

# LÄBERIT

PEOPLE + INNOVATION + TECHNOLOGY

# N4D

## Networking for dummies

LLIUUREX 

## ¿Que es N4D?

- Servicio XMLRPC por SSL en python3 que controla todos los sabores de LliureX
- Plugins de fácil desarrollo sin necesidad de grandes conocimientos del framework.
- Ofrece funciones de diversas clases cuyo acceso está protegido por autenticación de usuario o grupo.
- Autenticación a traves de PAM: es decir, usuarios UNIX, LDAP, SQL...
- Autenticación opcional a través de tickets de N4D o clave maestra.
- En esencia, es un sistema de DBUS en red.
- Pese a que es funcional con tan solo un cliente XMLRPC standard, aprovechando el protocolo de respuesta propio del servicio, se han programado librerías cliente en python y C++ para recibir fácilmente las respuestas y el estado de estas.

# Peculiaridades del servicio

- Sistema de variables que soporta una tipología servidor-cliente para poder consultar desde la red las variables del servidor directamente desde los clientes sin tener que dar saltos en la red.
- Notificación del cambio de valor de variables a los clientes conectados para programar eventos disparados por estos en los clientes o el propio servidor.
- Fases de arranque ordenadas.
- Sistema de arranque ordenado de plugins.
- Buzón de ejecución de scripts.
- Autocompletado de argumentos de función:
  - Usuario que ha autenticado para ejecutar la función.
  - IP del cliente que ha solicitado la petición.

# Plugin de ejemplo - /usr/share/n4d/python-plugins

/usr/share/n4d/python-plugins/MyFirstPlugin.py	/etc/n4d/conf.d/MyFirstPlugin.json
<pre>import n4d.responses  import n4d.server.core  class MyFirstPlugin:      def __init__(self):          self.core=n4d.server.core.get_core()          self.predepends=["MySecondPlugin"]      def startup(self,options):          if options["boot"]:  self.core.register_variable_trigger("VAR","MyFirstPlugin",self.my_trigger)      def myfunction(self,a,b):          return n4d.responses.build_successful_response(a)      def my_trigger(self,new_value):          print(new_value)</pre>	<pre>{      "SETUP":{          "type": "python",          "path": "MyFirstPlugin.py",          "class": "MyFirstPlugin"      },      "METHODS":      {          "myfunction":{              "allowed_groups": ["adm"]          },          "protected_args":{              "allowed_groups": ["adm"],              "protected_user": 0,              "protected_ip": 1          }      }  }</pre>

## LliureX y N4D: Llum

Llum - Lliurex User Manager

+ Añadir usuarios genericos

Añadir usuarios a este grupo ▼

Plantilla de usuarios Alumnos ▼

Numero de usuarios a crear 10 ▾

Generación de contraseñas

Contraseña aleatoria

Igual que el nombre de usuario

Misma contraseña para todos los usuarios

**Ha de crear grupos antes de añadir usuarios genéricos**

Conectado a LDAP como: lliurex

Salir

Llum - Lliurex User Manager

Borrar Grupos

Los siguientes usuarios van a ser borrados:

ID del Grupo	Descripción
alu	alu

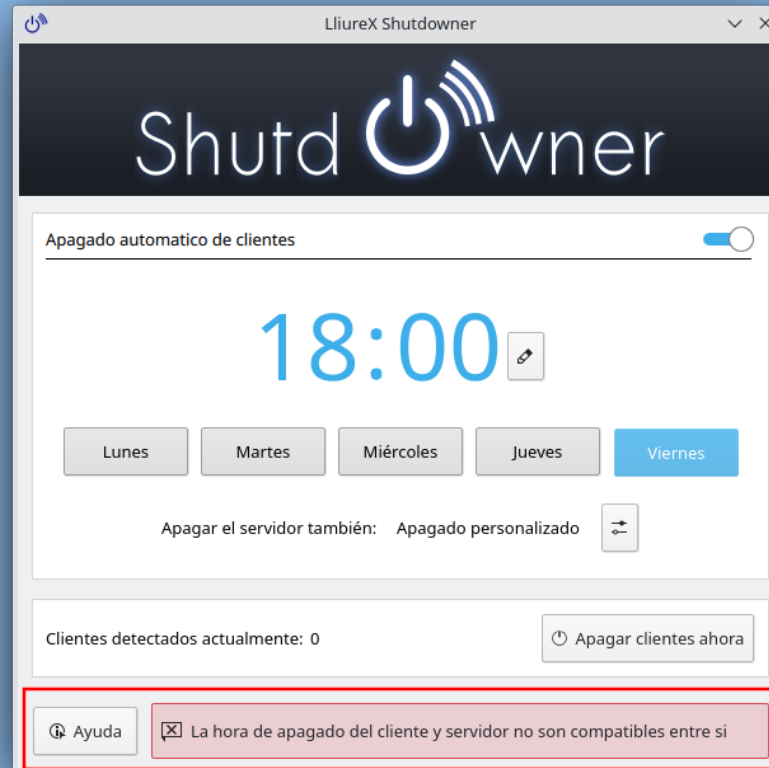
Borrar los usuarios  Borrar datos de usuario en /net también

