

cleanet

# CLASSADMIN

04/06/2022

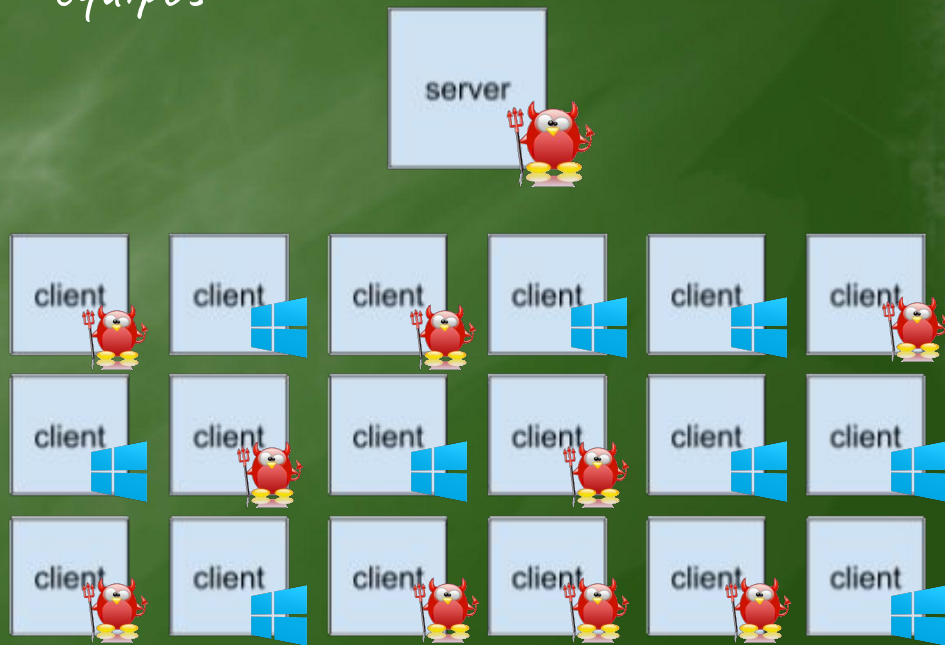
# ÍNDICE

1. Introducción
  - a. Justificación del proyecto
  - b. Objetivos
2. Desarrollo
  - a. Definición conceptual del problema
  - b. Análisis del sistema
  - c. Diseño
  - d. Implementación
3. Conclusiones
4. Líneas de investigación futuras
5. Bibliografía
6. GNU Free Document License

# INTRODUCCIÓN



*Dashboard web de control remoto de equipos*



# INTRODUCCIÓN - Justificación del proyecto

## Metas

Rendimiento



PYTHON SOCKET

entender funcionamiento



Crear servicios en Windows

django



framework



## módulos

ISO / ASO

Configuraciones del sistema, así como scripting para listar programas, bloquearlos, etc.

PAX

sockets, direcciones IP, puertos, arquitectura cliente-servidor

GDB / ADB

administración de base de datos, así como tablas, etc.

LLM / IAW

Despliegue de la aplicación Django así como Django Rest Framework

SXI

servicios MariaDB, Apache2, y despliegue en Microsoft Azure

SAD

Certificados SSL en servicios sockets, así como en Django

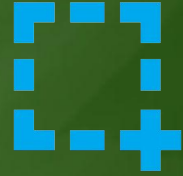
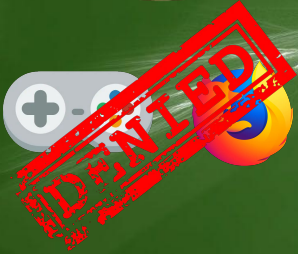




