

Laboratory #5

Security

Camilo Andres Dajer Piñerez
Software Architecture
2023-II

Nota: el laboratorio debe ser desarrollado en grupos de laboratorio.

Actividades

i. Requisitos

1. Laboratorio #3 finalizado en su totalidad.
2. Instalar la herramienta [Docker Compose](#).

ii. Lightweight Directory Access Protocol (LDAP)

a. OpenLDAP

OpenLDAP es una implementación libre y de código abierto del protocolo LDAP.

b. Servidor LDAP

1. Crear un directorio llamado **swarch2023ii_ldap**.
2. En la raíz, crear el siguiente archivo *docker-compose.yml*:

```
version: '2.1'
services:
  swarch2023ii-ldap:
    image: osixia/openldap:1.1.8
    container_name: swarch2023ii_ldap
    environment:
      COMPOSE_HTTP_TIMEOUT: 200
      LDAP_LOG_LEVEL: "256"
      LDAP_ORGANISATION: "Software Architecture"
      LDAP_DOMAIN: "arqsoft.unal.edu.co"
      LDAP_BASE_DN: ""
      LDAP_ADMIN_PASSWORD: "admin"
      LDAP_CONFIG_PASSWORD: "config"
      LDAP_READONLY_USER: "false"
      #LDAP_READONLY_USER_USERNAME: "readonly"
      #LDAP_READONLY_USER_PASSWORD: "readonly"
      LDAP_BACKEND: "hdb"
      LDAP_TLS: "true"
      LDAP_TLS_CRT_FILENAME: "ldap.crt"
      LDAP_TLS_KEY_FILENAME: "ldap.key"
      LDAP_TLS_CA_CRT_FILENAME: "ca.crt"
      LDAP_TLS_ENFORCE: "false"
      LDAP_TLS_CIPHER_SUITE: "SECURE256:-VERS-SSL3.0"
      LDAP_TLS_PROTOCOL_MIN: "3.1"
      LDAP_TLS_VERIFY_CLIENT: "demand"
      LDAP_REPLICATION: "false"
      #LDAP_REPLICATION_CONFIG_SYNCPROV: "binddn=cn=admin,cn=config"
      bindmethod=simple credentials=$LDAP_CONFIG_PASSWORD searchbase="cn=config"
      type=refreshAndPersist retry="60 +" timeout=1 starttls=critical"
      #LDAP_REPLICATION_DB_SYNCPROV: "binddn=cn=admin,$LDAP_BASE_DN"
      bindmethod=simple credentials=$LDAP_ADMIN_PASSWORD searchbase="$LDAP_BASE_DN"
      type=refreshAndPersist interval=00:00:00:10 retry="60 +" timeout=1
      starttls=critical"
      #LDAP_REPLICATION_HOSTS:
      "#PYTHON2BASH:['ldap://ldap.example.org','ldap://ldap2.example.org']"
      LDAP_REMOVE_CONFIG_AFTER_SETUP: "true"
      LDAP_SSL_HELPER_PREFIX: "ldap"
    tty: true
    stdin_open: true
    volumes:
      - /var/lib/ldap
      - /etc/ldap/slapd.d
      - /container/service/slapd/assets/certs/
    ports:
      - "389:389"
      - "636:636"
    hostname: "arqsoft.unal.edu.co"
  phpldapadmin:
    image: osixia/phpldapadmin:latest
    container_name: ldap_client
```

```
environment:
  PHPLDAPADMIN_LDAP_HOSTS: "swarch2023ii-ldap"
  PHPLDAPADMIN_HTTPS: "false"
ports:
  - "8085:80"
links:
  - swarch2023ii-ldap
```

3. Analizar la estructura y el contenido del archivo anterior.

4. El primer elemento agregado hace referencia al componente **LDAP**. A continuación se describen algunos de los elementos más relevantes de su configuración:

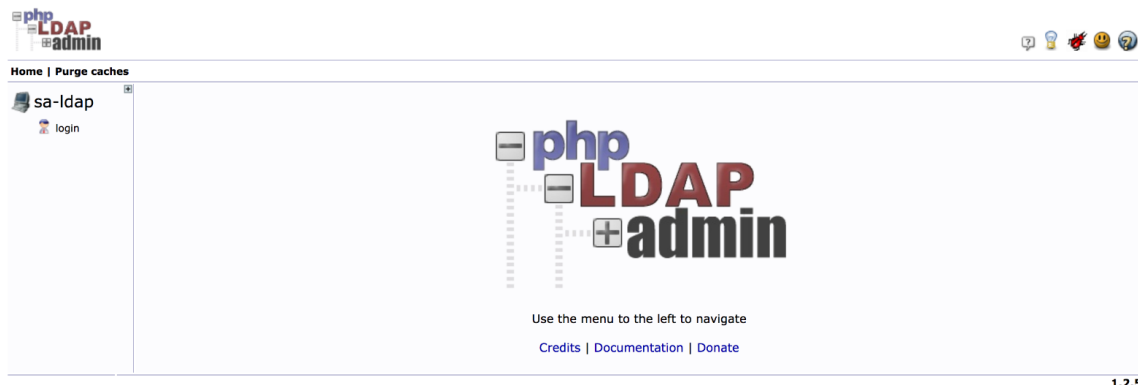
- ❑ **LDAP_ORGANISATION**: organización que utilizará el servidor LDAP.
- ❑ **LDAP_DOMAIN**: dominio que utilizará el servidor LDAP, la separación con puntos le indica al servidor LDAP como se estructurará dicho dominio. Para este caso particular, se tendrá la representación "arqsoft.unal.edu.co", que en lenguaje LDAP representa: "dc=arqsoft,dc=unal,dc=edu,dc=co", donde *dc* = *domain component*.
- ❑ **LDAP_BASE_DN**: al iniciar el servidor LDAP se le podrá asignar un valor de ruta de dominio para que siempre inicie desde ahí.
- ❑ **LDAP_ADMIN_PASSWORD**: contraseña asociada al administrador.
- ❑ La ruta del administrador será "cn=admin,dc=arqsoft,dc=unal,dc=edu,dc=co", donde *cn* = *common name*.

5. El segundo elemento agregado hace referencia a un gestor gráfico (phpLDAPadmin) que servirá para la administración del servidor LDAP.

6. Desplegar el componente:

`docker-compose up`

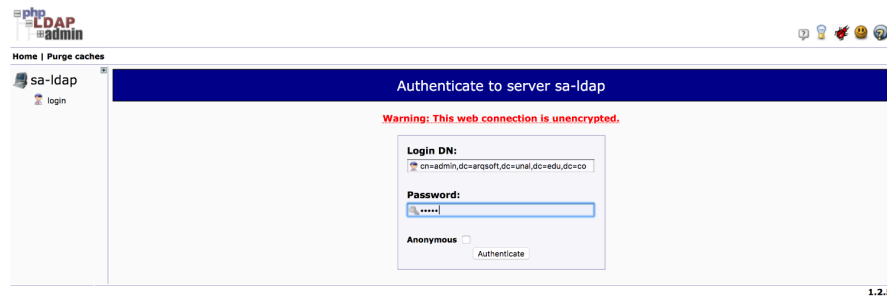
7. Verificar que el gestor gráfico se encuentre desplegado. Acceder a la ruta `localhost:8085` desde el navegador web.



