

Laboratorio de Sistemas Operativos y Redes

# GNU Mediagoblin

Profesor: José Luis Di Biase.

Integrantes: John Braian Lora.

Gian Franco Fioriello.



**Tutorial de instalación y uso**

# Índice:

2. Índice
3. Introducción y Requisitos
4. Guía de instalación. Instalando dependencias.
6. Configuración
7. Activando videos
8. Activando PDF

# Introducción

GNU MediaGoblin (también MediaGoblin) es una plataforma web de software libre para alojar y compartir multimedia digital, con el objetivo de proporcionar una alternativa extensible, adaptable, descentralizada y libre de restricciones de derechos de autor a otros servicios de internet relativos a la publicación de contenido informático como son Flickr, deviantArt, YouTube, etc.

Versión de Ubuntu: 12.04

## Requisitos

- Python 2.6 or 2.7
- python-lxml
- git
- SQLite/PostgreSQL
- Python Imaging Library (PIL)
- virtualenv

# Guía de instalación

Esta guía fue hecha en base a la siguiente pagina: [mediagoblin.readthedocs.org/en/v0.6.1](https://mediagoblin.readthedocs.org/en/v0.6.1)

## Instalando dependencias

1.- Instalamos git:

```
# apt-get install git
```

2.- Instalamos algunas dependencias necesarias:

```
# apt-get install python python-dev python-lxml python-imaging  
python-virtualenv
```

3.- Mediagoblin usa PostgreSQL como base de datos, así que lo instalamos:

```
# apt-get install postgresql postgresql-client python-psycopg2
```

4.- Nos logueamos con el usuario postgres para crear un usuario y la B.D:

```
# su postgres
```

5.- Ahora, con el usuario postgres, creamos el usuario de la B.D. Lo llamamos por ejemplo: mediagoblin

```
$ createuser mediagoblin
```

**Primer error: No nos deja entrar como usuario. Entonces creamos un usuario como super usuario y nos manejamos con ese.**

6.- Nos hará una pregunta, respondemos que si a la primera para poder usar el usuario nuevo como super usuario

```
Shall the new role be a superuser? (y/n) y
```

7.- A continuación creamos la B.D. donde se van a guardar todos los datos de mediagoblin:

```
$ createdb -E UNICODE -O mediagoblin mediagoblin
```

8.- Y para terminar con esta parte, abandonamos el usuario postgres:

```
$ exit
```

9.- A continuación, como administrador creamos una cuenta de usuario de sistema sin privilegios:

```
# adduser --shell /bin/bash --system --group mediagoblin
```

10.- Aunque es posible usar apache, Mediagoblin está pensado para usar por defecto nginx como servidor web, así que, para no complicarnos, haremos la configuración con nginx. Creamos un

directorio para la plataforma /srv/mediagoblin.example.org:

```
# mkdir /srv/mediagoblin.example.org
```

11.- Asignamos como propietario y grupo al usuario mediagoblin:

```
# chown -hR mediagoblin:mediagoblin /srv/mediagoblin.example.org
```

12.- Nos identificamos con el usuario mediagoblin y dentro de la carpeta donde vamos a alojar el servidor, descargamos el proyecto mediante git:

```
# su mediagoblin  
$ cd /srv/mediagoblin.example.org  
$ git clone git://gitorious.org/mediagoblin/mediagoblin.git  
$ cd mediagoblin  
$ git submodule init && git submodule update
```

13.- A continuación instalamos todos los recursos python que la aplicación va a necesitar:

```
$ (virtualenv --system-site-packages . || virtualenv .) && ./bin/python  
setup.py develop
```

14.- Por último, estando dentro del directorio mediagoblin.example.org/mediagoblin, instalamos flup:

```
$ ./bin/easy_install flup
```

Los próximos pasos son para la configuración del portal mediagoblin.

# Configuración

Nos situamos dentro del directorio de Mediagoblin, concretamente dentro de `/srv/mediagoblin.example.org/mediagoblin`.

1.- Copiamos `mediagoblin.ini` a `mediagoblin_local.ini`:

```
$ cp mediagoblin.ini mediagoblin_local.ini
```

2.- Editamos `mediagoblin_local.ini`:

Descomentamos la línea:

```
# sql_engine = postgresql:///mediagoblin
```

Cambiamos el mail por uno nuestro

```
email_sender_address = "tu.e-mail@tuservidor.com"
```

Desactivamos el envío de notificaciones de email en modo debug:

```
email_debug_mode = false
```

Desactivamos el registro de usuarios porque no queremos registro automático de usuarios:

```
allow_registration = false
```

3.- Copiamos `paste.ini` a `paste_local.ini`:

```
$ cp paste.ini paste_local.ini
```

4.- Ejecutamos el siguiente comando para actualizar las bases de datos de Mediagoblin:

```
$ bin/gmg dbupdate
```

Ahora probamos que el servidor esta funcionando:

```
$ ./lazyserver.sh --server-name=broadcast
```

Una vez que tenemos el servidor levantado podemos ver que solamente nos deja subir imagenes, solo queda instalar los plugins para compartir videos y pdfs (Hay muchos más plugins para otros tipos de medios como audio y objetos en 3d, solo probamos video y pdf).

# Activando videos

1.- Siguiendo las instrucciones del manual de Mediagoblin, instalamos los paquetes de gstreamer:

```
# apt-get install python-gst0.10 \  
gstreamer0.10-plugins-base \  
gstreamer0.10-plugins-bad \  
gstreamer0.10-plugins-good \  
gstreamer0.10-plugins-ugly \  
gstreamer0.10-ffmpeg
```

2.- Añadimos `[[mediagoblin.media_types.video]]` a la sección `[plugins]` de nuestro fichero `/srv/mediagoblin.example.org/mediagoblin/mediagoblin_local.ini`

3.- Ejecutamos:

```
# su mediagoblin  
$ cd /srv/mediagoblin.example.org/mediagoblin  
$ ./bin/gmg dbupdate
```

4.- Iniciamos el servidor:

```
$ ./lazyserver.sh --server-name=broadcast
```

# Activando pdf

1.- Según dice el manual de Mediagoblin, para poder mostrar archivos pdf es necesario disponer de pdftocairo y pdfinfo. Por lo tanto, en Debian tendremos que instalar el paquete poppler-utils:

```
# apt-get install poppler-utils
```

2.- También necesitamos el paquete unoconv para convertir documentos:

```
# apt-get install unoconv
```

3.- A continuación, para instalar el módulo pdf.js ejecutamos:

```
$ cd /srv/mediagoblin.example.org/mediagoblin
```

```
$ git submodule init
```

```
$ git submodule update
```

4.- Añadimos `[[mediagoblin.media_types.pdf]]` en la sección `[plugins]` de nuestro archivo de configuración `mediagoblin_local.ini`.

5.- Ejecutamos:

```
$ cd /srv/mediagoblin.example.org/mediagoblin
```

```
$ ./bin/gmg dbupdate
```

6.- Iniciamos el servidor:

```
$ ./lazyserver.sh --server-name=broadcast
```