

The background features a series of concentric circles in light gray, some solid and some dashed, creating a ripple effect. A large, solid red speech bubble is centered on the page, pointing downwards.

PROCESAMIENTO DE VOZ

Objetivos

- Crear aplicación real con botones para control de raspberry
- Control por voz de Raspberry

RPI texto a voz

Existe múltiples módulos que nos permiten realizar esta función, para este caso vamos a utilizar Pico Text to Speech-Google Android TTS engine. El cual tiene muy Buena calidad y no necesita acceder a fuentes externas para cumplir su proposito

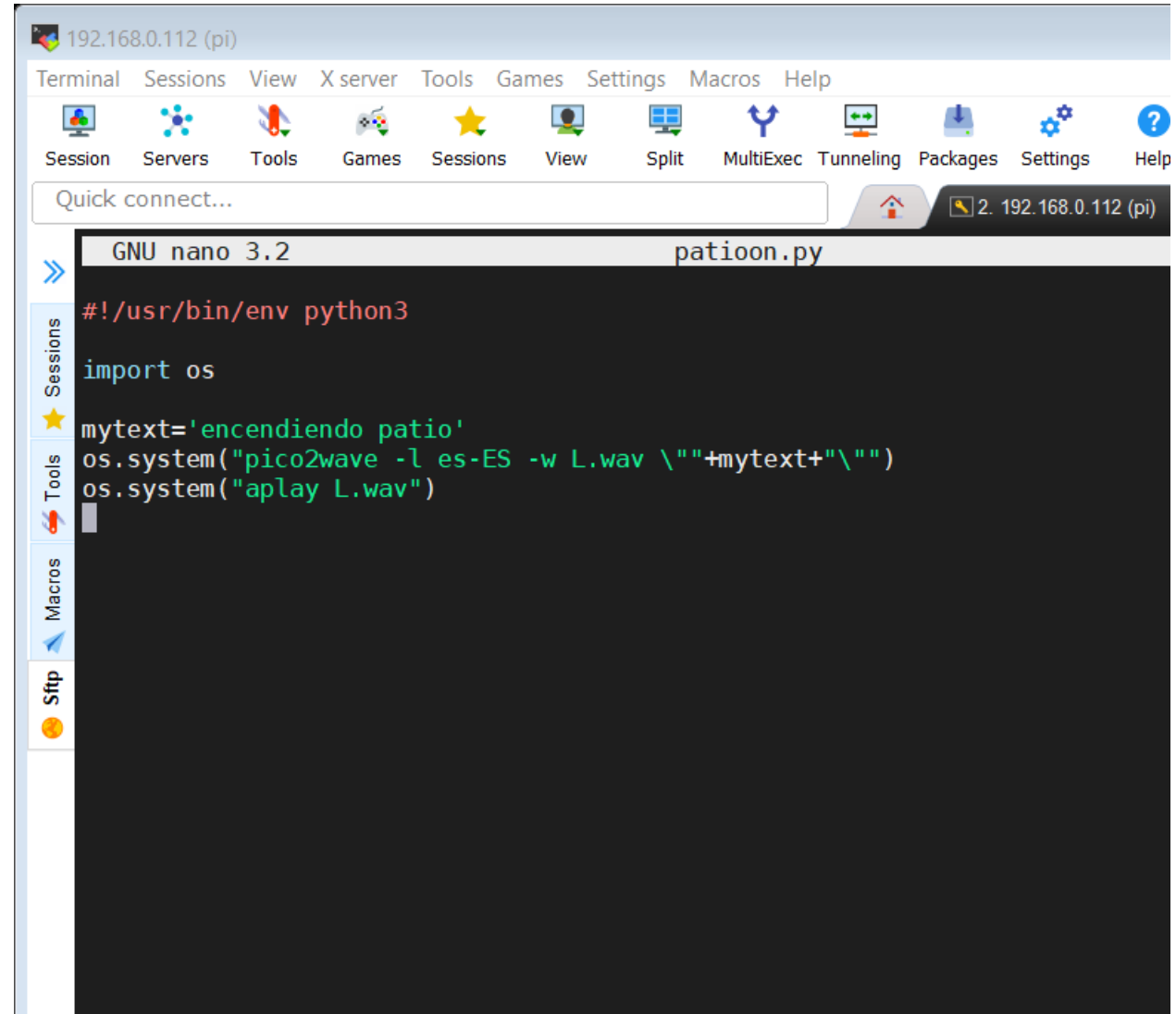
[https://elinux.org/RPi Text to Speech \(Speech Synthesis\)](https://elinux.org/RPi_Text_to_Speech_(Speech_Synthesis))

The background of the slide features several thin, curved lines in a light gray color, some solid and some dashed, creating a modern, abstract design.

