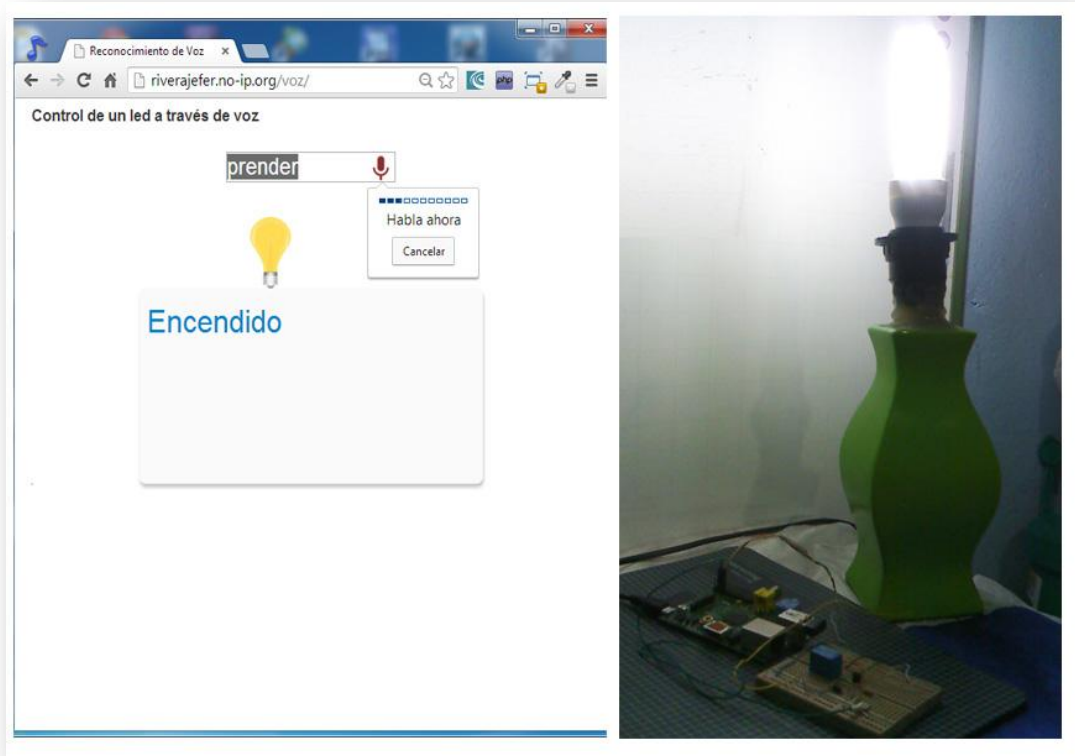


CONTROL DE UN BOMBILLO A TRAVÉS DE VOZ CON GOOGLE CHROME Y RASPBERRY PI



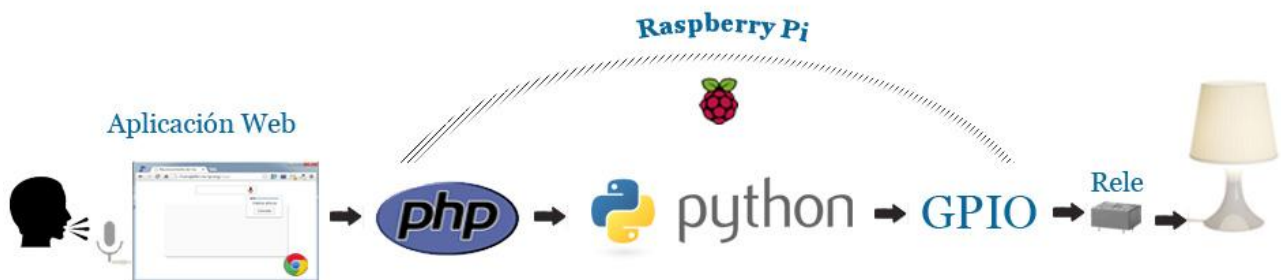
Desarrollado por:
Jefferson Rivera Patiño

