

## Unknown Title

---



Tú dijiste:

ldap\_bind: invalid credentilas (49) ===== dn: cn=suarez,ou=usuarios,dc=SUAREZ1,dc=abad2  
objectClass: inetOrgPerson objectClass: posixAccount objectClass: shadowAccount loginShell: /bin/bash  
homeDirectory: /home/suarez uid: suarez cn: suarez userPassword: suarez987 uidNumber: 10001  
gidNumber: 10000 sn: suarez mail: suarez@sistemas.edu

ChatGPT Plus

El error **ldap\_bind: Invalid credentials (49)** significa literalmente que **las credenciales (usuario o contraseña) no son válidas**. Es uno de los errores más comunes al autenticar en LDAP. Veamos las **causas posibles** y cómo solucionarlo en tu caso concreto 👇

---

### 🔍 1. Verifica el DN con el que haces el *bind*

El DN debe coincidir exactamente con el que aparece en tu entrada LDAP.

Según tu entrada:

makefile

```
dn: cn=suarez,ou=usuarios,dc=SUAREZ1,dc=abad2
```

Entonces el *bind DN* debe ser:

ini

```
cn=suarez,ou=usuarios,dc=SUAREZ1,dc=abad2
```

Ejemplo de comando correcto:

bash

```
ldapwhoami -x -D "cn=suarez,ou=usuarios,dc=SUAREZ1,dc=abad2" -w suarez987
```

**!** Si escribes mal una letra en el DN o usas uid=suarez en lugar de cn=suarez, el error 49 aparece.

---

## 🔑 2. Comprueba la contraseña real almacenada

Aunque en tu LDIF aparece:

makefile

```
userPassword: suarez987
```

LDAP puede haberla guardado **codificada** (por ejemplo, {SSHA}xxxxxxxx o {MD5}xxxxxxxx). Si se importó desde un archivo, a veces LAM o slapadd aplican cifrado automáticamente, y la contraseña en texto plano ya **no coincide**.

Prueba a cambiar o reasignar la contraseña:

bash

```
ldappasswd -x -D "cn=admin,dc=SUAREZ1,dc=abad2" -W \
"cn=suarez,ou=usuarios,dc=SUAREZ1,dc=abad2" -s suarez987
```

(Usa la contraseña del cn=admin cuando te la pida.)

Después, vuelve a probar el ldapwhoami.

---

## ✖ 3. Comprueba que el usuario existe realmente

Asegúrate de que la entrada esté cargada en el árbol LDAP:

bash

```
ldapsearch -x -b "ou=usuarios,dc=SUAREZ1,dc=abad2" "(cn=suarez)"
```

Debe mostrar la entrada completa.

Si no aparece, significa que no se importó correctamente o está en otra OU o dominio (por ejemplo, dc=sistemas,dc=edu en lugar de dc=SUAREZ1,dc=abad2).

---

## 4. Verifica si el servidor LDAP tiene acceso restringido

Algunos servidores solo permiten *binds* simples desde localhost, o requieren conexión segura (ldaps:// o -ZZ para STARTTLS).

Ejemplo con ldaps:

bash

```
ldapwhoami -x -H ldaps://localhost -D  
"cn=suarez,ou=usuarios,dc=SUAREZ1,dc=abad2" -w suarez987
```

## Resumen rápido

Possible causa	Solución
DN incorrecto	Usa cn=suarez,ou=usuarios,dc=SUAREZ1,dc=abad2
Contraseña incorrecta o cifrada	Usa ldappasswd para cambiarla
Entrada no existe	Confirma con ldapsearch
Servidor exige LDAPS o STARTTLS	Usa -ZZ o ldaps://
Error de mayúsculas/minúsculas	LDAP es sensible a mayúsculas en el DN

¿Podrías decirme **qué comando exacto** estás usando para hacer el *bind* (el que te devuelve el error)? Así te indico exactamente qué corregir.

Tú dijiste:

ldapadd -x -D cn=admin,dc=sistemas,dc=edu -W -f usuario1.ldif

ChatGPT Plus

¡Ah! Ahora todo tiene sentido. El problema es un **desajuste entre el DN de tu administrador y el dominio de la entrada que estás intentando añadir**.

Vamos a analizarlo paso a paso:

---

- ◆ Comando que estás usando:

bash

```
ldapadd -x -D cn=admin,dc=sistemas,dc=edu -W -f usuario1.ldif
```

- -D cn=admin,dc=sistemas,dc=edu → te estás autenticando como **admin del dominio dc=sistemas,dc=edu**.
- usuario1.ldif tiene esta entrada:

makefile

```
dn: cn=suarez,ou=usuarios,dc=SUAREZ1,dc=abad2
```

Fíjate: **el DN de la entrada pertenece a dc=SUAREZ1,dc=abad2, no a dc=sistemas,dc=edu.**

LDAP no te deja agregar una entrada fuera del árbol sobre el que tu DN de admin tiene permisos. Por eso da error de credenciales (49), porque desde la perspectiva de cn=admin,dc=sistemas,dc=edu, dc=SUAREZ1,dc=abad2 no existe o no tiene permisos.