Instalación de módulos de voz PICO Text

```
$ sudo apt-get install libttspico-utils
```

Script encenderv.py



192.168.0.112 (pi)

Terminal Sessions View X server Tools Games Settings Macros Help

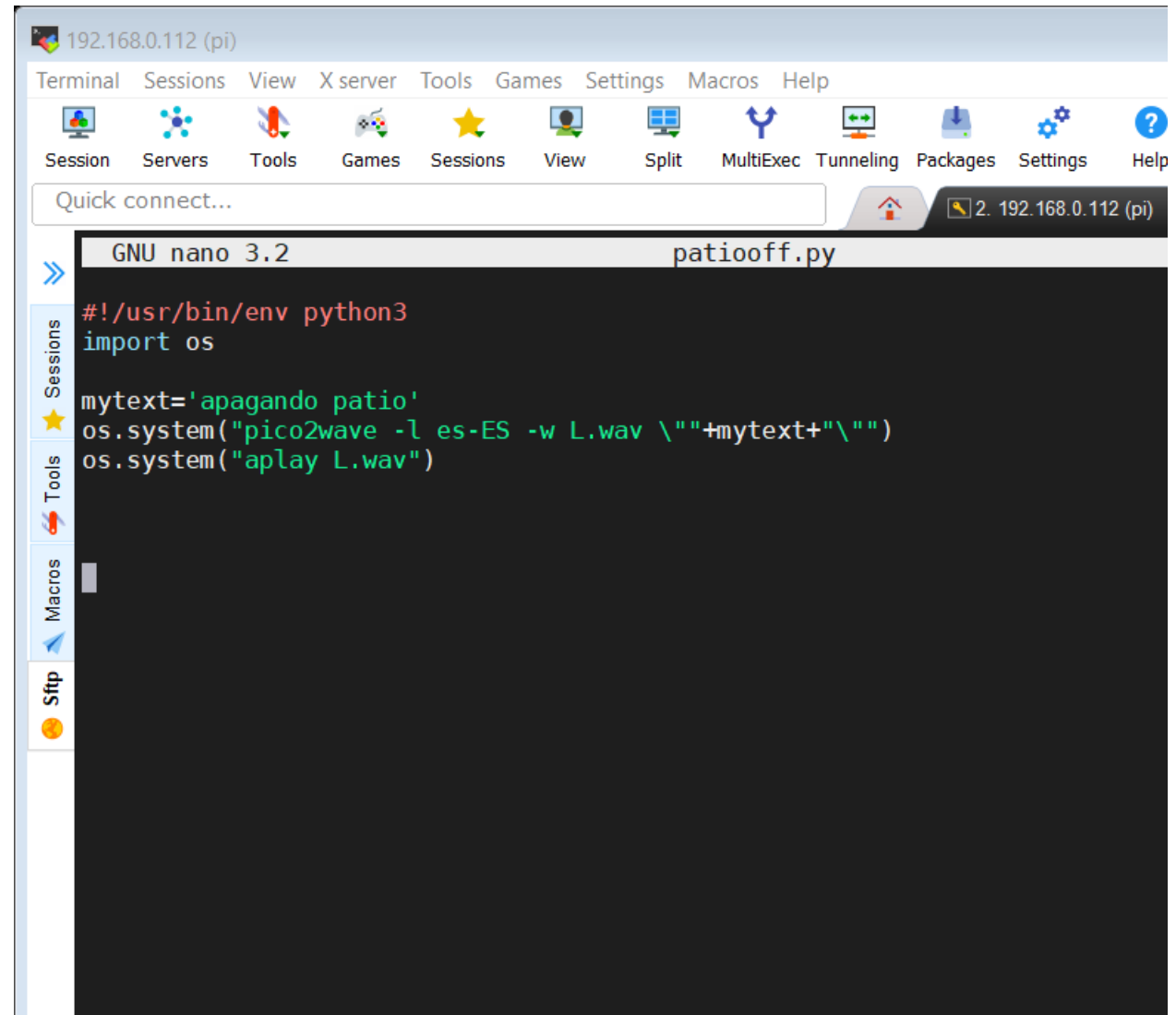
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help

Quick connect...