@riverajefer
riverajefer.blogspot.com
jeffersonrivera.com

Contenido

1. DESCRIPCIÓN DEL PROYECTO.....	3
2. EXPLICACIÓN DEL SOFTWARE	3
2.1 APLICACIÓN WEB	3
3. EXPLICACIÓN DEL HARDWARE.....	7
4. PRUEBA DE FUNCIONAMIENTO	8
5. BIBLIOGRAFÍA.....	8
ANEXOS.....	9

1. DESCRIPCIÓN DEL PROYECTO



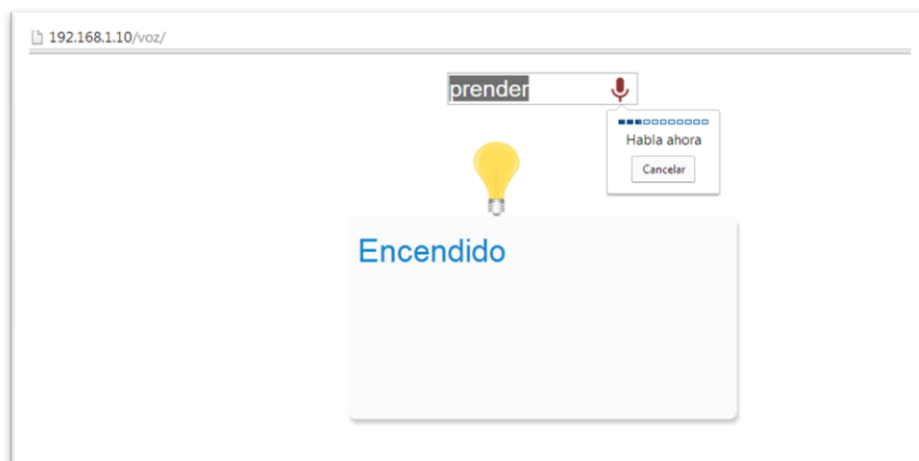
Como se visualiza en la imagen anterior, el flujo del proceso, va del cliente-navegador al hardware-lámpara 110v

El flujo es el siguiente, la persona por ejemplo pronuncia “prender” sobre el micrófono, que está conectado al PC, la señal que sale del micrófono la reconoce y procesa el navegador “chrome” luego la palabra pronunciada se convierte en una variable y se envía a PHP, donde se valida y se ejecuta un script python de acuerdo a la variable, en este caso “prende.py”, luego está señal digital que sale del GPIO, se pasa por un relevador y por último acciona el bombillo de la lámpara de 110V AC.

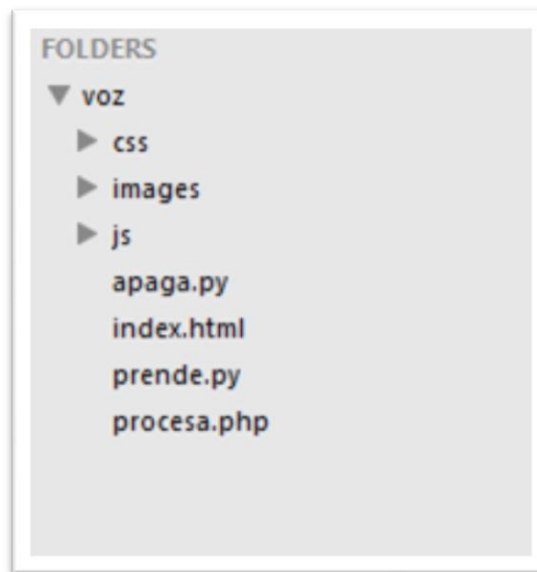
Importante el campo de reconocimiento de voz solo está disponible por el momento en Google Chrome.

