

Configuración GCP: Gluu Docker, Passport.js y OpenID Connect

Proceso de instalación Docker:

https://docs.google.com/document/d/1Lbek5JKxS5TEX_T97rDtsi2C0_LzMPFqqQMc_zd9CH9I/edit?usp=sharing

Máquina utilizada:

- VM Ubuntu 18.04 Its
- 2 Cores
- 4GB Ram
- 50 GB HDD

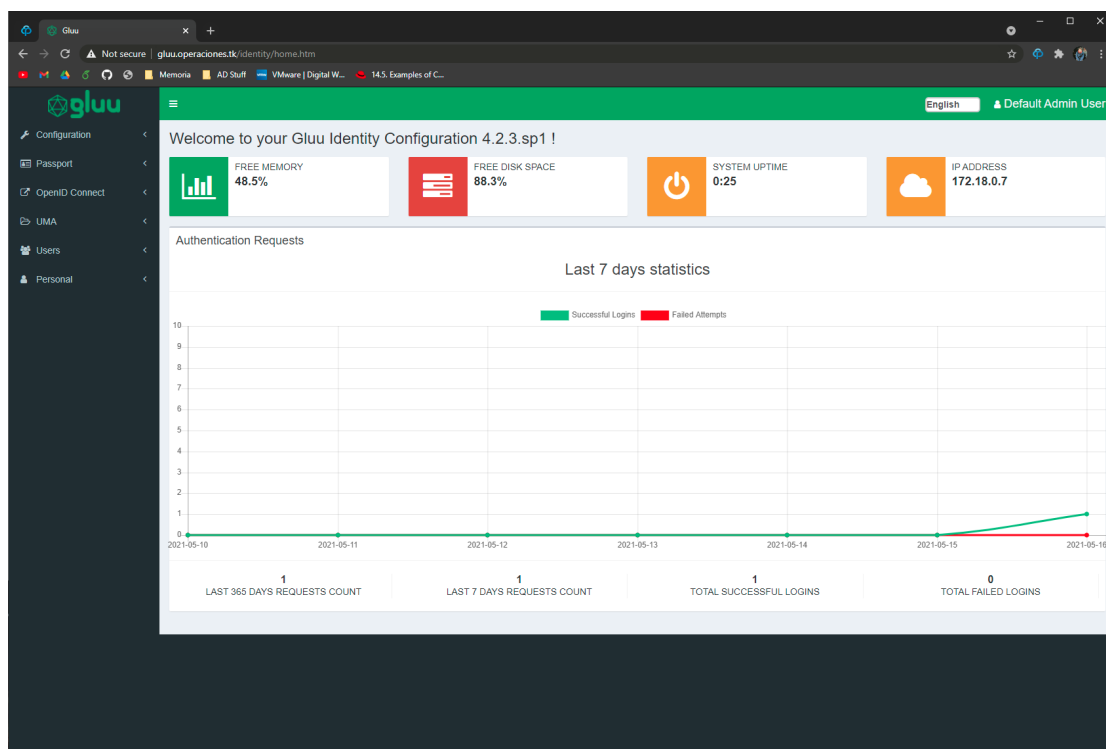
Registro de FQDN para la máquina virtual

<https://docs.google.com/document/d/1-l8eaUh2ZAEu0C7dULnsKEN-bbE5LeYZq5KC0WRQ-Ys/edit?usp=sharing>

FQDN: <http://gluu.operaciones.tk>

Default user: admin

Default pass: Operaciones1*



Vista principal de Gluu (Dominio <http://gluu.operaciones.tk>)

Pruebas con módulo Passport.js Gluu en Google Cloud

Documentación:

<https://gluu.org/docs/gluu-server/4.2/authn-guide/passport/>

Google OAuth Credentials

ID: -----

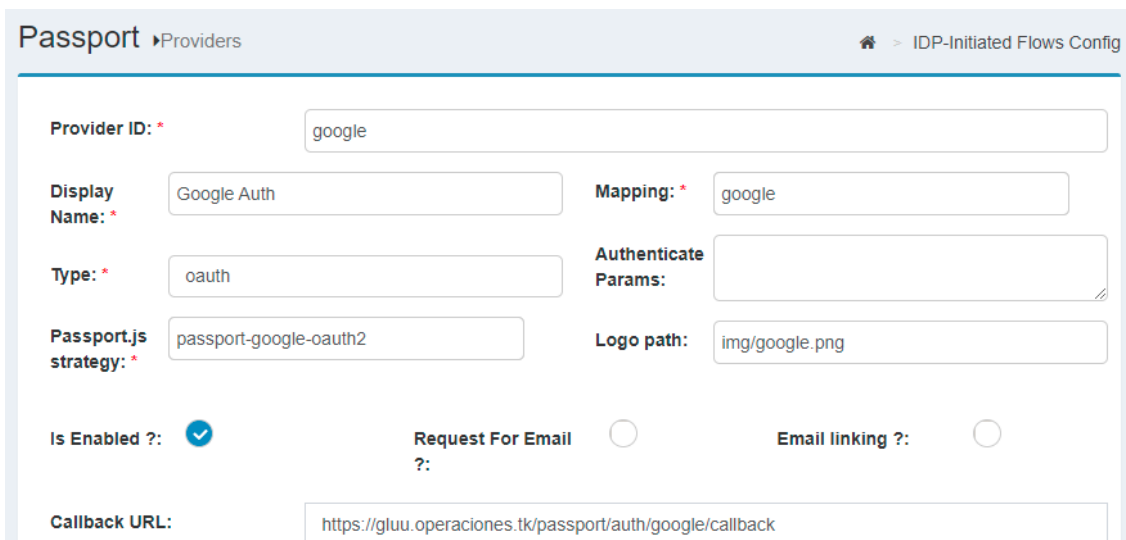
Secret: -----

Gluu ofrece el servicio de ***"Inbound Identity"***, que permite delegar la autenticación de usuarios a otros servicios que permitan esto a través de **SAML** u **OAuth2 (OIDC)**.

Los pasos siguientes corresponden a la sección ***"Inbound OAuth & OpenID Connect"***.

Luego de instalar Gluu junto al módulo Passport, todos los módulos necesarios de este vienen activados por defecto, por lo que los pasos de la sección ***"Enable Passport"*** se pueden saltar.

Para la configuración con Google, saltar a la sección ***"Integrate OAuth Authorization Servers"*** y seguir los pasos utilizando siempre la opción de Google como Provider. El resultado final debería ser algo similar a lo siguiente:



Passport Providers IDP-Initiated Flows Config

Provider ID: * google

Display Name: * Google Auth Mapping: * google

Type: * oauth Authenticate Params:

Passport.js strategy: * passport-google-oauth2 Logo path: img/google.png

Is Enabled ? ☒ Request For Email ? ☐ Email linking ? ☐

Callback URL: https://gluu.operaciones.tk/passport/auth/google/callback

Configuración Passport para Google

"Callback URL" se genera automáticamente y los campos ***"ClientID"*** y ***"ClientSecret"*** que se encuentran más abajo se pueden dejar en blanco de momento.

Ahora es necesario conseguir las credenciales del servicio a utilizar, en este caso Google con el fin de utilizar el correo USACH. Para esto es necesario ingresar a “<https://console.cloud.google.com/apis/dashboard>”, activar el servicio e ingresar a la pestaña “**OAuth consent screen**” y registrar una nueva aplicación.

Edit app registration

1

OAuth consent screen —

2

Scopes —

3

Summary

App information

This shows in the consent screen, and helps end users know who you are and contact you

App name *

Gluu

The name of the app asking for consent

User support email *

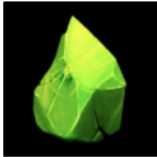
roberto.lillo@usach.cl

For users to contact you with questions about their consent

App logo

BROWSE

Upload an image, not larger than 1MB on the consent screen that will help users recognize your app. Allowed image formats are JPG, PNG, and BMP. Logos should be square and 120px by 120px for the best results.



App domain

To protect you and your users, Google only allows apps using OAuth to use Authorized Domains. The following information will be shown to your users on the consent screen.

Application home page

http://gluu.operaciones.tk

Provide users a link to your home page

Application privacy policy link

http://gluu.operaciones.tk

Provide users a link to your public privacy policy

Application terms of service link

http://gluu.operaciones.tk

Provide users a link to your public terms of service

Authorized domains ?

