# **LDAP**

## **Tutorial: Instalación y Configuración de LDAP con Wagtail y Django**

### **1. Instalación de Java**

Para poder abrir Apache Directory Studio, instala Java ejecutando:

sudo apt-get install openjdk-11-jdk

### **2. Instalación de Apache Directory Studio**

Descarga e instala Apache Directory Studio desde:  
  
[Apache Directory Studio](https://directory.apache.org/studio/download/download-linux.html)

### **3. Instalación de Docker y Docker Compose**

Ejecuta los siguientes comandos para instalar Docker y Docker Compose:

sudo apt-get update

sudo apt-get install ca-certificates curl

sudo install -m 0755 -d /etc/apt/keyrings

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc

sudo chmod a+r /etc/apt/keyrings/docker.asc

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \

$(. /etc/os-release && echo "$VERSION\_CODENAME") stable" | \

sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

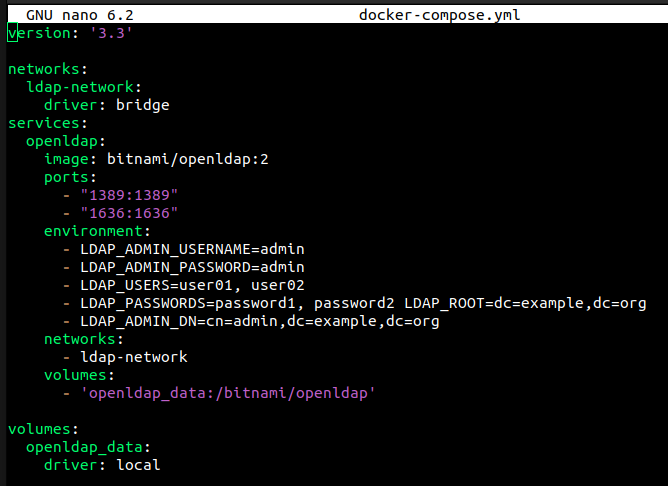
sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

sudo apt install docker-compose -y

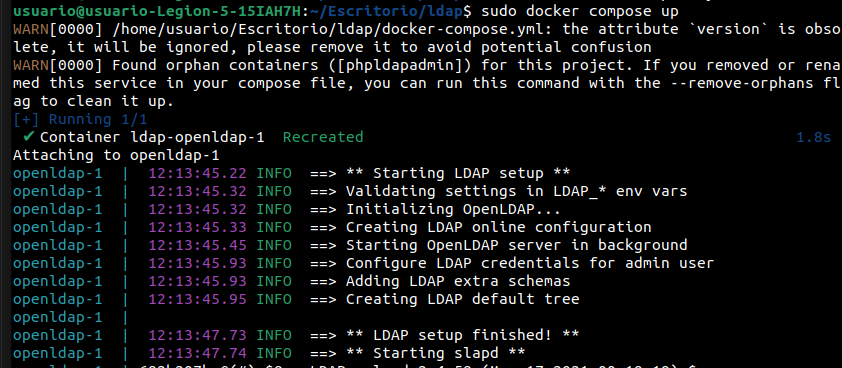
### **4. Levantar el Servidor OpenLDAP con Docker Compose**

Navega al directorio donde se encuentra tu archivo docker-compose.yml

****

y ejecuta:

sudo docker compose up

****

### **5. Configuración de Wagtail en Django**

Edita el archivo settings.py para incluir Wagtail:

INSTALLED\_APPS = [

...,

APPS YA EXISTENTES,

...,

'wagtail.contrib.forms',

'wagtail.contrib.redirects',

'wagtail.embeds',

'wagtail.sites',

'wagtail.users',

'wagtail.snippets',

'wagtail.documents',

'wagtail.images',

'wagtail.search',

'wagtail.admin',

'wagtail',

'taggit',

'modelcluster',

]

Agrega también la configuración de middleware:

MIDDLEWARE = [

...,

MIDDLEWARE YA EXISTENTE,

...,

'wagtail.contrib.redirects.middleware.RedirectMiddleware',

]

