



Proyecto Integrado
Ayuda Abuelo

Proyecto Integrado

Autor/a: Mercedes Barea Morales

Tutor: Luis González

Curso 2021/2022

**I.E.S. Francisco Romero
Vargas**



Administración de Sistemas Informáticos en Red

ÍNDICE

1. Introducción

1.1. Introducción	3
1.2. Finalidad	3
1.3. Objetivos	4
1.4. Medios necesarios	6
1.5. Planificación	7

2. Realización del Proyecto.

2.1. Trabajos realizados	8
2.2. Problemas encontrados	27
2.3. Modificaciones sobre el proyecto planteado inicialmente	29
2.4. Posibles mejoras al proyecto	29
2.5. Bibliografía	30

1. INTRODUCCIÓN

Actualmente, la sociedad ha avanzado respecto a la implementación de la tecnología como medio de comunicación y gestión en diferentes entornos, no solo laborales, también domésticos y socioeconómicos. Ante esta realidad, hay un gran número de personas que no han sabido o podido adaptarse a estos cambios, por ende, se han quedado atrasados, y poco a poco la sociedad ha dejado abandonados a su suerte a estas personas, sin ninguna ayuda para facilitarles el acceso y los conocimientos necesarios para poder realizar ciertas gestiones esenciales.

Este problema ha llevado a estas personas, mayoritariamente adultos de avanzada edad, a una situación de abandono casi total por parte de las instituciones y entidades bancarias, que desde el 2020 con motivo de la Covid19, se fueron digitalizando todo tipo de gestiones a pasos agigantados. Este proyecto nace ante la necesidad de cubrir ese abandono de manera parcial, con los medios y conocimientos que dispongo.

1.2. Finalidad

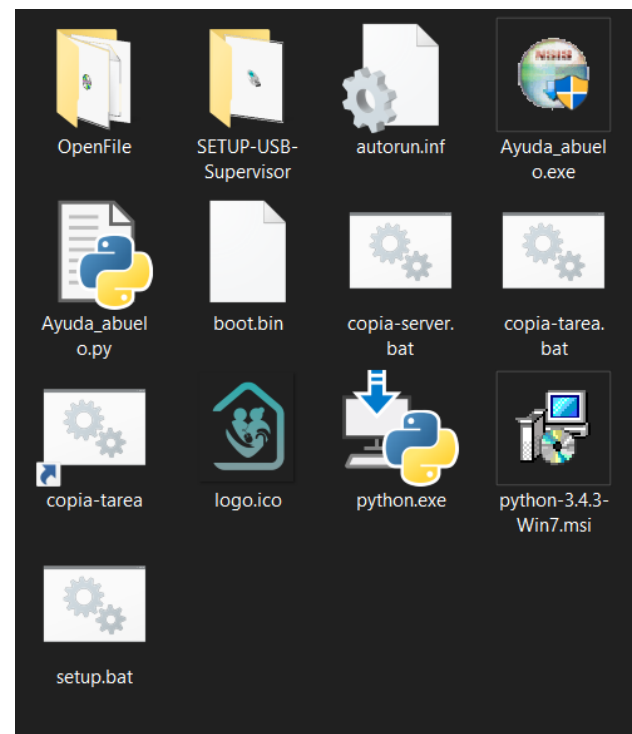
La finalidad de este proyecto es acercar la información justa y necesaria para que las personas desconocedoras de esta, puedan introducirse en la nueva era digital en la que nadie se ha parado a prestarles atención, ya que el cambio ha sido radical y no todos han sabido adaptarse, acercandoles una forma de fácil acceso a la red, en cuestión a una página web, con una interfaz intuitiva y sencilla en la que podrá visualizar guías sobre diferentes contenidos esenciales y de ocio digital.

1.3. Objetivos

Hay diferentes escalas de dificultad en la que se puede clasificar a las personas que les cuesta acceder a la información, para ello se ha pensado desde un primer momento en las personas que solo saben encender el ordenador.

Para empezar, necesita una previa instalación en el equipo a través de un **pendrive autoarrancable** que contiene los siguientes archivos:

- Los “.exe” como el de **python**; que instala el programa para leer archivos “.py”. Está “Ayuda_abuelo.exe” que tiene comprimido todos los archivos y los deja en “C:\Program Files\ayuda_abuelo” por si hiciera falta una reinstalación.
- El archivo “.py” que es el **servidor http en python** que escucha las peticiones http del mando a distancia.
- Los “.bat” contienen las **órdenes de instalación** de estos archivos y los mueve, como el “.py” al escritorio y a la carpeta que inicia programas en windows cuando inicia sesión.
- Y algún **acceso directo** para la ejecución de algún “.bat” con permisos de administrador.
- El otro python.msi es para la instalación de python en Windows 7, está comprobado que tiene total compatibilidad con esa versión de python.



El servidor http en python interactúa con un **mando a distancia**; con una placa de arduino con el **módulo esp8266**(módulo Wifi) que se conecta al **Wifi** (previa configuración de dicho dispositivo) y manda una **petición http** al equipo que contiene el servidor http instalado (es recomendable tener una IP fija en el equipo que se realiza la instalación del servidor).

Tras mandar dicha petición al **programa**(servidor http) creado en **python**, que se abre automáticamente al iniciar sesión en el equipo, dicho programa escucha la petición y responde abriendo el **navegador chrome** (elegido por ser el más popular y utilizado) con la página web www.ayuda.abuelo.tk, donde se alojan las guías con la información necesaria para desenvolverse en redes sociales, correo electrónicos, etc.

También este programa de python se queda instalado en el escritorio para esas otras personas que sepan abrirlo puedan darle, abrir el servidor http de python y pulsar el botón del mando a distancia, para así abrirse la página web automáticamente.

Siempre pueden acceder a la página directamente desde cualquier navegador escribiendo www.ayuda.abuelo.tk

Hay diferentes **vías de acceso** a la web dispuestas para el usuario, solo depende del tipo de dificultad que tenga, usará una u otra.

Tras acceder a la página web, la **navegación** sobre ella es sencilla, ya que dispone de paneles y botones de gran tamaño y el tamaño de la letra creo que es el adecuado para aquellas personas que tengan dificultad en la vista. Dichos **tutoriales** dentro de cada sección de la web también se pueden visualizar en gran tamaño, son ampliables y reducibles y se pueden descargar para poder ser impresos.