When a domain is used on the consent screen or in an OAuth client's configuration, it must be pre-registered here. If your app needs to go through verification, please go to the [Google Search Console](#) to check if your domains are authorized. [Learn more](#) about the authorized domain limit.

operaciones.tk

Configuración de registro de APP en Google Cloud

En la sección de Scopes se puede definir qué información se comparte en el momento que una cuenta inicia sesión.

Edit app registration

✓ OAuth consent screen — **2 Scopes** — 3 Summary

Scopes express the permissions you request users to authorize for your app and allow your project to access specific types of private user data from their Google Account. [Learn more](#)

[ADD OR REMOVE SCOPES](#)

Your non-sensitive scopes

API ↑	Scope	User-facing description	
	.. ./auth/userinfo .email	See your primary Google Account email address	🗑
	.. ./auth/userinfo .profile	See your personal info, including any personal info you've made publicly available	🗑

🔒 Your sensitive scopes

Sensitive scopes are scopes that request access to private user data.

API ↑	Scope	User-facing description
No rows to display		

🔒 Your restricted scopes

Restricted scopes are scopes that request access to highly sensitive user data.

API ↑	Scope	User-facing description
No rows to display		

Configuración de scopes de información

OAuth consent screen

[EDIT](#)

User type

Internal

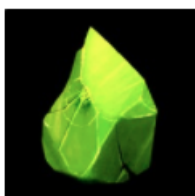
App name

Gluu

Support email

roberto.lillo@usach.cl

App logo



Application homepage link

<http://gluu.operaciones.tk>

Application privacy policy link

<http://gluu.operaciones.tk>

Application terms of service link

<http://gluu.operaciones.tk>

Authorized domains

operaciones.tk

Contact email addresses

roberto.lillo@usach.cl

Scopes

[EDIT](#)

API ↑	Scope	User-facing description
	.../auth/userinfo.email	See your primary Google Account email address
	.../auth/userinfo.profile	See your personal info, including any personal info you've made publicly available

Resumen de configuración de APP

(IMPORTANTE) Hay una información a destacar que no se aprecia en las screenshots anteriores, al inicio del registro se consulta si se desea hacer interno o público:

- Interno: permite que sólo cuentas dentro del dominio puedan autenticarse, en este caso permite que solo cuentas **@usach.cl** tengan acceso.
- Público: cualquier cuenta de Google tiene acceso a autenticación.

En la configuración anterior se especificó que el registro sea **interno**, permitiendo así solo a usuarios USACH autenticarse.

Luego de esta configuración es necesario ir a la pestaña de “**Credenciales**” y presionar arriba en el botón “**Crear credenciales**” y seleccionar “**Oauth client ID**”.

En el dropdown seleccionar que es una aplicación de tipo web e ingresar los siguientes valores:

Name *

Gluu Server

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

i The domains of the URIs you add below will be automatically added to your [OAuth consent screen](#) as [authorized domains](#).

Authorized JavaScript origins ?

For use with requests from a browser

+ ADD URI

Authorized redirect URIs ?

For use with requests from a web server

URIs *

<https://gluu.operaciones.tk/passport/auth/google/callback>

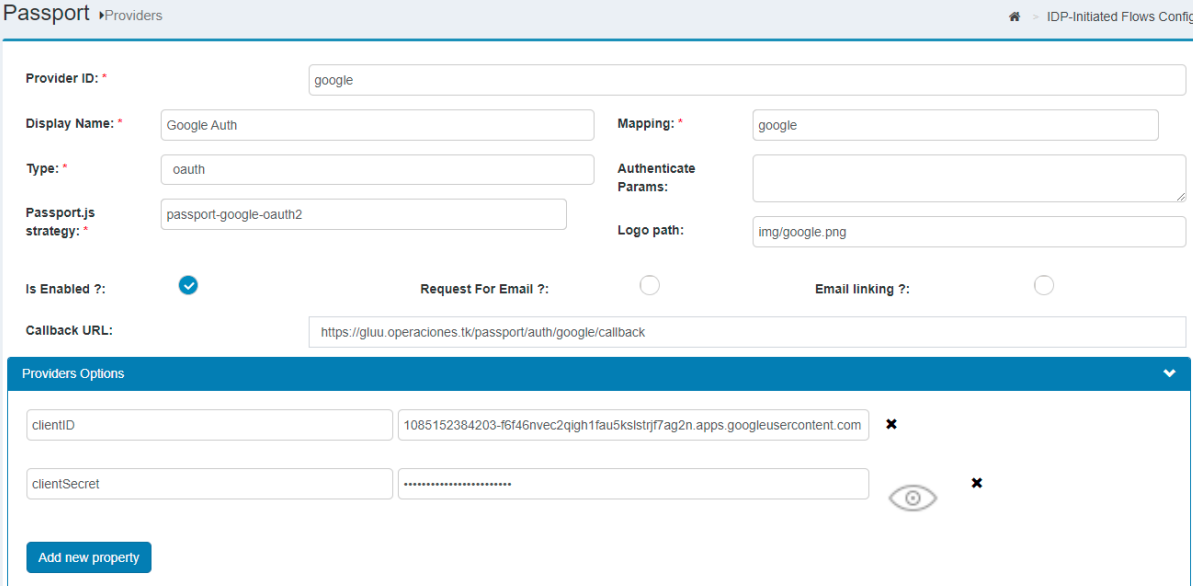
+ ADD URI

SAVE CANCEL

Registro de credenciales para la aplicación

La dirección de **callback** es la que generó Gluu automáticamente en el proceso previo de configuración de **Passport**.

Al finalizar, se obtienen los dos valores **ID** y **Secret**, que deben ser ingresados en Gluu como los valores **clientID** y **clientSecret** respectivamente.



Passport Providers IDP-Initiated Flows Config

Provider ID: * google

Display Name: * Google Auth Mapping: * google

Type: * oauth Authenticate Params:

Passport.js strategy: * passport-google-oauth2 Logo path: img/google.png

Is Enabled?: ☒ Request For Email?: ☐ Email linking?: ☐

Callback URL: https://gluu.operaciones.tk/passport/auth/google/callback

Providers Options

clientID 1085152384203-f6f46nvec2qigh1fauskslstrj7ag2n.apps.googleusercontent.com ✕

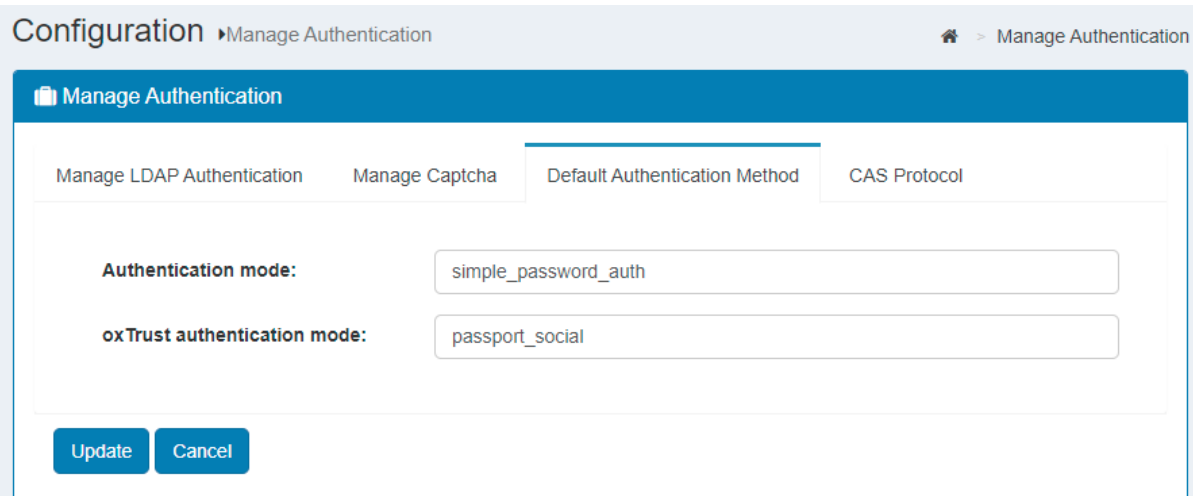
clientSecret 👁 ✕

Add new property

Configuración final Passport

Finalmente, para poder hacer testing de la funcionalidad, se puede configurar oxTrust (la GUI de Gluu) para que los usuarios puedan autenticarse mediante **Social Login**, es decir, el módulo que configuramos recientemente.