Configura las variables de Wagtail:

WAGTAIL\_SITE\_NAME = "HerguTrucks"

WAGTAILADMIN\_BASE\_URL = "http://localhost:8000"

WAGTAILDOCS\_EXTENSIONS = ['csv', 'docx', 'key', 'odt', 'pdf', 'pptx', 'rtf', 'txt', 'xlsx', 'zip']

TAGGIT\_CASE\_INSENSITIVE = True

### **6. Configuración de URLs en Django**

Edita urls.py para incluir las rutas de Wagtail:

from django.contrib import admin

from django.urls import path, include

from wagtail.admin import urls as wagtailadmin\_urls

from wagtail import urls as wagtail\_urls

from wagtail.documents import urls as wagtaildocs\_urls

urlpatterns = [

path('admin/', admin.site.urls),

path('', include('app.urls')),

path('wagtail-admin/', include(wagtailadmin\_urls)),

path('documents/', include(wagtaildocs\_urls)),

path('cms/', include(wagtail\_urls)),

]

### **7. Configuración de LDAP en Django**

Edita settings.py para conectar Django con OpenLDAP:

AUTH\_LDAP\_SERVER\_URI = "ldap://127.0.0.1:1389"

AUTH\_LDAP\_BIND\_DN = "cn=admin,dc=example,dc=org"

AUTH\_LDAP\_BIND\_PASSWORD = "admin"

AUTH\_LDAP\_USER\_SEARCH = LDAPSearch(

"ou=users,dc=example,dc=org", ldap.SCOPE\_SUBTREE, "(uid=%(user)s)"

)

AUTH\_LDAP\_GROUP\_SEARCH = LDAPSearch(

"ou=users,dc=example,dc=org",

ldap.SCOPE\_SUBTREE,

"(objectClass=groupOfNames)",

)

AUTH\_LDAP\_USER\_ATTR\_MAP = {

"first\_name": "givenName",

"last\_name": "sn",

"email": "mail",

}

AUTH\_LDAP\_GROUP\_TYPE = GroupOfNamesType(name\_attr="cn")

AUTH\_LDAP\_USER\_FLAGS\_BY\_GROUP = {

"is\_active": "cn=readers,ou=users,dc=example,dc=org",

"is\_staff": "cn=bloggers,ou=users,dc=example,dc=org",

"is\_superuser": "cn=superusers,ou=users,dc=example,dc=org",

}

### **8. Configuración de Notificaciones por Correo**

EMAIL\_BACKEND = "django.core.mail.backends.smtp.EmailBackend"

EMAIL\_HOST = 'smtp.gmail.com'

EMAIL\_PORT = 587

EMAIL\_USE\_TLS = True

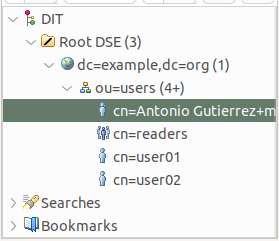
EMAIL\_HOST\_USER = 'antonio.gutierrez.munoz.alu@iesfernandoaguilar.es'

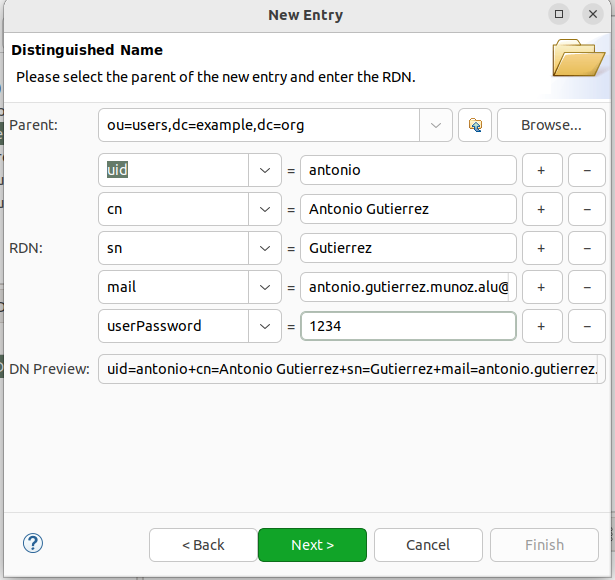
EMAIL\_HOST\_PASSWORD = 'hzwk ddck tizf ovcw'

