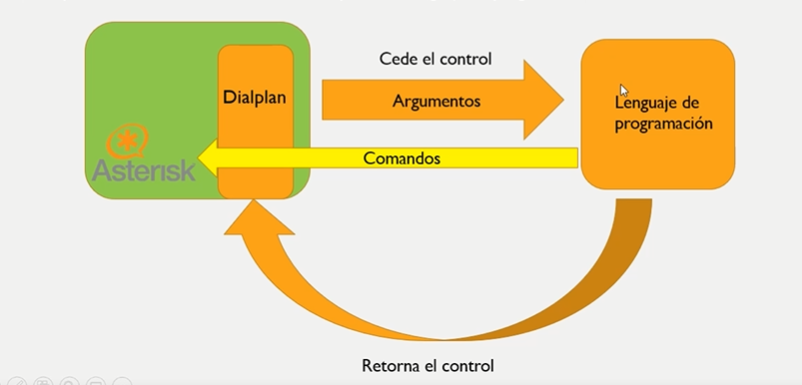
# Descripción de AGI

El AGI es una API similar al AMI, a diferencia del AMI que era un programa externo que enviaba ordenes o recibía información mediante un determinado socket, en este caso el AGI nos permite pasar el control de la llamada en el dialplan a un lenguaje de programación tradicional. El dialplan le cede el control al programa, luego el programa cuando se está ejecutando podría enviar algunos comandos al Asterisk y cuando el programa llega al final le devuelve el control al dialplan, depende del dialplan que es lo que va a hacer.



La parte del control seria de la sig. manera:

* Entramos en el “extensions.conf” y entramos en él.
* La extensión pudiera ser de la sig. manera:

exten = 400,n,AGI,(archivo,arg1,arg2,…)

* En la parte del archivo ponemos su nombre o la ruta completa.
* Los argumentos son opcionales si deseamos pasarle información adicional.
* El archivo por defecto esta ubicado en el directorio “/var/lib/asterisk/agi-bin/”.
* A los archivos hay que darles permiso con el comando “chmod 777 archivo” para que el asterisk los pueda ejecutar,
* Hay argumentos que se pasan por default, cuya información seria:
  + agi\_variable: valor
  + agi\_request: valor
  + agi\_channel: SIP/200
  + agi\_callerid:
  + agi\_context:
  + agi\_extension:
  + agi\_priority:
* En el programa y/o script, esto mandara el programa a Asterisk
  + ASNWER: contestar la llamada.
  + HANGUP: cortar la llamada.
  + SAY ALPHA: decir las letras de un texto.
  + SAY DIGITS: decir los dígitos de un número.
  + VERBOSE: mandar algo al CLI de Asterisk.
  + WAIT FOR DIGIT: esperar del llamante que presione un digito.
  + SAY DATE
  + SAY TIME
  + SET CALLERID
  + SET CONTEXT
  + SET EXTENSION
  + SET PRIORITY
  + SET VARIABLE

Ya sea cualquier lenguaje de programación, hay tres variables involucradas.

* STDIN: es como recibimos la información del Asterisk.
* STDOUT: es como envía algo hacia el Asterisk.
* STDERR: es para visualizar los errores.

En Python solamente utilizaremos:

* import sys: es la función que nos va permitir acceder a las variables.
* sys.stdin.readline().stripe(): servirá para el STDIN para leer la información que ha venido del Asterisk.
* sys.stdout.write(comando): para enviar un comando o una información del programa hacia el Asterisk
* sys.stdout.flush(): limpiar el buffer, es necesario después de un sttdout.
* sys.stderr.write(data):
* sys.stderrt.flush():limpiar el buffer, es necesario después de un sttderr.

# Ejemplo de envió comandos Python – Asterisk

Vamos a hacer un ejemplo donde desde el Asterisk al dialplan cedamos el control a un programa que va a estar en Python y ese programa a su vez puede enviar algunos comandos de ejemplo que vamos a realzar.