Para esto ingresar en la pestaña **“Configuration”**, luego a **“Manage Authentication”** y finalmente a **“Default Authentication Method”**, donde es necesario cambiar la entrada **oxTrust authentication mode** por **“passport_social”**.



Configuration Manage Authentication Manage Authentication

Manage Authentication

Manage LDAP Authentication Manage Captcha Default Authentication Method CAS Protocol

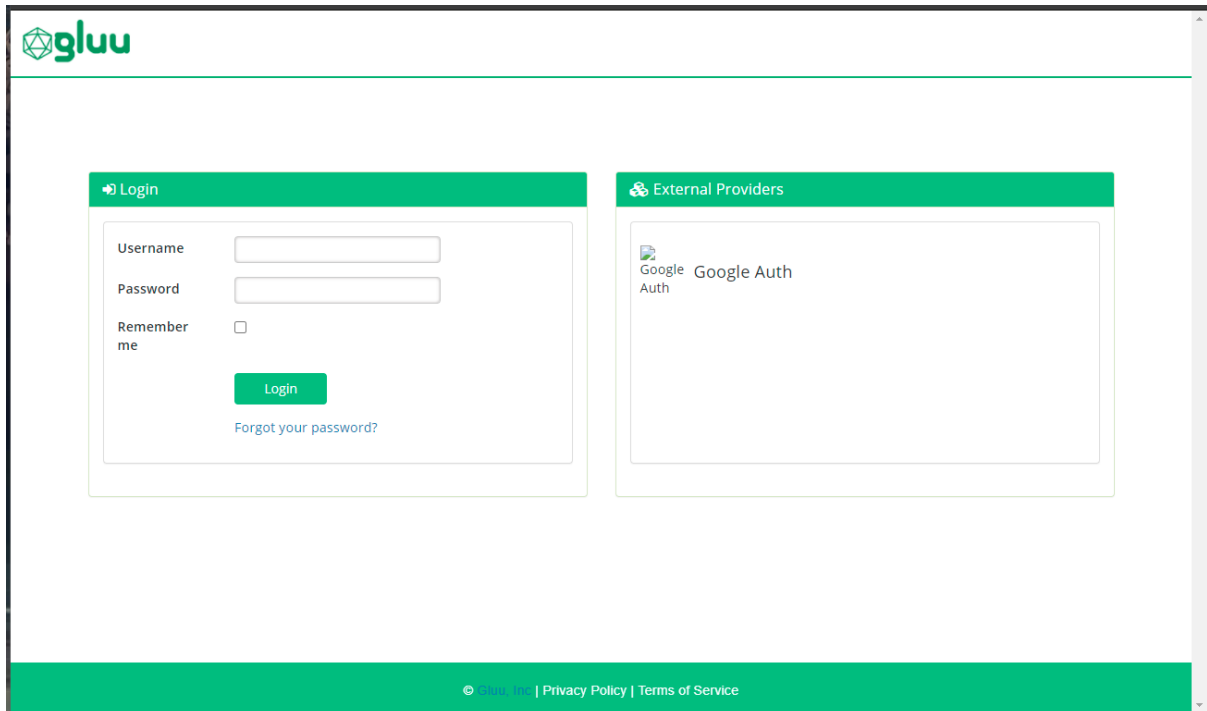
Authentication mode: simple_password_auth

oxTrust authentication mode: passport_social

Update Cancel

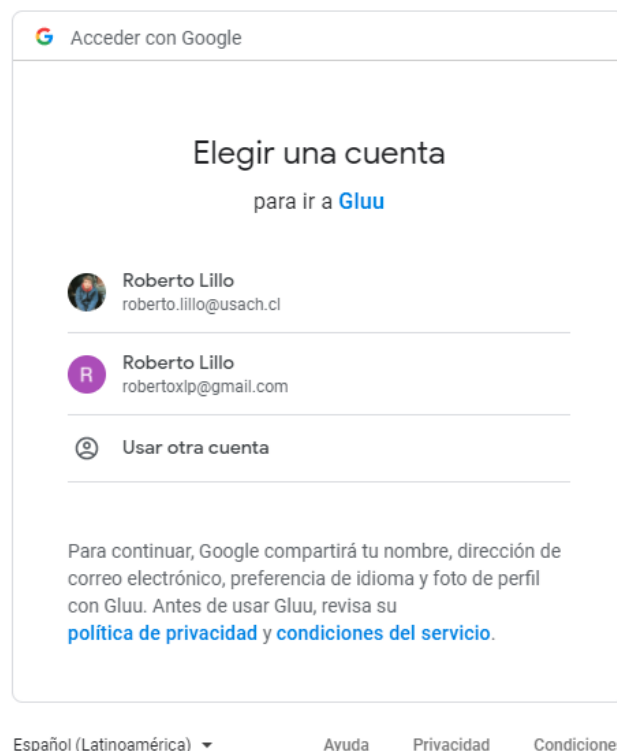
Configuración opciones de login oxTrust

De esta forma, al intentar hacer login a oxTrust se puede ver la siguiente vista:

The image shows the Gluu login interface. At the top left is the Gluu logo. Below it, there are two main sections: 'Login' and 'External Providers'. The 'Login' section contains fields for 'Username' and 'Password', a 'Remember me' checkbox, a green 'Login' button, and a link for 'Forgot your password?'. The 'External Providers' section shows a 'Google Auth' button. At the bottom of the page, there is a green footer bar with links for 'Gluu, Inc.', 'Privacy Policy', and 'Terms of Service'.

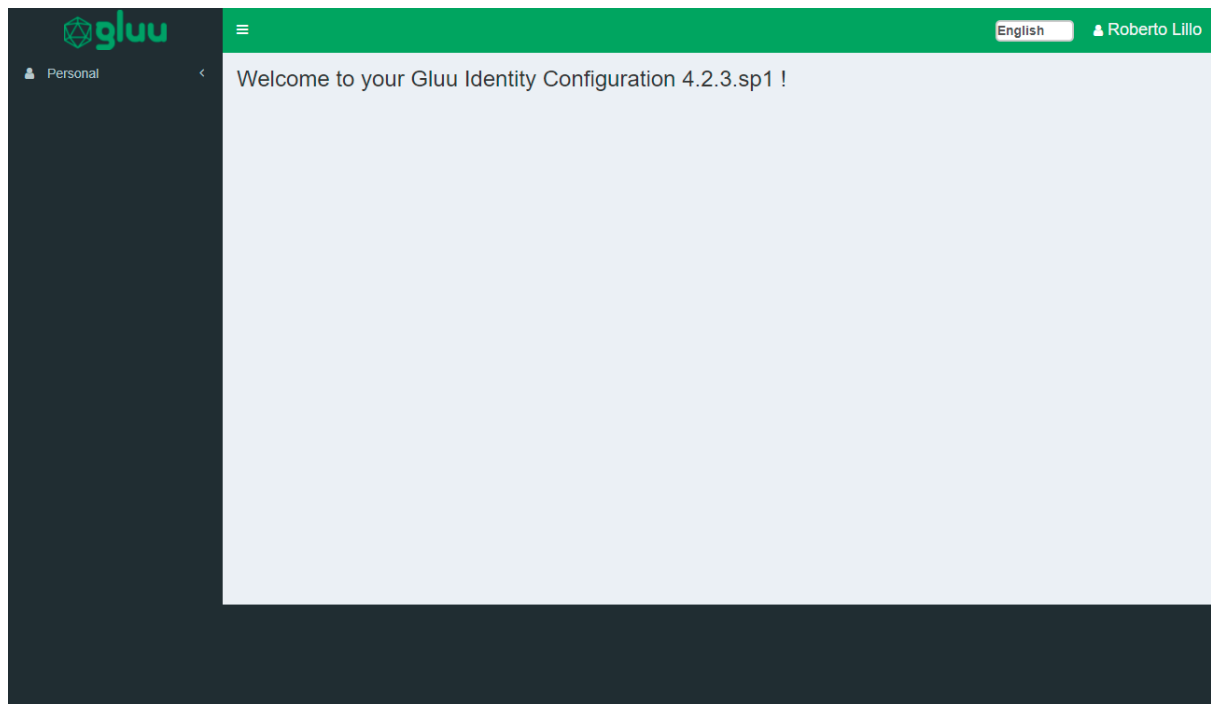
Login de Gluu post Passport

Al presionar en Google Auth, se redirecciona a Google y se puede seleccionar la cuenta con la que se quiere ingresar:

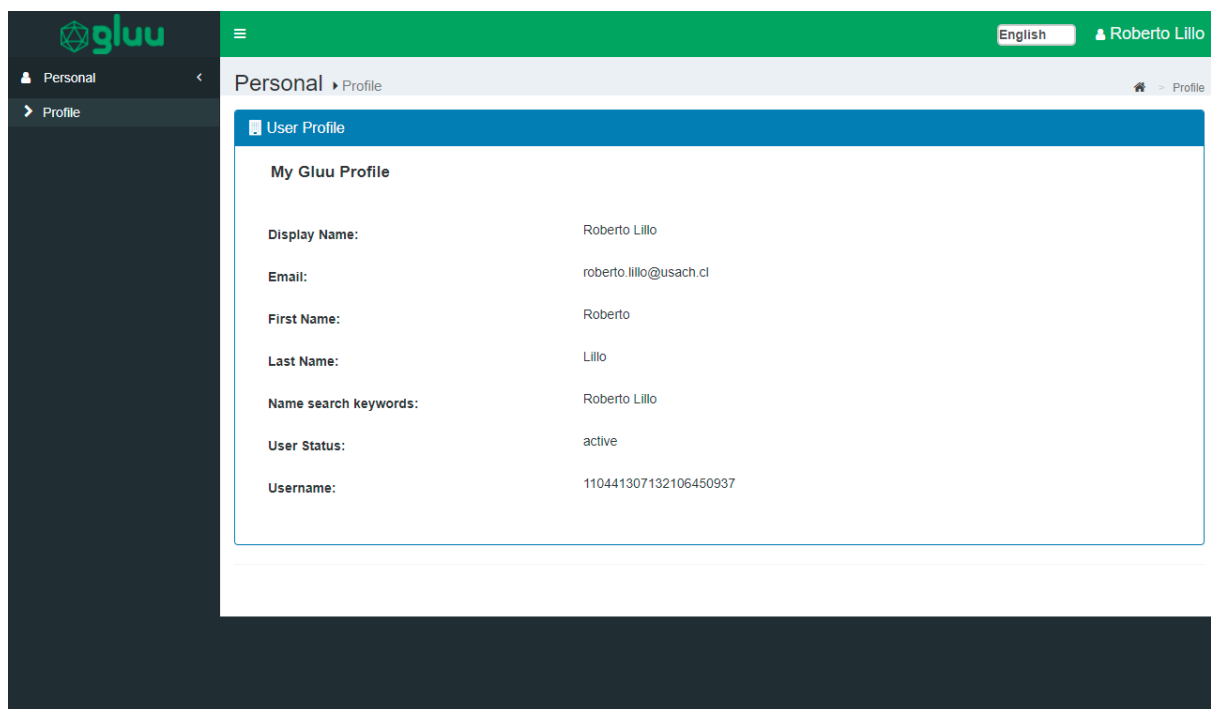
The image shows a Google account selection screen. At the top, it says 'Acceder con Google'. The main heading is 'Elegir una cuenta para ir a Gluu'. There are two account options listed: 'Roberto Lillo' with email 'roberto.lillo@usach.cl' and another 'Roberto Lillo' with email 'robertoxlp@gmail.com'. Below these is a link 'Usar otra cuenta'. At the bottom, there is a paragraph explaining that Google will share the user's name, email, language preference, and profile photo with Gluu, and a link to the 'política de privacidad y condiciones del servicio'. At the very bottom, there are links for 'Español (Latinoamérica)', 'Ayuda', 'Privacidad', and 'Condiciones'.

Login de Gluu post Passport

En este caso si escojo mi correo USACH, me redirecciona a Gluu ya con mi sesión iniciada y mi usuario creado en el LDAP interno de Gluu.



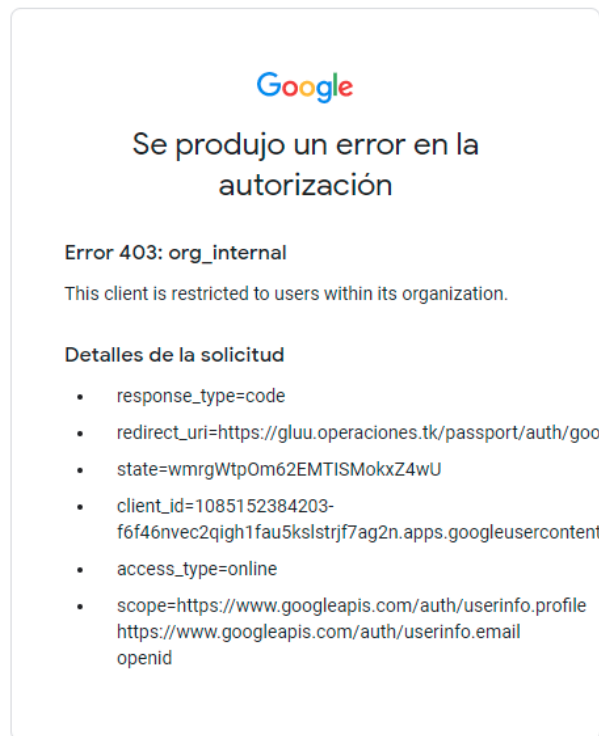
Vista principal Gluu, inicio de sesión mediante Google



Perfil de usuario, inicio de sesión mediante Google

Se puede apreciar que los datos que son entregados por Google son el correo, primer nombre y apellido.

En el caso de seleccionar un correo que no sea parte del dominio **@usach.cl** se aprecia el siguiente resultado:



Español (Latinoamérica) ▼

Ayuda

Privacidad

Condiciones

Error de inicio de sesión, no es parte de la organización

Luego con una cuenta de administrador, se puede apreciar la información del usuario que se creó automáticamente en Gluu:

The screenshot shows the 'Update User' page in the Gluu Users management interface. The page has a header 'Users > Update User' and a breadcrumb 'Add/Update User'. The main content area is titled 'Person Add Form' and contains several input fields for user information:

- Display Name: Roberto Lillo
- Email: roberto.lillo@usach.cl
- First Name: Roberto
- Inum: 38b30127-800a-40ad-acbe-d865bd9e44bd
- Last Name: Lillo
- Name search keywords: Roberto Lillo
- User Status: active
- Username: 110441307132106450937

To the right of these fields is a section titled 'Available User Claims' with three tabs: 'gluuPerson', 'gluuCustomPerson', and 'eduPerson'. The 'gluuPerson' tab is selected, showing a list of claims including Birthdate, CIBA Device Registration Token, CIBA User code, Country, Email Verified, Enrollment code, Gender, male or female, IMAP Data, Iname, Last Updated, Locale, memberOf, Middle Name, Nickname, OpenID Connect JSON formatted address, Organization, Password, Persistentid, Phone Number Verified, Picture URL, Preferred Language, Preferred Username, Profile URL, Secret Answer, Secret Question, Time zone info, Transientid, User certificate, User Permission, and Website URL.

Below the form is a section titled 'Authentication Methods' with a dropdown menu. It shows a table with one entry:

ID	Nickname	Modality	Date Added	Remove
1104413...	passport-google	Passport	-	X

Usuario creado a partir de Inicio de sesión en Google

Pruebas a realizar

- Qué pasa si un usuario con el mismo correo ya se encuentra creado en Gluu.
- Qué pasa con los usuarios que se pueden sincronizar mediante **Cache Refresh (LDAP Sync)**.
- Qué tanta libertad de manejo de información se le puede entregar al usuario (como para que ingrese otros datos a su usuario).

Pruebas sobre mapping de atributos de Passport

Documentación:

<https://gluu.org/docs/gluu-server/4.2/tutorials/passport-attributes-mapping/>

La documentación anterior muestra la flexibilidad que tiene Passport para hacer mapping de los atributos que se reciben desde el **External Identity Provider** con el usuario que será creado dentro de Gluu.

Gluu por defecto trae varios archivos de mapping en formato **js** para proveedores como Google, Facebook, Github, etc. Pero así mismo uno tiene la posibilidad de crear versiones propias de estos archivos modificando los parámetros según sea necesario.

(IMPORTANTE) Gluu hace hincapié en no modificar los archivos que vienen por defecto.

Por ejemplo, el archivo de mapping “**google.js**” contiene lo siguiente:

```
module.exports = profile => {  
  return {  
    uid: profile.username || profile.id,  
    mail: profile.email,  
    cn: profile.displayName,  
    displayName: profile.displayName,  
    givenName: profile.name.givenName,  
    sn: profile.name.familyName  
  }  
}
```

Archivo google.js

En la imagen se puede apreciar que Google provee los valores **id**, **email**, **displayName**, **givenName** y **familyName**, los que corresponde a la información que luego es guardada en el usuario que se crea en Gluu.