DEFAULT\_FROM\_EMAIL = EMAIL\_HOST\_USER

### **9. Creación de Usuarios y Grupos en LDAP**

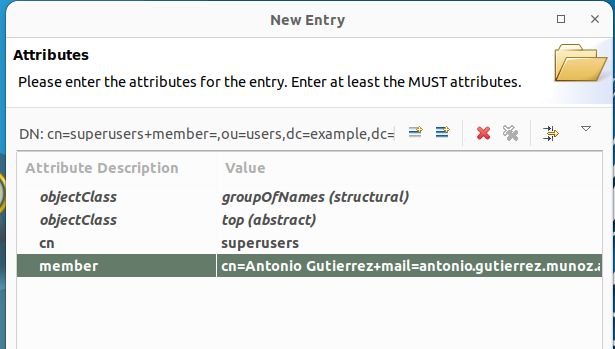
Para crear usuarios y grupos en OpenLDAP, sigue estos pasos en Apache Directory Studio:

1. Abre Apache Directory Studio y conéctate a tu servidor LDAP.
2. Navega hasta ou=users,dc=example,dc=org.
3. Para crear un usuario:
   * Haz clic derecho en ou=users y selecciona New Entry.
   * Usa inetOrgPerson como objectClass.
   * Define uid, cn, sn, mail y userPassword.



* + Agrega el usuario al grupo correspondiente.

1. Para crear un grupo:
   * Haz clic derecho en ou=users y selecciona New Entry.
   * Selecciona Create entry from scratch y usa groupOfNames como objectClass.
   * Define cn como el nombre del grupo (superusers, bloggers, readers).
   * Agrega member para vincular usuarios al grupo.



### **10. Implementación de Señales para LDAP**

Crea signals.py:

from django.contrib.auth.signals import user\_logged\_in

from django.contrib.auth.models import Group

from django.dispatch import receiver

from django\_auth\_ldap.backend import LDAPBackend

from django.core.mail import send\_mail

@receiver(user\_logged\_in)

def user\_add\_to\_bloggers(sender, request, user, \*\*kwargs):

ldap\_backend = LDAPBackend()

ldap\_user = ldap\_backend.populate\_user(user.username)

if ldap\_user:

ldap\_groups = ldap\_user.ldap\_user.group\_names

if "bloggers" in ldap\_groups:

bloggers\_group, created = Group.objects.get\_or\_create(name="Bloggers")

user.groups.add(bloggers\_group)

user.save()