GNU nano 3.2 patioon.py

```
#!/usr/bin/env python3
import os
mytext='encendiendo patio'
os.system("pico2wave -l es-ES -w L.wav \""+mytext+"\"")
os.system("aplay L.wav")
```

Script
apagarv.py



```
192.168.0.112 (pi)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
GNU nano 3.2 patiooff.py
#!/usr/bin/env python3
import os

mytext='apagando patio'
os.system("pico2wave -l es-ES -w L.wav \""+mytext+"\"")
os.system("aplay L.wav")
```

RPI voz a texto



www.bujarra.com

En nuestra Raspberry debemos tener instalado Apache y le pondremos una web que permitirá que el navegador que la visite, conecte su micrófono y le hable. La web tratará lo que escuche ejecutando cualquier comando que configuremos previamente. A esta web nos podremos conectar desde un PC/Portatil/móvil/tableta y con el Chrome, nos solicitará acceso al micrófono y nos estará escuchando en todo momento.



Bien, todo esto es posible gracias a una librería de Javascript que tendrá la web (html) llamada **Annyang**, añadiendo la posibilidad de usar reconocimiento de voz a los usuarios que se conecten. Reenviará lo que escuche a una pequeña web de php que ejecutará los comandos.



- El servidor web solicitará acceso al micrófono, si hacemos que el sitio sea seguro (HTTPS) e instalamos un certificado autofirmado, el navegador confiará en el sitio y ya no pedirá acceso al microfono más que la primera vez.
- Los pasos necesarios dar seguridad web, crearemos un directorio para guardar ahí los certificados, crearemos el certificado y la clave privada, habilitamos los módulos en apache, creamos enlace a la configuración de Apache y la editaremos para indicar las rutas correctas de los certificados:

Configuraciones necesarias

Cambiamos el usuario que ejecutará apache2:

```
$ sudo apt-get install apache2 libapache2-mod-wsgi
```

```
$ sudo apt-get install php
```

```
$ sudo systemctl restart apache2
```

```
$ sudo nano /etc/apache2/envvars
```

```
export APACHE_RUN_USER=pi
```

```
export APACHE_RUN_GROUP=pi
```

Crearemos un directorio para guardar los certificados

```
$ sudo mkdir /etc/apache2/ssl
```

3. Crearemos el certificado y la clave privada con encriptación RSA

```
$ sudo openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -out  
/etc/apache2/ssl/server.crt -keyout /etc/apache2/ssl/server.key
```

```
pi@raspberrypi:/etc/apache2 $ sudo openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -out /etc/apache2/ssl/server.
out /etc/apache2/ssl/server.key
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to '/etc/apache2/ssl/server.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:EC
State or Province Name (full name) [Some-State]:pichincha
Locality Name (eg, city) []:Quito
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Espe
Organizational Unit Name (eg, section) []:espe
Common Name (e.g. server FQDN or YOUR name) []:Lj
Email Address []:luipmejia@gmail.com
```

Habilitamos los módulos en Apache
Creamos un enlace a la configuración
de Apache
Editaremos las rutas correctas de los
certificados:

```
$ sudo systemctl restart apache2
```

```
$ sudo a2enmod ssl
```

```
$ sudo ln -s /etc/apache2/sites-available/default-ssl.conf  
/etc/apache2/sites-enabled/000-default-ssl.conf
```

```
$ sudo nano /etc/apache2/sites-enabled/000-default-ssl.conf
```

```
SSLCertificateFile /etc/apache2/ssl/server.crt
```

```
SSLCertificateKeyFile /etc/apache2/ssl/server.key
```

