



# UNIREMINGTON®

CORPORACIÓN UNIVERSITARIA REMINGTON

RES. 2661 MEN JUNIO 21 DE 1996



# **Aliviando un entorno fascista con gotas de colaboración.**

Tatiana Paola Hernández Nieto.lacuerpa@riseup.net  
Bioingeniera UdeA  
Maestría en Educación y Desarrollo Humano

Cristian Alexander Ospina Sánchez  
Daniel Estiven González García  
Estudiantes Ingeniería de Sistemas



# ABP: FEDERACIÓN

Desde marzo del año en curso, me encuentro enseñando en UNIREMINGTON de Rionegro y en específico en la materia Arquitectura de Software, realizo una motivación de Aprendizaje Basado en Problemas (ABP), con el objetivo de lograr generar una Federación de GNUHealth, puesto que en mi país, la salud es pésima y observamos como diariamente le niegan la atención y los medicamentos a los pacientes y acontecen sendas denuncias sobre el robo masivo de los recursos aportados por el estado a unas intermediarias llamadas EPS, que viven enriqueciéndose y acumulando fortunas particulares a merced de la vida de las ciudadanas y ciudadanos que cotizan y contribuyen con sus impuestos para garantizar ese derecho fundamental contemplado en nuestra constitución nacional.



# CONTEXTO

Para dar un poco de contexto, Rionegro es la ciudad donde nació Pablo Escobar, y donde vive Álvaro Uribe Velez, expresidente colombiano de ultra-derecha, hijo del terrateniente del gran narcotraficante en mención y quién creó la Ley 100 según la cuál, hay que pagarle a éstas intermediarias EPS por adelantado, aunque año tras año dejan desfalcadas a las clínicas y hospitales y a su vez, a las médicas y médicos y demás personal de dichas instituciones, a quienes muchas veces se les adeudan hasta tres años de salario. Durante su mandato ocurrieron los “falsos positivos”, dónde 6402 jóvenes que buscaban empleo, fueron asesinados por fuerzas del estado para justificar la política de “seguridad democrática” del expresidente.



# SOLUCIÓN

- No existe en nuestro país un sistema integrado de salud mediante el cuál toda la información sea accesible en cualquier ciudad o entidad de salud, además, el software con los que se manejan las compras, la facturación, las historias clínicas y demás, son privativos, suelen estar difuminados y no se puede ni siquiera perteneciendo a una misma EPS, acceder a los datos de un mismo paciente de una institución a otra.
- Por último y todavía más grave, el software médico es privativo, lo cuál impide la transparencia y trazabilidad necesarias para superar éste problema.
- **GNUHealth es una verdadera solución a éste problema**, sin embargo, la dependencia a Windows es evidente, y la inercia para migrar es muy fuerte.



# AVANCE

- Servidor-cliente autónomo desde una máquina virtual con el Sistema Operativo PureOS
- trabajo colaborativo
- <https://we.riseup.net/arquitecturadesoftware>



# DOCUMENTACIÓN

Daniel Gonzalez

Arquitectura de Software

Inicio Páginas Personas Configuración

## Guía de instalación de GNU Health en PureOS (VirtualBox)

Esta guía resume paso a paso la instalación de GNU Health 5.0 en un entorno VirtualBox con PureOS. Incluye los problemas más comunes que surgieron durante la instalación real y cómo se resolvieron.

Mostrar Editar versiones Imprimir

1. Configuración inicial de la máquina virtual
1. Crear una nueva máquina virtual en VirtualBox.
  - Asigne un nombre (p. ej., pureos-gnuhealth) y elija tipo Linux → Debian (64-bit).
  - Asigne al menos 2 GB de RAM y 32 GB de disco dinámico. Se recomienda usar particionamiento MBR (opción por defecto) para evitar complicaciones.
2. Descargar la ISO de PureOS (debian basada en PureOS) y arrancar la VM con la ISO.
3. Instalar PureOS siguiendo el instalador, seleccionando idioma español, región Bogotá/Colombia y distribución del teclado latinoamericano.
4. Crear cuentas:
  - Cuenta de superusuario (root) con contraseña segura.
  - Cuenta normal gnuhealth (rol posterior del servidor). Añadirla al grupo sudo con sudo usermod -aG sudo gnuhealth para permitir elevación.
2. Preparación del sistema



# DOCUMENTACIÓN

## Daniel Gonzalez

1. Abra una terminal y actualice paquetes:

```
sudo apt update
```

```
sudo apt full-upgrade
```

2. Instale dependencias básicas:

```
sudo apt install python3 python3-venv python3-pip git unzip \
postgresql postgresql-contrib nginx curl build-essential \
libpq-dev libxml2-dev libxslt1-dev libldap2-dev libsasl2-dev libjpeg-dev zlib1g-dev
```

3. Configurar PostgreSQL

GNU Health usa PostgreSQL como base de datos. Debe configurar un clúster 13 (main) y crear un usuario y base de datos.