(DETALLE) Se puede apreciar que en **uid** se prioriza el atributo **username** por sobre **id**, no obstante, Google solamente provee **id** en la respuesta.

Con esta información se realizaron los siguientes pasos:

1. Hacer una copia del archivo **google.js** con el nombre **google-operaciones.js**
2. Modificar el script para utilizar el **email** como **uid** en Gluu.
3. Modificar la configuración del provider en Gluu, para cambiar el script de mapping de **google** a **google-operaciones.js**.

De esta forma, se configuró el siguiente script de mapping:

```
module.exports = profile => {  
  return {  
    uid: profile.email,  
    mail: profile.email,  
    cn: profile.displayName,  
    displayName: profile.displayName,  
    givenName: profile.name.givenName,  
    sn: profile.name.familyName  
  }  
}  
/opt/gluu/node/passport/server/mappings #
```

Archivo google-operaciones.js

Y luego al hacer inicio de sesión en Gluu se obtiene el siguiente perfil:

The screenshot shows the 'Personal > Profile' section of the Gluu interface. It features a 'User Profile' header and a 'My Operaciones Profile' section. The profile details are as follows:


Field	Value
Display Name:	Roberto Lillo
Email:	roberto.lillo@usach.cl
First Name:	Roberto
Last Name:	Lillo
Name search keywords:	Roberto Lillo
User Status:	active
Username:	roberto.lillo@usach.cl

Perfil de usuario en Gluu

En donde se puede apreciar que el **username** (**uid** en LDAP) ahora es el correo institucional.

Prueba con usuarios creados previamente

Después de cambiar el script anterior, se borró el usuario creado automáticamente y se creó manualmente el siguiente usuario en Gluu:

 Person Add Form

Display Name:	<input type="text" value="Roberto Lillo"/>
Email:	<input type="text" value="roberto.lillo@usach.cl"/>
First Name:	<input type="text" value="Roberto"/>
Inum:	bc39b9d6-d02d-44f4-9e4c-df6ae7b3f6d9
Last Name:	<input type="text" value="Lillo"/>
Name search keywords:	<input type="text" value="Roberto Roberto Lillo"/>
User Status:	<input type="text" value="active"/>
Username:	<input type="text" value="roberto.lillo@usach.cl"/>

Usuario creado manualmente en Gluu

Posteriormente, al intentar ingresar sesión mediante Google se obtuvo el siguiente mensaje de error:

OOPS

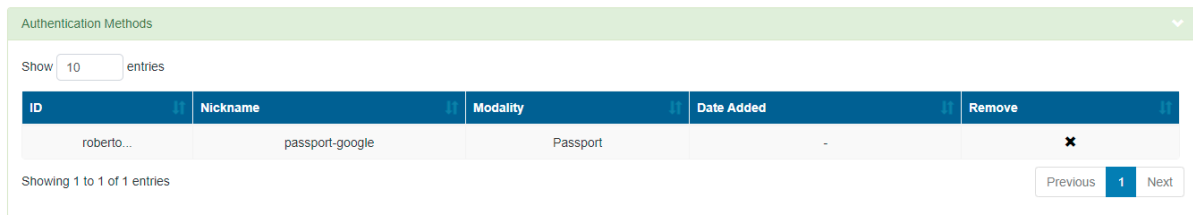
An unexpected error has occurred at null

Email value corresponds to an already existing account. If you already have a username and password use those instead of an external authentication site to get access.

Error de inicio de sesión mediante Google

Lo que implica que la creación de un usuario **manual** y uno que es creado **automáticamente** por un script al momento de iniciar sesión por primera vez mediante Google presenta alguna diferencia, lo que crea una distinción entre estos dos tipos de usuarios.

De hecho se puede apreciar que los usuarios creados automáticamente presentan la siguiente sección en su perfil:



ID	Nickname	Modality	Date Added	Remove
roberto...	passport-google	Passport	-	✕

Showing 1 to 1 of 1 entries

Previous 1 Next

Sección de Authentication Methods en usuario

(Prueba) Al remover la autenticación mostrada en la imagen anterior y luego iniciar sesión nuevamente con el usuario mediante Google, se encuentra el mismo error de inicio de sesión anterior.

Verificar diferencias entre usuarios manuales y automáticos

Gluu contiene la base de datos LDAP en uno de los contenedores y puede ser consultada mediante ldapsearch con los siguientes comandos:

1. Comando de testing de deploy

```
docker exec -it ldap /opt/openssh/bin/ldapsearch -h localhost -p 1636 -Z -X -D  
"cn=directory manager" -w Operaciones1* -b "o=gluu" -s base "objectClass="
```

```
dn: o=gluu  
description: Welcome to oxTrust!  
displayName: Operaciones  
gluuManagerGroup: inum=60B7,ou=groups,o=gluu  
gluuOrgShortName: Operaciones  
gluuThemeColor: 166309  
objectClass: gluuOrganization  
objectClass: top  
o: gluu
```

Resultado comando de testing de deploy

2. Objeto específico en base a un atributo

```
docker exec -it ldap /opt/openssh/bin/ldapsearch -h localhost -p 1636 -Z -X -D  
"cn=directory manager" -w Operaciones1* -b "o=gluu" -s sub  
"uid=roberto.lillo@usach.cl"
```

Donde **"uid=roberto.lillo@usach.cl"** puede ser reemplazado por cualquier otro atributo que se quiera buscar.

Resultados de búsqueda de usuario automático y manual en Gluu.

```
dn: inum=9147e689-857a-4a67-8bdc-c045e757120a,ou=people,o=gluu
cn: Roberto Lillo
displayName: Roberto Lillo
givenName: Roberto
gluuStatus: active
inum: 9147e689-857a-4a67-8bdc-c045e757120a
mail: roberto.lillo@usach.cl
objectClass: gluuCustomPerson
objectClass: gluuPerson
objectClass: top
oxCreationTimestamp: 20210520214421.357Z
oxExternalUid: passport-google:roberto.lillo@usach.cl
oxTrustEmail: {"value":"roberto.lillo@usach.cl","primary":false}
sn: Lillo
uid: roberto.lillo@usach.cl
updatedAt: 20210520214658.114Z
```

Resultado de búsqueda roberto.lillo@usach.cl (automático)

```
dn: inum=59edbe45-1d0b-4ac6-8228-8567d25a71e9,ou=people,o=gluu
cn: Testing Testing User
displayName: Testing User
givenName: Testing
gluuStatus: active
inum: 59edbe45-1d0b-4ac6-8228-8567d25a71e9
mail: testinguser@usach.cl
objectClass: eduPerson
objectClass: gluuCustomPerson
objectClass: gluuPerson
objectClass: top
oxCreationTimestamp: 20210520220808.038Z
oxTrustEmail:: ewogICJ2YWx1ZSIgOiAidGVzdGluZ3VzZXJAdXNhY2guY2wiLAogICJka
XNwbGF5IiA6IG51bGwsCiAgInR5cGUiIDogbnVsbCwKICAicHJpYWYySgOiBmYWxzZQp9
sn: User
uid: testinguser
userPassword: {SSHA512}ttM7ABuBAawDGnLUVNqRN1PDwRBFu8+JHpWx72rOjL6MsMxnf
nrv9I3DMPY7WRaRSzJecFclr46570OzVZHOT1NJqKf0UxMR
```

Resultado de búsqueda testinguser (manual)

Diferencias apreciables:

1. El usuario automáticamente creado presenta el atributo **oxExternalUid**, que contiene información sobre passport.
2. El atributo **oxTrustEmail** en el usuario de automático contiene un JSON con el valor del correo que se obtuvo desde la respuesta de Google y el atributo *primary* que no sé que es. Por otro lado, la versión del usuario manual guarda el correo probablemente encriptado.
3. El usuario manual también contiene la contraseña encriptada.