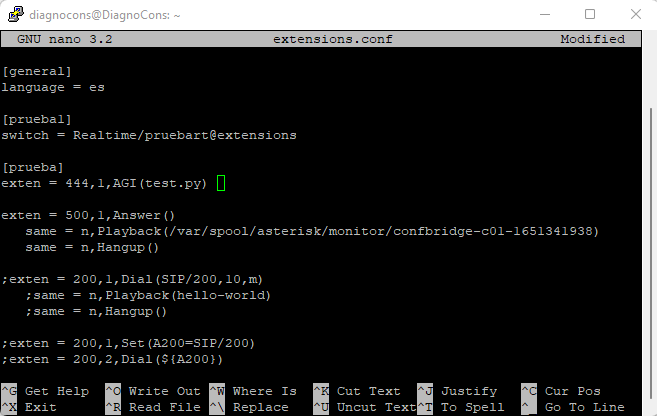
Empezaremos con la parte de como le cedemos el control en el dialplan.

Lo primero será entrar en el archivo “extensions.conf” usando el sig. comando:

nano extensions.conf

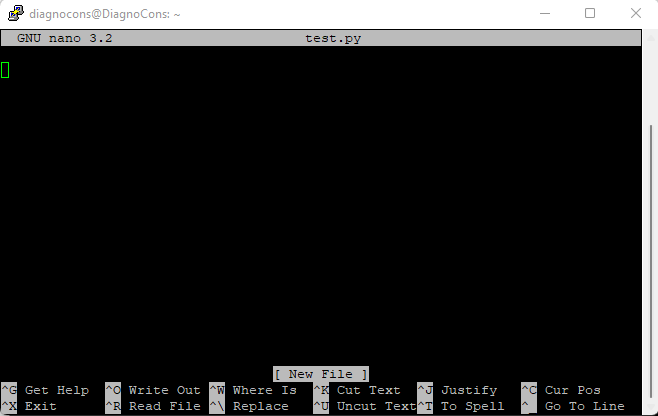
Después escogemos una extensión para poder mandarlo y colocamos lo sig.:

exten = 444,1,AGI(test.py) (“py” es la extensión de Python)



Luego guardamos y vamos a la ruta del Python que es “/var/lib/asterisk/agi-bin/” y aquí vamos a crear el archivo “test.py” usando el sig. comando:

nano test.py



Lo primero será indicarle en que ruta se encuentra el Python y le escribimos lo sig.:

#!/usr/bin/python3



Luego importamos la librería “sys”.

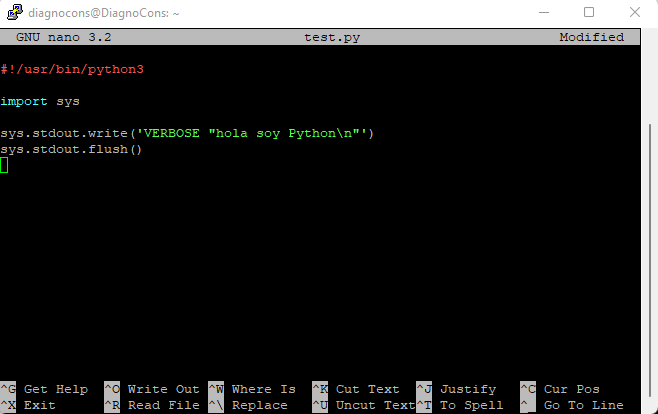
import sys



Después vamos a enviar un comando para ver algo en CLI que seria “VERBOSE” para ello escribimos lo sig.:

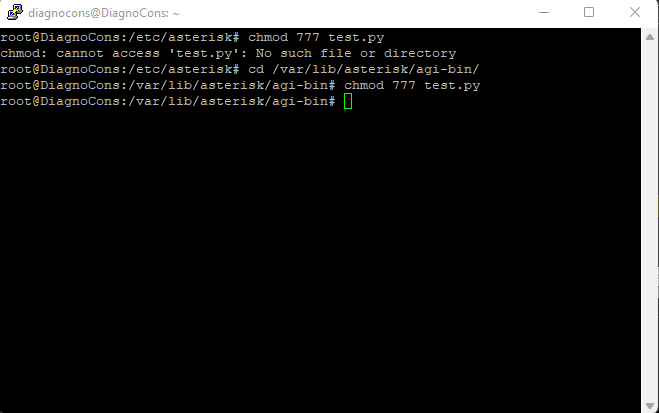
sys.stdout.write('VERBOSE "hola soy Python\n"')

sys.stdout.flush()

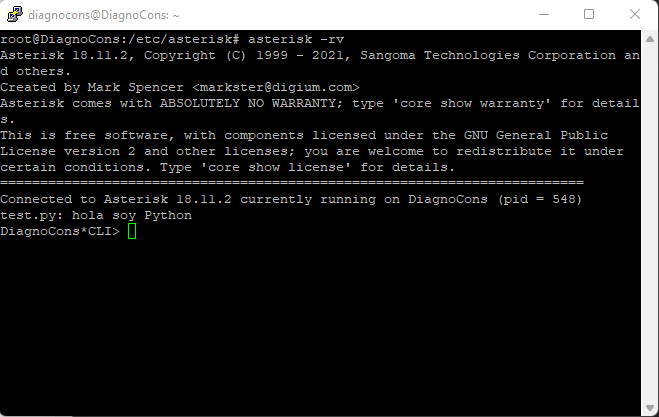


Luego le damos los permisos al programa usando el sig. comando:

chmod 777 test.py



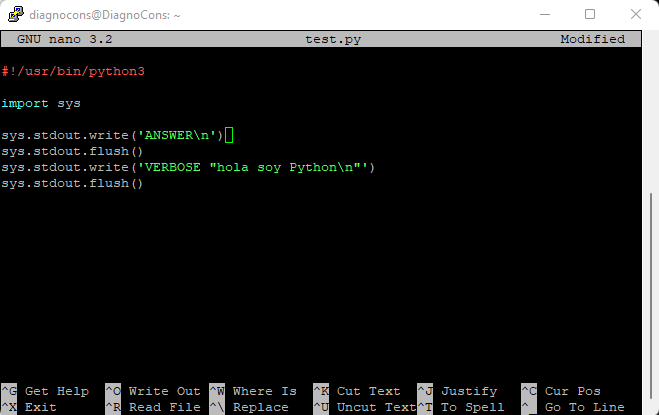
Lo que estamos realizando es simplemente que cuando le ceda el control que le mande algo al CLI. Luego vamos a la consola Asterisk le hacemos un “dialplan reload” y vemos los resultados.



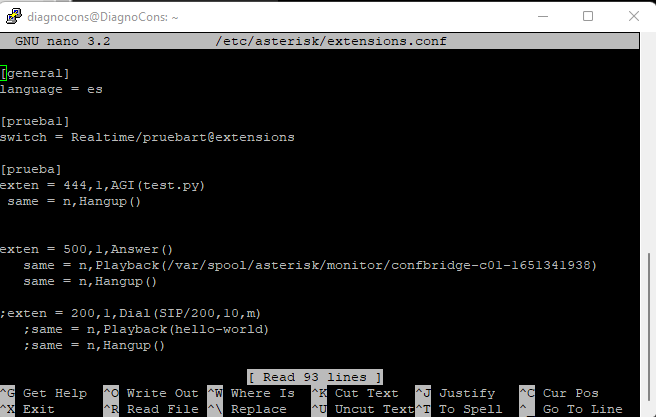
Como vemos funciona, ahora lo que haremos será que antes de mandar la información vamos a contestar la llamada, para lo cual en el “test.py” escribimos lo sig.:

sys.stdout.write('ANSWER \n')