1. Compruebe que el clúster existe y está iniciado:

```
sudo pg_lsclusters
```

```
sudo pg_ctlcluster 13 main start
```

El comando pg\_lsclusters debe mostrar el clúster 13 main 5432 online.



# DOCUMENTACIÓN

# Daniel Gonzalez



# DOCUMENTACIÓN

## Daniel Gonzalez

1. ... y todos los módulos extra de gnuhealth ...

- Error ModuleNotFoundError: No module named 'PIL': se solucionó instalando Pillow, la librería de Python para imágenes (pip install Pillow).
- Error al compilar psycopg2 psycopg2cffi (pyconfig.h no existe): se resolvió instalando paquetes de desarrollo (libpq-dev y python3-dev) o utilizando psycopg2-binary como arriba.

3. Descargue el código fuente de GNU Health 5.0:

```
cd /opt/gnuhealth
```

```
git clone github.com/gnuhealth/gnuhealth.git —depth 1 -b stable/5.0
```

4. Copie el archivo de configuración por defecto y editelo:

```
cp gnuhealth/tryton/server/trytond.conf /opt/gnuhealth/his-50/etc/trytond.conf
```

```
nano /opt/gnuhealth/his-50/etc/trytond.conf
```

Ajuste estos valores:

```
db_type = postgresql
db_host = localhost
db_port = 5432
db_user = gnuhealth
db_password =
db_name = gnuhealth
admin_passwd =
```

Guarde y cierre.



# DOCUMENTACIÓN

## Daniel Gonzalez

### 5. Inicializar la base de datos y módulos:

```
trytond-admin -c /opt/gnuhealth/his-50/etc/trytond.conf —all -d gnuhealth
```

Error habitual: "password cannot be longer than 72 bytes"

Al ejecutar trytond-admin se produjo un error relacionado con la longitud de la contraseña de admin. La biblioteca bcrypt limita las contraseñas a 72 bytes [\[12202144063270↑screenshot\]](#). Esto se solucionó con dos alternativas:

1. Usar una contraseña más corta (con menos de 72 caracteres). Ejemplo: Gnh2025!Col.

2. Generar un hash pbkdf2 con passlib y actualizar directamente la tabla res\_user en PostgreSQL. Se ejecutó un script de Python para obtener el hash pbkdf2:

```
python - <<'PY'
```

```
from passlib.hash import pbkdf2_sha256
```

```
print(pbkdf2_sha256.hash("Gnh2025!Col"))
```

```
PY
```

Luego se actualizó la base de datos:

```
sudo -u postgres psql -d gnuhealth -c "UPDATE res_user SET password_hash=''" WHERE login='admin';"
```

Esta solución evita el límite de 72 bytes porque pbkdf2 no lo tiene. El archivo trytond.conf se puede ajustar para usar PBKDF2.

Tras aplicar una contraseña corta o pbkdf2, el comando trytond-admin se ejecutó correctamente y creó todas las tablas y módulos.

### 6. Ejecutar el servidor Tryton y el cliente

1. Para lanzar el servidor en el entorno virtual:

```
cd /opt/gnuhealth/his-50
```

```
source venv/bin/activate
```

```
trytond -c /opt/gnuhealth/his-50/etc/trytond.conf
```

Si el comando trytond no se encuentra, instale el paquete trytond dentro del entorno virtual (pip install "trytond==6.0.0").

2. Instale el cliente de GNU Health en el host (PureOS) o en otro equipo:

```
pip3 install —user "gnuhealth-client>=4.4,<4.5"
```

El comando se ejecuta con el Python de sistema (python3 -m pip install —user ...) para no depender del entorno virtual. Si pip no está instalado, ejecute sudo apt install python3-pip.

3. Inicie el cliente con el comando gnuhealth-client-qt. Configure los parámetros de conexión:

- Servidor: localhost (o la IP de la VM).

- Puerto: 8000 (puerto por defecto del servidor Tryton).