ID Usuario	Nombre	Apellido
alu01	alu01	alu01
alu02	alu02	alu02
alu03	alu03	alu03

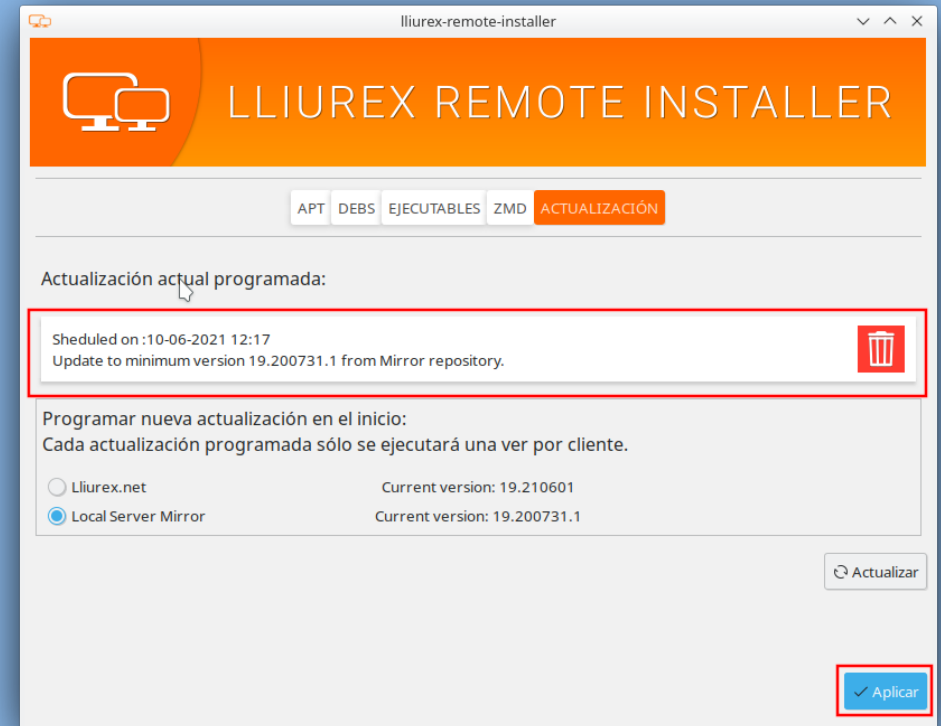
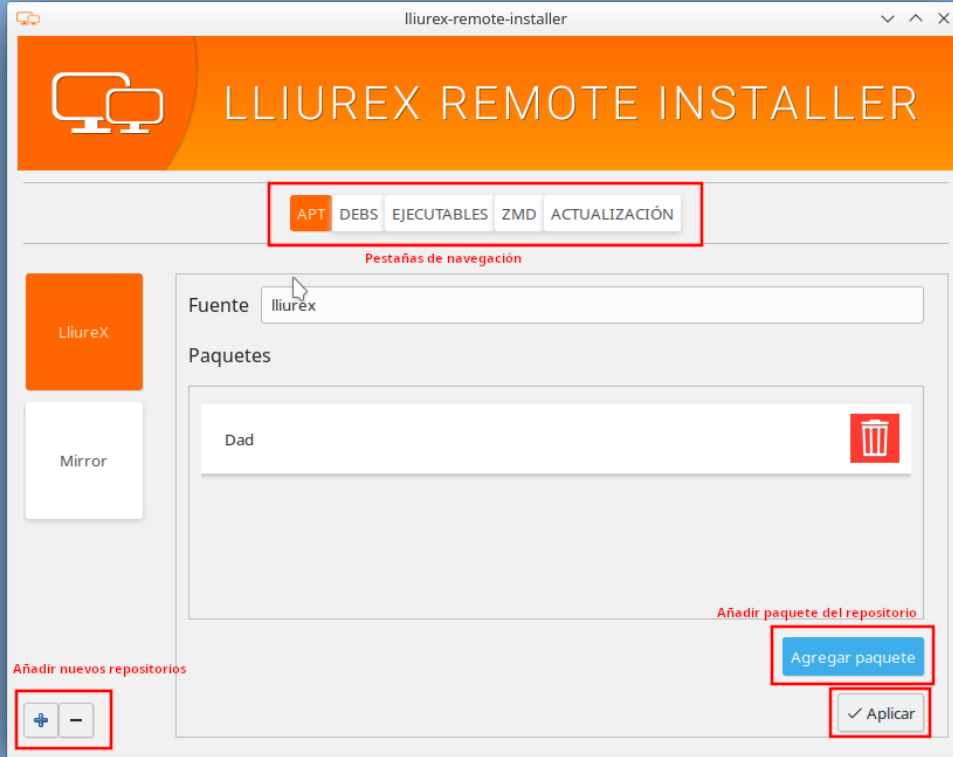
Conectado a LDAP como: lliurex

