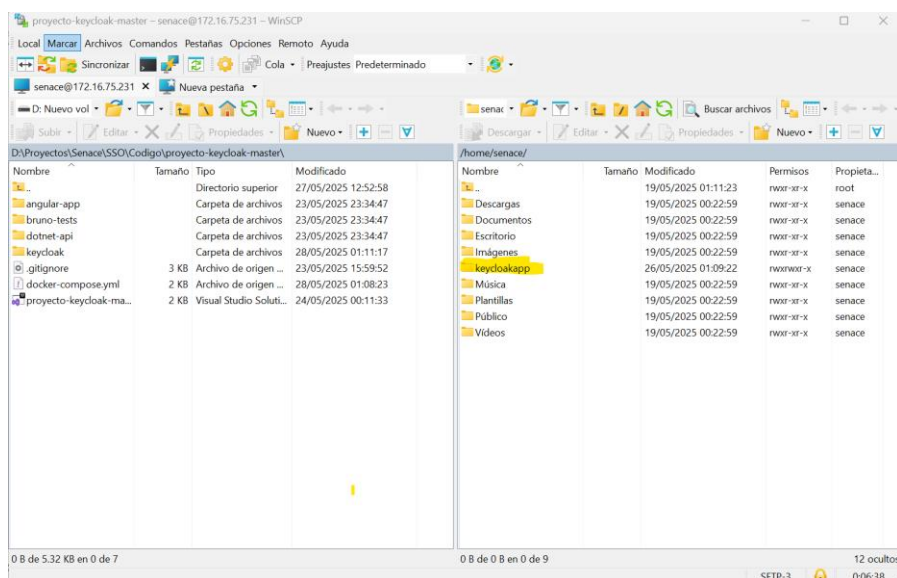
4.2 Despliegue de aplicación

4.2.1 Conexión mediante WinSCP

Abrir la aplicación e ingresar las credenciales:



Copiar el contenido del archivo compartido hacia la carpeta creada en el servidor Linux.

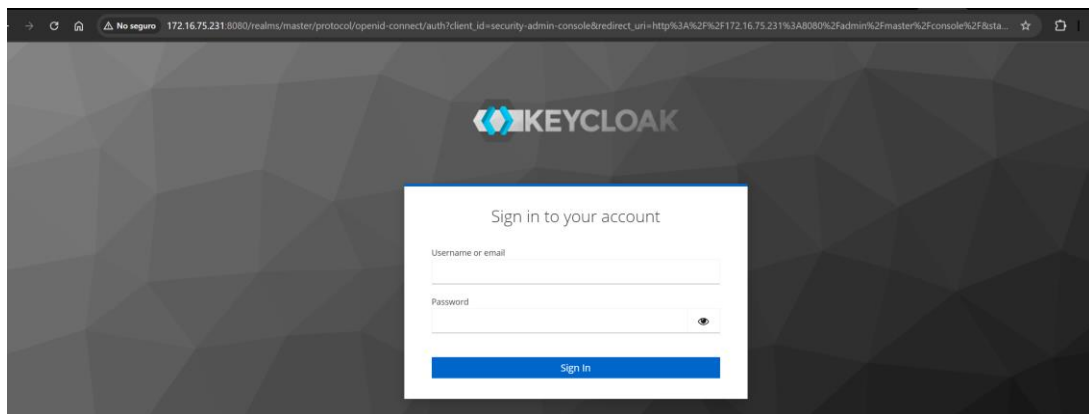


Volver a la consola y posicionarse en la ruta donde está el contenido copiado. Sobre esta ruta, ejecutar el comando **docker-compose up --build -d**



```
[senace@nc-ss0-desa-proyecto-keycloak-master]# sudo docker compose up --build --d
[+] Building 46.2s (16/24)
=> [dotnetapi build 1/4] FROM mcr.microsoft.com/dotnet/sdk:8.0sha256:e6a5a8d84609907fa8d468927d765967f6b22f890ce92bd3ae614ca4ae87e 44.9s
=> sha256:19c1e22f6e65829627f5077f53ace1aeb658c4fa83b682901aeb0b3 154B / 154B 0.0s
=> sha256:c7b7b706c55870bc2343909194d9ba7cca4d818d5fa3c1f53ce41ae1b652c5d 18.73MB / 18.73MB 12.3s
=> sha256:4e39187fd9a9654377c6675318b02d0b556036224a5f01780918e2036a2e6ac 32.25MB / 32.25MB 21.7s
=> sha256:b69e1e22f6e65829627f5077f53ace1aeb658c4fa83b682901aeb0b3 154B / 154B 12.5s
=> sha256:ef919660cf490da619ebe5eedf2ddcbea1e57760bacc8680a30ea2515f7236 11.08MB / 11.08MB 18.3s
=> extracting sha256:dad67da3f26bce15939543965e09c405953b025f707aad72ed3d3fa09c66f8 1.4s
=> sha256:21ae5ae03a98523d090408b2c9ea39bbbbb4f388dcfa02b8b90319aac03ded 2.28kB / 2.28kB 30.0s
=> extracting sha256:c7b7b706c55870bc2343909194d9ba7cca4d818d5fa3c1f53ce41ae1b652c5d 0.5s
=> extracting sha256:f8a78fa15fface87144ed7c21ce9e22d4247a3181277685ab705bfb76304d9 0.0s
=> sha256:1cd4496a93825155d249ace050415059fee04ad64007235dddf2ae08d 50.33MB / 177.34MB 44.9s
=> extracting sha256:4e39187fd9a9654377c6675318b02d0b556036224a5f01780918e2036a2e6ac 0.7s
=> sha256:12577d033b0c8a8d373b05c99ae43fb41d0fa1960f38b9aff053d79a97e3d8c0 2.64kB / 2.64kB 21.9s
=> sha256:bc37b2ebd5bbe2f504af05d720d018e13e9b51d0c1ae0fb59af52205118d7fd 16.97MB / 16.97MB 35.2s
=> extracting sha256:b69e1e22f6e65829627f5077f53ace1aeb658c4fa83b682901aeb0b3 0.0s
=> extracting sha256:ef919660cf490da619ebe5eedf2ddcbea1e57760bacc8680a30ea2515f7236 0.2s
=> extracting sha256:814c5ae03a98523d090408b2c9ea39bbbbb4f388dcfa02b8b90319aac03ded 1.2s
[dotnetapi stage-1 1/3] FROM mcr.microsoft.com/dotnet/aspnet:8.0sha256:d5c0d01bc8fe887684b9763409036270ed78cd2a5121436e842a8114e64d584 22.8s
=> resolve mcr.microsoft.com/dotnet/aspnet:8.0sha256:d5c0d01bc8fe887684b9763409036270ed78cd2a5121436e842a8114e64d584 0.0s
=> sha256:45d88189dd454102e8e94a6a1974b9a02817b9142046dfdb4a7c6a5d441017d 1.58kB / 1.58kB 0.0s
=> sha256:dad67da3f26bce15939543965e09c405953b025f707aad72ed3d3fa09c66f8 28.23MB / 28.23MB 14.0s
=> sha256:f8a78fa15fface87144ed7c21ce9e22d4247a3181277685ab705bfb76304d9 3.28kB / 3.28kB 0.5s
=> sha256:d5cd9d1bc9fe887684b9763409036270ed78cd2a5121436e842a8114e64d584 1.08kB / 1.08kB 0.0s
=> sha256:5c2c398550182e42b11bd46be6d3b9f98b3d03073747370647e6872f426d257 2.52kB / 2.52kB 0.0s
=> sha256:c7b7b706c55870bc2343909194d9ba7cca4d818d5fa3c1f53ce41ae1b652c5d 18.73MB / 18.73MB 12.3s
=> sha256:4e39187fd9a9654377c6675318b02d0b556036224a5f01780918e2036a2e6ac 32.25MB / 32.25MB 21.7s
=> sha256:b69e1e22f6e65829627f5077f53ace1aeb658c4fa83b682901aeb0b3 154B / 154B 12.5s
=> sha256:ef919660cf490da619ebe5eedf2ddcbea1e57760bacc8680a30ea2515f7236 11.08MB / 11.08MB 18.3s
=> extracting sha256:f8a78fa15fface87144ed7c21ce9e22d4247a3181277685ab705bfb76304d9 0.0s
=> extracting sha256:b69e1e22f6e65829627f5077f53ace1aeb658c4fa83b682901aeb0b3 0.2s
=> extracting sha256:ef919660cf490da619ebe5eedf2ddcbea1e57760bacc8680a30ea2515f7236 0.2s
[dotnetapi internal] load build context
=> transferring context: 2.08kB 0.0s
[angular internal] load dockerignore
=> transferring context: 2B 0.0s
[angular build 1/4] FROM docker.io/library/node:20-alpinesha256:d3507a213936fe4ef54760a186e113db5188472d9efdf491686bd94580a1c1e8 33.8s
=> resolve docker.io/library/node:20-alpinesha256:d3507a213936fe4ef54760a186e113db5188472d9efdf491686bd94580a1c1e8 0.0s
=> sha256:d3507a213936fe4ef54760a186e113db5188472d9efdf491686bd94580a1c1e8 7.67kB / 7.67kB 0.0s
=> sha256:af573be8b995ef79d83764af1d607c1edbc97139090246edc8775ea1d3b072ed 1.72kB / 1.72kB 0.0s
=> sha256:367a28bb5439cb0fdb1c9a3ab6275e84flec7ab8c61d8e89580ebd2bd7f00b54 6.21kB / 6.21kB 0.0s
=> sha256:fe07684d1b82247c3539ed86a5ff37a76138ec25d380bd80c809a1a4c73236 3.80MB / 3.80MB 29.6s
=> extracting sha256:fe07684d1b82247c3539ed86a5ff37a76138ec25d380bd80c809a1a4c73236 0.2s
=> sha256:1cd4496a93825155d249ace050415059fee04ad64007235dddf2ae08d 50.33MB / 177.34MB 44.9s
```