2. EXPLICACIÓN DEL SOFTWARE

2.1 APLICACIÓN WEB



La aplicación cuenta con los siguientes directorios y archivos:



- **css/** contiene las hojas de estilo,
- **js/** contiene jQuery
- **images/** contiene la imagen del bombilo.
- **index.html** -> contiene el campo de reconocimiento de voz que nos ofrece Google Chrome, y la conexión ajax a PHP
- **procesa.php** recibe las variables del navegador "index.html", y de acuerdo a su valor, ejecuta prende.py, apaga.py o no hace nada.
- **apaga.py**, **prende.py** envía por el GPIO de la RPi 0 y 1 respectivamente.

Descarga desde:

<http://jeffersonrivera.com/pi/voz.zip>

index.html

```
index.html
1  <html>
2  <head>
3    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
4    <title>Reconocimiento de Voz</title>
5    <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css" />
6    <link rel="stylesheet" type="text/css" href="css/estilos.css" />
7    <script src="js/jquery-1.10.1.min.js"></script>
8  </head>
9  <body>
10   <script type="text/javascript">
11     function onChange(val) {
12       var valor = $("#entrada").val();
13       $("#caja").text(valor);
14       console.log("valor: "+valor);
15       $.ajax({
16         type: "POST",
17         url: "procesa.php",
18         data: "valor="+valor,
19         success: function(datos){
20           console.log( "Salida: " + datos);
21           switch(datos)
22           {
23             case "1":
24               $("#bombillo").show('slow');
25               $("#caja").text('Encendido');
26               console.log("Encendido");
27               break;
28             case "2":
29               $("#bombillo").hide('slow');
30               $("#caja").text('Apagado');
31               console.log("Apagado");
32               break;
33             case "3":
34               $("#caja").text('Sigue Intentando');
35               $("#bombillo").hide('slow');
36               console.log("Sigue Intentando");
37               break;
38           }
39         }
40       });
41     }
42   </script>
```

Index.html Parte 1

```
index.html
40   });
41   }
42 </script>
43 <h4>Control de un led a trav&eacute;s de voz</h4>
44 <br>
45 <div align="center">
46   <input onwebkitspeechchange="onChange(this.value)" x-webkit-speech="undefined" style="font-size: 25px;" id="entrada" autofocus>
47 </div>
48 <br><br>
49 <div id="bombillo"></div>
50 <div id="caja">
51 </div>
52 </body>
53 </html>
```

Index.html parte 2

Procesa.php

```
procesa.php
1 <?php
2 $valor=$_POST['valor'];
3 if($valor=='on' || $valor=='activar' || $valor=='prender' || $valor=='encender' || $valor=='enciende')
4 {
5     exec('sudo python /var/www/voz/prende.py');
6     $retorna=1;
7 }
8 elseif($valor=='off' || $valor=='apaga' || $valor=='apagar' || $valor=='desactivar')
9 {
10    exec('sudo python /var/www/voz/apaga.py');
11    $retorna=2;
12 }
13 else {
14     $retorna=3;
15 }
16
17 echo $retorna;
18 ?>
19
```

Procesa.php

prende.py

```
apaga.py  x  prende.py  x
1 #prende.py
2 import RPi.GPIO as GPIO
3 GPIO.setup(7, GPIO.OUT)
4 GPIO.output(7, False)
5
6
```