# INTRODUCCIÓN - Objetivos

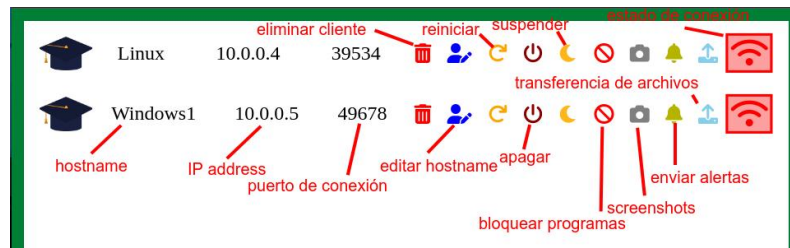
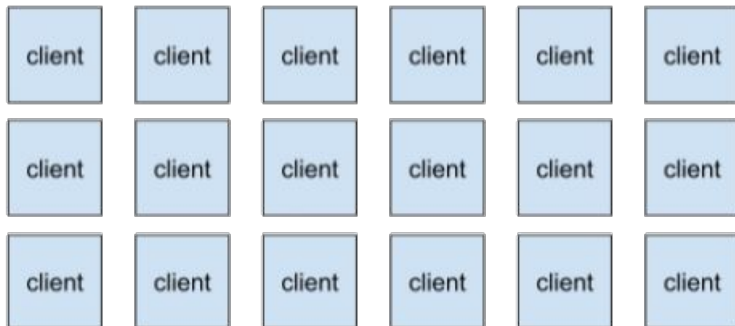
- ✓ Control remoto de un equipo desde una webapp
- ✓ Bloquear aplicaciones remotamente
- ✓ Transferencia de archivos
- ✓ Practicar mucho más en el lenguaje Python 3
- ✓ Profundizar en el módulo socket de Python 3
- ✓ Licenciar proyecto bajo GPLv3 así como GFDL
- ✓ Aprender a manejar Microsoft Azure
- ✓ Entender mejor el funcionamiento de Windows
- ✓ Mejora y conceptos de Powershell
- ✓ Automatización de instalación y desinstalación de un programa.
- ✓ Crear servicios nativos de Windows con un programa de terceros
- ✓ Superación personal

# DESARROLLO - Definición conceptual del problema



# DESARROLLO - Análisis del sistema

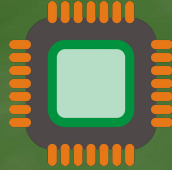
	Linux	10.0.0.4	39534										
	Windows1	10.0.0.5	49678										



El sistema ClassAdmin trabaja con las siguientes tecnologías:



- ✓ Versatil, legible
- ✓ multiplataforma
- ✓ compilar código (híbrido)



- ✓ Multiprocesos
- ✓ Hilos



- ✓ Framework Python
- ✓ Desarrollo Web
- ✓ MVC



- ✓ Scripting para listar los programas y bloquearlos, así como apagarlos, etc. En Windows



PYTHON SOCKET

- ✓ Conexiones TCP/IP entre nodos.

django



framework

- ✓ Parte de Django Framework para hacer APIs



- ✓ Scripting para listar los programas y bloquearlos, así como apagarlos, etc. En GNU/Linux



*El sistema ClassAdmin, también utiliza programas terceros:*



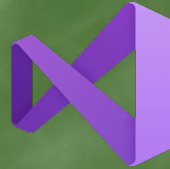
*Creación de servicios para Windows.*



*Creación de Base de datos tanto en Linux como Windows (XAMPP)*



*Servicio web en Linux que permite el despliegue de aplicaciones web. En nuestro caso Django.*



*Utilización de Microsoft C++ Build Tools en Windows para el funcionamiento de los servicios (módulo psutil)*



*Servicio web en Windows que permite el despliegue de aplicaciones web. En nuestro caso Django. Incluye MariaDB*

La aplicación Django utiliza una API creada con Django Rest Framework.