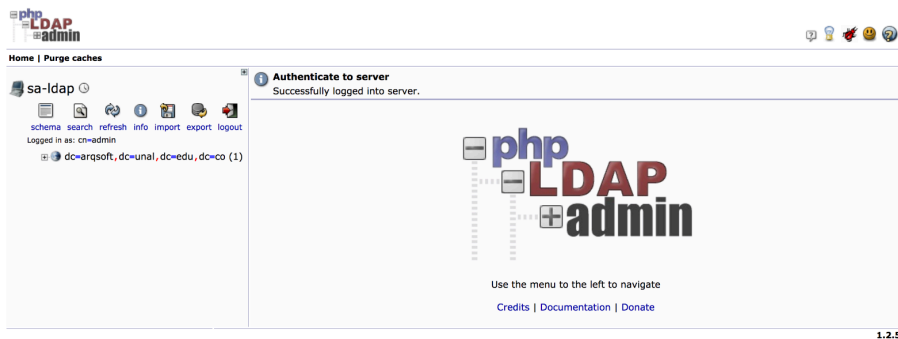
c. Configuración del Directorio

1. Hacer clic en **login** e ingresar con los siguientes datos:

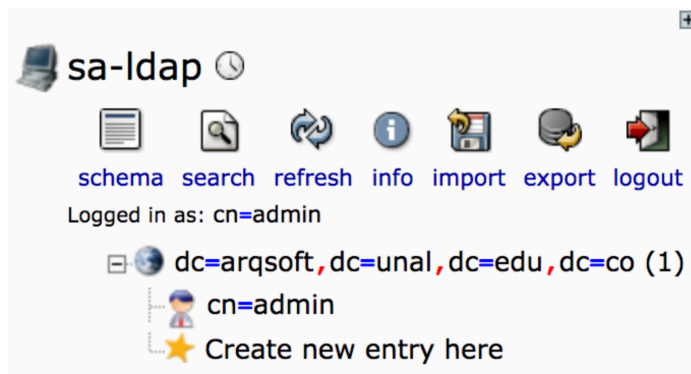
- ☐ **Login DN:** cn=admin,dc=arqsoft,dc=unal,dc=edu,dc=co
- ☐ **Password:** admin



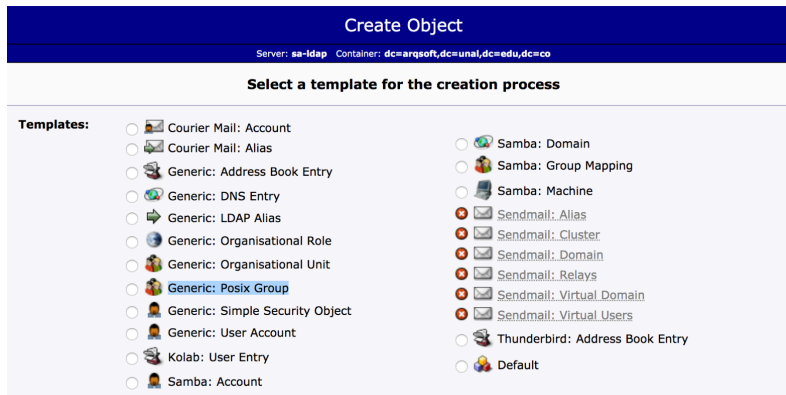
2. Si los datos fueron ingresados correctamente, se podrá ver la página de inicio con la estructura del servidor LDAP.



3. Ir al panel izquierdo y desplegar la pestaña del dominio, posteriormente hacer clic en **Create new entry here:**



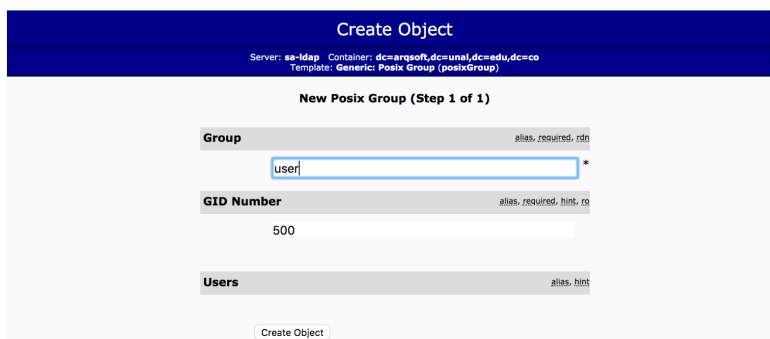
4. Seleccionar **Generic: Posix Group**. Este grupo permitirá diferenciar tipos de usuario:



The screenshot shows the 'Create Object' window with the following details:

- Server: **sa-ldap** Container: **dc=argsoft,dc=unal,dc=edu,dc=co**
- Section: **Select a template for the creation process**
- Templates list (left column):
 - ☐ Courier Mail: Account
 - ☐ Courier Mail: Alias
 - ☐ Generic: Address Book Entry
 - ☐ Generic: DNS Entry
 - ☐ Generic: LDAP Alias
 - ☐ Generic: Organisational Role
 - ☐ Generic: Organisational Unit
 - ☒ **Generic: Posix Group**
 - ☐ Generic: Simple Security Object
 - ☐ Generic: User Account
 - ☐ Kolab: User Entry
 - ☐ Samba: Account
- Templates list (right column):
 - ☐ Samba: Domain
 - ☐ Samba: Group Mapping
 - ☐ Samba: Machine
 - ☒ Sendmail: Alias
 - ☒ Sendmail: Cluster
 - ☒ Sendmail: Domain
 - ☒ Sendmail: Relays
 - ☒ Sendmail: Virtual Domain
 - ☒ Sendmail: Virtual Users
 - ☐ Thunderbird: Address Book Entry
 - ☐ Default

5. Crear un nuevo grupo llamado **user**, hacer clic en **Create Object** y luego en **Commit**.

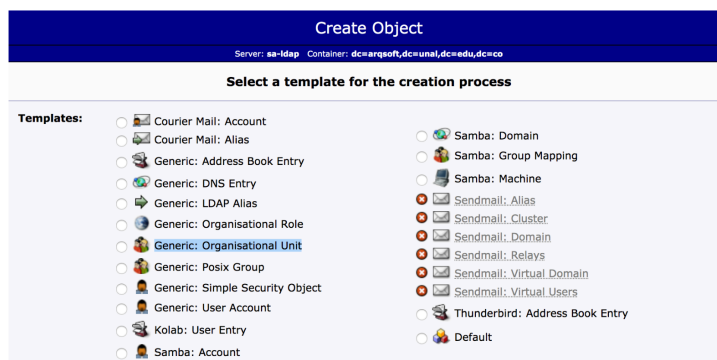


The screenshot shows the 'Create Object' window with the following details:

- Server: **sa-ldap** Container: **dc=argsoft,dc=unal,dc=edu,dc=co** Template: **Generic: Posix Group (posixGroup)**
- Section: **New Posix Group (Step 1 of 1)**
- Form fields:
 - Group**: (alias, required, rdn)
 - GID Number**: (alias, required, hint, ro)
 - Users**: (alias, hint)
- Buttons: **Create Object**

6. Ir nuevamente al panel izquierdo y hacer clic en **Create new entry here**.

7. Seleccionar **Generic: Organisational Unit**. Estas unidades hacen referencia principalmente a los diferentes sistemas de software que harán uso del servidor LDAP.



The screenshot shows the 'Create Object' window with the following details:

- Server: **sa-ldap** Container: **dc=argsoft,dc=unal,dc=edu,dc=co**
- Section: **Select a template for the creation process**
- Templates list (left column):
 - ☐ Courier Mail: Account
 - ☐ Courier Mail: Alias
 - ☐ Generic: Address Book Entry
 - ☐ Generic: DNS Entry
 - ☐ Generic: LDAP Alias
 - ☐ Generic: Organisational Role
 - ☒ **Generic: Organisational Unit**
 - ☐ Generic: Posix Group
 - ☐ Generic: Simple Security Object
 - ☐ Generic: User Account
 - ☐ Kolab: User Entry
 - ☐ Samba: Account
- Templates list (right column):
 - ☐ Samba: Domain
 - ☐ Samba: Group Mapping
 - ☐ Samba: Machine
 - ☒ Sendmail: Alias
 - ☒ Sendmail: Cluster
 - ☒ Sendmail: Domain
 - ☒ Sendmail: Relays
 - ☒ Sendmail: Virtual Domain
 - ☒ Sendmail: Virtual Users
 - ☐ Thunderbird: Address Book Entry
 - ☐ Default

8. Crear una nueva unidad organizacional llamada **sa**, hacer clic en **Create Object** y luego en **Commit**.

The screenshot shows the 'Create Object' wizard for a new Organisational Unit. The header bar is dark blue with the text 'Create Object'. Below it, a status bar shows 'Server: sa-ldap', 'Container: dc=argsoft,dc=una,dc=edu,dc=co', and 'Template: Generic: Organisational Unit (ou)'. The main content area is titled 'New Organisational Unit (Step 1 of 1)'. It features a form labeled 'Organisational Unit' with a text input field containing 'sa'. To the right of the input field is a hint: 'alias, required, rdn, hint'. Below the input field is a 'Create Object' button.