El **vocabulario** utilizado en dicho tutorial está cuidado para que sea fácil de entender para personas tanto mayores como menores de edad, sin el uso de palabras muy técnicas, y en caso de haber, habiéndose explicado anteriormente para conocer su referencia.

1.4. Medios Necesarios

Medios Físicos:

- Un ordenador (otro de prueba con Windows 7)
- Un pendrive configurado con auto-arranque
- Placa de arduino Nodemcu con modulo ESP3266
- 4 cables
- Pila de petaca de 9V
- Hilo de material 3D
- Impresora 3D Ender 3 Pro
- Botón pulsador arduino
- Embellecedor de botón arduino
- Cinta de pegamento de doble cara
- Cable USB

Medios Lógicos:

- Arduino IDE
- SamLogic USB Autorun Creator
- Visual Studio Code
- Tinkercad (Online)
- Python
- Google Chrome
- Amazon Web Service(AWS)
- NSIS
- Apache
- Winrar
- CMD
- Bloc de notas

1.5. Planificación

- Investigación y búsqueda de la placa de arduino más favorable: 1H
- Investigación y desarrollo del código para el funcionamiento del arduino: 4H
- Investigación, prueba y realización del servidor http en python: 5H
- Creación del diseño de la página web: 20H
- Creación de tutoriales para la página web: 5'30H
- Investigación y configuración del pendrive autoarrancable: 9H
- Uso del NSIS: 45 min
- Diseño de la caja 3D: 2'30H
- Impresión 3D: 4H
- Montaje del circuito eléctrico: 20 min
- Configuración de instancia en AWS e instalación del servidor web: 3H

TOTAL: 55'05 HORAS

2. REALIZACIÓN DEL PROYECTO

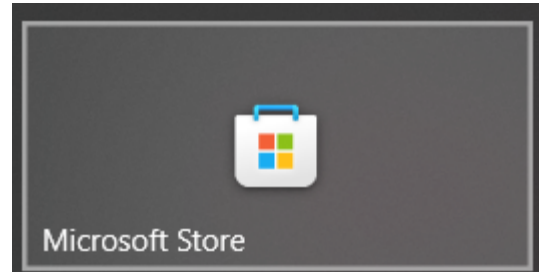
2.1. Trabajos realizados

Arduino

Primero de todo, para que el equipo detecte el la placa de arduino Nodemcu V2 ESP8266 tendremos que instalar su driver:

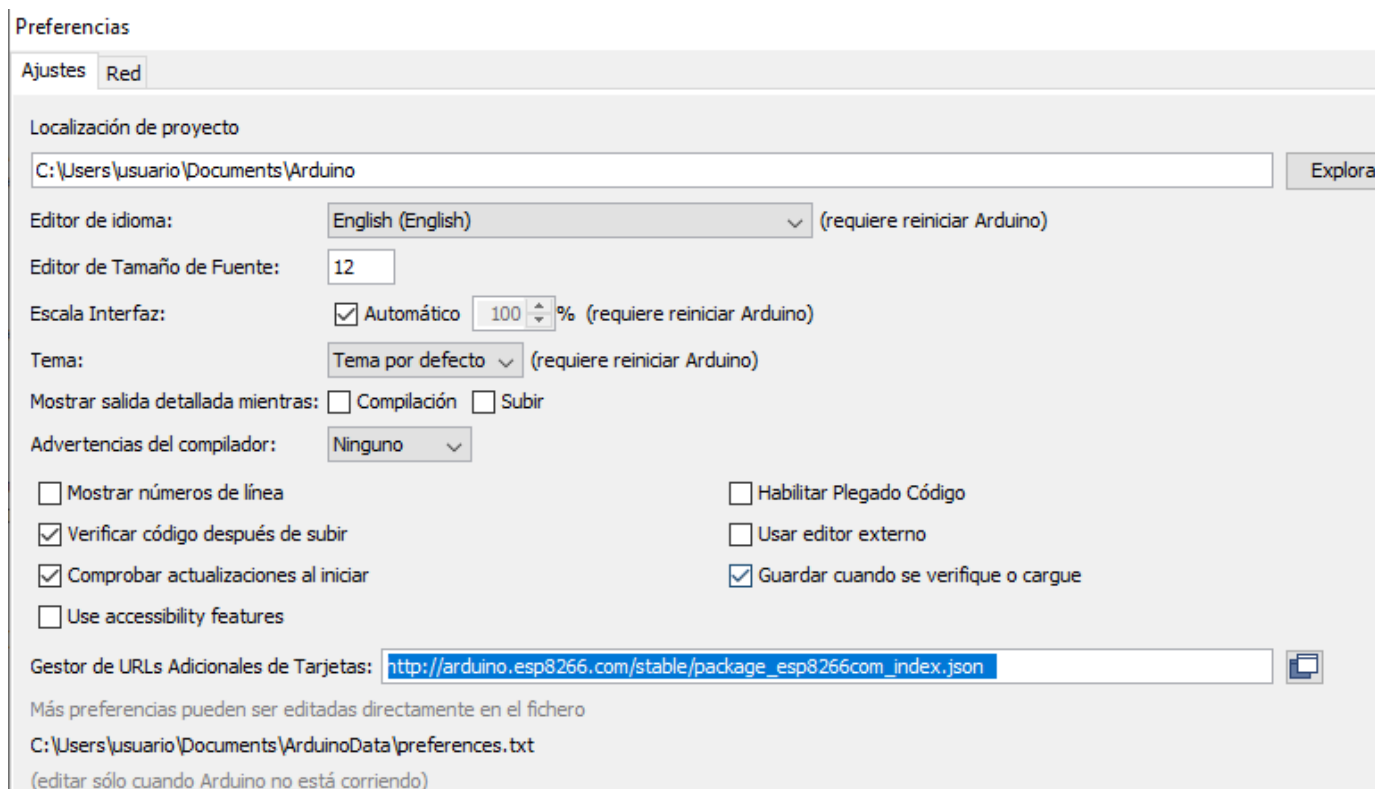
[CP210x USB to UART Bridge VCP Drivers - Silicon Labs \(silabs.com\)](http://silabs.com/CP210x_USB_to_UART_Bridge_VCP_Drivers)