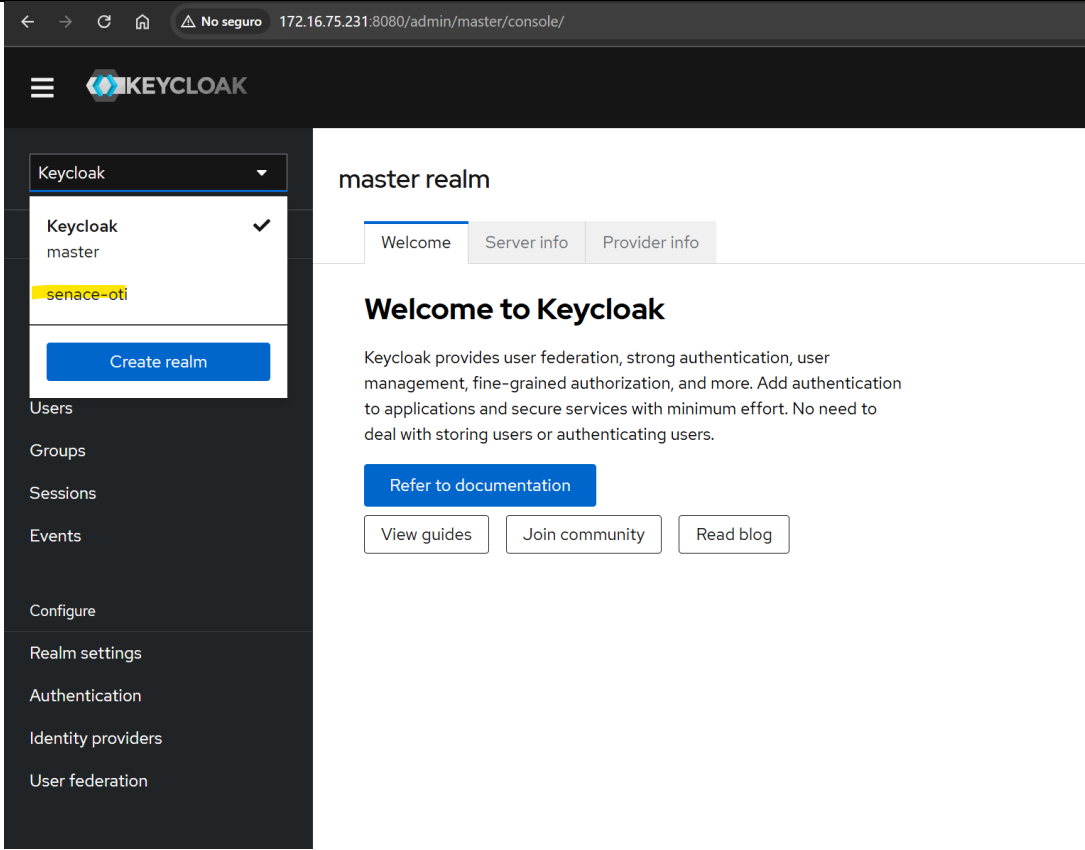
Luego de completada el comando, ingresaremos a la url <http://172.16.75.231:8080> en el navegador local de la PC y veremos la imagen del Keycloak ejecutándose



4.3 Pruebas

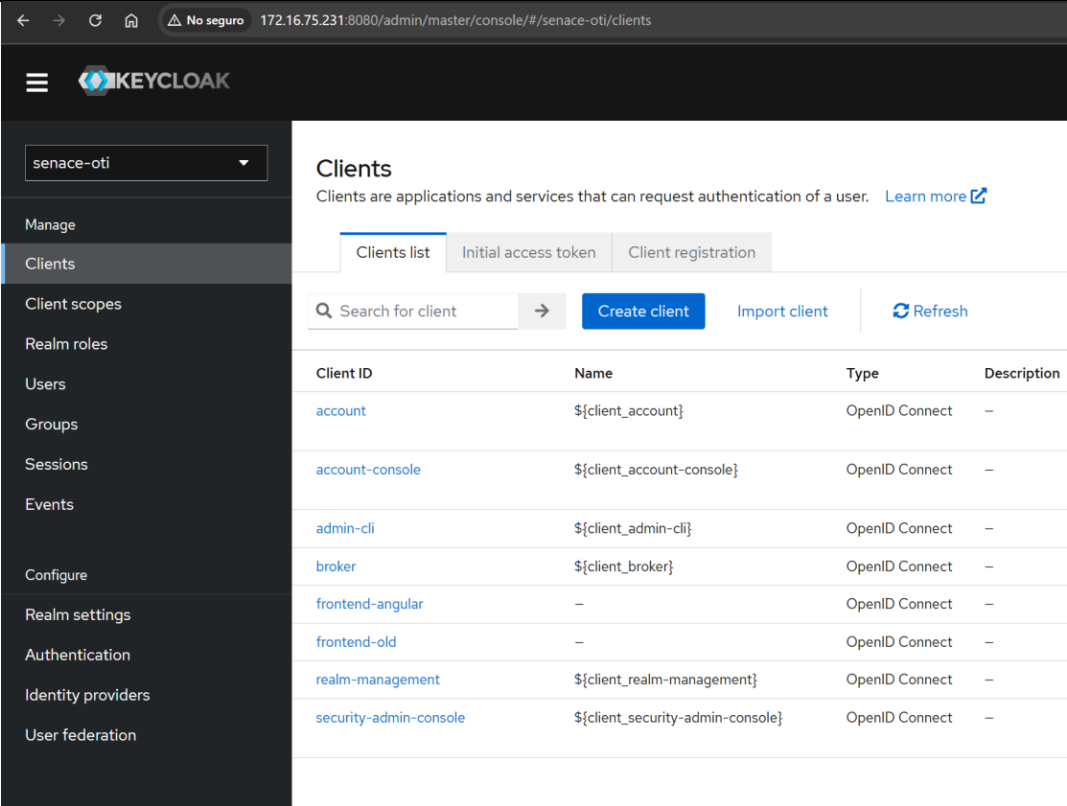
4.3.1 Autenticacion

Luego de tener la página activa, ingresar con las credenciales de prueba. Usuario: **admin** y clave **admin**. Se debe mostrar la ventana de configuración de Keycloak, en la que se debe poder ingresar a ver el Realm personalizado creado (senace-oti, como se muestra en la imagen).



Validar que cumpla los requisitos siguientes requisitos:

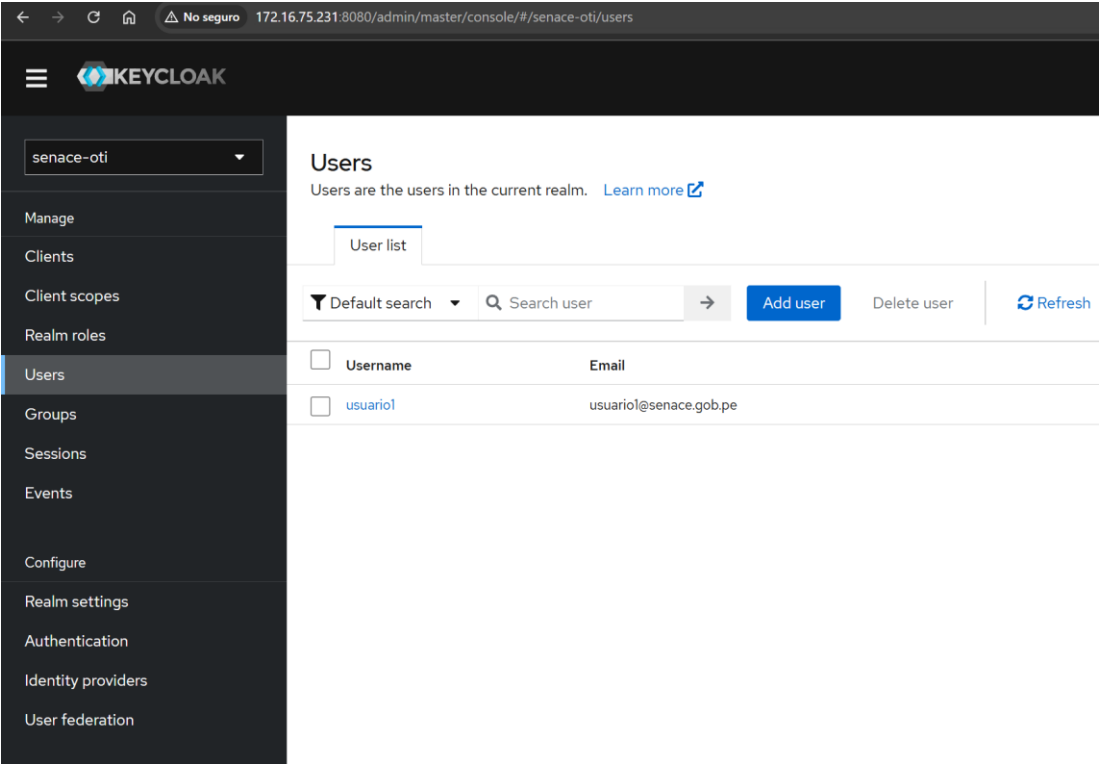
- Registro de aplicaciones
- Registro de Clientes.