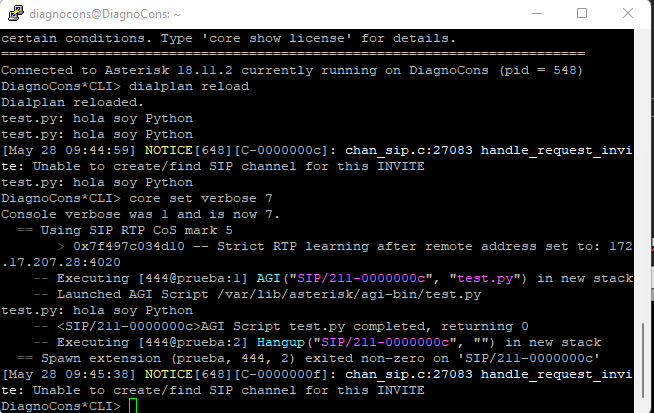
sys.stdout.flush()



Se puede hacer un mix, es decir que el answer este en el python y el hangup en el dialplan.



Luego vamos a la consola de Asterisk, realizamos un “dialplan relaod” y probamos los resultados.



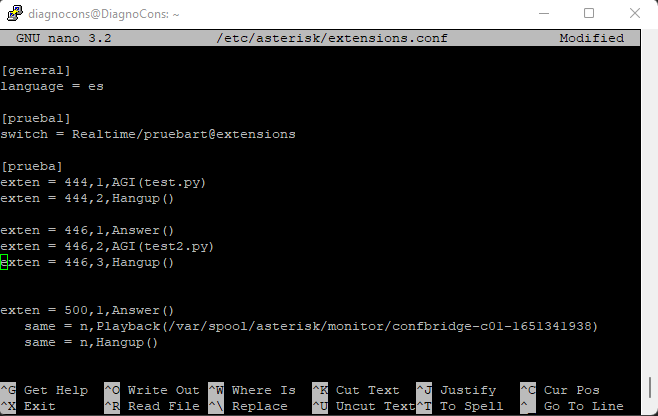
# Ejemplo de lectura de variables enviadas del Asterisk al Python

En esta ocasión vamos a leer la información que Asterisk envía al comienzo por defecto. Lo primero será hacer unas variaciones al “extensions.conf”, le creamos una nueva extensión, escribiendo lo sig.:

exten = 446,1,Answer()

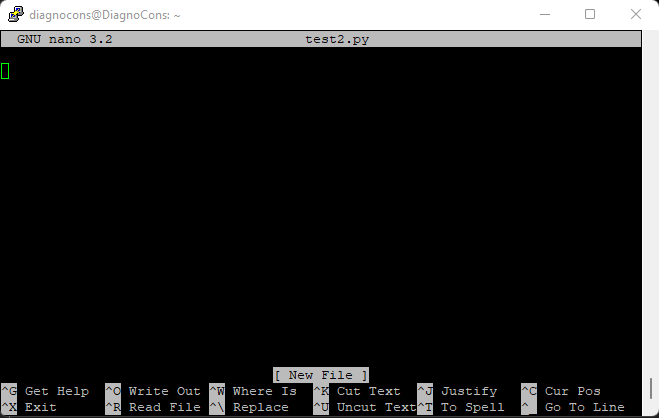
exten = 446,2,AGI(test2.py)

exten = 446,3,Hangup()