Tras esto, instalar arduino IDE desde la Microsoft Store



Para poder tener los parámetros de la tarjeta en arduino IDE, nos vamos a Archivo < Preferencias y en Gestor de URLs adicionales de tarjetas pegamos el siguiente enlace:

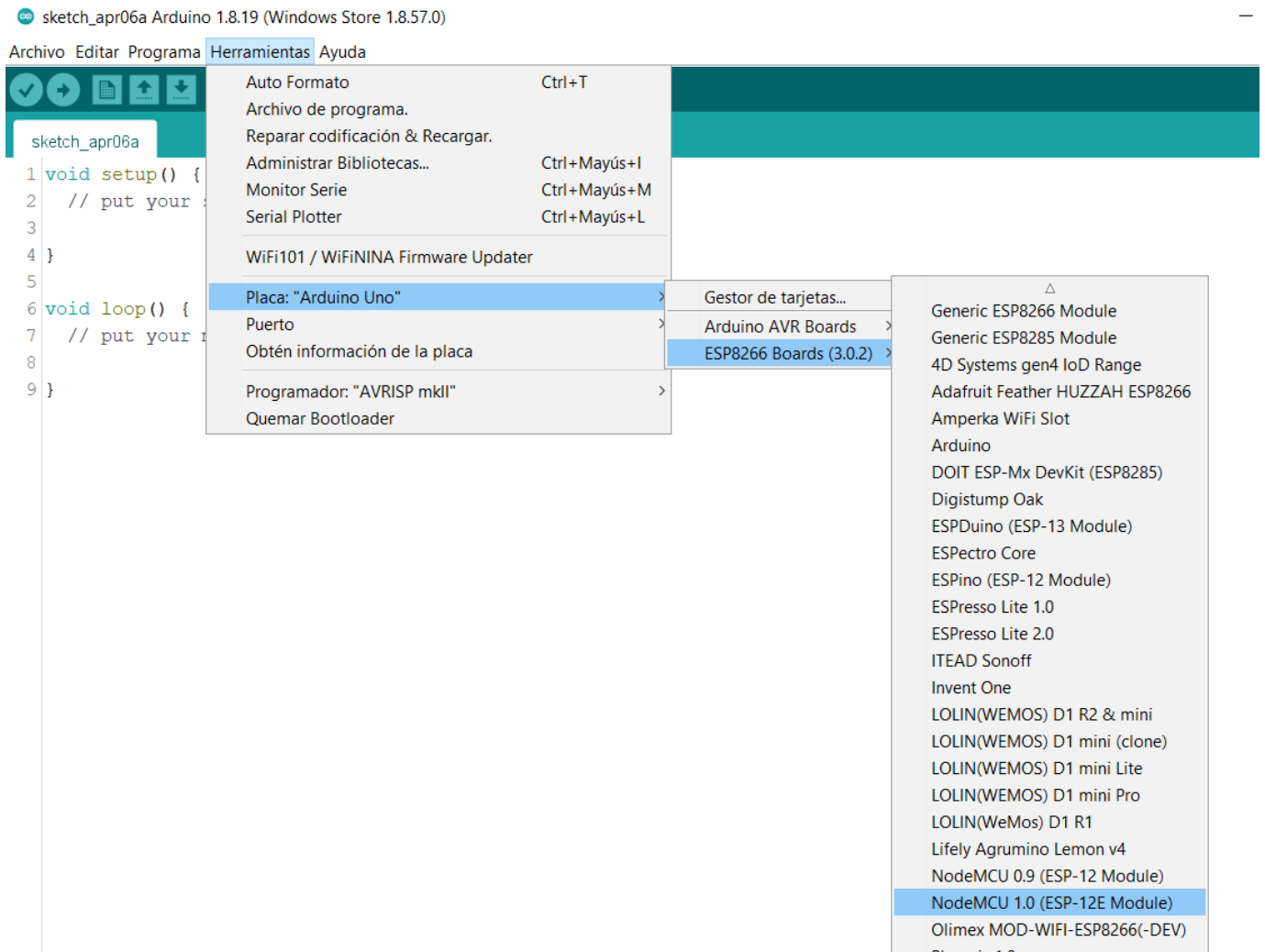
http://arduino.esp8266.com/stable/package_esp8266com_index.json



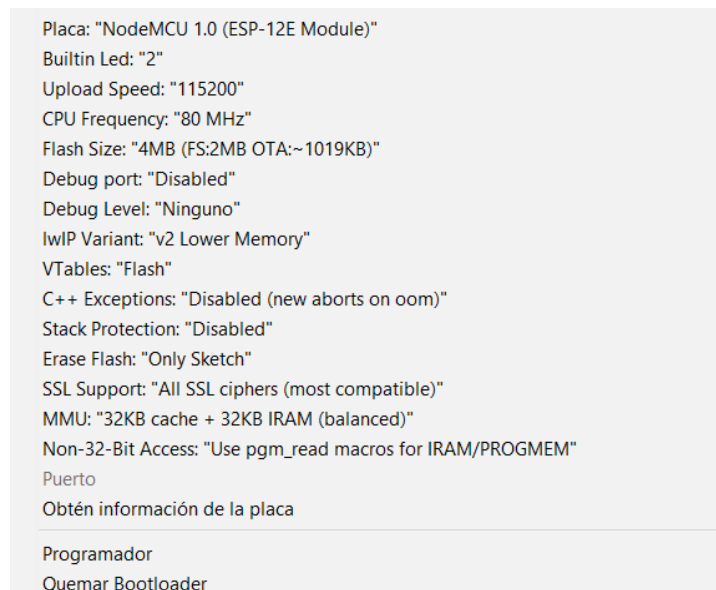
Ahora en Herramientas < Placa < Gestor de tarjetas buscamos esp8266



Y cambiamos la placa ahora por la Nodemcu 1.0.