The screenshot shows the Keycloak administration console for the 'senace-oti' realm. The left sidebar contains navigation options: Manage, Clients (selected), Client scopes, Realm roles, Users, Groups, Sessions, Events, Configure, Realm settings, Authentication, Identity providers, and User federation. The main content area is titled 'Clients' and includes a description: 'Clients are applications and services that can request authentication of a user.' Below this are tabs for 'Clients list' (active), 'Initial access token', and 'Client registration'. A search bar and buttons for 'Create client', 'Import client', and 'Refresh' are present. A table lists the following clients:

Client ID	Name	Type	Description
account	`\${client_account}`	OpenID Connect	–
account-console	`\${client_account-console}`	OpenID Connect	–
admin-cli	`\${client_admin-cli}`	OpenID Connect	–
broker	`\${client_broker}`	OpenID Connect	–
frontend-angular	–	OpenID Connect	–
frontend-old	–	OpenID Connect	–
realm-management	`\${client_realm-management}`	OpenID Connect	–
security-admin-console	`\${client_security-admin-console}`	OpenID Connect	–

Registrar los usuarios que sean necesarios.



The screenshot shows the Keycloak administration console for the 'senace-oti' realm, specifically the 'Users' section. The left sidebar is the same as the previous screenshot, with 'Users' selected. The main content area is titled 'Users' and includes a description: 'Users are the users in the current realm.' Below this is a 'User list' tab. A search bar and buttons for 'Add user', 'Delete user', and 'Refresh' are present. A table lists the following user:

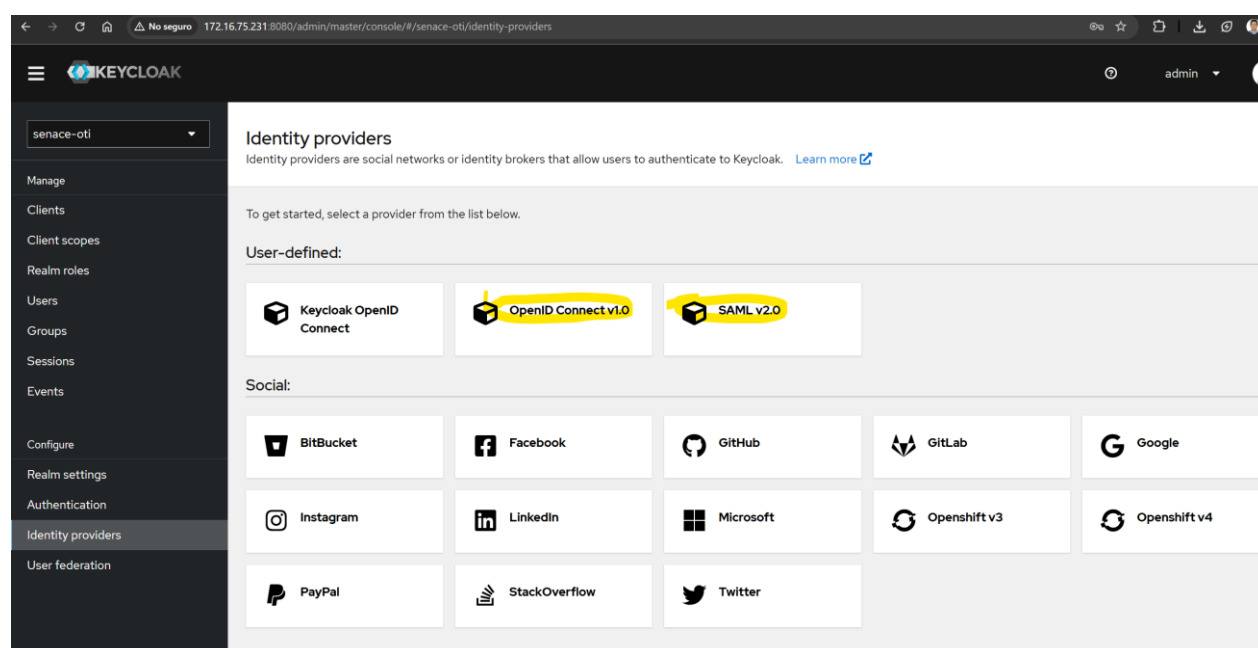
Username	Email
usuario1	usuario1@senace.gob.pe

4.3.2 Evidencias según TDR:


Con fines de poder validar lo requerido en el TDR en la sección 4.2.c se lista los siguientes puntos y sus evidencias:

- i. **Compatibilidad con Estándares Abiertos: Soporte para protocolos como SAML 2.0, OAuth y OpenID Connect, facilitando la interoperabilidad con diversas aplicaciones.**

En la sección de **Identity Providers** encontraremos las opciones de Federación con proveedores basados en OAuth .20















Por ejemplo, si hacemos clic en la opción OpenId Connect podríamos entrar a configurar una integración de login con un tercero



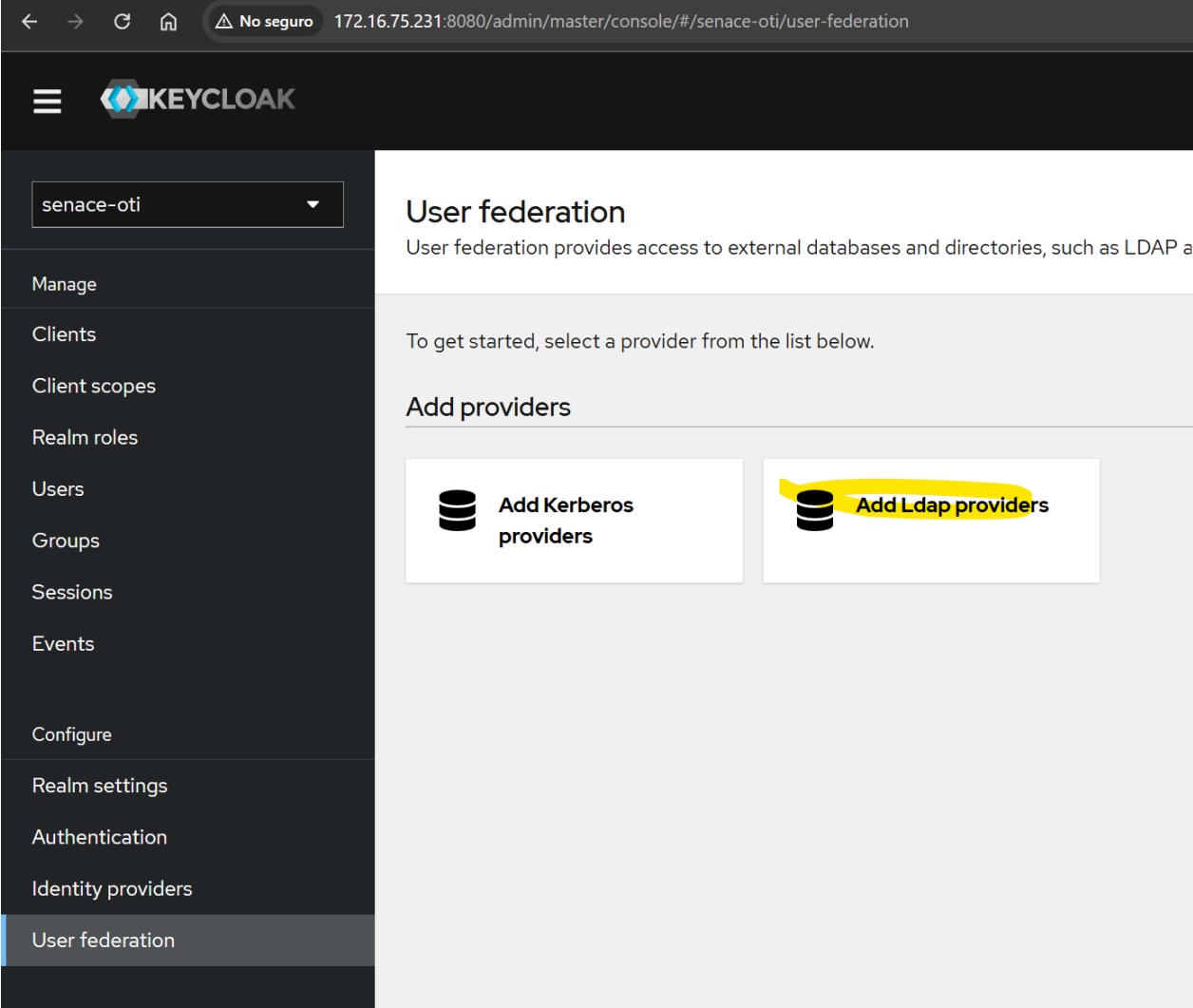
senace-oti

Manage
 Clients
 Client scopes
 Realm roles
 Users
 Groups
 Sessions
 Events
 Configure
 Realm settings
 Authentication
 Identity providers
 User federation

Identity providers > Add OpenID Connect provider
 Add OpenID Connect provider
 Redirect URI  http://172.16.75.231:8080/realms/senace-oti/broker/oidc/endpoint 
 Alias *  oidc
 Display name 
 Display order 
 OpenID Connect settings
 Use discovery  On
 Use discovery endpoint 
 Discovery endpoint *  https://hostname/auth/realms/master/.well-known/openid-configuration
 > [Show metadata](#)
 Client authentication  Client secret sent as post
 Client ID * 
 Client Secret *  