● POST ● PUT ● PATCH  
● GET ● DELETE



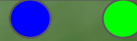
/status



/status/<name>



/clients



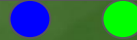
/clients/<id>



/clients/<address>/<hostname>



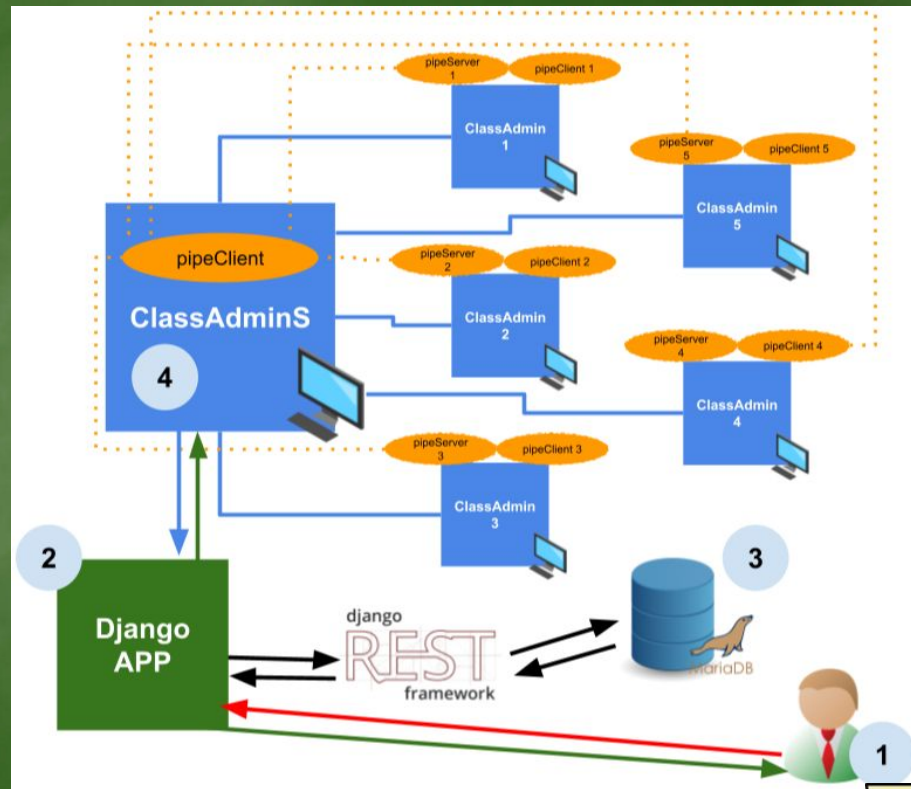
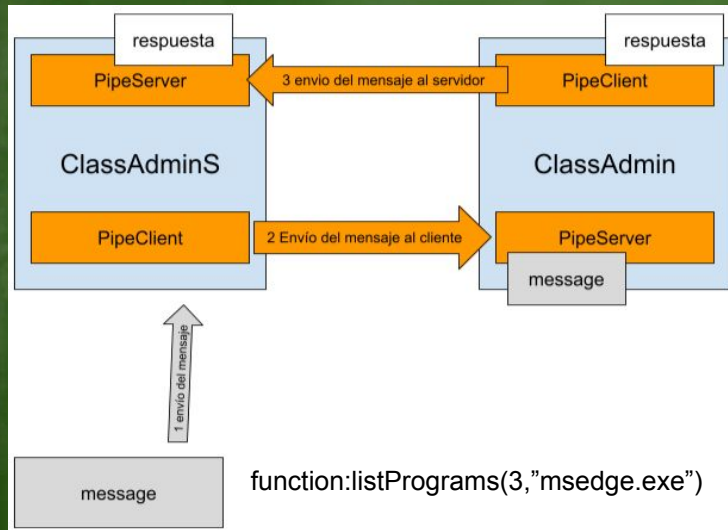
/servers