Tras esto debería verse así, a continuación, pon los parámetros como muestra a continuación:



Una vez así, se le añadirá el siguiente código:

```
#include <ESP8266WiFi.h>

//Nombre de red casa
const char *nombre = "MiFibra-B24A";
//Password
const char *password = "HSzLs9Fc";

//Nombre de red clase
//const char *nombre = "informatica-inf1-2";
//Password
//const char *password = "hdphdj1234";

WiFiClient client;
IPAddress server(192, 168, 1, 85); // IP pc casa
// IPAddress server(172, 30, 20, 5); // IP pc clase

void setup()
{
  Serial.begin(115200);
  delay(50); // breve espera para que se concrete la conexion

  // Conexion al WiFi
  WiFi.begin(nombre, password);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(100); //aquí debemos esperar unos instantes
  }

  Serial.println("Conexión exitosa al WiFi");

  Serial.print("la IP es: ");
  Serial.print(WiFi.localIP());
  Serial.println("");

  pinMode(BUILTIN_LED, OUTPUT); //LED azul de la placa
  pinMode(4, INPUT_PULLUP);
  //0 en el flash 4 D2
}

void loop()
{
  // put your main code here, to run repeatedly:
  if (digitalRead(4) == LOW)
  {
```

```

digitalWrite(BUILTIN_LED, LOW);
if (client.connect(server, 8080))
{
  Serial.println("connected");
  // Make a HTTP request:
  client.println("GET / HTTP/1.0");
  client.println();
  Serial.println("Conectado a www.ayuda.abuelo.tk");
}

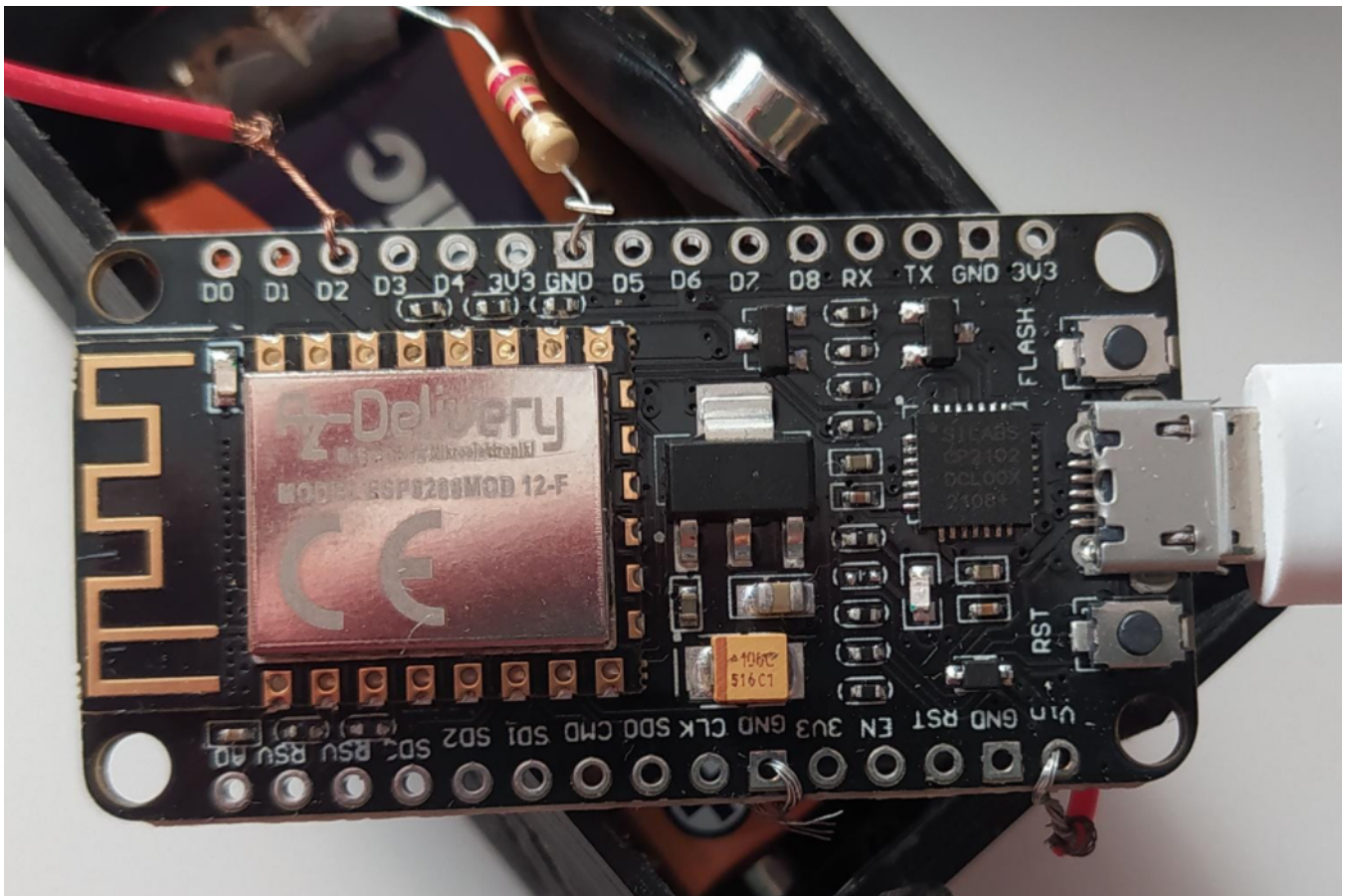
  delay(2000);
}else{
  digitalWrite(BUILTIN_LED, HIGH);
}
}

```

Y este sería el código del arduino: Conecta al Wifi, se le dará una IP por DHCP del router, y teniendo el servidor http Python escuchando en el PC, poniendo su IP en "**WiFiClient client; IPAddress server(192, 168, 1, 85);**" (la ip fija del equipo), le hará una petición HTTP al pulsar el botón configurado del mando de arduino; "**pinMode(4, INPUT_PULLUP);**" y "**if (digitalRead(4) == LOW)**", haciendo la petición con la linea del código **client.println("GET / HTTP/1.0");** .