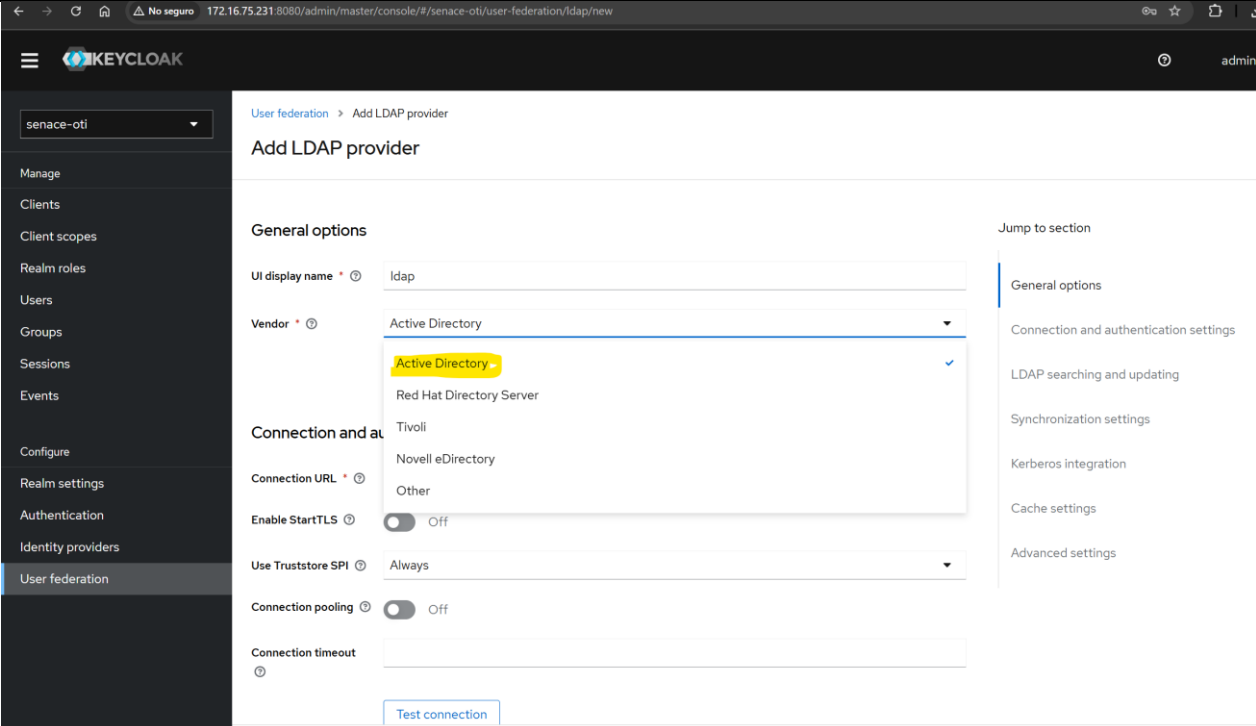
ii. **Integración con Directorios de Usuarios: Capacidad para conectarse con servicios como Active Directory u otros sistemas de gestión de identidades.**

En la sección User Federation podemos encontrar las secciones que nos permitirán integrarnos con diferentes proveedores de LDAP.



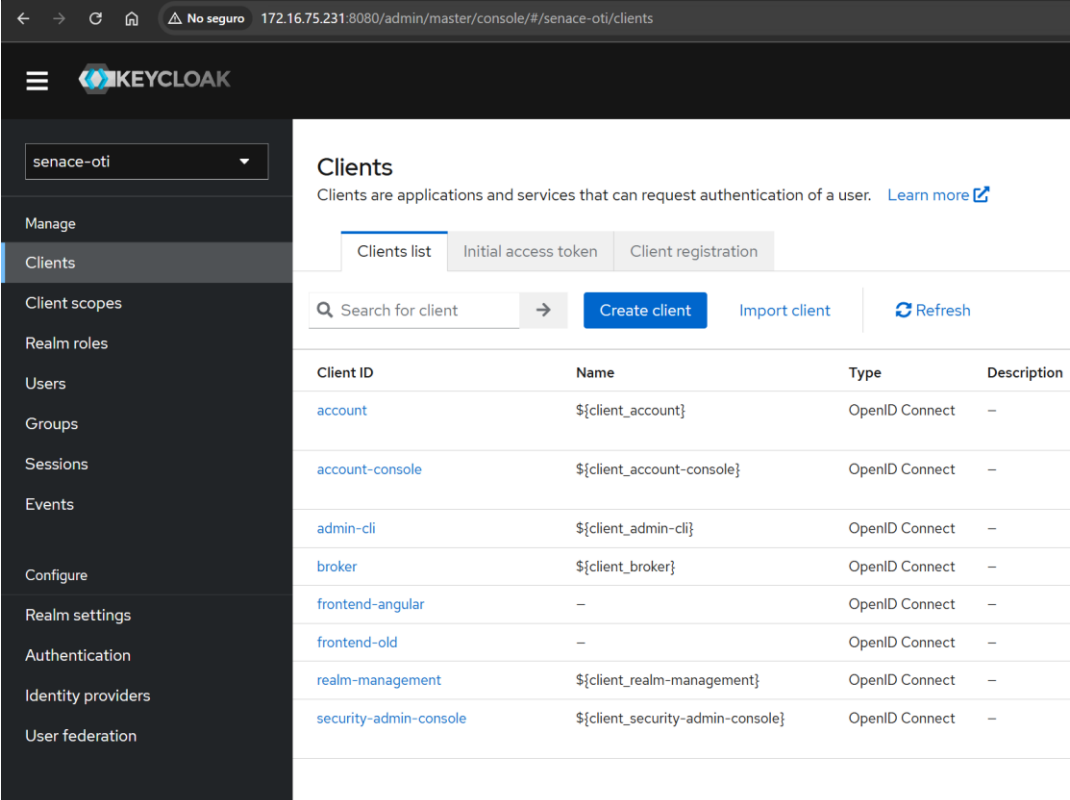
The screenshot shows the Keycloak administration console for the realm 'senace-oti'. The left sidebar contains a menu with options: Manage, Clients, Client scopes, Realm roles, Users, Groups, Sessions, Events, Configure, Realm settings, Authentication, Identity providers, and User federation (which is highlighted). The main content area is titled 'User federation' and includes a description: 'User federation provides access to external databases and directories, such as LDAP and Active Directory'. Below this, it says 'To get started, select a provider from the list below.' and 'Add providers'. There are two buttons: 'Add Kerberos providers' and 'Add Ldap providers'. The 'Add Ldap providers' button is highlighted with a yellow circle.

Por ejemplo, si ingresamos a la opción **Add Ldap Providers** encontraremos las opciones para federarnos a un Active Directory y otros servicios similares:



iii. Interfaz de Administración Centralizada: Herramienta que permita gestionar usuarios y políticas de acceso basados en roles o permisos.

Como interfaz centralizada podemos ver que se pueden configurar las aplicaciones (Clients)



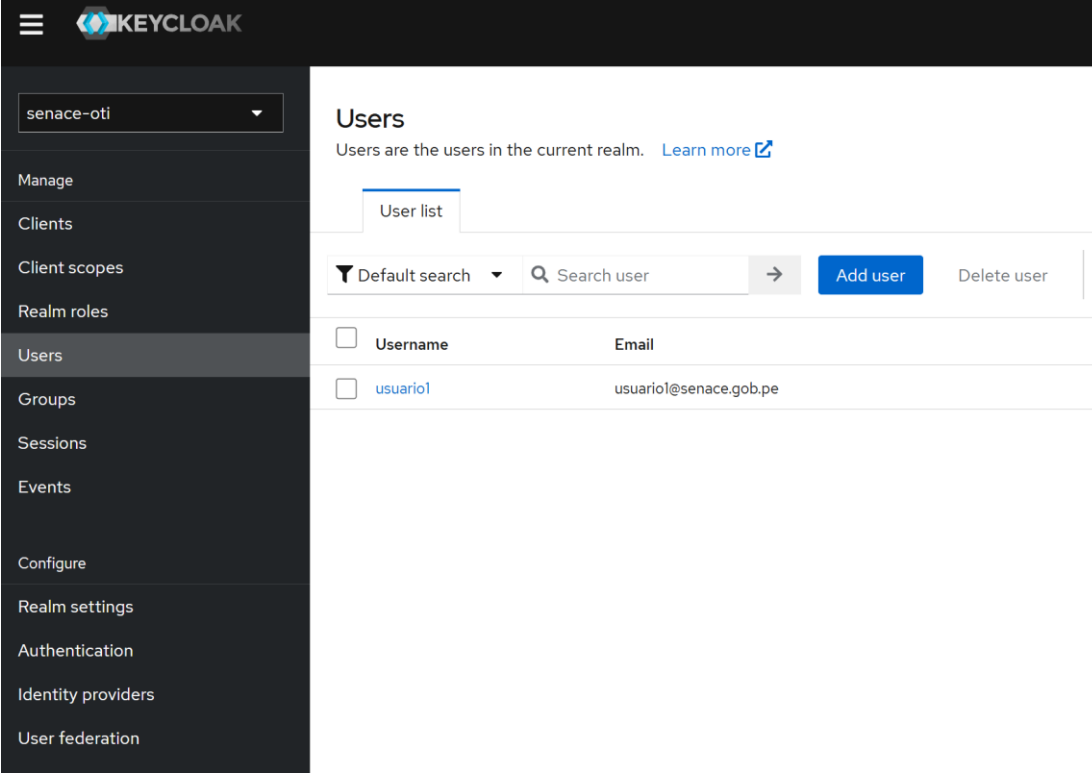
Clients
Clients are applications and services that can request authentication of a user. [Learn more](#)