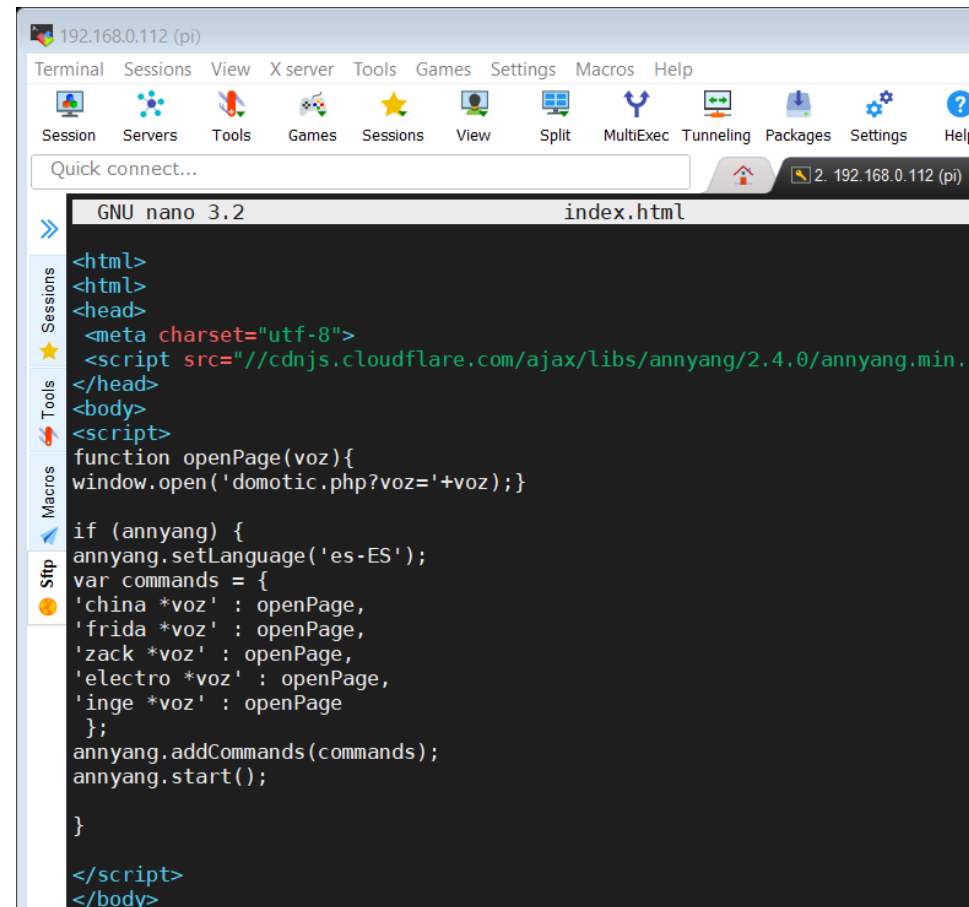
Finalmente, reiniciamos
Apache2

```
$ sudo /etc/init.d/apache2  
restart
```

```
# A self-signed (snakeoil) certificate can be created by installing  
# the ssl-cert package. See  
# /usr/share/doc/apache2/README.Debian.gz for more info.  
# If both key and certificate are stored in the same file, only the  
# SSLCertificateFile directive is needed.  
SSLCertificateFile      /etc/ssl/certs/ssl-cert-snakeoil.pem  
SSLCertificateKeyFile   /etc/ssl/private/ssl-cert-snakeoil.key
```

```
# A self-signed (snakeoil) certificate can be created by installing  
# the ssl-cert package. See  
# /usr/share/doc/apache2/README.Debian.gz for more info.  
# If both key and certificate are stored in the same file, only the  
# SSLCertificateFile directive is needed.  
SSLCertificateFile      /etc/apache2/ssl/server.crt  
SSLCertificateKeyFile   /etc/apache2/ssl/server.key
```