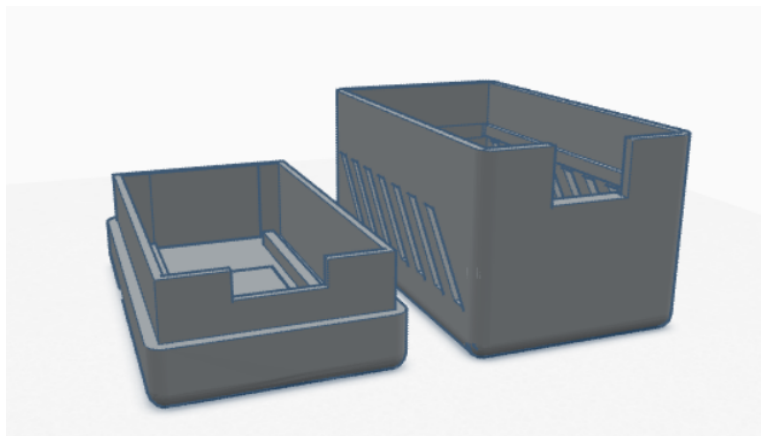
La respuesta te la debe dar el servidor web python, que ahora veremos cómo se configura.

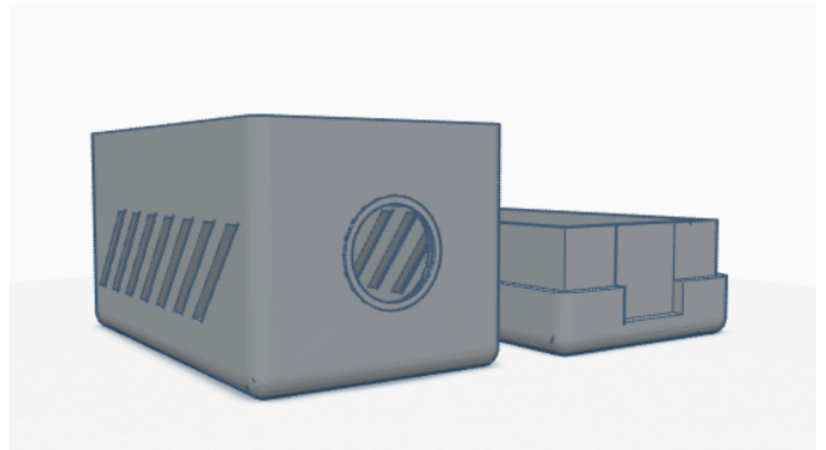
El botón se le ha puesto en la entrada física **D2**, en la entrada digital 4, de ahí ese números. Se le ha puesto una resistencia en la GND (tierra) y conectado una **pila de petaca de 9V** como batería.



Caja 3D

Para que quede todo todo bien recogido, se le ha hecho una impresión en 3D donde almacenar la placa, la pila y su botón, con el **orificio** para sacar el embellecedor y poder pulsarlo, y una **rendija cuadrada** para poder dar acceso a la cabeza de un cable USB para así cambiar el código; cambia el Wifi y la contraseña o la IP del equipo al que se conecta al servidor http que escucha. Aquí os muestro los diseños realizados en Tinkercad:





Servidor Http Python

La petición que realiza el arduino debe ser recibida por este programa que es un servidor HTTP en lenguaje Python.

Lo único que habría que cambiar es la ruta donde se tiene el archivo **chrome.exe**, ya que este puede instalarse en "**C:\Program Files**" o a veces "**C:\Program Files(x86)**". También puede ser este navegador o puede ser cualquier otro, solo copiar y pegar la ruta del archivo que abre el navegador que quieras realmente en el "subprocess.Popen(["

```
#!/usr/bin/env python3
```

```
"""
```

Very simple HTTP server in python for logging requests

Usage::

```
./server.py [<port>]
```

```
"""
```

```
from http.server import BaseHTTPRequestHandler, HTTPServer
```

```
import logging
```

```
import subprocess
```

```
class S(BaseHTTPRequestHandler):
```

```
    def _set_response(self):
```

```
        self.send_response(200)
```

```
        self.send_header('Content-type', 'text/html')
```

```
        self.end_headers()
```

```

def do_GET(self):
    logging.info("GET request,\nPath: %s\nHeaders:\n%s\n", str(self.path), str(self.headers))
    self._set_response()
    self.wfile.write("GET request for {}".format(self.path).encode('utf-8'))
    subprocess.Popen(['C:\Program Files (x86)\Google\Chrome\Application\chrome.exe',
'http://www.ayuda.abuelo.tk'])
    #subprocess.Popen(['C:\Program File\Google\Chrome\Application\chrome.exe',
'http://www.ayuda.abuelo.tk'])

def do_POST(self):
    content_length = int(self.headers['Content-Length']) # <--- Gets the size of data
    post_data = self.rfile.read(content_length) # <--- Gets the data itself
    logging.info("POST request,\nPath: %s\nHeaders:\n%s\n\nBody:\n%s\n",
        str(self.path), str(self.headers), post_data.decode('utf-8'))

    self._set_response()
    self.wfile.write("POST request for {}".format(self.path).encode('utf-8'))

def run(server_class=HTTPServer, handler_class=S, port=8080):
    logging.basicConfig(level=logging.INFO)
    server_address = ("", port)
    httpd = server_class(server_address, handler_class)
    logging.info('Starting httpd...\n')
    try:
        httpd.serve_forever()
    except KeyboardInterrupt:
        pass
    httpd.server_close()
    logging.info('Stopping httpd...\n')

if __name__ == '__main__':
    from sys import argv

    if len(argv) == 2:
        run(port=int(argv[1]))
    else:
        run()

```


Este al abrirse se muestra así:

```
C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.10_3.10.1520.0_x64__qbz5n2kfra8p0\python3.10.exe
INFO:root:Starting httpd...
```

Y al escuchar, muestra el GET realizado por el arduino, la IP que ha cogido, y la fecha y hora de la petición. Tras esto, abre automáticamente la página web www.ayuda.abuelo.tk

Servidor Web en AWS

Primero, antes de ponernos con la configuración del servidor web, cojamos de Freenom un nombre de dominio:

 A Name for Everyone

Services ▾

Managing abuelo.tk

Information Upgrade Management Tools ▾ Manage Freenom DNS

Information

To the right you can find the details of your domain.
You can manage your domain using the tabs above.