Luego guardamos y vamos a la ruta del Python que es “/var/lib/asterisk/agi-bin/” y aquí vamos a crear el archivo “test.py” usando el sig. comando:

nano test.py



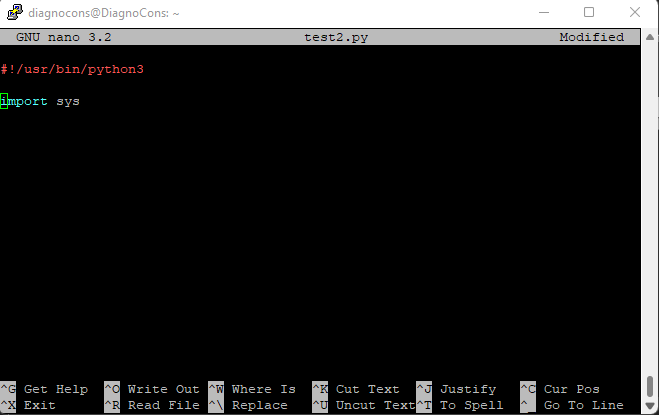
Lo primero será indicarle en que ruta se encuentra el Python y le escribimos lo sig.:

#!/usr/bin/python3



Luego importamos la librería “sys”.

import sys



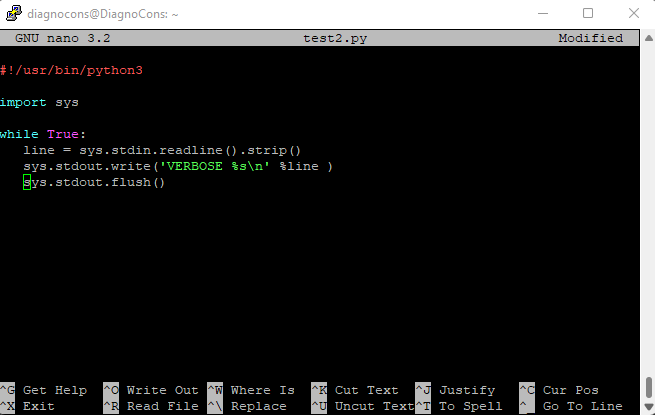
Al inicio Asterisk le envía una serie de variable con su respectivo, para poder leer esos valores escribiremos lo sig.:

while True:

line = sys.stdin.readline().strip()

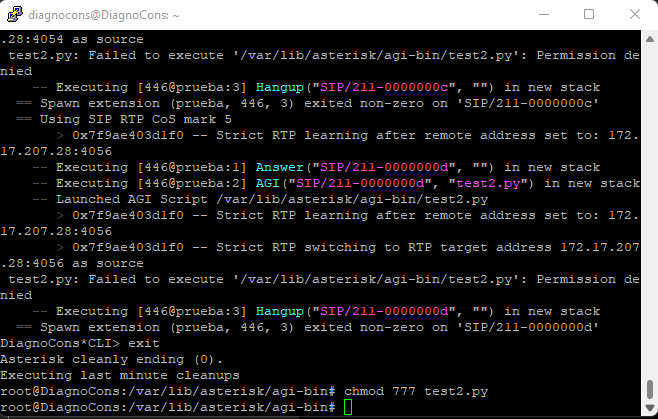
sys.stdout.write('VERBOSE %s\n' %line )

sys.stdout.flush()

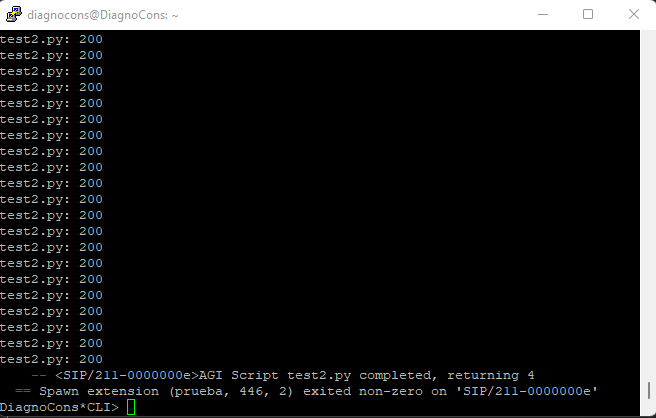


Luego le damos los permisos al programa usando el sig. comando:

chmod 777 test2.py



Después vamos a la consola del Asterisk, hacemos “dialplan reload” y vemos los resultados.