9. Una vez creada la unidad organizacional, se deben crear los usuarios que harán parte de ella. En este caso, crear un nuevo usuario para el sistema de software que hará uso del servicio LDAP. Para ello, seleccionar **ou=sa** y hacer clic en **Create a child entry**.

The screenshot shows the 'OU=sa' entry page. The header bar is dark blue with the text 'OU=sa'. Below it, a status bar shows 'Server: sa-ldap', 'Distinguished Name: ou=sa,dc=argsoft,dc=una,dc=edu,dc=co', and 'Template: Default'. The main content area lists various actions: 'Refresh', 'Switch Template', 'Copy or move this entry', 'Rename', 'Create a child entry' (highlighted with a star), 'Show internal attributes', 'Export', 'Delete this entry', 'Compare with another entry', and 'Add new attribute'. There are also two hints: 'Hint: To delete an attribute, empty the text field and click save.' and 'Hint: To view the schema for an attribute, click the attribute name.'

10. Seleccionar **Generic: User Account**:

The screenshot shows the 'Create Object' wizard for selecting a template. The header bar is dark blue with the text 'Create Object'. Below it, a status bar shows 'Server: sa-ldap', 'Container: ou=sa,dc=argsoft,dc=una,dc=edu,dc=co', and 'Template: Default'. The main content area is titled 'Select a template for the creation process'. It features a list of templates under the heading 'Templates:'. The templates are arranged in two columns. The first column includes: 'Courier Mail: Account', 'Courier Mail: Alias', 'Generic: Address Book Entry', 'Generic: DNS Entry', 'Generic: LDAP Alias', 'Generic: Organisational Role', 'Generic: Organisational Unit', 'Generic: Posix Group', 'Generic: Simple Security Object', 'Generic: User Account' (highlighted with a blue bar), 'Kolab: User Entry', and 'Samba: Account'. The second column includes: 'Samba: Domain', 'Samba: Group Mapping', 'Samba: Machine', 'Sendmail: Alias', 'Sendmail: Cluster', 'Sendmail: Domain', 'Sendmail: Relays', 'Sendmail: Virtual Domain', 'Sendmail: Virtual Users', 'Thunderbird: Address Book Entry', and 'Default'.


11. En el formulario de creación de la cuenta de usuario, diligenciar la información del usuario que se desea registrar en el servidor LDAP, hacer clic en **Create Object** y luego en **Commit**.

Create Object

Server: sa-ldap Container: ou=sa,dc=arqsoft,dc=unal,dc=edu,dc=co
Template: Generic: User Account (posixAccount)

New User Account (Step 1 of 1)

First namealias

 Jeisson Andrés

Last namealias, required

Vergara Vargas*


Common Namealias, required, rdn


javergarav@unal.edu.co*

User IDalias, required

javergarav*


Passwordalias, hint

 ... md5

 ... (confirm)

Check password...