« Back to Domains List

Domain:
abuelo.tk ACTIVE

Registration Date:
16/05/2022

Expiry date:
16/05/2023

Tras esto, ya sí podemos ponernos en marcha con **AWS**:

Asignamos una IP elástica (IP pública) para la instancia;

EC2 > Direcciones IP elásticas > Asignar la dirección IP elástica

Asignar la dirección IP elástica [Información](#)

Configuraciones de la dirección IP elástica [Información](#)

Grupo fronterizo de red [Información](#)

us-east-1

Grupo de direcciones IPv4 públicas

☒ Grupo de direcciones IPv4 de Amazon

✔ **La dirección IP elástica se ha asignado correctamente.**
Dirección IP elástica 35.174.187.167

Asociar esta dirección IP elástica

✔	publica tfg 🔗	35.174.187.167
---	-------------------------------	----------------

Lanzaremos una instancia donde montaremos el servidor web:

EC2 > Instancias > Lanzar una instancia

Lanzar una instancia [Información](#)

Amazon EC2 le permite crear máquinas virtuales, o instancias, que se ejecutan en la nube de AWS. Comience rápidamente siguiendo los sencillos pasos que se indican a continuación.

Nombre y etiquetas [Información](#)

Nombre

tfg: www.abuelo.tk

[Agregar etiquetas adicionales](#)


Para ello usaremos la plantilla gratuita con la AMI que expone la siguiente imagen:

Recientes

Inicio rápido

☒ Ejecutado recientemente

☐ Actualmente en uso



Buscar más AMI

Incluidas las AMI de AWS, Marketplace y la comunidad

Amazon Machine Image (AMI)

amzn2-ami-kernel-5.10-hvm-2.0.20211223.0-x86_64-gp2

ami-08e4e35cccc6189f4

2022-01-05T22:28:32.000Z

arquitectura: 64 bits (x86)

Virtualización: hvm

Habilitado para ENA: true

Tipo de dispositivo raíz: ebs

Descripción

Amazon Linux 2 Kernel 5.10 AMI 2.0.20211223.0 x86_64 HVM gp2

Arquitectura

ID de AMI

x86_64

ami-08e4e35cccc6189f4

Creamos su par de claves para iniciar sesión y la exportamos.

▼ Par de claves (inicio de sesión)

Puede utilizar un par de claves para conectarse al servidor seleccionado antes de lanzar la instancia.

Nombre del par de claves - obligatorio

clavetfg

Descargas

 clavetfg.pem

[Abrir archivo](#)

Ahora la configuración de red y los grupos de seguridad

▼ Configuraciones de red

VPC - obligatorio [Información](#)

vpc-069634539b3d06417 (privada)
172.31.0.0/16

(predeterminado) ▼



Subred [Información](#)

Sin preferencias ▼



[Crear una nueva subred](#)

Asignar automáticamente IP pública [Información](#)

Desactivar ▼

Firewall (grupos de seguridad) [Información](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☐ Crear grupo de seguridad

☒ Seleccionar un grupo de seguridad existente

Grupos de seguridad comunes [Información](#)

Seleccionar grupos de seguridad ▼



[Comparar reglas de grupo de seguridad](#)

cms1 sg-0dbcb4609294d0819 ✕
VPC: vpc-069634539b3d06417

Los grupos de seguridad que agrega o elimine aquí se agregarán a todas las interfaces de red o se eliminarán de ellas.

Asignar un grupo de seguridad: ☒ Crear un nuevo grupo de seguridad

☐ Seleccionar un grupo de seguridad existente

Nombre del grupo de seguridad:

cms1

Descripción:

cms security 2022-01-25T00:19:49.334+01:00


Tipo	Protocolo	Rango de puertos	Origen
SSH ▼	TCP	22	Personaliz... 0.0.0.0/0
HTTPS ▼	TCP	443	Personaliz... 0.0.0.0/0, ::/0
HTTP ▼	TCP	80	Personaliz... 0.0.0.0/0, ::/0

Dirección IP elástica: 35.174.187.167

Tipo de recurso

Elija el tipo de recurso al que desea asociar la dirección IP elástica.

- ☒ Instancia
- ☐ Interfaz de red

 Si asocia una dirección IP elástica a una instancia que ya tiene asociada una dirección de este tipo, la dirección IP elástica anterior se desasociará, pero aun así se asignará a su cuenta. [Más informa](#)

Instancia

Q Elegir una instancia



i-05187279670b473f0 (NGINX NO FUNCIONA) - stopped

i-06d5ab0de3763533a (cms nginx) - running

i-0b5a0b98d90b2521c (PHP-WORDPRESS-CMS1) - running

i-059f6c13289c58ed2 (tfg: www.abuelo.tk) - running

Y ya estaría su configuración. Ahora observamos un resumen de dicha instancia con su ip pública (elástica) y privada.

	Name	ID de la instancia	Estado de la i...	Tipo de inst...	Comprobación ...	Es
<input checked="" type="checkbox"/>	tfg: www.abuelo.tk	i-059f6c13289c58ed2	✓ En ejecución	t2.micro	✓ 2/2 comprobaci...	Si
<input type="checkbox"/>	PHP-WORDPRESS-CMS1	i-0b5a0b98d90b2521c	✓ En ejecución	t2.micro	✓ 2/2 comprobaci...	Si
<input type="checkbox"/>	cms nainx	i-06d5ab0de3763533a	✓ En ejecución	t2.micro	✓ 2/2 comprobaci...	Si

Instancia: i-059f6c13289c58ed2 (tfg: www.abuelo.tk)

Detalles	Seguridad	Redes	Almacenamiento	Comprobaciones de estado	Monitoreo	Etiquetas
▼ Resumen de instancia Información						
ID de la instancia		Dirección IPv4 pública		Direcciones IPv4 privadas		
 i-059f6c13289c58ed2 (tfg: www.abuelo.tk)		 35.174.187.167 dirección abierta 		 172.31.91.105		

Y vamos a conectarnos:

Conectarse a la instancia [Información](#)

Conéctese a la instancia i-059f6c13289c58ed2 (tfg: www.abuelo.tk) mediante cualquiera de estas opciones


Conexión de la instancia EC2

Administrador de sesiones

Cliente SSH

Consola de serie de EC2

ID de la instancia

 i-059f6c13289c58ed2 (tfg: www.abuelo.tk)

Dirección IP pública

 35.174.187.167

Nombre de usuario

ec2-user

Conéctese con un nombre de usuario personalizado o use el nombre de usuario predeterminado ec2-user para la AMI empleada en el lanzamiento de la instancia.