send\_mail(

subject="Has sido registrado en el grupo bloggers",

message="Has sido registrado en HERGUTRUCKS FELICIDADES GUAPO/A.",

from\_email="antonio.gutierrez.munoz.alu@iesfernandoaguilar.es",

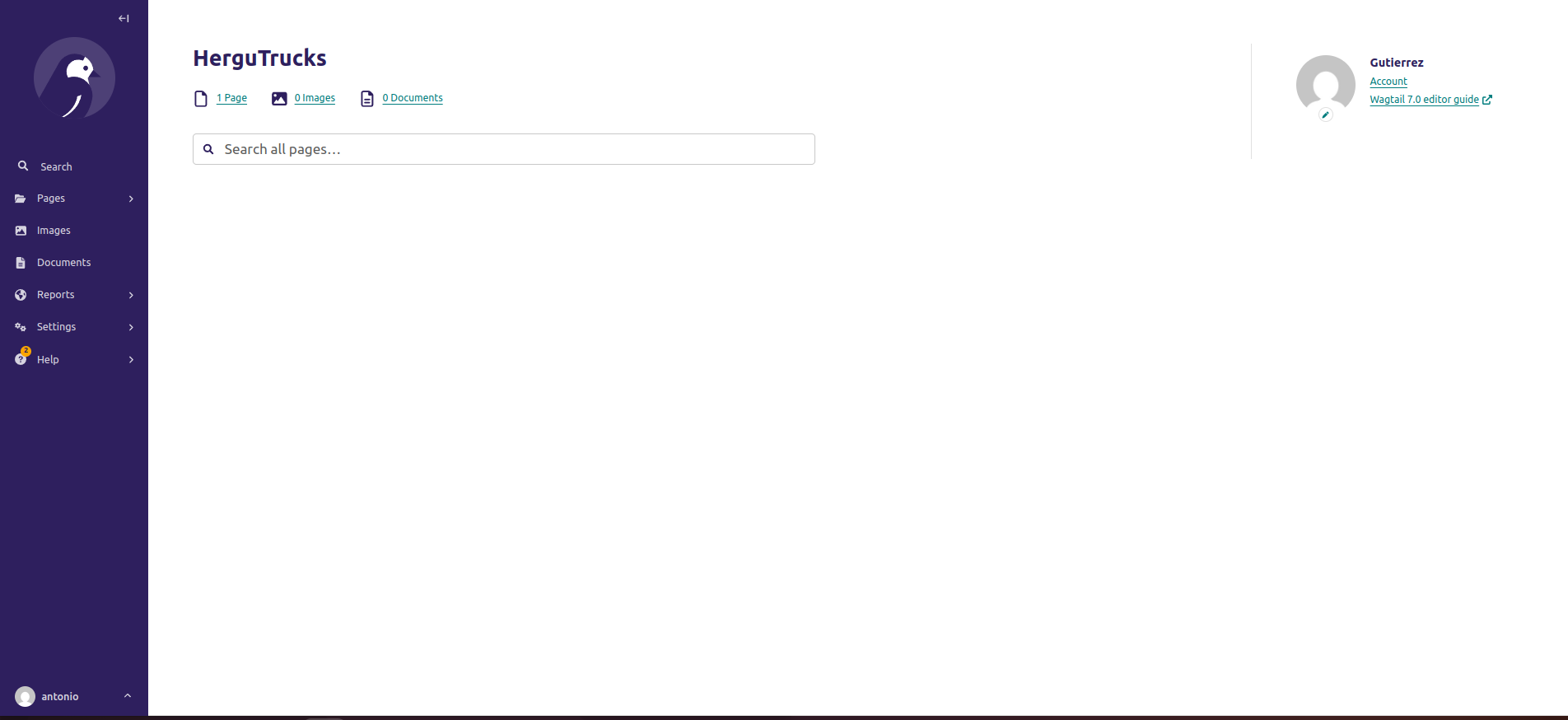
recipient\_list=[user.email, "antonio.gutierrez.munoz.alu@iesfernandoaguilar.es"],

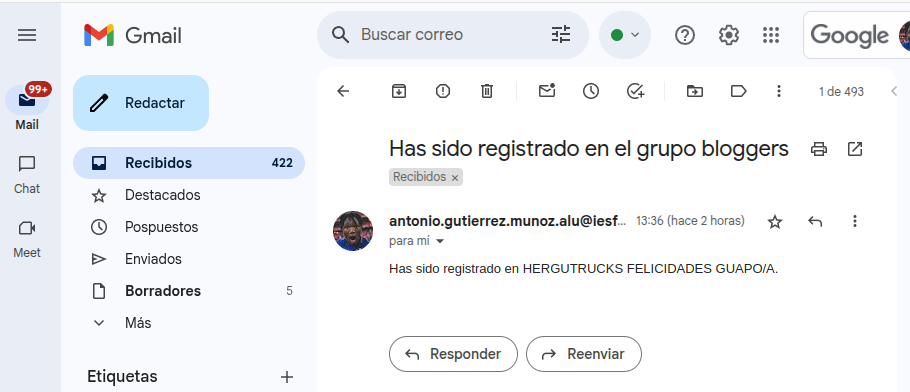
fail\_silently=False

)

### **11. Prueba de funcionamiento**

### 

****

****