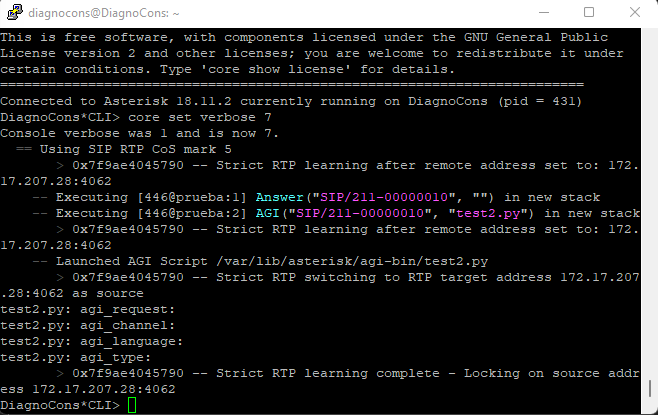
Nota: salieron los datos, pero después de eso solo manda el número 200.

Para arreglar esto vamos haciendo que solo muestre la información que empiece con “agi”, para ello le agregamos lo sig.:



Nota: esto nos quiere decir que si los primeros cuatro dígitos es igual a “agi\_” hay recién imprimimos.

Después vamos a la consola de Asterisk y vemos los resultados.



Como vemos muestra las variables, pero no muestra los valores, para ello cambiaremos lo sig.:

while True:

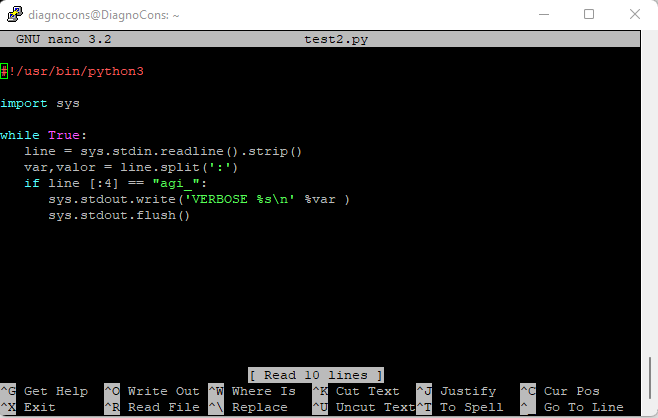
line = sys.stdin.readline().strip()

var,valor = line.split(':')

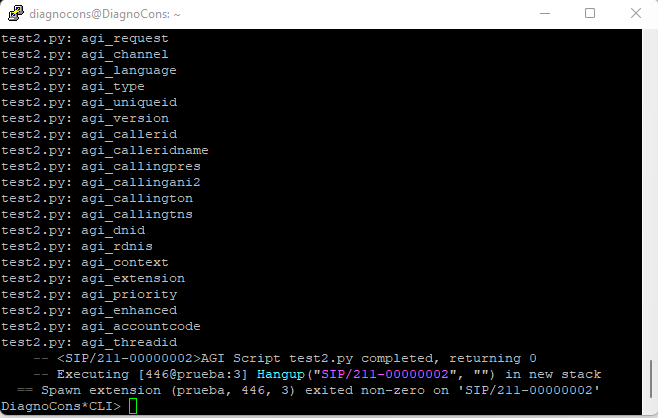
if line [:4] == "agi\_":

sys.stdout.write('VERBOSE %s\n' %var )

sys.stdout.flush()