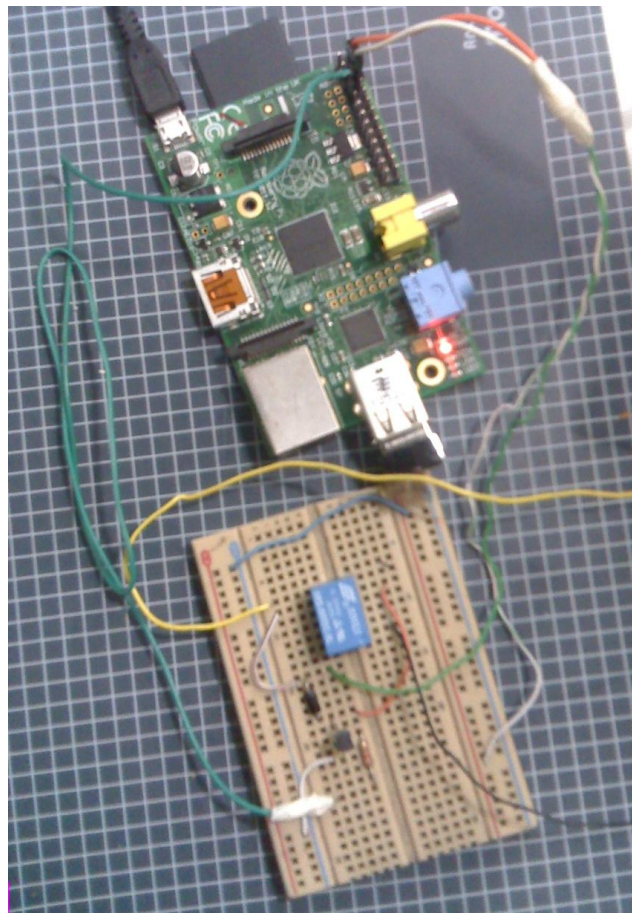
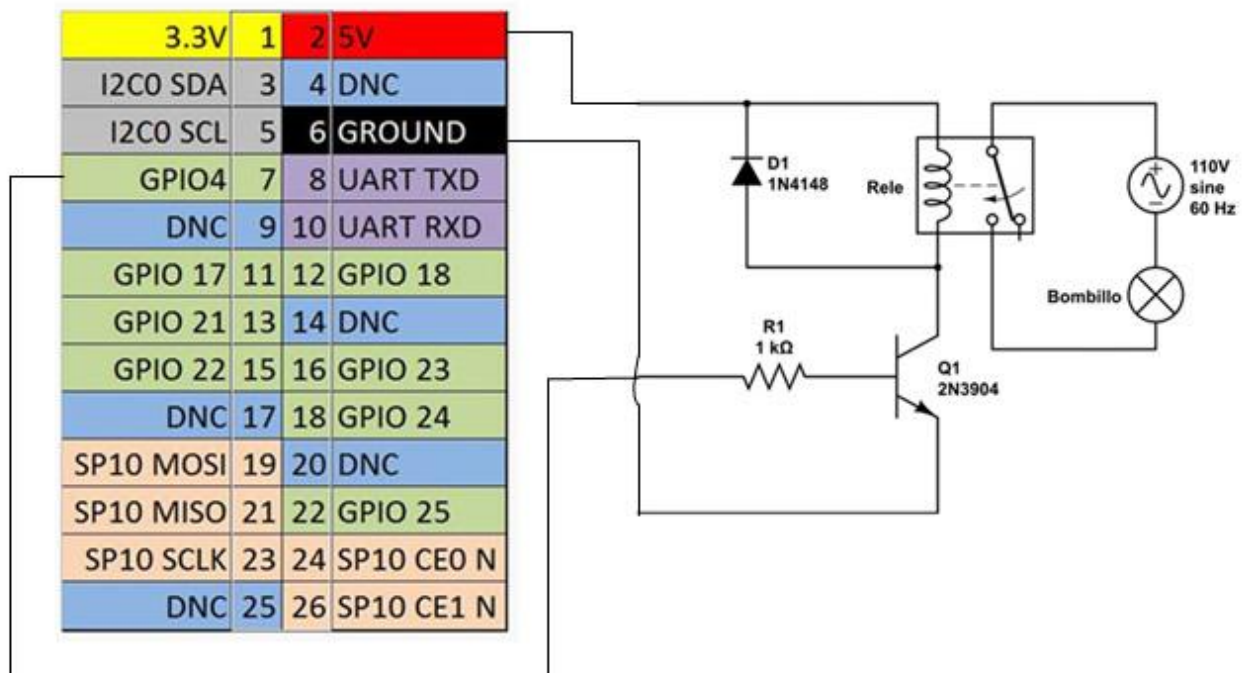
Prende.py