Clients list | Initial access token | Client registration

Search for client → [Create client](#) [Import client](#) [Refresh](#)

Client ID	Name	Type	Description
account	\${client_account}	OpenID Connect	–
account-console	\${client_account-console}	OpenID Connect	–
admin-cli	\${client_admin-cli}	OpenID Connect	–
broker	\${client_broker}	OpenID Connect	–
frontend-angular	–	OpenID Connect	–
frontend-old	–	OpenID Connect	–
realm-management	\${client_realm-management}	OpenID Connect	–
security-admin-console	\${client_security-admin-console}	OpenID Connect	–

Puedo trabajar con usuarios



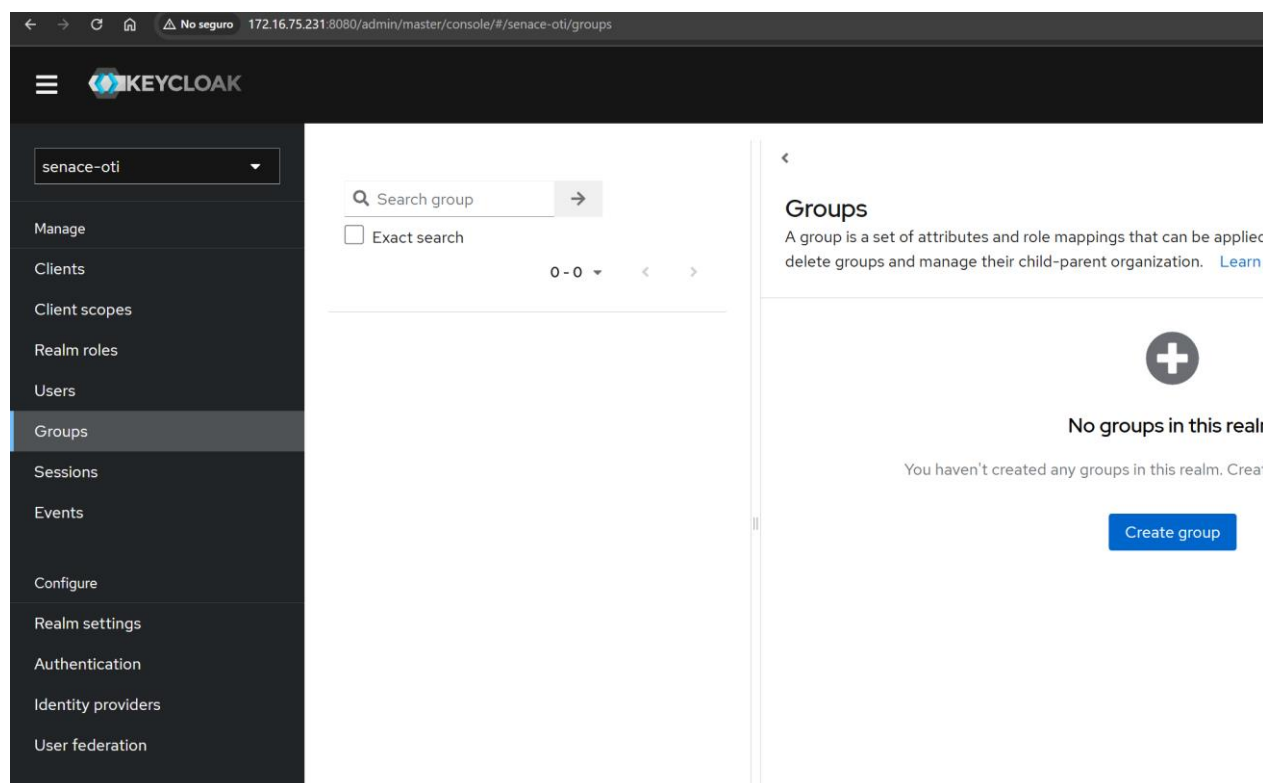
Users
Users are the users in the current realm. [Learn more](#)

User list

Default search Search user → [Add user](#) [Delete user](#)

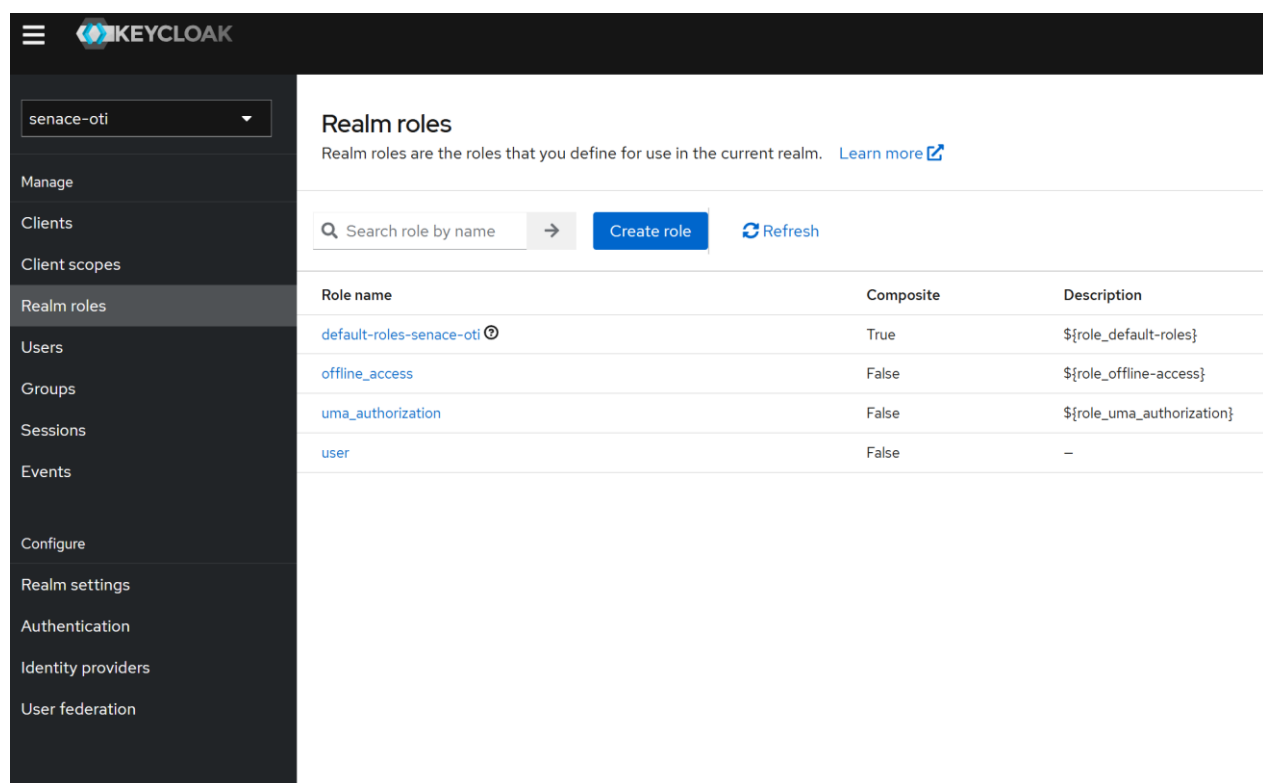
<input type="checkbox"/> Username	Email
<input type="checkbox"/> usuario1	usuario1@senace.gob.pe

Grupo de usuarios:



The screenshot shows the Keycloak administration console for the 'senace-oti' realm. The left sidebar contains a menu with options: Manage, Clients, Client scopes, Realm roles, Users, Groups (selected), Sessions, Events, Configure, Realm settings, Authentication, Identity providers, and User federation. The main content area is titled 'Groups' and displays 'No groups in this realm'. It includes a search bar, an 'Exact search' checkbox, and a 'Create group' button. A message states: 'You haven't created any groups in this realm. Create'.

Puedo gestionar roles de acceso al Realm

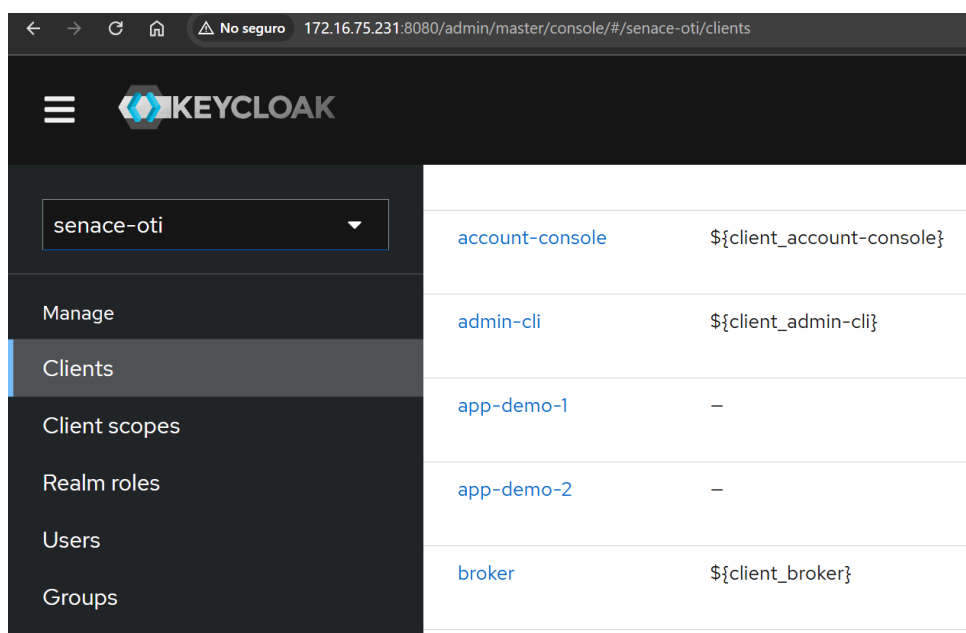


The screenshot shows the Keycloak administration console for the 'senace-oti' realm, specifically the 'Realm roles' page. The left sidebar is identical to the previous screenshot, with 'Realm roles' selected. The main content area is titled 'Realm roles' and includes a description: 'Realm roles are the roles that you define for use in the current realm. Learn more'. Below this is a search bar, a 'Create role' button, and a 'Refresh' button. A table lists the roles:

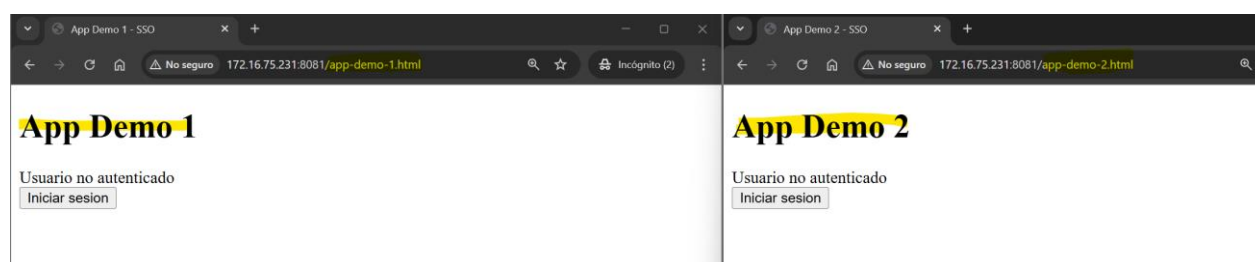
Role name	Composite	Description
default-roles-senace-oti	True	`\${role_default-roles}`
offline_access	False	`\${role_offline-access}`
uma_authorization	False	`\${role_uma_authorization}`
user	False	—

iv. Gestión de sesión única: Una vez autenticado el usuario podrá acceder a otras aplicaciones sin necesidad de ingresar nuevamente las credenciales.

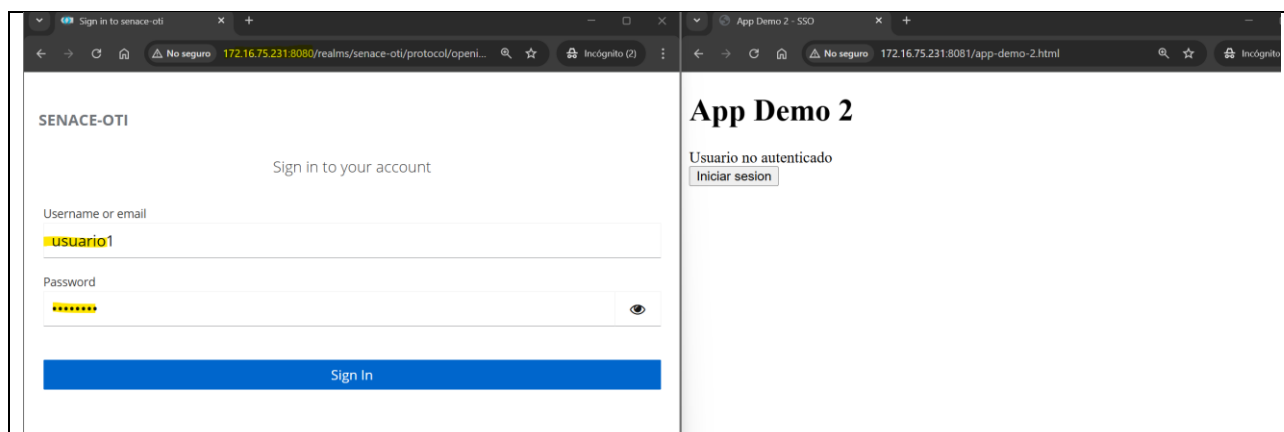
Se puede probar con dos aplicaciones de prueba llamada app-demo-1 (<http://172.16.75.231:8081/app-demo-1.html>) y app-demo-2 (<http://172.16.75.231:8081/app-demo-2.html>) Ambas están registradas en el KeyCloak



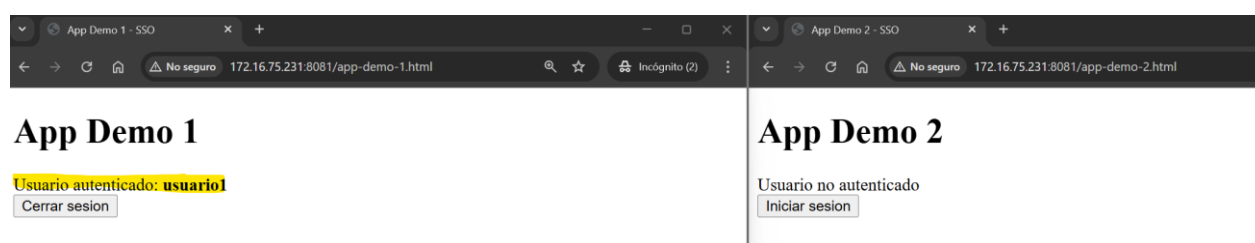
Abrimos ambas aplicaciones en ventanas compartidas confirmando que ninguna tiene autenticación:



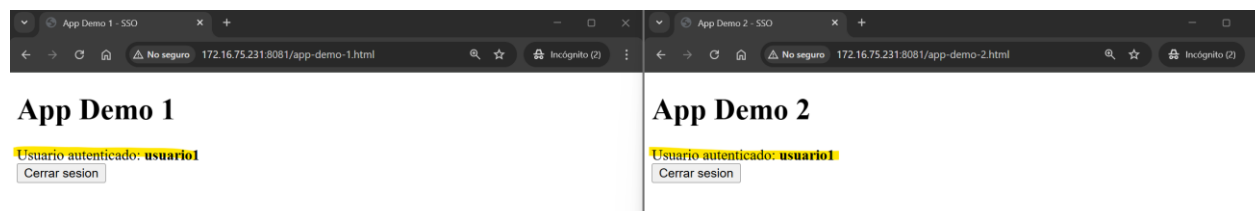
En la app-demo-1 iniciaremos sesión y nos redirigirá a la venta de login centralizada y en la que ingresamos el usuario y clave:



Luego de validar el login regresara a la app-demo-1 pero indicando el usuario autenticado

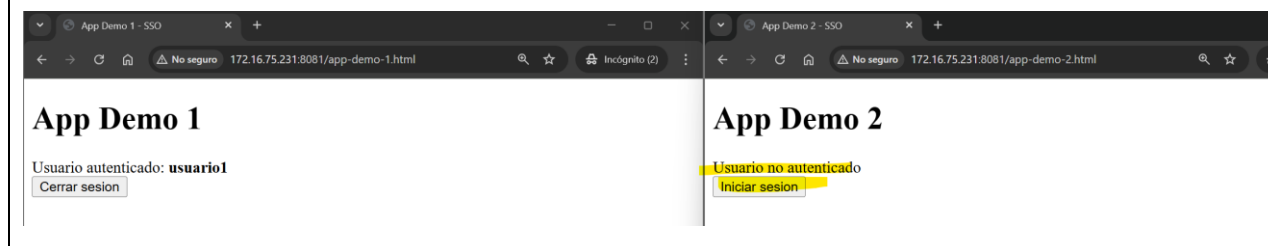


Haremos clic en Iniciar sesión de app-demo-2: y automáticamente inicia sesión sin pedirme las credenciales

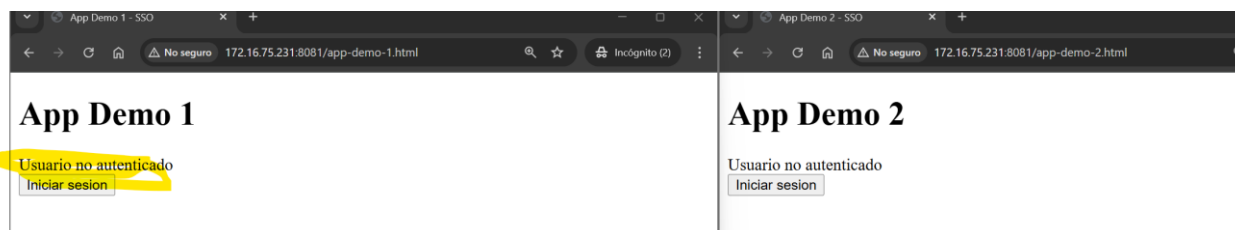


- v. **Token de sesión:** Luego de verificar la identidad, el SSO debe generar un token de sesión (como JWT o SAML), el cual será propagado a todas las aplicaciones conectadas. El token debe tener un tiempo de expiración y debe ser renovado para mantener la sesión activa. Al cerrar la sesión, debe finalizar la sesión de todas las aplicaciones.