Programación HTML

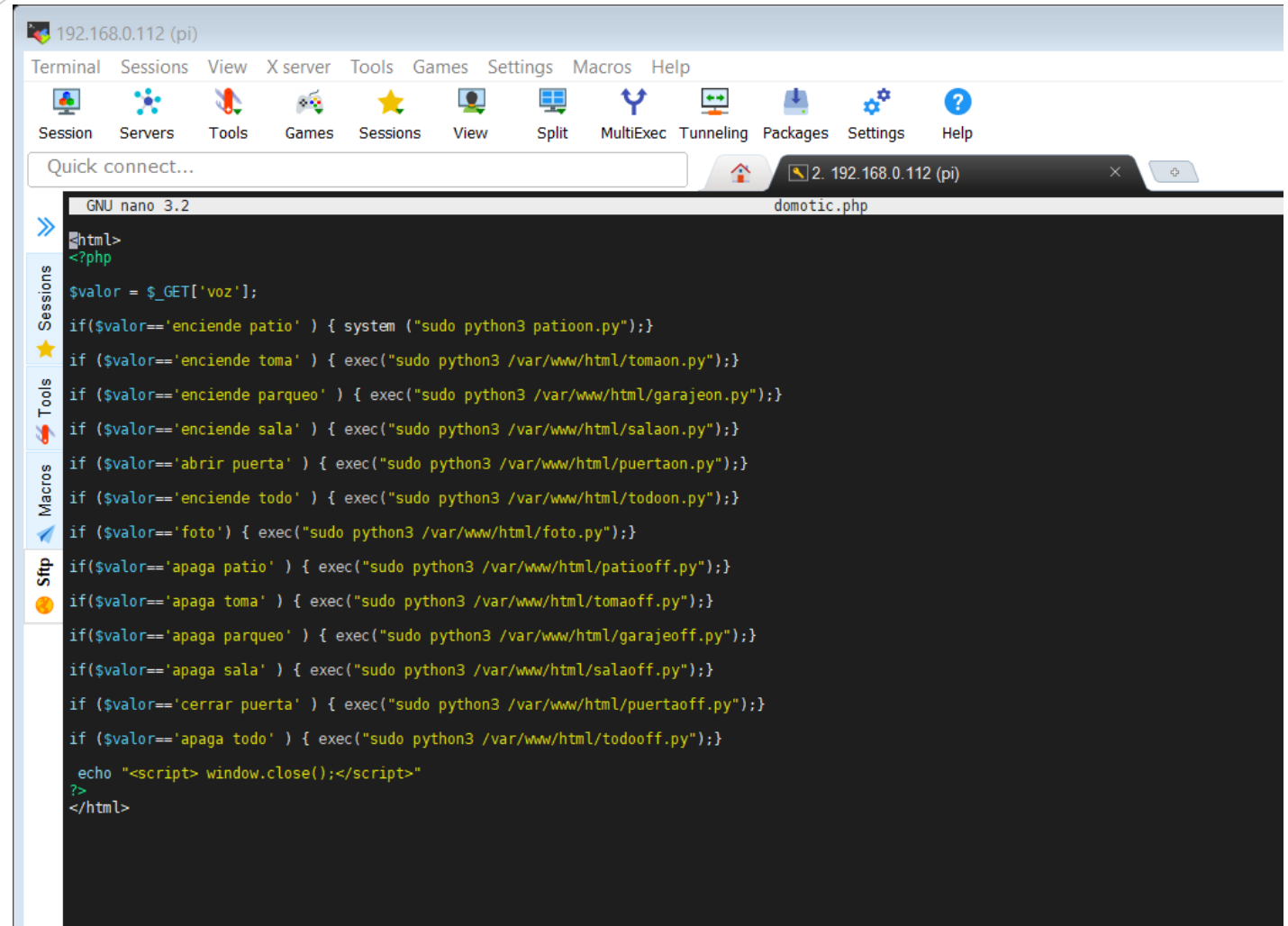


```
GNU nano 3.2 index.html
<html>
<html>
<head>
  <meta charset="utf-8">
  <script src="//cdnjs.cloudflare.com/ajax/libs/anyang/2.4.0/anyang.min.js">
</head>
<body>
  <script>
    function openPage(voz){
      window.open('domotic.php?voz='+voz);}

    if (anyang) {
      anyang.setLanguage('es-ES');
      var commands = {
        'china *voz' : openPage,
        'frida *voz' : openPage,
        'zack *voz' : openPage,
        'electro *voz' : openPage,
        'inge *voz' : openPage
      };
      anyang.addCommands(commands);
      anyang.start();
    }
  </script>
</body>
```