apaga.py

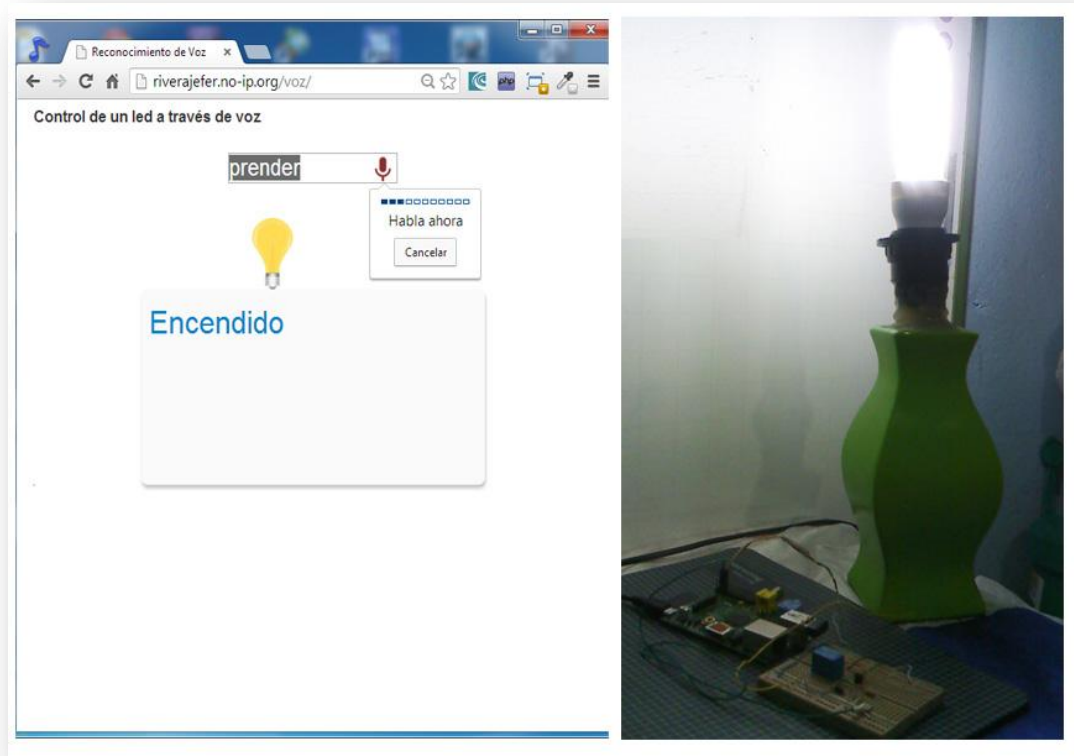
```
apaga.py  x  prende.py  x
1 #apaga.py
2 import RPi.GPIO as GPIO
3 GPIO.setup(7, GPIO.OUT)
4 GPIO.output(7, True)
5
```

apaga.py

3. EXPLICACIÓN DEL HARDWARE



4. PRUEBA DE FUNCIONAMIENTO



Ver video

5. BIBLIOGRAFÍA

- <http://slides.html5rocks.com/#speech-input>
- <http://www.sitepoint.com/html5-speech-input-fields/>
- <http://getbootstrap.com/2.3.2/>
- <https://www.circuitlab.com/editor/>
- <http://www.scoop.it/t/raspberry-pi/p/3994797536/raspberry-pi-driving-a-relay-using-gpio>
- http://img.scoop.it/NjdOaNV3M5_Uv4qdkSZInDI72eJkfbmt4t8yenImKBVaiQDB_Rd1H6kmuBWtceBJ

ANEXOS

Código fuente de la aplicación:

<http://jeffersonrivera.com/pi/voz.zip>

Desarrollado por:
Jefferson Rivera Patiño

@riverajefer
riverajefer.blogspot.com
jeffersonrivera.com