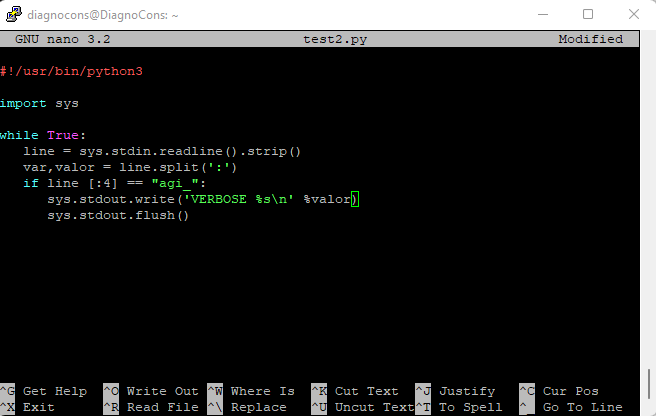


Después vamos a la consola de Asterisk y vemos los resultados.

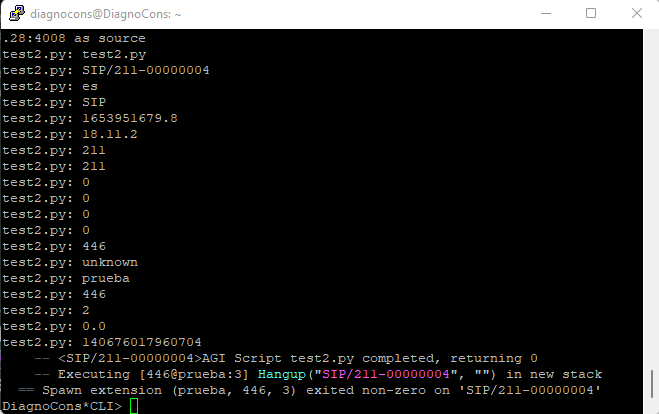


Nota: imprimió la lista completa de variables, pero ahora corta la llamada.

Como vimos si imprime las variables pero falta el valor, para obtener el valor en la línea que dice “sys.stdout.write('VERBOSE %s\n' %var)” cambiaremos el “%var” por “%valor”.



Después vamos a la consola de Asterisk y vemos los resultados.



Ahora si como vimos nos muestra el valor de las variables como lo pueden ser la terminal, extensión, idioma etc.

# Ejemplo de lectura de dígitos en el Python enviado por el Asterisk

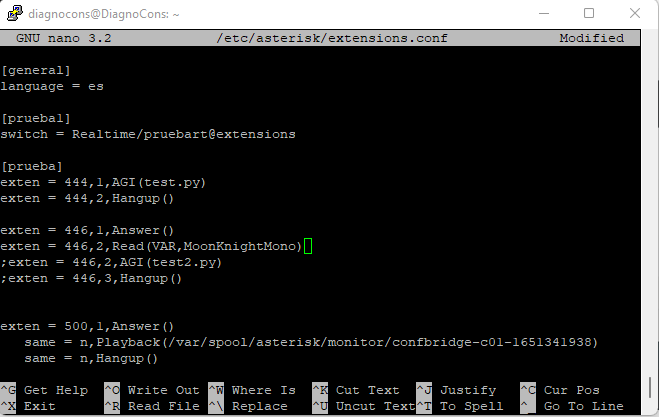
Vamos a hacer una aplicación donde el programa nos reproduce un audio, pero para reproducir ese audio nos pide que presionemos algunos dígitos, es decir, el valor que presionemos se lo vamos a mandar al AGI y el AGI lo va a imprimir en el verbose y luego el programa me va a pedir con el mismo audio que vuelva a digitar otros dígitos, una vez que los presiona, va al AGI y el AGI lee ese valor, se lo devuelve al Asterisk en el verbose y así sucesivamente hasta que corte la llamada.