Este fichero es la web inicial, que carga el Javascript y permite grabar desde el audio de un navegador. Podremos cambiar ahí el idioma de origen y poner cualquier otro, así como la(s) palabra(s) que tiene que escuchar y tras ello lanzar una función que abre la web **domotic.php**
https://drive.google.com/drive/folders/1aHOP9RC9fqI4JgzfeEdBYTma_nNtTUGL?usp=sharing

Programación PHP



The screenshot shows a terminal window titled '192.168.0.112 (pi)' with a menu bar (Terminal, Sessions, View, X server, Tools, Games, Settings, Macros, Help) and a toolbar. A sidebar on the left contains icons for Session, Servers, Tools, Games, Sessions, View, Split, MultiExec, Tunneling, Packages, Settings, and Help. The main area displays a nano 3.2 editor with a file named 'domotic.php'. The code is a PHP script that uses the \$_GET superglobal to trigger various system commands via exec(). The commands are categorized by voice commands: 'enciende' (turn on) and 'apaga' (turn off) for patio, toma, parqueo, sala, and puerta. The script also includes an 'abrir puerta' (open door) command and an 'echo' statement to close the browser window.

```
GNU nano 3.2 domotic.php
<html>
<?php
$valor = $_GET['voz'];

if($valor=='enciende patio' ) { system ("sudo python3 patioon.py");}
if ($valor=='enciende toma' ) { exec("sudo python3 /var/www/html/tomaon.py");}
if ($valor=='enciende parqueo' ) { exec("sudo python3 /var/www/html/garajeon.py");}
if ($valor=='enciende sala' ) { exec("sudo python3 /var/www/html/salaon.py");}
if ($valor=='abrir puerta' ) { exec("sudo python3 /var/www/html/puertaon.py");}
if ($valor=='enciende todo' ) { exec("sudo python3 /var/www/html/todoon.py");}
if ($valor=='foto') { exec("sudo python3 /var/www/html/foto.py");}
if($valor=='apaga patio' ) { exec("sudo python3 /var/www/html/patiooff.py");}
if($valor=='apaga toma' ) { exec("sudo python3 /var/www/html/tomaoff.py");}
if($valor=='apaga parqueo' ) { exec("sudo python3 /var/www/html/garajeoff.py");}
if($valor=='apaga sala' ) { exec("sudo python3 /var/www/html/salaoff.py");}
if ($valor=='cerrar puerta' ) { exec("sudo python3 /var/www/html/puertaoff.py");}
if ($valor=='apaga todo' ) { exec("sudo python3 /var/www/html/todooff.py");}

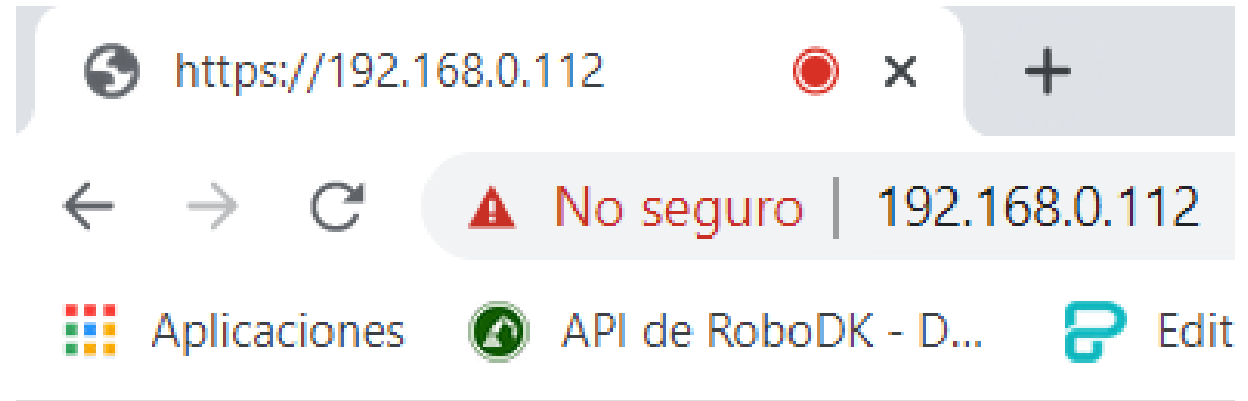
echo "<script> window.close();</script>"
?>
</html>
```

Este fichero se abre con el comando "Inge o cualquier nombre + encendiendo o apaga patio"
De acuerdo al estado encender o apagar, el código PHP llama a ejecutar los archivos de python.