También para verificar el cambio al eliminar la autenticación por passport en el perfil del usuario se realizó la siguiente búsqueda:

```
dn: inum=9147e689-857a-4a67-8bdc-c045e757120a,ou=people,o=gluu
cn: Roberto Lillo
displayName: Roberto Lillo
givenName: Roberto
gluuStatus: active
inum: 9147e689-857a-4a67-8bdc-c045e757120a
mail: roberto.lillo@usach.cl
objectClass: eduPerson
objectClass: gluuCustomPerson
objectClass: gluuPerson
objectClass: top
oxCreationTimestamp: 20210520214421.357Z
oxTrustEmail:: ewogICJ2YWx1ZSIgOiAicm9iZXJ0by5saWxsB1c2FjaC5jbCIsCiAgImRp
c3BsYXkiIDogbnVsbCwKICaidHlwZSIgOiBudWxsLAogICJwcm9tYXJ5IiA6IGZhbHNlCn0=
sn: Lillo
uid: roberto.lillo@usach.cl
updatedAt: 20210520223428.545Z
```

Resultado de búsqueda roberto.lillo@usach.cl

Y se aprecia que precisamente se elimina el atributo **oxExternalUid** y también el JSON en **oxTrustEmail** el cual cambia probablemente al correo encriptado.

(INFORMACIÓN) Luego se encontró esto en la documentación de Gluu:

Identifiers

When adding a provider, administrators will be asked to assign an ID to it. Instead of automatically generate one, we decided to let admins enter a value explicitly. This is for convenience since IDs end up being part of URLs and even ldap data such as the `oxExternalUid` attribute used to correlate external accounts with local accounts. Ideally IDs should be short, compact, and self explanatory instead of random obscure strings.

Información sobre los identificadores de passport

How user onboarding works

As stated in the [sample flow](#), after a user has logged in at an external provider a new record is added in local LDAP - or updated if the user is known.

To determine if a user was already added, a string is composed with the provider name and the user ID. For example, if user "MrBrown123" has logged in at Twitter, the string would look like `passport-twitter:mrbrown123`. An LDAP search is performed for a match in the people branch for an entry where attribute `oxExternalUid` equals `passport-twitter:mrbrown123`.

If there are no matches, an entry is added using the values received from the external provider (after having applied the corresponding attribute mapping) attaching the computed value for `oxExternalUid`. The user profile can contain single or multivalued attributes.

Información sobre el usuario en Gluu LDAP

Prácticamente confirmando que **oxExternalUid** dentro de LDAP tiene la función identificar los usuarios externos y diferenciarlos de los locales.

Finalmente, se encuentra la siguiente información en la sección “***Altering flow behavior***”

Email account linking

There are cases in which an external provider is trusted, so you can change the default behavior of adding a new user entry locally, but binding an existing account to the person logging in. This linking can be done via `mail` attribute.

For example, suppose you have 3 users in your Gluu local LDAP: Larry (`larry@acme.com`), Moe (`moe@acme.com`), and Curly (`curly@acme.com`). When you enable email account linking for provider "XYZ" and certain user logs in through XYZ to access your application, he will be logged as Moe as long as his email is "moe@acme.com" at XYZ.

To enable account linking, follow these steps:

- In oxTrust, navigate to `Passport > Providers` and click on the provider of interest
- Tick the `Email Linking` checkbox
- Hit Update button

Información sobre el usuario en Gluu LDAP

Por lo tanto, esta parte de la documentación soluciona el error anterior al crear un usuario manualmente con un correo y posteriormente iniciar sesión mediante Google con el mismo.

Se activó el **Email Linking** y se realizaron las siguientes pruebas:

1. Se creó el usuario “**roberto.lillo@usach.cl**”, donde el **uid** e **email** tienen el valor anterior.
2. Se inició sesión mediante Google y el proceso se realizó exitosamente.
3. Se revisó el perfil del usuario creado y se encontró que automáticamente se agregó la sección de **Authentication Methods**, incluyendo el atributo **passport-google**.

De esta forma se completa el proceso de autenticación mediante Google, configurado de manera correcta para que usuarios ya creados en Gluu puedan iniciar sesión mediante el correo USACH.

Configuración de Cache Refresh

Documento de configuración (2019)

https://docs.google.com/document/d/1cufekP52v0WQxbQKb6nsS8-W9gX0_wtZrnvyXjDkLUU/edit?usp=sharing

Cache Refresh Form

Cache Refresh

Customer Backend Key/Attributes

Source Backend LDAP Servers

Inum DB server

Last run:

May. 21 2021 02:17 AM

Updates at the last run:

0

Problems at the last run:

0

Refresh Method: *

Copy

Add source attribute to destination attribute mapping: *

sAMAccountName

uid

✕

cn

cn

✕

givenName

givenName

✕

sn

sn

✕

displayName

displayName

✕

mail

mail

✕

Add source attribute to destination attribute mapping

Polling interval (minutes):

2

Server IP Address:

172.18.0.4

Snapshot Folder: *

/var/identity/cr-snapshots

Snapshots count: *

10

Keep external persons:

☒

Load source data with limited search:

☐

Search size limit:

1000

Cache Refresh:

☒

Update

Cancel

Update & Validate script

Configuración de Cache Refresh

Cache Refresh Form

Cache Refresh Customer Backend Key/Attributes Source Backend LDAP Servers Inum DB server

Key attribute: *

SAMAccountName

✕

Add key attribute

Object class: *

user

✕

Add object class

Source Attribute: *

cn

✕

givenName

✕

sn

✕

displayName

✕

mail

✕

Add source attribute

Custom LDAP filter:

Update

Cancel

Update & Validate script

Configuración de atributos a copiar desde SAMBA

Cache Refresh Form

Cache Refresh Customer Backend Key/Attributes Source Backend LDAP Servers Inum DB server

Name: *

source

Remove source server

Bind DN: *

CN=Administrator,CN=Users,DC=operaciones,DC=lan

Max connections: *

3

Server:Port: *

10.128.0.3:636

✕

Add server

Base DN: *

DC=operaciones,DC=lan

✕

Add base DN

Change Bind Password

Use SSL: *

✓

Test LDAP Connection

Add source LDAP server

Update

Cancel

Update & Validate script

Configuración de dirección de servidor SAMBA

Users List

+ Add Person

Q Search

Show entries

Search:

UID	Display Name	First Name	Group Count	Email	Status
cesar.valenzuela	Cesar Valenzuela	Cesar	0	cesar.valenzuela@usach.cl	active
javier.perez	Javier Perez	Javier	0	javier.perez@usach.cl	active
jorge.ayala	Jorge Ayala	Jorge	0	jorge.ayala@usach.cl	active
ricardo.hernandez	Ricardo Hernandez	Ricardo	0	ricardo.hernandez@usach.cl	active
roberto.lillo	Roberto Lillo	Roberto	0	roberto.lillo@usach.cl	active

Showing 1 to 5 of 5 entries

CSV

Excel

PDF

Print

Previous

1

Next

Usuarios sincronizados desde SAMBA hacia Gluu

Luego al iniciar sesión mediante Google con el correo USACH con el usuario **“roberto.lillo”**, se logra correctamente el link entre los usuarios gracias al atributo de correo.

User Profile

My Operaciones Profile

Display Name:

Roberto Lillo

Email:

roberto.lillo@usach.cl

First Name:

Roberto

Last Name:

Lillo

Name search keywords:

Roberto Lillo

User Status:

active

Username:

roberto.lillo

Perfil del usuario que inició sesión

Ya con estas dos configuraciones es posible:

1. Crear un usuario en SAMBA que tenga la información del correo USACH.
2. Sincronizar este usuario hacia Gluu.
3. Iniciar sesión con el usuario utilizando la autenticación Google USACH.

Luego al realizar pruebas con otros correos USACH (Jorge Ayala y Francisco Guajardo) se puede apreciar que los usuarios que iniciaron sesión mediante Google tienen su nombre de usuario en azul.

UID	Display Name	First Name	Group Count	Email	Status
cesar.valenzuela	Cesar Valenzuela	Cesar	0	cesar.valenzuela@usach.cl	active
francisco.guajardo.v	Francisco Guajardo Villa	Francisco	0	francisco.guajardo.v@usach.cl	active
javier.perez	Javier Perez	Javier	0	javier.perez@usach.cl	active
jorge.ayala	jorge ayala aceval	jorge	0	jorge.ayala@usach.cl	active
ricardo.hernandez	Ricardo Hernandez	Ricardo	0	ricardo.hernandez@usach.cl	active
roberto.lillo	Roberto Lillo	Roberto	0	roberto.lillo@usach.cl	active

Usuarios registrados en Gluu

Configuración y pruebas OpenID Connect Provider (OP)

Documentación:

<https://gluu.org/docs/gluu-server/4.2/admin-guide/openid-connect/>
<https://gluu.org/docs/gluu-server/4.2/api-guide/openid-connect-api/>

Debugger:

<https://oauthdebugger.com/>

La idea de esto es probar si el endpoint **OAuth 2.0** de Gluu está funcionando correctamente mediante la capa superior que provee **OpenID Connect**.