#### ◆ Cómo solucionarlo

##### 1. Usar el DN de admin correcto para el dominio dc=SUAREZ1,dc=abad2.

Si tu servidor tiene un administrador para ese dominio, el comando sería algo así:

bash

```
ldapadd -x -D cn=admin,dc=SUAREZ1,dc=abad2 -W -f usuario1.ldif
```

##### 2. Si no existe un admin para ese dominio, puedes:

- Crear un administrador en dc=SUAREZ1,dc=abad2.
- O cambiar el LDIF para que coincida con tu dominio actual dc=sistemas,dc=edu.

#### ◆ Alternativa: modificar la entrada LDIF para tu dominio

Si quieres mantener tu admin cn=admin,dc=sistemas,dc=edu, cambia la entrada:

ldif

```
dn: cn=suarez,ou=usuarios,dc=sistemas,dc=edu
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
loginShell: /bin/bash
homeDirectory: /home/suarez
uid: suarez
cn: suarez
userPassword: suarez987
uidNumber: 10001
```

```
gidNumber: 10000
sn: suarez
mail: suarez@sistemas.edu
```

Luego puedes ejecutar tu `ldapadd` sin errores.

---

Siquieres, puedo **reescribirte tu LDIF usuario1.ldif** listo para usar con tu admin actual para que no tengas que tocar nada más.

¿Quieres que lo haga?

Tú dijiste:

`ldapadd: invalid format (line 1) entry: ""`

ChatGPT Plus

Ah, este error **`ldapadd: invalid format (line 1) entry: ""`** significa que **LDAP no puede interpretar tu archivo LDIF**, y normalmente ocurre por alguno de estos motivos:

---

- ◆ **1. Línea en blanco al inicio**

El LDIF no puede empezar con una línea vacía.

Ejemplo incorrecto:

`ldif`

```
<BLANCO>
dn: cn=suarez,ou=usuarios,dc=sistemas,dc=edu
...
```

Solución: elimina cualquier línea en blanco antes del `dn:`.

---

- ◆ **2. Saltos de línea incorrectos**

- Cada entrada debe estar separada por **una línea en blanco**, pero **no debe haber líneas en blanco dentro de la entrada**.
- Cada atributo va en su propia línea.

Ejemplo correcto:

`ldif`

```
dn: cn=suarez,ou=usuarios,dc=sistemas,dc=edu
objectClass: inetOrgPerson
```

```
objectClass: posixAccount
objectClass: shadowAccount
loginShell: /bin/bash
homeDirectory: /home/suarez
uid: suarez
cn: suarez
userPassword: suarez987
uidNumber: 10001
gidNumber: 10000
sn: suarez
mail: suarez@sistemas.edu
```

Después de esta entrada puedes añadir otra entrada dejando **una línea vacía** entre ellas.

---

### ◆ 3. Codificación del archivo

- Asegúrate de que el archivo esté en **UTF-8 sin BOM**.
  - Evita caracteres invisibles que algunos editores ponen al inicio.
- 

### ◆ 4. Comprobación rápida

Puedes probar:

bash

```
cat -A usuario1.ldif
```

- Esto muestra todos los caracteres especiales y líneas vacías.
  - La primera línea debe ser exactamente dn: cn=suarez, ... sin nada encima.
- 

Si quieras, puedo **reescribirte tu LDIF completamente limpio y listo para ldapadd** para que funcione sin errores.

¿Quieres que haga eso?

Tú dijiste:

me aparece esto cuando ejecuto "cat -A": dn cn=suarez,ou=usuarios,dc=SQUAREZ1,dc=abad2\$  
objectClass: inetOrgPerson\$ objectClass: posixAccount\$ objectClass: shadowAccount\$ loginShell:  
/bin/bash\$ homeDirectory: /home/suarez\$ uid: suarez\$ cn: suarez\$ userPassword: suarez987\$  
uidNumber: 10001\$ gidNumber: 10000\$ sn: suarez\$ mail: suarez@sistemas.edu\$

¡Ah, ahora está claro! 😊

El problema es que la **primera línea de tu LDIF no tiene los dos puntos después de dn.**

Actualmente tienes:

nginx

```
dn: cn=suarez,ou=usuarios,dc=SUAREZ1,dc=abad2
```

Debe ser:

makefile

```
dn: cn=suarez,ou=usuarios,dc=SUAREZ1,dc=abad2
```

- En LDIF, **cada línea de atributo tiene formato atributo: valor**, y el dn **también necesita el :**.

## ✓ LDIF corregido

ldif

```
dn: cn=suarez,ou=usuarios,dc=SUAREZ1,dc=abad2
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
loginShell: /bin/bash
homeDirectory: /home/suarez
uid: suarez
cn: suarez
userPassword: suarez987
uidNumber: 10001
gidNumber: 10000
sn: suarez
mail: suarez@sistemas.edu
```

- Asegúrate de que **no haya líneas en blanco al principio del archivo**.
- Guarda en UTF-8 y luego ejecuta:

bash

```
ldapadd -x -D "cn=admin,dc=SUAREZ1,dc=abad2" -W -f usuario1.ldif
```

Si quieras, puedo hacer una **versión lista para tu dominio actual dc=sistemas,dc=edu**, que además funcione con tu administrador existente, para que no tengas que cambiar nada más.