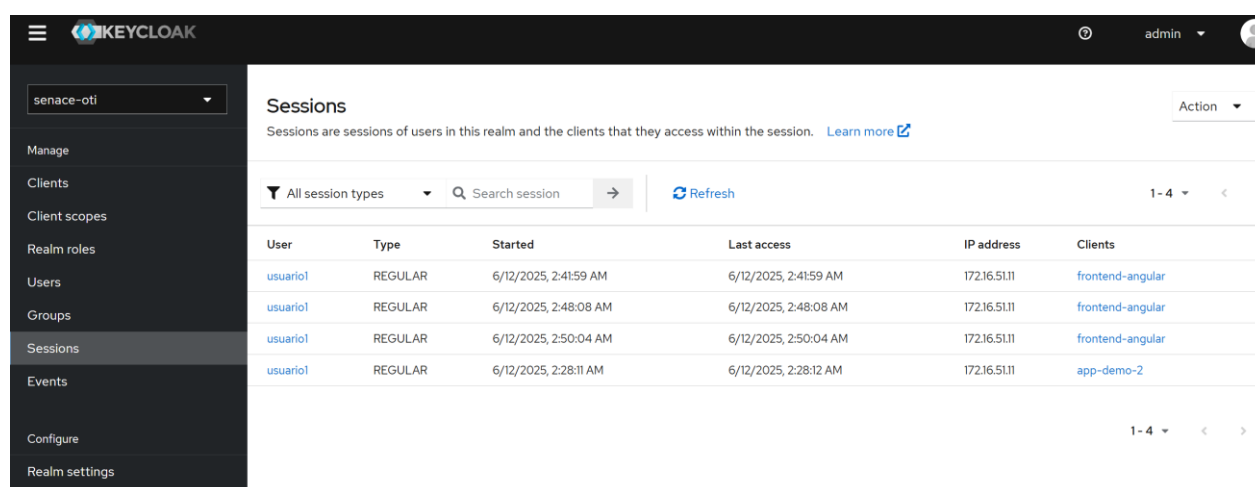
Siguiente el ejercicio anterior, haremos clic en Cerrar Sesión del app-demo-2 el cual hará un redirect rápido el SSO y luego retornará:



Para verificar que sucedió con la app-demo-1 podemos presionar F5 para refrescar la pagina y notaremos que verifico que no hay sesión en el SSO y muestra la opción de iniciar sesion



Debemos mencionar que el SSO Keycloak administra la sesiones pudiendo revocarlas manualmente o por aplicación



vi. APIs del SSO: El SSO debe contar con APIs Rest como interfaces de comunicación para ser consumidas desde otros sistemas.

El SSO con keycloak de forma nativa expone servicios rest para poder exponer información del usuario y que estas puedan ser consumidas desde otro sistema.

Por ejemplo, si generamos un Access token para validar a un usuario enviando su contraseña y la aplicación podríamos hacerlo con el siguiente comando CURL

```
curl --location 'http://172.16.75.231:8080/realms/senace-oti/protocol/openid-connect/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=password' \
--data-urlencode 'client_id=frontend-angular' \
--data-urlencode 'username=usuario1' \
--data-urlencode 'password=clave123' \
--data-urlencode 'scope=openid'
```

Esto también puede ser ejecutado desde Postman, obteniendo el siguiente resultado:

Donde como respuesta tendremos los valores del usuario

GET

http://172.16.75.231:8080/realms/senace-oti/protocol/openid-connect/userinfo

Send

Params

Authorization

Headers (7)

Body

Scripts

Settings

Cookies

Headers

6 hidden

Key	Value	Description
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUiiwia2kiIA6ICJSUI...	
Key	Value	Description

Body

Cookies

Headers (8)

Test Results

200 OK • 174 ms • 483 B •

JSON

Preview

Visualize

```

1  {
2    "sub": "a4bad228-42bf-4edb-a34e-259bb06c6cf1",
3    "email_verified": true,
4    "name": "Juan Pérez",
5    "preferred_username": "usuario1",
6    "given_name": "Juan",
7    "family_name": "Pérez",
8    "email": "usuario1@senace.gob.pe"
9  }

```

En el segundo entregable se personalizará ese servicio para entregar datos adicionales.

vii. Compatibilidad con Aplicaciones: Asegurar que el SSO funcione correctamente en aplicaciones web y móviles.

En los puntos iv) y v) se hizo las pruebas con aplicaciones web. Para la parte de aplicaciones móviles se ha creado un Client para aplicaciones móviles. En esta configuración veremos que el Redirect no lo envía hacia un http o https como es una aplicación web sino hacia un myapp://callback

← → ↺ ⚠ No seguro 172.16.75.231:8080/admin/master/console/#/senace-oti/clients/732ec174-699c-4efd-97f1-43c16c0bfe2b/settings

☰

KEYCLOAK

senace-oti

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

Events

Configure

Realm settings

Authentication

Identity providers

User federation

app-demo-mobile OpenID Connect

Clients are applications and services that can request authentication of a user.

Settings

Roles

Client scopes

Sessions

Advanced

General settings

Client ID ⓘ app-demo-mobile

Name ⓘ

Description ⓘ

Always display in UI ⓘ ☐ Off

Access settings

Root URL ⓘ

Home URL ⓘ

Valid redirect URIs ⓘ myapp://callback

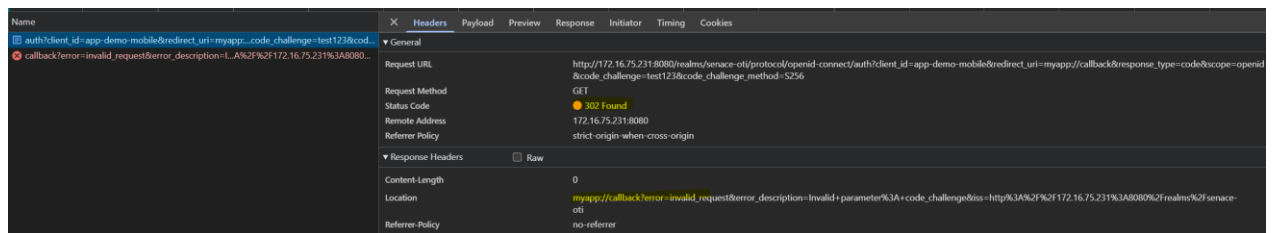
[+ Add valid redirect URIs](#)

Podemos hacer pruebas construyendo una ruta como la siguiente:

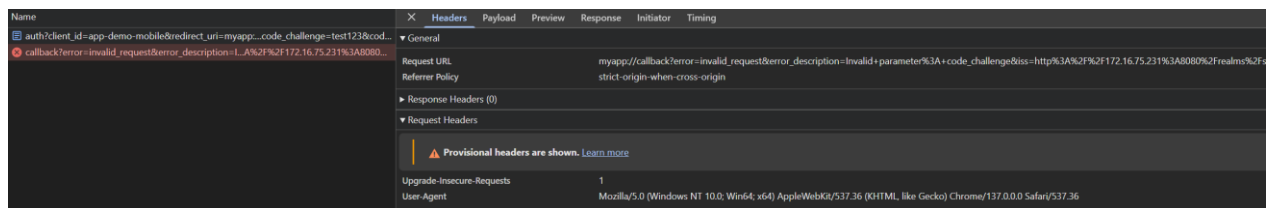
http://172.16.75.231:8080/realms/senace-oti/protocol/openid-connect/auth?client_id=app-demo-mobile&redirect_uri=myapp://callback&response_type=code&scope=openid&code_challenge=test123&code_challenge_method=S256

Donde se puede notar que estamos apuntando a la aplicación app-demo-mobile y le decimos que redireccione hacia myapp://callback

El navegador no esta preparado para resolver ese tipo de protocolos, pero si entramos a la herramienta de desarrollador presionando el F12 veremos que Keycloak si resolvió la llamada e intento redireccionar a la app.

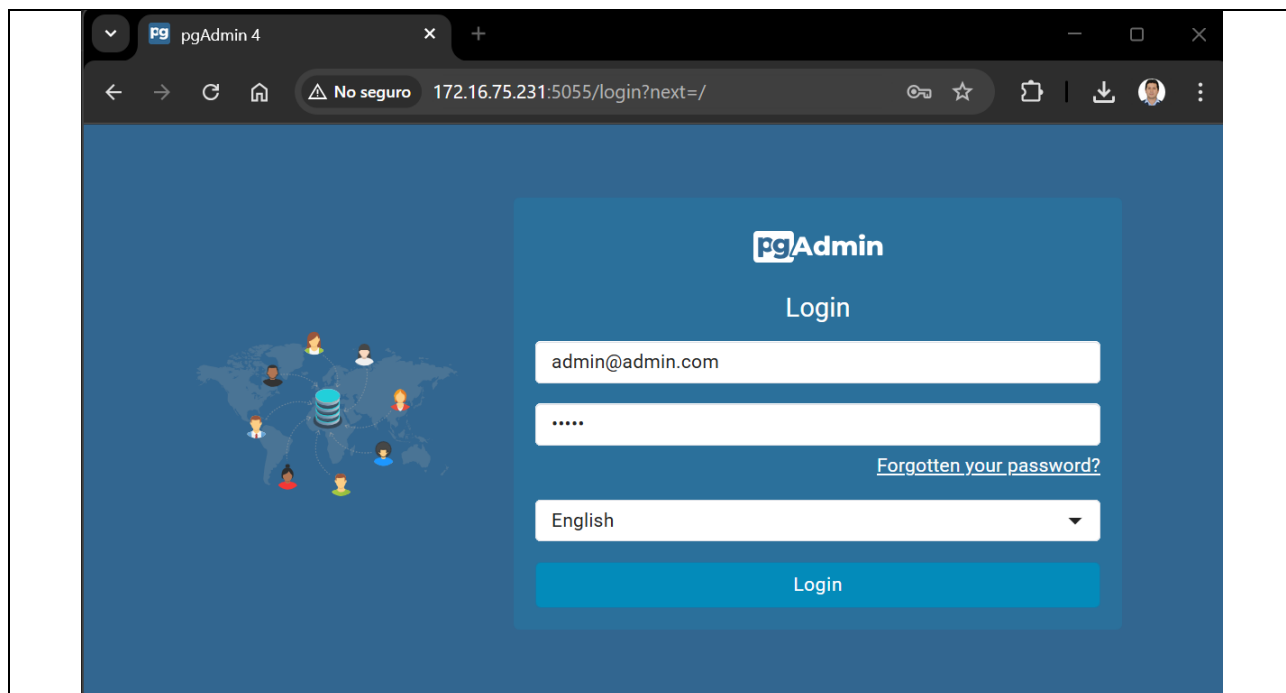


En la segunda línea donde se está haciendo el reenvío se encuentra que el navegador no pudo resolver esto. (Aunque cuando se prueba con una App si funcionara)