Instalación y configuración de paquetes:

Nada mas conectar:

sudo su y yum update.

yum install httpd (para el servidor web Apache)

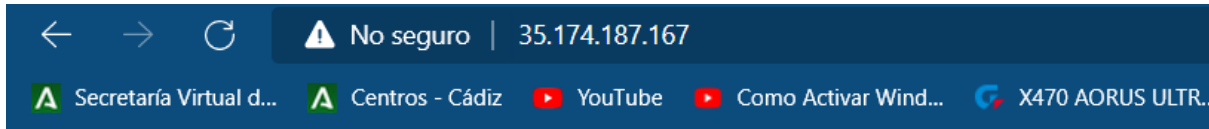
```
[ec2-user@ip-172-31-91-105 ~]$ sudo su
[root@ip-172-31-91-105 ec2-user]# yum install httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
amzn2extra-docker
```

```
[root@ip-172-31-91-105 ec2-user]# systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target
emd/system/httpd.service.
[root@ip-172-31-91-105 ec2-user]# systemctl start httpd
[root@ip-172-31-91-105 ec2-user]#
```

En /etc/httpd/conf/httpd.conf

```
<VirtualHost *:80>
    ServerName www.ayuda.abuelo.tk
    DocumentRoot "/var/www/abuelo"
</VirtualHost>
```

Y en /var/www/abuelo añadí un index.html de muestra para visualizar algo:



Bienvenido a www.ayuda.abuelo.tk

En cuanto se validó en freenom el nombre de ayuda para abuelo.tk, se pudo realizar la búsqueda a través de www.ayuda.abuelo.tk

Una vez realizado esto, empecé el diseño de la página web y los tutoriales.

Pendrive Autoarrancable

Ya tendríamos el mando a distancia, el servidor http y el servidor web montado pero, para que esto funcione desde un equipo hay que instalarlo en un equipo y de manera automática, ya que el usuario cliente no va saber hacerlo. Para ello surgió la idea de usar un pendrive que al auto-arrancar, instalase y moviera los archivos a los diferentes sitios de utilidad. Veamos cómo he creado este pendrive y sus diferentes archivos:

.bat

Primero tienes que saber que tenemos ya el "Ayuda_abuelo.py" creado. Para que se pueda leer, he descargado el **python.exe** que instala ese lenguaje en el sistema para poder ejecutarlo. Vamos a ir viendo los .bat que ejecutan diferentes órdenes durante la instalación:

- **copia-server.bat:** Este abre un cmd y copia el "Ayuda_abuelo.py" en el escritorio del usuario, de esta forma, si no se abre automáticamente, puede abrirse pulsando desde el escritorio.
- **copia-tarea.bat:** Este abre un cmd y copia el "Ayuda_abuelo.py" en esta ruta;

"%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup"

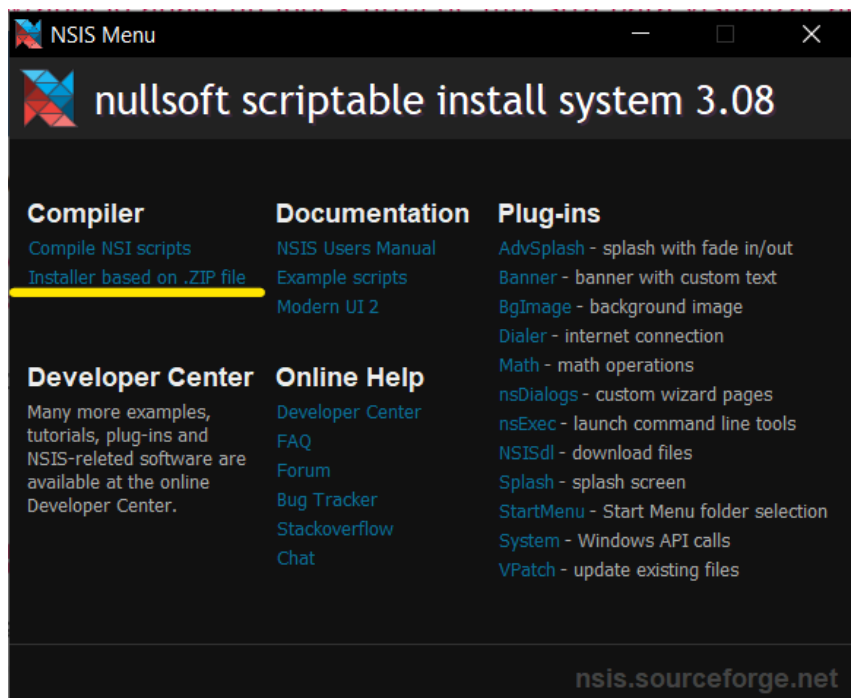
Dicha ruta hace que todo archivo de su interior se ejecute o abra al iniciar sesión el usuario. De esta forma, escuchará automáticamente el servidor http, a la espera de que pulse el botón del mando a distancia. a este ".bat" tuve que crearle un **acceso directo** con permisos de administrador ya que si no, no se copia dicho archivo en dicha ruta, de ahí que haya un archivo llamado **copia-tarea.lnk**

- **setup.bat:** Es el ".bat" que se ejecuta al auto-arrancar el pendrive, va llamando a cada archivo del pendrive en orden. Este llama dos veces a copia-tarea.lnk (que es copia-tarea.bat) ya que a veces se estanca. Contiene un timeout de 10 segundos para descomprimir bien el "ayuda_abuelo.exe" y 5 segundos para "python.exe".

NSIS

Es un "manejador de script" Windows de código abierto con requerimientos mínimos, desarrollado por Nullsoft. Este convierte un ".zip" en un **ejecutable** en el que puedes seleccionar donde descomprimirlo al ejecutarlo.

Primero coges el .zip con los documentos que vas a descomprimir, el nombre que le pones al ejecutable, la ruta y el nombre de la carpeta donde vas a descomprimirlo y donde vas a almacenar dicho ejecutable al crearse.



Zip2Exe converts a ZIP file to a self extracting archive (a simple NSIS installer). To customize the NSIS script code, modify the header files in the 'Contrib\zip2exe' folder.