UID Numberalias, required, hint, ro

 1000

GID Numberalias, required, hint

user*

Home directoryalias, required

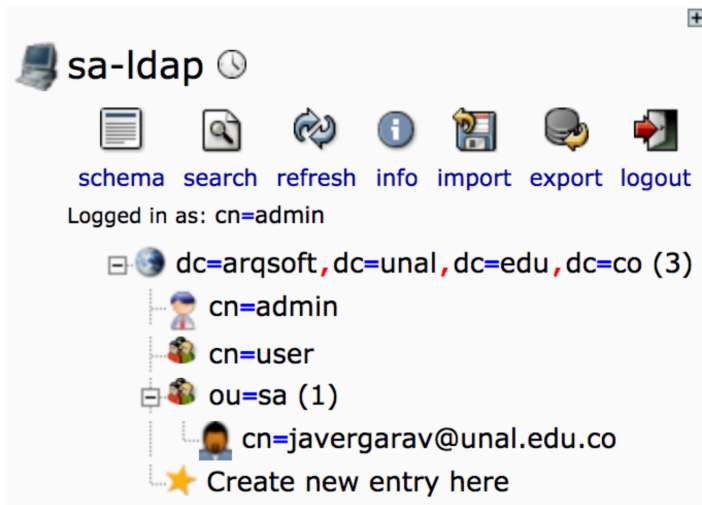
/home/users/javergarav*

Login shellalias

Create Object

Password: 123

12. Una vez creado el usuario, éste se podrá evidenciar en la estructura del directorio LDAP creado:



iii. Proxy Inverso

a. Nginx

Nginx es un servidor web y proxy inverso, es decir que su función es ser un intermediario en las peticiones de recursos. De esta manera, se pueden implementar medidas de control de acceso, registro del tráfico, restricción a determinados tipos de archivos, mejora de rendimiento, caché web, entre otras.

b. Implementación

1. Crear un directorio llamado **swarch2023ii_proxy**.
2. En la raíz, crear el siguiente archivo **Dockerfile**:

```
FROM nginx

RUN apt-get update -qq && apt-get -y install apache2-utils
ENV NODE_ROOT /var/www/api-gateway
WORKDIR $NODE_ROOT
RUN mkdir log
COPY app.conf /tmp/app.nginx
RUN envsubst '$NODE_ROOT' < /tmp/app.nginx > /etc/nginx/conf.d/default.conf

EXPOSE 80

CMD [ "nginx", "-g", "daemon off;" ]
```

3. Crear un archivo llamado **app.conf**:

```
upstream api_gateway_node {
    server localhost:5000;
}
```



```

server {
    listen 80;
    proxy_buffers 64 16k;
    proxy_max_temp_file_size 1024m;
    proxy_connect_timeout 5s;
    proxy_send_timeout 10s;
    proxy_read_timeout 10s;

    location ~ /\. {
        deny all;
    }

    location ~* ^.+\. (rb|log)$ {
        deny all;
    }

    # serve static (compiled) assets directly if they exist (for node
    production)
    location ~ ^/(assets|images|javascripts|stylesheets|swfs|system)/ {
        try_files $uri @api_gateway_node;

        access_log off;
        gzip_static on; # to serve pre-gzipped version

        expires max;
        add_header Cache-Control public;

        # Some browsers still send conditional-GET requests if there's a
        # Last-Modified header or an ETag header even if they haven't
        # reached the expiry date sent in the Expires header.
        add_header Last-Modified "";
        add_header ETag "";
        break;
    }

    location / {
        try_files $uri $uri/ @api_gateway_node;
    }

    location @api_gateway_node {
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
        proxy_redirect off;
        proxy_pass http://api_gateway_node;
        access_log /var/www/api-gateway/log/nginx.access.log;
        error_log /var/www/api-gateway/log/nginx.error.log;
    }
}

```

c. Despliegue

1. Desplegar el componente:

```
docker build -t swarch2023ii_proxy .
```

```
docker run -p 80:80 swarch2023ii_proxy
```

Antes de realizar esta acción, asegurarse de que el API Gateway se encuentre desplegado correctamente.

2. Acceder a la ruta <http://localhost/graphiql> y verificar que el **Proxy Inverso** recibe la petición y la traslada al **API Gateway**.

Entrega

Entregable: archivo (en formato *.pdf*) con nombre **I5.pdf**, el cual debe contener:

1. Nombre completo de los integrantes del grupo.
2. Soporte visual del despliegue de los dos componentes de software: *Servidor LDAP* y *Proxy Inverso*.
3. Soporte visual de las configuraciones realizadas en el Servidor LDAP.
4. Soporte visual de la ejecución de las peticiones HTTP sobre la API-GraphQL del API Gateway, pasando primero por el Proxy Inverso.

Forma de Entrega: por medio de la plataforma virtual [Moodle](#).

Fecha de Entrega: Lunes, 30 de octubre de 2023, antes de las: 23:59.

Nota: se debe realizar una única entrega por cada grupo, es decir, solo uno de los integrantes del grupo debe realizar el envío del archivo.