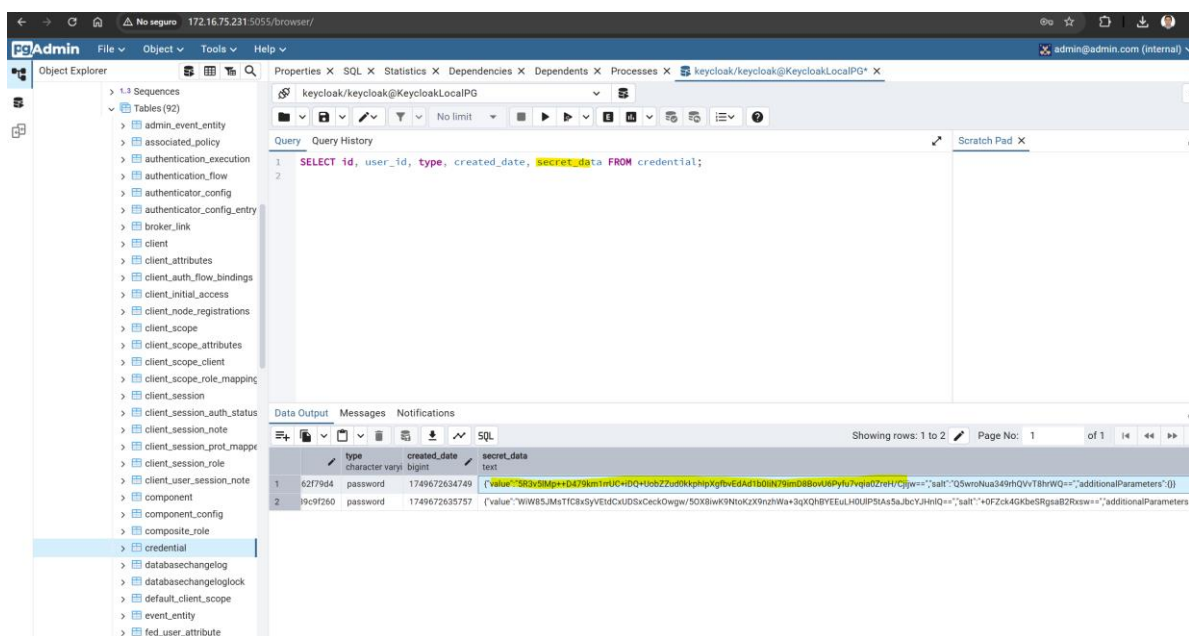


viii. Seguridad y protección de datos: El sistema debe garantizar que las credenciales y la sesión estén protegidas contra accesos no autorizados.

La credencial por defecto viene encriptada, si ingresamos a la url <http://172.16.75.231:5055> ingresaremos a la aplicación PGAdmin para poder hacer búsquedas sobre la BD de Keycloak

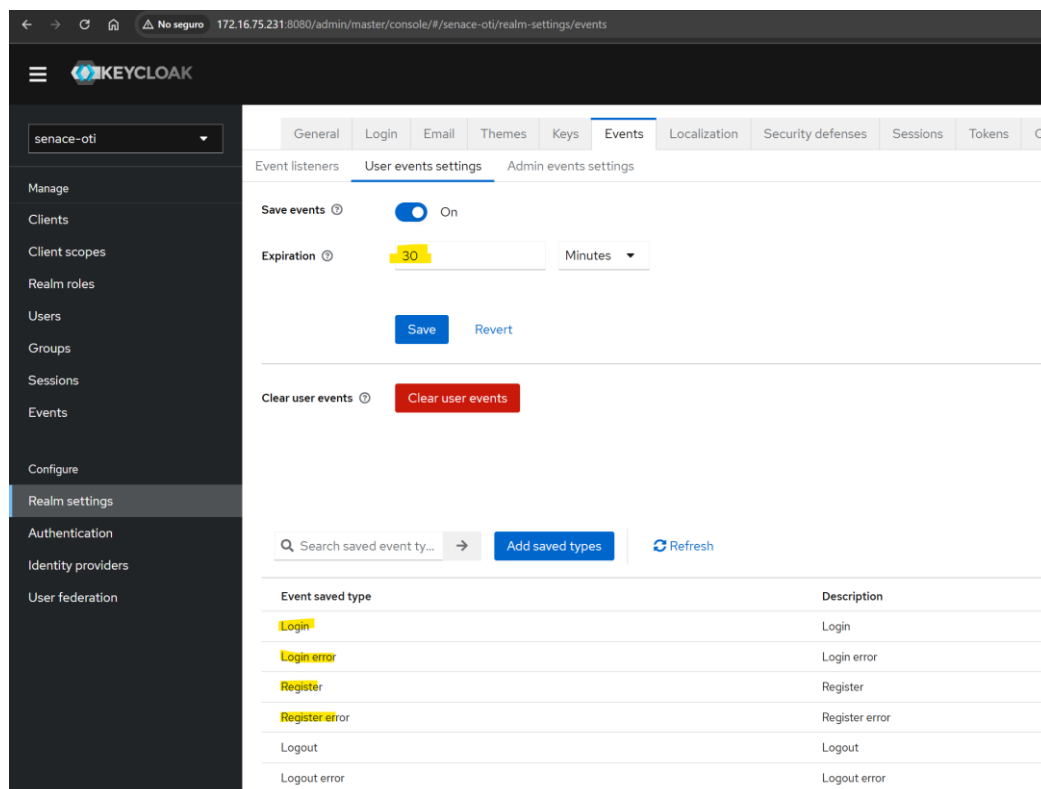


Luego de ingresar las credenciales de usuario, hacemos un Select a la tabla Credentials. Donde podremos ver que la columna secret_data tiene el valor encriptado.

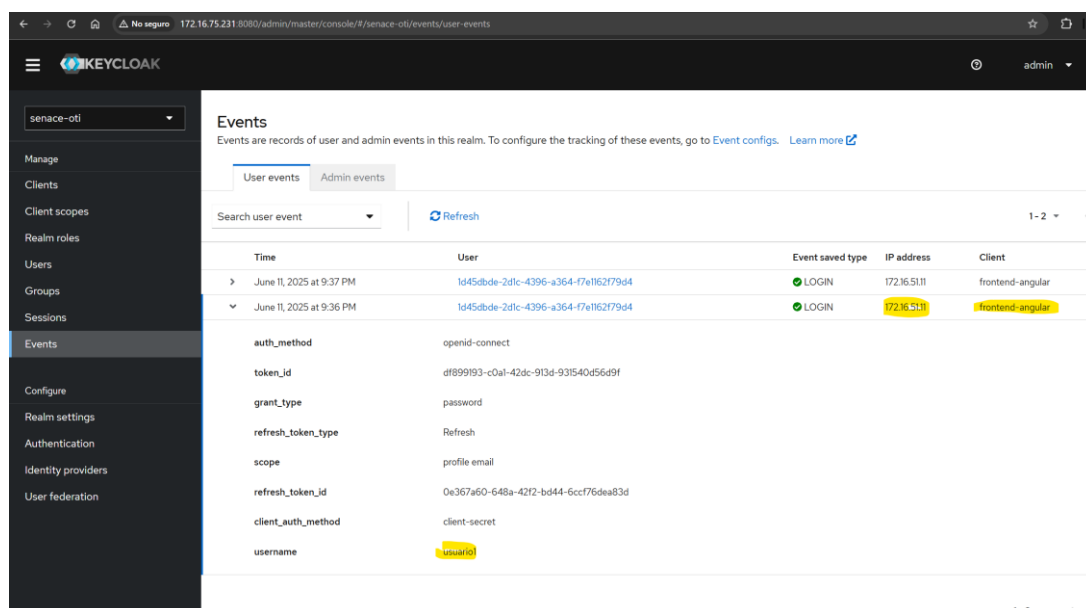


ix. Logs detallados: El SSO debe tener un registro y auditoría de intentos de acceso.

Dentro de la consola de Keycloak podemos configurar la política de logs que se quiere grabar.



Luego de eso podemos ir viendo los logs de diferente tipo dentro de la consola.



DESPLIEGUE DE SOFTWARE EN ENTORNO DE DESARROLLO		
5. REVERSIÓN O ROLLBACK DEL SOFTWARE		
<p>En caso de requerir rollback y reiniciar todo deberíamos ejecutar los siguientes comandos:</p> <ul style="list-style-type: none"> • docker-compose down Detiene la ejecución de los contenedores • docker volume rm proyecto-keycloak-master_pgdata Elimina los volúmenes creados para evitar dejar datos que hagan conflicto. 		
6. FIRMANTES DEL DOCUMENTO		
	Nombre: Jaime Alfredo Enero Antonio Rol: Coordinador de Proyectos SENACE	
	Nombre: Carlos Enrique Pérez Sinticala Rol: Especialista II en Sistemas De Información Digital	