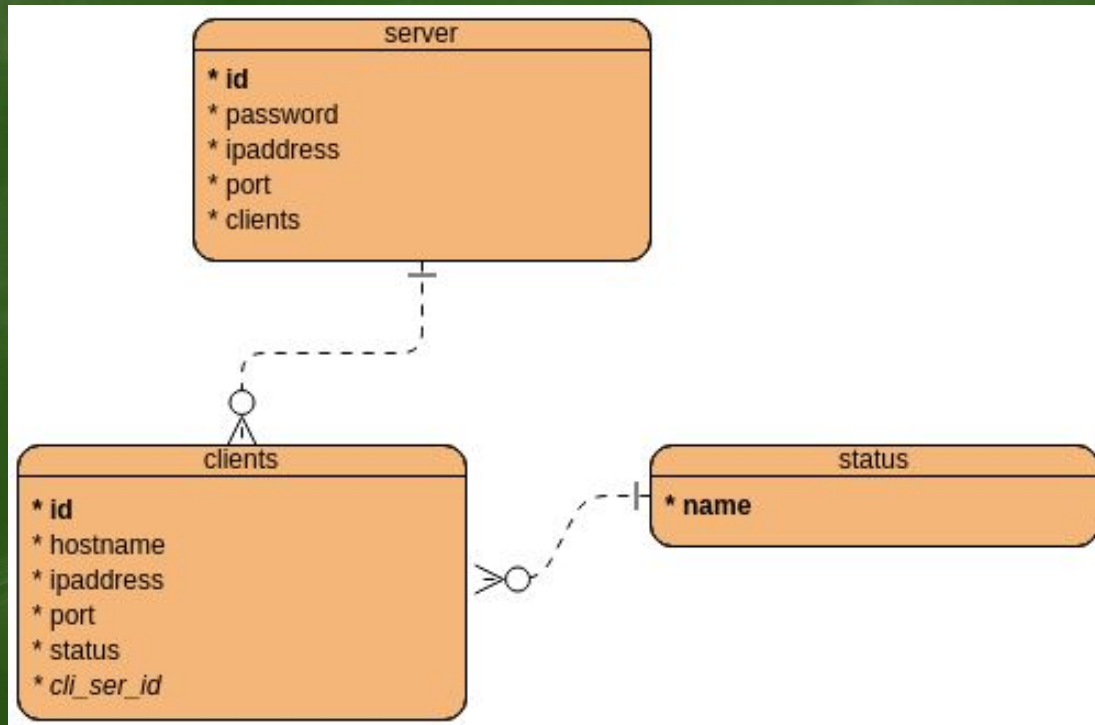


/servers/<id>



# DISEÑO

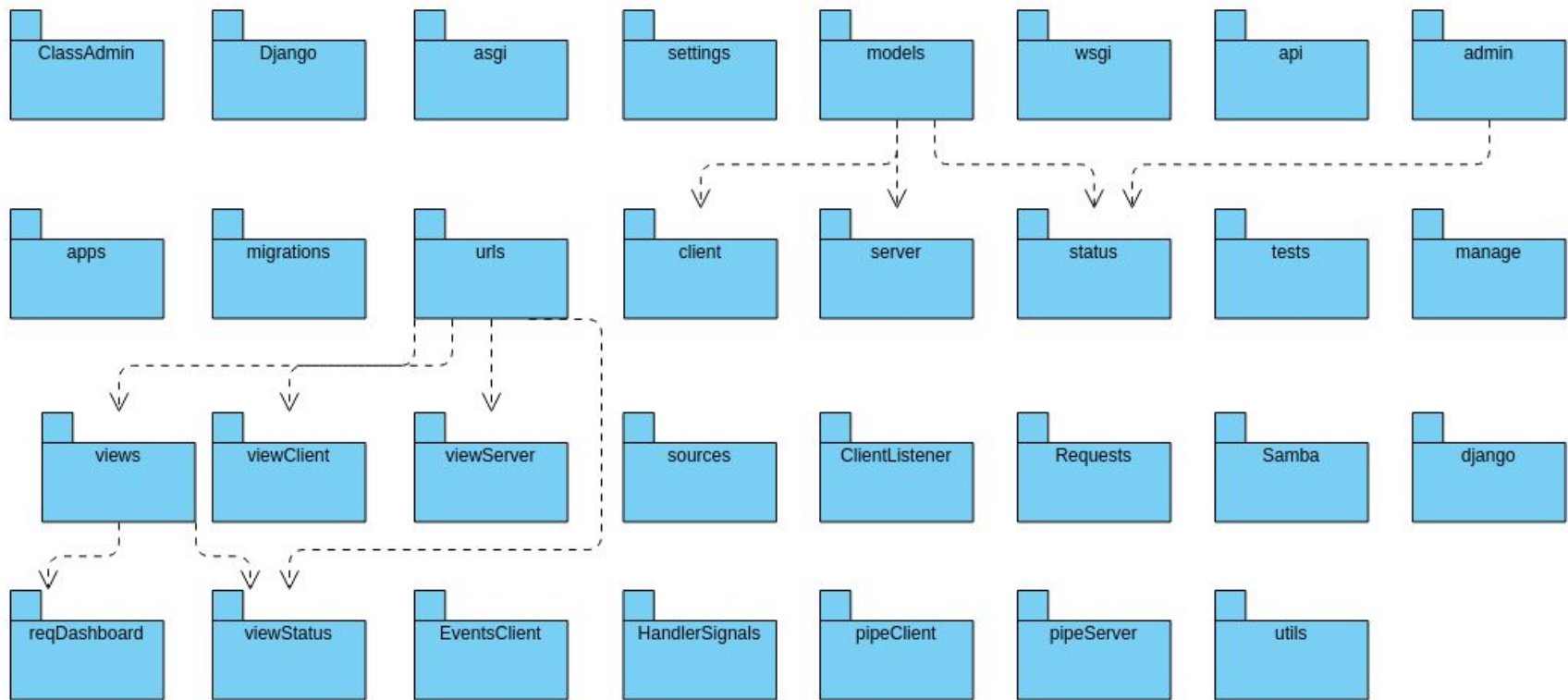




▼ **ClassAdmin** /etc/ClassAdmin

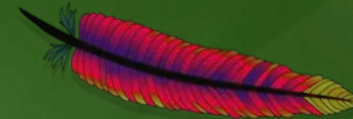
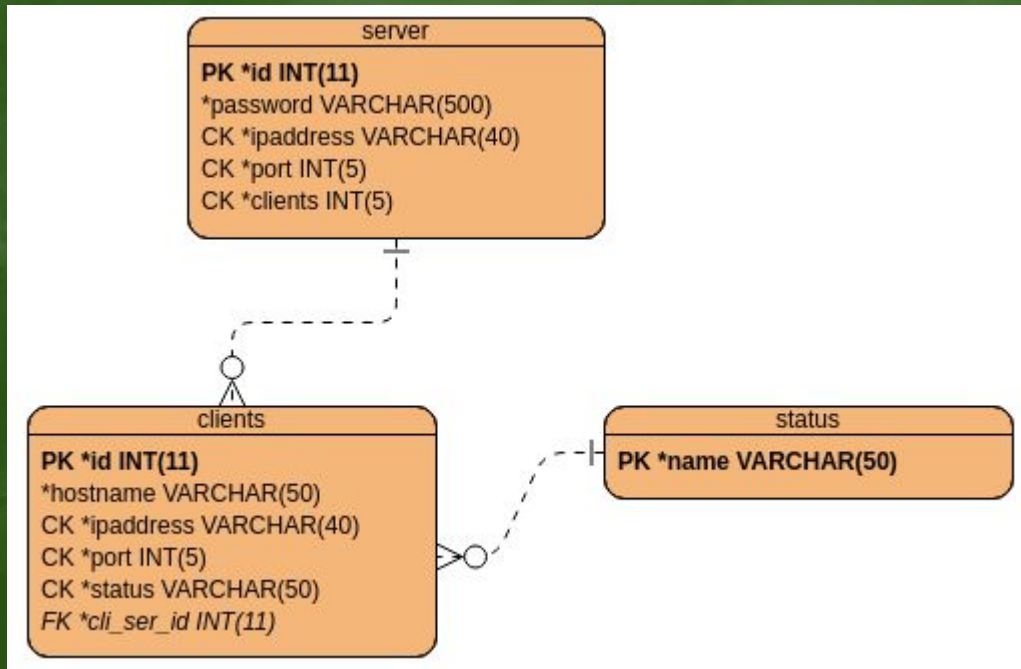
- > `apache2.conf.d`
- > `commands`
- > `Daemon`
- > `Django`
- > `services`
- > `sources`
- > `ssl`
- > `transfers`
- `.gitignore`
- `__init__.py`
- `COPYING`
- `init.sql`