# DOCUMENTACIÓN

## Daniel Gonzalez

– Base de datos: gnuhealth.

– Usuario: admin.

– Contraseña: la usada al inicializar.

### 7. Conclusiones y buenas prácticas

- No utilice contraseñas largas para el superusuario admin en GNU Health/Tryton cuando se emplea bcrypt. El límite de 72 bytes provoca errores graves. Use contraseñas seguras pero más cortas o cambie a pbkdf2 como algoritmo.

- Verifique siempre permisos en carpetas como /opt/gnuhealth y /var/lib/postgresql para evitar fallos de "permiso denegado" al crear entornos virtuales o bases de datos.

- Si el error indica que falta un módulo (No module named ...), instálelo con pip install ... dentro del entorno virtual.

- Puede automatizar el inicio del servidor con un servicio systemd para que se ejecute al arrancar la máquina. Consulte la documentación oficial.



# DOCUMENTACIÓN

Cristian Ospina

## Documentacion Cliente-Servidor GNUHealth

Durante la instalación y configuración del sistema GNU Health en una máquina virtual con PureOS , se presentaron varios inconvenientes relacionados con dependencias, permisos y conexión a la base de datos. Este documento explica el proceso completo desde que el sistema abre la ventana del cliente GNU Health solicitando usuario y contraseña, hasta que se logra ingresar correctamente.

[Mostrar](#)

[Editar](#)

[versiones](#)

[Imprimir](#)



# DOCUMENTACIÓN

## Cristian Ospina

### Fase 1 — Configuración del Entorno

#### 1. Configuración del PATH

El comando trytond y gnuhealth-client no eran reconocidos.

Solución:

```
echo 'export PATH="$HOME/.local/bin:$PATH"' >> ~/.bashrc
source ~/.bashrc
```

Esto permitió que el sistema reconociera los comandos de GNU Health instalados localmente.

### Fase 2 — Configuración de la Base de Datos PostgreSQL

#### 2. Instalación y activación del servicio

```
sudo apt install -y postgresql postgresql-contrib
sudo systemctl start postgresql
sudo systemctl enable postgresql
```

#### 3. Creación del usuario y la base de datos

```
sudo -u postgres createuser --createdb --no-createrole --no-superuser gnuhealth
sudo -u postgres psql -c "ALTER USER gnuhealth WITH PASSWORD 'gnuhealth';"
createdb -U gnuhealth gnuhealth
```

Error presentado: el rol «gnuhealth» ya existe.

Solución: Continuar con la configuración, ya que el usuario estaba previamente creado.

Verificación:

```
psql -U gnuhealth -h localhost gnuhealth
```

Resultado: conexión exitosa.

### Fase 3 — Archivo de Configuración del Servidor Tryton

Error inicial: mkdir: no se puede crear el directorio «/etc/gnuhealth»: Permiso denegado.

Solución:

```
sudo mkdir -p /etc/gnuhealth
sudo tee /etc/gnuhealth/trytond.conf > /dev/null << 'EOF'
database
uri = postgresql://gnuhealth:gnuhealth@localhost:5432/
web
```



# DOCUMENTACIÓN

## Cristian Ospina

```
host = localhost
port = 8000
session
timeout = 600
EOF
sudo chown -R $USER:$USER /etc/gnuhealth

Fase 4 — Ejecución del Servidor Tryton

Error inicial: trytond: command not found
Causa: El comando no estaba en el PATH.
Solución: Ejecutarlo directamente:
~/local/bin/trytond -c /etc/gnuhealth/trytond.conf -v

Error: ModuleNotFoundError: No module named 'psycopg2'

Solución: pip3 install psycopg2-binary
Luego de instalarlo, el servidor inició correctamente:
trytond -c /etc/gnuhealth/trytond.conf -v
Resultado exitoso: servidor ejecutándose en localhost:8000

Fase 5 — Inicialización de la Base de Datos

Comando ejecutado:
trytond-admin -c /etc/gnuhealth/trytond.conf -d gnuhealth --all

Problema: no mostraba salida ni pedía datos.

Solución: verificar con curl -v localhost:8000 (respuesta 405 correcta)

Reintentar con:
trytond-admin -c /etc/gnuhealth/trytond.conf -d gnuhealth --all -v --email "admin@hospital.local" -p
Contraseña: Admin123!

Resultado: inicialización completada exitosamente.

Fase 6 — Conexión del Cliente GNU Health

Ejecución del cliente:
gnuhealth-client
```

# DOCUMENTACIÓN

## Cristian Ospina

### Configuración:

Profile: Default (o Local)

Host: localhost

Database: gnuhealth

Username: admin

Password: Admin123!