Primero se registra un nuevo cliente en Gluu, en la pestaña **OpenID Connect** y luego en **Clients**.

The screenshot shows the 'Clients' configuration page in Gluu. The top navigation bar includes tabs for 'Standard settings', 'Advanced settings', 'Encryption/Signing settings', 'Client Attributes', and 'Custom Scripts'. The 'Standard settings' tab is active.

The configuration form is divided into two main columns. The left column contains fields for 'Client ID' (905b69b9-5e3b-4f6a-9dd1-eca0c72d1faf), 'Client Secret' (masked), 'Client Name' (OAuth 2.0 debugger), 'Client Description', and 'Sector URI'. Below these are expandable sections for 'Redirect Login URIs' (containing https://oauthdebugger.com/debug), 'Scopes' (containing profile), and 'Response Types' (containing code). The right column contains fields for 'Disabled' (unchecked), 'Pre-Authorization' (unchecked), 'Persist Client Authorizations' (checked), 'Application Type' (Web), 'Subject Type' (pairwise), 'Authentication method for the Token Endpoint' (client_secret_basic), 'Expirable client' (unchecked), 'Grant Types' (containing authorization_code), 'Redirect Logout URIs' (with a 'Post Logout Redirect URI' button), 'Logo URI', and 'Policy URI'.

Luego, se entregan los valores correspondientes en el debugger (la URI de autorización se encuentra en la documentación de la API de OpenID Connect de Gluu).

Authorize URI (required)

`http://gluu.operaciones.tk/oxauth/restv1/authorize`

Redirect URI (required)

`https://oauthdebugger.com/debug`

Client ID (required)

`905b69b9-5e3b-4f6a-9dd1-eca0c72d1faf`

Scope (required)

`profile`

State

Nonce

`vlnf22cs74s`

Response type (required)

☒ code ☐ token

Authorization code flow

The authorization server will respond with a `code`, which the client can exchange for tokens on a secure channel. This flow should be used when the application code runs on a secure server (common for MVC and server-rendered pages apps).

Response mode (required)


☐ query ☒ form_post ☐ fragment

```
http://gluu.operaciones.tk/oxauth/restv1/authorize
?client_id=905b69b9-5e3b-4f6a-9dd1-eca0c72d1faf
&redirect_uri=https://oauthdebugger.com/debug
&scope=profile
&response_type=code
&response_mode=form_post
&nonce=vlnf22cs74s
```

SEND REQUEST 


Preparación de consulta en OAuth 2.0 debugger

Al presionar el botón “**SEND REQUEST**” se enviará la petición al servidor de Gluu y se despliega la siguiente página.



Request for Permission

OAuth 2.0 debugger is requesting permission to do the following:

 View your basic profile info.

Powered by [Gluu](#). Free and open source access management.

Consulta de acceso a información

Se puede apreciar que sólo se requiere acceso a la **información básica de perfil** debido a que se configuró de tal manera. Si se permite el acceso se llega al siguiente resultado.

 **Success!**

 [Start over](#)

The flow was successful. The authorization server responded with an authorization code because the flow was started with the **code** response type.

The returned state is **83a77fc1-3cf1-4b18-bc73-1b7f28ff333e**.

Authorization code
901a88a9-cef9-42ac-9bff-54d9c6f7971f



Resultado de consulta por OpenID Connect

Dependiendo del tipo de aplicación que se está utilizando, existen distintos flujos ya sea usando **id_token**, **code** o **hybrid_flow**. En el caso anterior, se configuró para hacer la consulta mediante el flujo de **code**.

Step 2: Exchange code for tokens

Now you need to exchange the authorization code for tokens using the token endpoint. We can't do this step for you because it involves your client secret.

```
POST {tokenEndpoint}
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code&
code=901a88a9-cef9-42ac-9bff-54d9c6f7971f&
client_id= 487dfd42-df16-42d6-af71-c7f8de94bfc4&
client_secret={clientSecret}&
redirect_uri=https%3A%2F%2Foauthdebugger.com%2Fdebug
```

POST request details

Form body values

```
code=901a88a9-cef9-42ac-9bff-54d9c6f7971f
acr_values=passport_social
scope=profile
session_id=73f2a550-718c-483c-9ca1-859365c5b85f
state=83a77fc1-3cf1-4b18-bc73-1b7f28ff333e
session_state=9616548644652c5a337c418909a1725727f1a363b20de95cb6499e189b9ee0c6.7428be28-
699e-4885-a91d-8bf065f0ebaf
sid=1d4087df-2f39-4670-b5b6-e268289fd981
```

Otros resultados por parte de OAuth 2.0 debugger

(IMPORTANTE) Esto solo demuestra que de momento es posible conectar una aplicación externa hacia Gluu, con el fin de que esta pueda autenticar usuarios mediante OAuth 2.0 (técnicamente OpenID Connect).

El proceso de hacer pruebas en las que un usuario inicie sesión en tal aplicación corresponde a la siguiente sección de este documento.

Conexión aplicación de prueba e inicio de sesión

Docs:

<http://www.passportjs.org/packages/passport-oauth2/>

<https://www.freecodecamp.org/news/a-quick-introduction-to-oauth-using-passport-js-65ea5b621a/>

<https://flaviocopes.com/express-https-self-signed-certificate/>

Para probar el inicio de sesión mediante Gluu se levantó una nueva máquina virtual en GCP con el FQDN <https://testapp1.operaciones.tk>, la cual contiene una aplicación javascript hecha con nodejs que funcionará como cliente para Gluu.

Se utilizó el código presente en el link anterior de <https://www.freecodecamp.org>, pero se modificó para utilizar la estrategia de Passport.js “**passport-oauth2**” y no “**passport-google-oauth20**”, además de cambios extras para utilizar el puerto 443 y poder utilizar **https** (se utiliza un certificado autofirmado).

OpenID Connect > Clients

Client Form

Client Config Summary Delete

Standard settings Advanced settings Encryption/Signing settings Client Attributes Custom Scripts

Client ID: 821b776a-b038-49f1-a9d2-bd80e297e0dc Disabled: ☐

Client Secret: [Redacted] Change Client Secret

Client Name: OAuth 2.0 Test App Pre-Authorization: ☐

Client Description: [Empty] Persist Client Authorizations: ☒

Sector URI: [Empty] Application Type: Web

Subject Type: pairwise

Authentication method for the Token Endpoint: client_secret_basic

Expirable client: ☐

Redirect Login URIs

https://testapp1.operaciones.tk/auth/oauth2/callback

Add Login Redirect URI

Scopes

profile

Add Scope

Response Types

code

Add Response Type

Grant Types

authorization_code

Add Grant Type

Redirect Logout URIs

Post Logout Redirect URI

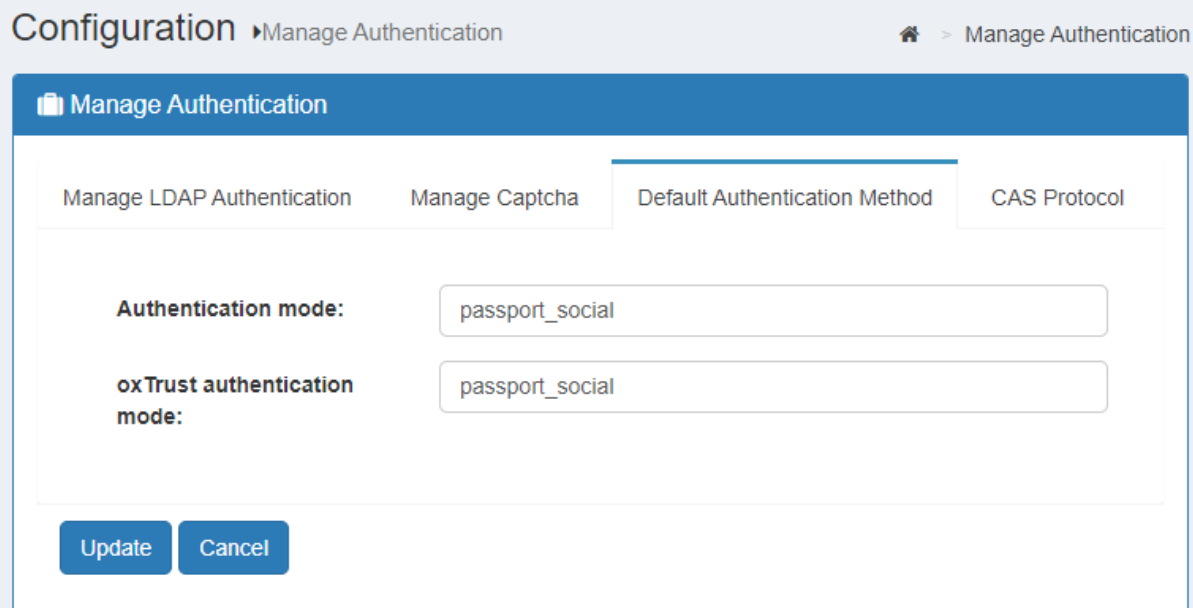
Logo URI: [Empty]

Policy URI: [Empty]

Registro del cliente en Gluu

Fue necesario utilizar **https** debido a que Gluu solo acepta **Redirect Login URIs** que sean bajo este protocolo, permitiendo **http** solamente para **localhost** o **127.0.0.1**.