¿Quieres que haga eso?

Tú dijiste:

como comprobar que los usuarios pertenecen al grupo?

ChatGPT Plus

Para comprobar si los usuarios pertenecen a un grupo en **LDAP**, hay varias formas dependiendo de cómo estén definidos los grupos (por **atributo member**, memberUid, etc.). Te explico paso a paso:

---

## 1 Verifica la definición de tus grupos

Un grupo LDAP típico puede tener:

a) Grupo tipo **posixGroup** (más común en Linux):

ldif

```
dn: cn=alumnos,ou=grupos,dc=sistemas,dc=edu
objectClass: posixGroup
cn: alumnos
gidNumber: 10000
memberUid: suarez
memberUid: miguel
```

- Los usuarios se asocian con **memberUid** que contiene el uid del usuario.

b) Grupo tipo **groupOfNames**:

ldif

```
dn: cn=profesores,ou=grupos,dc=sistemas,dc=edu
objectClass: groupOfNames
cn: profesores
member: cn=suarez,ou=usuarios,dc=sistemas,dc=edu
member: cn=miguel,ou=usuarios,dc=sistemas,dc=edu
```

- Los usuarios se asocian con **member** que contiene el DN completo del usuario.

---

## 2 Buscar miembros de un grupo

a) Si es **posixGroup**

bash

```
ldapsearch -x -b "ou=grupos,dc=sistemas,dc=edu" "(cn=alumnos)" memberUid
```

- Esto te devuelve la lista de usuarios (uid) que pertenecen al grupo alumnos.

## b) Si es groupOfNames

bash

```
ldapsearch -x -b "ou=grupos,dc=sistemas,dc=edu" "(cn=profesores)" member
```

- Te devuelve los DN de los usuarios que pertenecen al grupo profesores.

## 3 Buscar grupos a los que pertenece un usuario

Si quieras ver todos los grupos de un usuario, usa el filtro inverso:

### a) Para posixGroup

bash

```
ldapsearch -x -b "ou=grupos,dc=sistemas,dc=edu" "(&(objectClass posixGroup)(memberUid=suarez))" cn
```

- Devuelve todos los grupos donde suarez es miembro.

### b) Para groupOfNames

bash

```
ldapsearch -x -b "ou=grupos,dc=sistemas,dc=edu" "(&(objectClass=groupOfNames)(member=cn=suarez,ou=usuarios,dc=sistemas,dc=edu))" cn
```

Si quieras, puedo hacerte **un comando único que liste todos los grupos de un usuario** de forma directa, listo para copiar y pegar en tu servidor LDAP. Esto te ahorra varios pasos.

¿Quieres que haga eso?

Tú dijiste:

si

ChatGPT Plus

Perfecto ✓

Aquí tienes los **comandos exactos** para listar los grupos a los que pertenece un usuario en LDAP,

según el tipo de grupos que estés usando.

---

## Si tus grupos son del tipo posixGroup

(es decir, usan memberUid para los miembros)

👉 Comando para ver todos los grupos de un usuario (por ejemplo, suarez):

bash

```
ldapsearch -x -b "ou=grupos,dc=sistemas,dc=edu" "(&(objectClass=posixGroup)(memberUid=suarez))" cn
```

### Qué hace:

- Busca en la rama ou=grupos, dc=sistemas, dc=edu.
- Filtra solo las entradas con objectClass=posixGroup y memberUid=suarez.
- Devuelve el cn de los grupos a los que pertenece.

💡 Resultado esperado:

makefile

```
# alumnos, grupos, sistemas, edu
dn: cn=alumnos,ou=grupos,dc=sistemas,dc=edu
cn: alumnos
```

---

## Si tus grupos son del tipo groupOfNames

(es decir, usan member con el DN completo del usuario)

👉 Comando equivalente:

bash

```
ldapsearch -x -b "ou=grupos,dc=sistemas,dc=edu" "(&(objectClass=groupOfNames)(member=cn=suarez,ou=usuarios,dc=sistemas,dc=edu))" cn
```

### Qué hace:

- Busca todos los grupos (groupOfNames) donde el atributo member contenga el DN del usuario suarez.

💡 Resultado esperado:

makefile

```
# profesores, grupos, sistemas, edu  
dn: cn=profesores,ou=grupos,dc=sistemas,dc=edu  
cn: profesores
```

---

## Extra: detectar automáticamente el tipo de grupo

Si no sabes qué tipo de grupos tienes, ejecuta:

bash

```
ldapsearch -x -b "ou=grupos,dc=sistemas,dc=edu" "(objectClass=*)"  
objectClass cn
```

y mira si los grupos tienen posixGroup o groupOfNames.

---

¿Quieres que te ayude a **crear un grupo y añadir usuarios a él** (por ejemplo, alumnos con suarez y miguel) para probar que estos comandos funcionan?

ChatGPT puede cometer errores. Considera verificar la información importante. Ver [preferencias de cookies](#).