Resultado: conexión establecida correctamente.

### Errores principales y soluciones

trytond: command not found → PATH no configurado → Agregar ~/.local/bin al PATH

mkdir: Permiso denegado → Falta de permisos → Usar sudo mkdir

ModuleNotFoundError: psycopg2 → Falta conector PostgreSQL → pip3 install psycopg2-binary

trytond-admin sin respuesta → Parámetros incompletos → Usar —email y -p

405 Method Not Allowed → Petición inválida → El servidor responde correctamente

### Conclusiones

GNU Health requiere una correcta configuración de PostgreSQL, Tryton Server y módulos de Python.

La mayoría de errores se debieron a rutas no incluidas en el PATH o dependencias faltantes.

Una vez corregidos los permisos y configurado el servidor, el sistema funcionó correctamente en el puerto 8000.

El cliente pudo conectarse exitosamente usando el usuario admin.

### Recomendaciones finales

- Mantener PostgreSQL y Python actualizados.
- No cerrar la terminal donde corre el servidor Tryton.



Escuchemos las voces de éstos dos estudiantes en particular, quienes consiguieron éstos logros antes del tiempo estimado para hacerlo gracias a su pasión en la realización del ABP. Transcribiré sus audios a continuación:



## Daniel

Bueno, al inicio del curso lo primero que nos dieron a conocer o que nos plantearon fue que supiéramos qué era un software libre, aprendimos a instalar PureOS, después de ésto nos plantearon buscar o poder instalar y hacer un buen funcionamiento de la aplicación GNUHealth, de mi parte yo busqué mucho en internet pero no encontré como tutoriales o guías de cómo hacerlo en pureos entonces me tocó ayudarme de la IA, me ayudó bastante y todo fue prueba y error. Mas que todo hubieron dos puntos críticos, uno de los puntos críticos fue que en el momento de crear el usuario de la aplicación, el usuario administrador, cuando le ingresaba contraseña, le salía un error de 72 bits, entonces supuestamente la contraseña que muy grande, tenía espacios o signos que no podía llevar entonces la solución fue usar una librería de Python para ingresar la contraseña y que me la pudiera aceptar evadiendo ese error y otro de los puntos críticos fue con la base de datos que cuando crea la base de datos creaba el usuario y en el momento de ir a crear el servidor no me reconocía el usuario. Ésto lo pude solucionar editando la base de datos y creando directamente ahí el usuario. Fue muy difícil, tuve que investigar mucho, pero bueno, al final lo pude lograr y se aprendió bastante sobre qué es un software libre, cómo poder usarlo y bueno, nos puso dificultades que hasta ahora en la facultad de ingeniería no habíamos visto.



## Cristian

Al inicio del curso se nos planteó qué era software libre, las ramas que éste tiene y digamos el objetivo que buscábamos era la finalidad de GNUHealth, de instalarlo, de correrlo y en algún momento lograr una Federación, ésto lo hicimos en base en PureOS, el cuál nunca había escuchado, lo conocí, tuve que investigar bastante. Había usado algunos sistemas Linux y Debian, pero no ésta distribución exactamente. La instalación fue bastante complicada ya que se presentaban problemas de Hardware y de Software, los cuales la mayoría de compañeros, incluyéndome, no teníamos tanta experiencia en éste. Luego de tener el PureOS instalado, empezamos con la configuración del servidor, de las bases de datos y ésto para mí fue la parte más difícil de implementar ya que es bastante conflictivo el servidor con la base de datos, en ocasiones hay que crear usuarios y contraseñas directamente en la base de datos modificándola. El servidor genera problemas al crear la contraseña pero ésto se corrige con unas librerías. Una de las grandes ayudas fue la inteligencia artificial DeepSeek, la cuál también es de código abierto y nos permitió terminar la instalación del servidor, del PureOS inclusive y también del cliente GNUHealth. Éste fue un gran apoyo ya que sin ésto hubiéramos tenido que investigar mucho más y tal vez por temas de tiempo no hubiéramos alcanzado. Fue bastante complicada la experiencia pero gratificante cuando lo logramos debido a que es algo, digamos un área que no conocíamos tanto, pero muy interesante, más de algo tan importante como la salud, que está en todos los países y sería muy gratificante ver ésto algún día en el sistema público de salud.

**¡Gracias!**