- `INSTALL.ps1`
- `INSTALL.sh`
- `LICENSE`
- `README.md`
- `requirements.txt`
- `UNINSTALL`
- `UNINSTALL.ps1`
- External Libraries
- Scratches and Consoles







# IMPLEMENTACIÓN



APACHE  
HTTP SERVER

**MOD\_WSGI**  
**REWRITE**



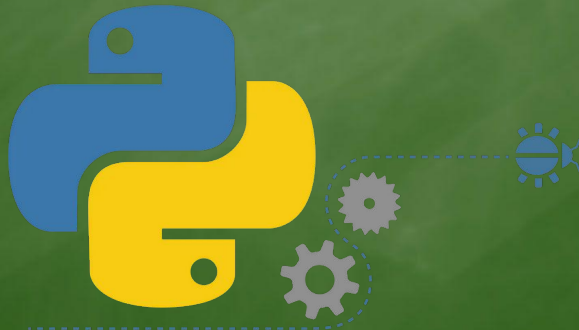
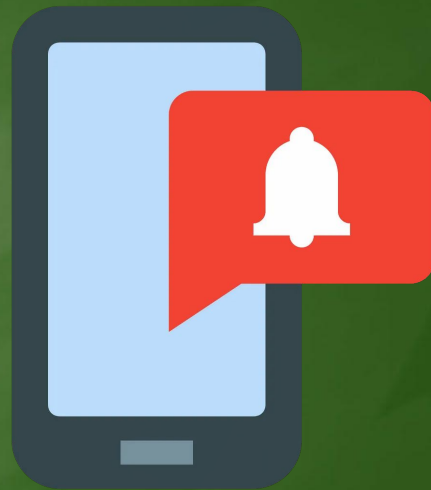
bind-address = 0.0.0.0  
max\_allowed\_packet = 1G  
max\_connections = 1000

