

CONEXIÓN CON NAS

Un servidor NAS es un dispositivo de almacenamiento conectado a la red.

Función: Hacer copias de seguridad de los archivos que se le indique en la configuración, tanto los del ordenador personal como los de cualquier otro dispositivo móvil.

Estructura

Un NAS es un ordenador con su propio sistema operativo y que está adaptado para estar todo el día funcionando. En ellos puedes distinguir dos conjuntos de componentes:

- Por una parte lo que es el NAS en sí con su RAM, su procesador y toda su circuitería
- Los discos duros que puedes añadir a sus ranuras. Dependiendo del modelo o el fabricante, estos discos duros pueden venir incluidos cuando los compres o tendrás que comprarlos aparte

Formas de conexión (Tutoriales importantes marcados con “**”)

<https://youtu.be/qNP588hGOYA>

<https://quintagroup.com/cms/python/freenas>

<https://youtu.be/1MX6t2L2mdE>

<https://youtu.be/1MX6t2L2mdE> **

<https://youtu.be/rd9-AhPwsLM> **

EJEMPLOS DE CONEXIÓN

Para conectar un NAS a código de Python, se puede utilizar una biblioteca llamada "pyNAS" que proporciona una interfaz Python para varios protocolos de almacenamiento de red, incluyendo SMB/CIFS, NFS y FTP.

Ejemplo de cómo utilizar esta biblioteca para conectarse a un NAS SMB/CIFS:

```
```python
import pynas

Conectar a un NAS SMB/CIFS
nas = pynas.SMBNas("nombre_del_servidor", "nombre_de_usuario", "contraseña")

Obtener una lista de archivos en una carpeta compartida
archivos = nas.listdir("/carpeta_compartida")
```

```
Descargar un archivo de la carpeta compartida
nas.download_file("/carpeta_compartida/archivo.txt", "/ruta/local/archivo.txt")

Subir un archivo a la carpeta compartida
nas.upload_file("/ruta/local/archivo.txt", "/carpeta_compartida/archivo.txt")

Desconectar del NAS
nas.disconnect()
'''
```

**Este es solo un ejemplo básico de cómo utilizar la biblioteca pyNAS para conectarse a un NAS y realizar algunas operaciones de archivo. La biblioteca también proporciona otras funciones para trabajar con archivos y carpetas, así como para conectarse a NAS que utilizan otros protocolos de almacenamiento de red.**

## **EJM 2**

Para conectarse a un NAS y descomprimir/comprimir archivos de texto en Python, se podría utilizar la biblioteca pyNAS mencionada anteriormente para conectarse al NAS y luego utilizar la biblioteca integrada `zipfile` de Python para descomprimir y comprimir archivos.

A continuación se muestra un ejemplo de cómo podría hacerse esto:

```
```python
import pynas
import zipfile

# Conexión al NAS SMB/CIFS
nas = pynas.SMBNas("nombre_del_servidor", "nombre_de_usuario", "contraseña")

# Ubicación del archivo comprimido en el NAS
archivo_zip_nas = "/carpeta_compartida/archivo.zip"

# Descarga del archivo comprimido desde el NAS
archivo_zip_local = "archivo.zip"
nas.download_file(archivo_zip_nas, archivo_zip_local)

# Descomprimir el archivo descargado
with zipfile.ZipFile(archivo_zip_local, 'r') as zip_ref:
    zip_ref.extractall()

# Procesar el archivo descomprimido
archivo_descomprimido = "archivo.txt"
```

```

with open(archivo_descomprimido, "r") as archivo:
    contenido = archivo.read()
    # Procesar el contenido del archivo descomprimido aquí...

# Comprimir el archivo procesado
archivo_comprimido = "archivo_procesado.zip"
with zipfile.ZipFile(archivo_comprimido, 'w', compression=zipfile.ZIP_DEFLATED) as
zip_ref:
    zip_ref.write(archivo_descomprimido)

# Subir el archivo comprimido al NAS
nas.upload_file(archivo_comprimido, "/carpeta_compartida/archivo_procesado.zip")

# Eliminar archivos locales
os.remove(archivo_zip_local)
os.remove(archivo_descomprimido)
os.remove(archivo_comprimido)

# Desconexión del NAS
nas.disconnect()
'''

```

Este es solo un ejemplo básico de cómo utilizar la biblioteca pyNAS y la biblioteca `zipfile` integrada de Python para descomprimir y comprimir archivos de texto en un NAS. Dependiendo de sus requisitos específicos, es posible que necesite ajustar el código de ejemplo para adaptarlo a su caso de uso particular.