Finalmente ingresamos al navegador e ingresamos

<https://192.168.137.2/>

La dirección IP varía según la asignación
Damos permisos de acceso al micrófono y
acceso a ventanas emergentes



Control Raspberry desde web- botones

```
GNU nano 3.2 botones.php
<html>
<head>
<meta name="viewport" content="width=device-width" />
<title>PRACTICA 12 SISTEMAS EMBEBIDOS.</title>
</head>
<body>
<center><h1>Control web SISTEMAS EMBEBIDOS 2020 IIT</h1>
<form method="get" action="botones.php">
  <input type="submit" style = "font-size: 16 pt" value="ApagarPatio" name="offPatio">
  <input type="submit" style = "font-size: 16 pt" value="EncenderPatio" name="onPatio">
</form></center>

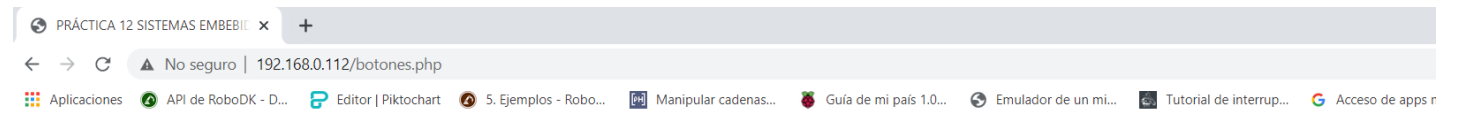
<center><form method="get" action="botones.php">
  <input type="submit" style = "font-size: 16 pt" value="ApagarParqueo" name="offParqueo">
  <input type="submit" style = "font-size: 16 pt" value="EncenderParqueo" name="onParqueo">
</form></center>

<center><form method="get" action="botones.php">
  <input type="submit" style = "font-size: 16 pt" value="ApagarSala" name="offSala">
  <input type="submit" style = "font-size: 16 pt" value="EncenderSala" name="onSala">
</form></center>

<center><form method="get" action="botones.php">
  <input type="submit" style = "font-size: 16 pt" value="ApagarEntrada" name="offEntrada">
  <input type="submit" style = "font-size: 16 pt" value="EncenderEntrada" name="onEntrada">
</form></center>

<?php
if(isset($_GET['offPatio']))
{
    echo "Patio apagado";
    system ("sudo python3 patiooff.py");
}
else if(isset($_GET['onPatio']))
{
    echo "Patio encendido";
    exec("sudo python3 patioon.py");
}
else if(isset($_GET['onParqueo']))
{
    echo "Parqueo encendido";
    exec("sudo python3 garajeon.py");
}
else if(isset($_GET['offParqueo']))
{
    echo "Parqueo apagado";
    exec("sudo python3 garajeoff.py");
}
else if(isset($_GET['onSala']))
{
    echo "Sala encendida";
    exec("sudo python3 salaon.py");
}
else if(isset($_GET['offSala']))
{
    echo "Sala Apagada";
    exec("sudo python3 salaoff.py");
}
else if(isset($_GET['onEntrada']))
{
    echo "Entrada encendida";
    exec("sudo python3 entradaon.py");
}
else if(isset($_GET['offEntrada']))
{
    echo "Entrada apagada";
    exec("sudo python3 entradaoff.py");
}
else{
    echo "SELECCIONE ACCIÓN";
}
?>
</body>
</html>
```

En el navegador
ingresar
<https://suip/botones.php>



Control web SISTEMAS EMBEBIDOS 2020 IIT



SELECCIONE ACCIÓN

PREGUNTAS?