# CONCLUSIONES

- ✓ Entender y mejorar los sockets en Python 3
- ✓ Instalar servicios nativos de Windows
- ✓ Profundizar más en Django
- ✓ Aprender a desplegar una API con DRF
- ✓ Implementación multiplataforma (Windows / Linux)
- ✓ Implementación de licencias GPLv3 y GFDL



# LÍNEAS DE INVESTIGACIÓN FUTURAS



# BIBLIOGRAFÍA



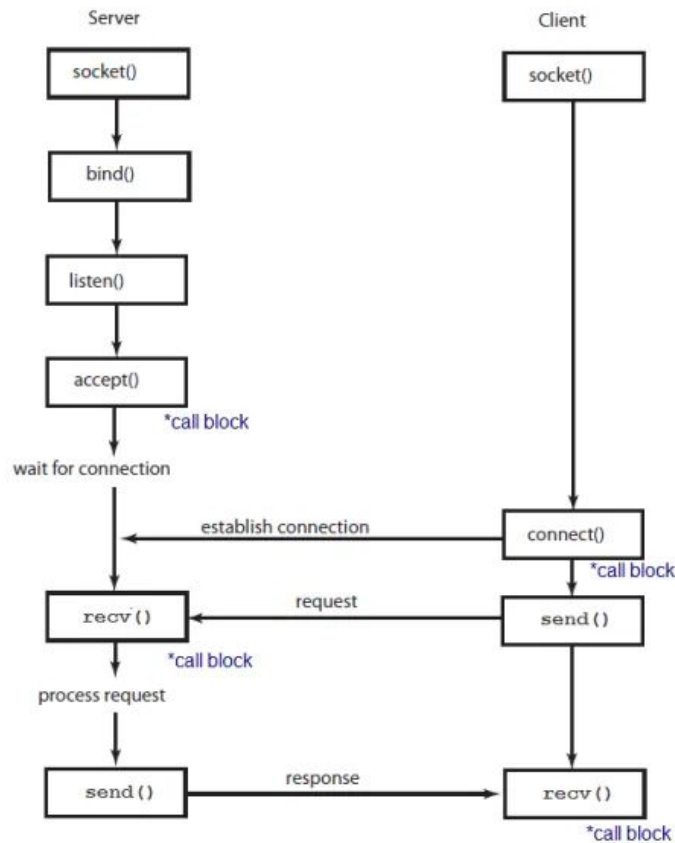
Django documentation Django REST framework Python Docs Requests library  
Command line -NSSM Usage - NSSM PowerShell Module Browser  
socket – Low-level networking interface – Python 3.10.4 documentation

Cómo utilizar las licencias de GNU para su propio software

Consejos para la utilización de la Licencia de Documentación Libre de GNU -  
Proyecto GNU - Free Software Foundation



REST API con Python, Django y MySQL (GET, POST, PUT, DELETE) ✓



[Ver perfil de Drakaris - Foro el hacker](#)

[Stopping python using ctrl+c - Stack Overflow](#)

[Signal handling: Catch Ctrl-C in Python](#)

[NSSM - the Non-Sucking Service Manager](#)

[Apache Friends](#)

[Microsoft C++ Build Tools - Visual Studio](#)



# GNU FREE DOCUMENT LICENSE



FREE SOFTWARE  
FOUNDATION

*Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, Versión 1.3 o cualquier versión posterior publicada por la Free Software Foundation; sin secciones invariantes, sin textos de portada y sin textos de contraportada. Se incluye una copia de la licencia en el archivo GFDL.txt. Se debe de mantener la licencia original, así como nombrar al autor original quien creó el software.*





??????



*Muchas gracias por vuestra atención!!*