También fue necesario ir a la pestaña **Configuration > Manage Authentication > Default Authentication Method** y cambiar en valor de **Authentication mode** a *passport_social*.



The screenshot shows the Gluu Configuration interface. At the top, there's a breadcrumb trail: Configuration > Manage Authentication. Below this is a tabbed interface with four tabs: Manage LDAP Authentication, Manage Captcha, Default Authentication Method (which is selected), and CAS Protocol. The Default Authentication Method tab contains two input fields. The first is labeled 'Authentication mode:' and the second is labeled 'oxTrust authentication mode:'. Both fields have 'passport_social' entered. At the bottom of the form are two buttons: 'Update' and 'Cancel'.

Configuración método de autenticación

La aplicación de prueba es un simple html con tres links:

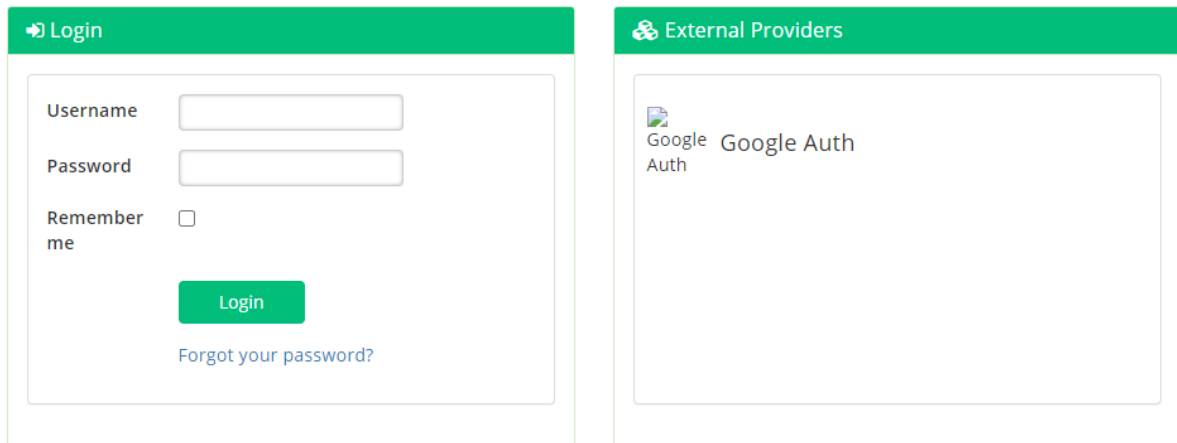
1. Login: redirecciona al usuario hacia Gluu para que inicie sesión.
2. Secret: muestra un mensaje dependiendo de si el usuario inició sesión o no.
3. Logout: termina la sesión del usuario.

- [Login](#)
- [Secret](#)
- [Logout](#)

Opciones en <https://testapp1.operaciones.tk>

Si el usuario no ha iniciado sesión, se le muestra el mensaje “*You must login!*”, mientras que si ya lo hizo verá el mensaje “*You have reached the secret route*”.

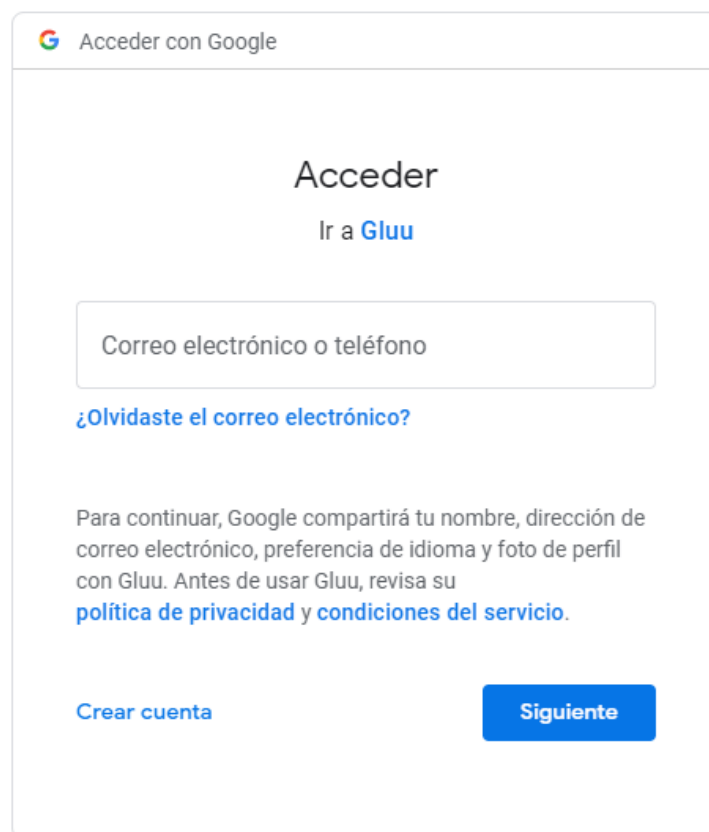
Al presionar el login se le redirecciona a la vista de Gluu para iniciar sesión.



The image shows two side-by-side login panels. The left panel, titled "Login", has a green header bar. Below it, there are input fields for "Username" and "Password", a "Remember me" checkbox, a green "Login" button, and a link "Forgot your password?". The right panel, titled "External Providers", also has a green header bar. It contains a box with a Google logo and the text "Google Auth".

Login Gluu

Al presionar en Google Auth es redireccionado a Google,



The image shows a Google login screen. At the top, there is a Google logo and the text "Acceder con Google". Below this, the word "Acceder" is prominently displayed, followed by a link "Ir a Gluu". There is a text input field for "Correo electrónico o teléfono". Below the input field, there is a link "¿Olvidaste el correo electrónico?". A paragraph of text follows, stating: "Para continuar, Google compartirá tu nombre, dirección de correo electrónico, preferencia de idioma y foto de perfil con Gluu. Antes de usar Gluu, revisa su política de privacidad y condiciones del servicio." At the bottom, there are two links: "Crear cuenta" and "Siguiente".

Login Google

Donde al ingresar un correo USACH es redireccionado al login institucional.



Acceso correo corporativo
USACH

Estimad@ Usuari@:
Recuerde que la contraseña de su correo debe ser renovada
al menos cada 180 días.

Usuario

Contraseña

Ingresar

[Recuperación de contraseña](#) | [Cambiar informaciones recuperación contraseña](#)

[Guía para cambio de contraseña](#)

 UNIVERSIDAD
DE SANTIAGO
DE CHILE

Login USACH

Luego de ingresar las credenciales correspondientes, se hace todo el camino de respuesta de vuelta mediante los Callbacks (Primer Callback desde Google a Gluu y luego desde Gluu a TestApp).

Al igual que con **OAuth 2.0 debugger**, si es la primera vez que el usuario está ingresando, se pedirá el permiso para entregar información como el perfil.

Como se puede apreciar, se producen varios saltos entre distintos proveedores de identidad, por lo que probablemente sea necesario buscar formas de simplificar algunos pasos, tal vez **configurar Gluu para que directamente vaya a Google**, pero esto dejaría a usuarios creados localmente sin capacidad de ingresar, o buscar la forma de **llegar inmediatamente al login USACH sin pasar primero por Google**.