Salir

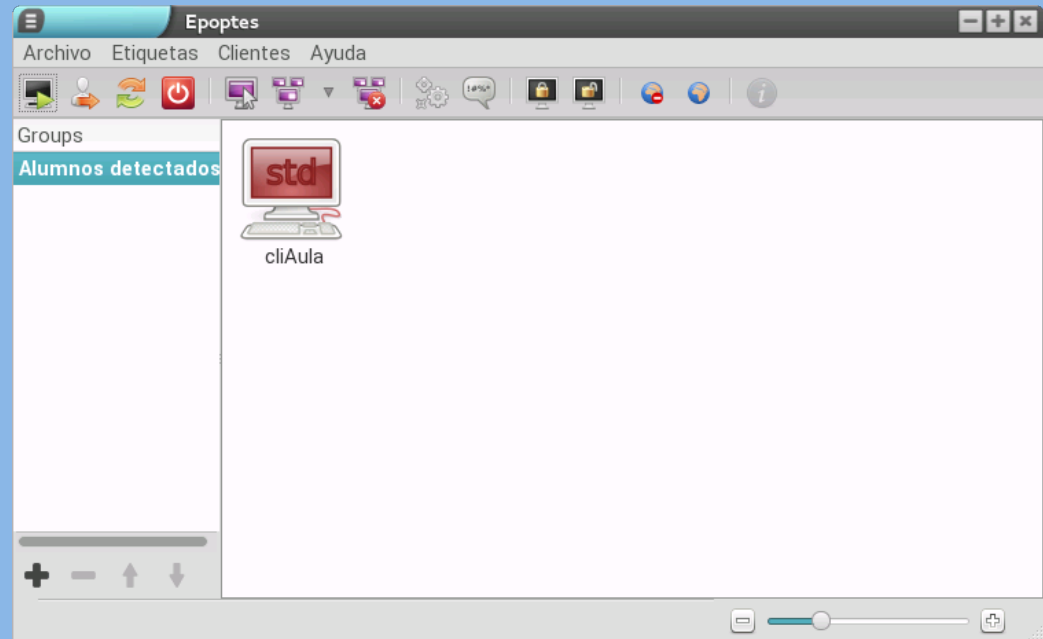
# LliureX y N4D - LliureX Shutdowner



## LliureX y N4D: LliureX Remote Installer



# LliureX y N4D: Epoptes





# LÄBERIT

PEOPLE + INNOVATION + TECHNOLOGY

# Gracias

Héctor García Huertas

- [hgarcia@laberit.com](mailto:hgarcia@laberit.com)
- GitHub: hectorgh

Enrique Medina Gremaldos

- [emedina@laberit.com](mailto:emedina@laberit.com)
- GitHub: Lt-Henry

Raúl Rodrigo Segura

- [rrodrigo@laberit.com](mailto:rrodrigo@laberit.com)
- GitHub: raurodse

Daniel Durán Vecino

- [dduran@laberit.com](mailto:dduran@laberit.com)
- GitHub: daduve