Lo primero que haremos será abrir el “extensions.conf” con el sig. comando (no cambiamos de directorio):

nano /etc/asterisk/extensions.conf

Después comentamos el “test2.apy” y el “Hangup” y escribimos lo sig.:

exten = 446,2,Read(VAR,MoonKnightMono)



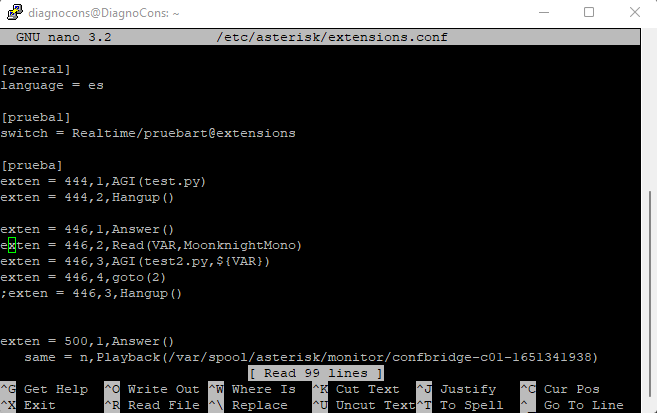
Luego probamos, sonara el audio, digitamos los dígitos y la variable “VAR” guardara el valor. Después de eso el “exten” que tiene el AGI lo ponemos como prioridad 3 y a su vez le pasamos el valor de la variable (usar los sig.: ${}), de modo que nos quede de la sig. manera:

exten = 446,1,Answer()

exten = 446,2,Read(VAR,MoonknightMono)

exten = 446,3,AGI(test2.py,${VAR})

exten = 446,4,goto(2) (Sirve para enviar a alguna parte del dialplan, el numero 2 lo hará a la prioridad 2)



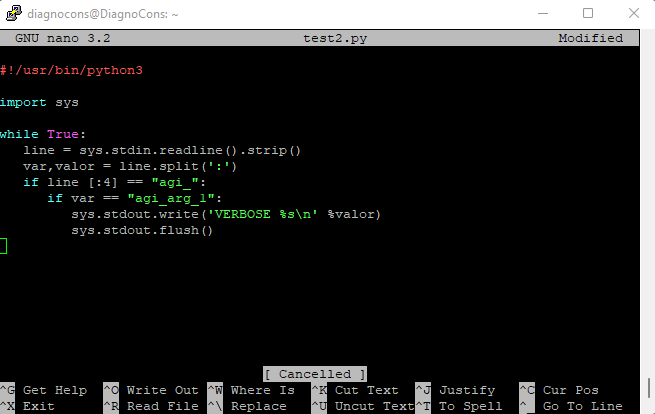
Aquí lo que sucede es que responde, pide la variable, le manda el valor de la variable al AGI, regresa y nuevamente salta a la prioridad 2. Luego en el “test2.py” le agregamos lo sig.:

if line [:4] == "agi\_":

if var == "agi\_arg\_1":

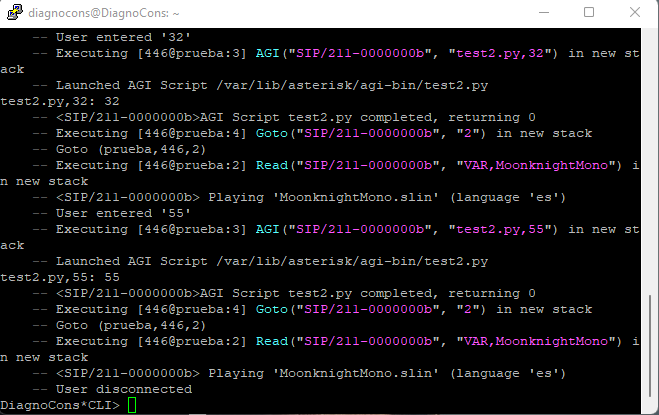
sys.stdout.write('VERBOSE %s\n' %valor)

sys.stdout.flush()



Nota: lo que hará es que en el verbose solo me imprima la variable “agi\_arg\_1”.

Después hacemos un “dialplan reload” y probamos.



Nota: como vemos generamos un bucle en el que siempre mostrara en el verbose los dígitos que presionemos.