Source ZIP File

Open... F:\Ayuda_abuelo.zip

Extracted: 10 files, 53059KB

- autorun.inf
- Ayuda_abuelo.py
- boot.bin
- copia-server.bat
- copia-tarea.bat

Output Installer Options

Installer Name Ayuda_abuelo

Interface ☒ Modern ☐ Classic ☒ Unicode

Default Folder C:\Program Files\ayuda_abuelo

Output EXE File H:\Little Kanéa\Documents\Gr. Informatica\proyecto\avance\ Browse...

Compression ☒ LZMA ☐ BZip2 ☐ ZLib ☐ Solid

Generate

MakeNSIS v3.08 - Copyright 1999-2021 Contributors
See the file COPYING for license details.
Credits can be found in the Users Manual.

Processing config: H:\Little Kanéa\Documents\Gr. Informatica\proyecto\nssr
Processing script file: "C:\Users\LITTLE~1\AppData\Local\Temp\zne9421.tmp"

Processed 1 file, writing output (x86-unicode):

Output: "H:\Little Kanéa\Documents\Gr. Informatica\proyecto\avance\web-avance\ayuda_abuelo.exe"
Install: 2 pages (128 bytes), 1 section (1 required) (2072 bytes), 67 instructions

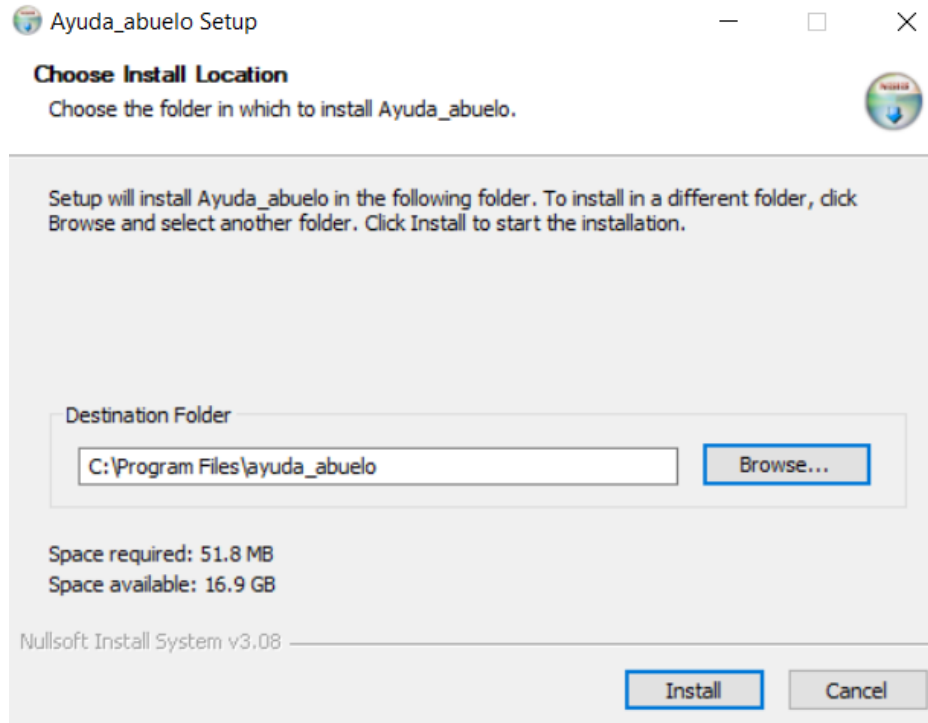
Using lzma compression.

EXE header size:	51200 / 38400 bytes
Install code:	1474 / 7538 bytes
Install data:	53956569 / 54327027 bytes
CRC (0x82884C70):	4 / 4 bytes

Total size: 54009247 / 54372969 bytes (99.3%)
(source ZIP size was 53550271 bytes)

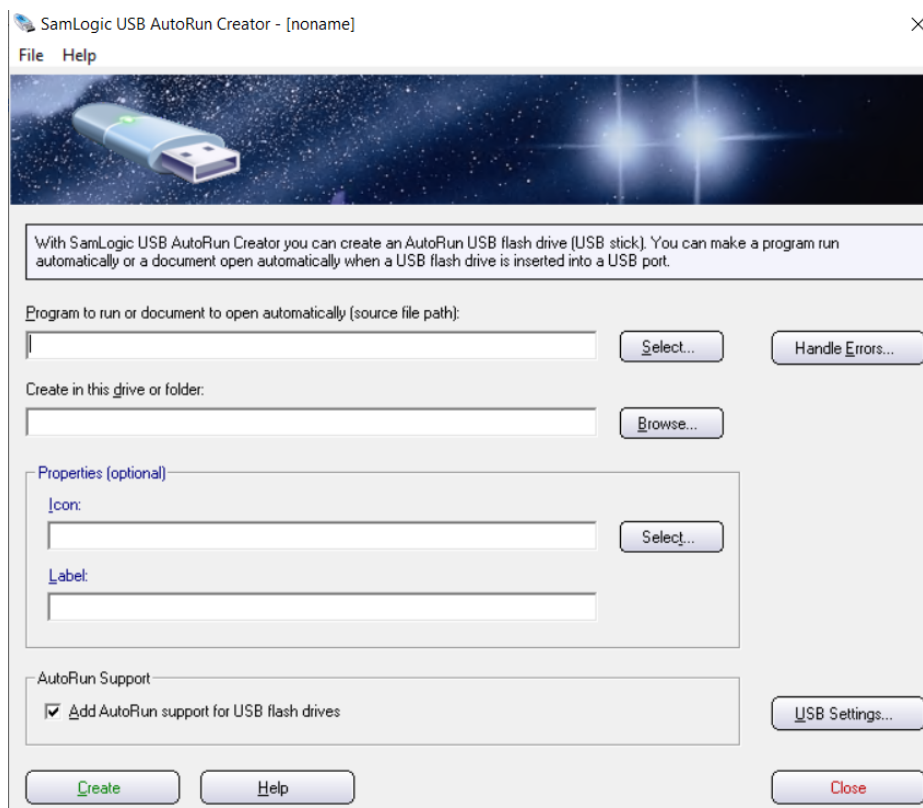
< Back Test Close

Al ejecutarse, se mostraría así y solo habría que darle a instalar:



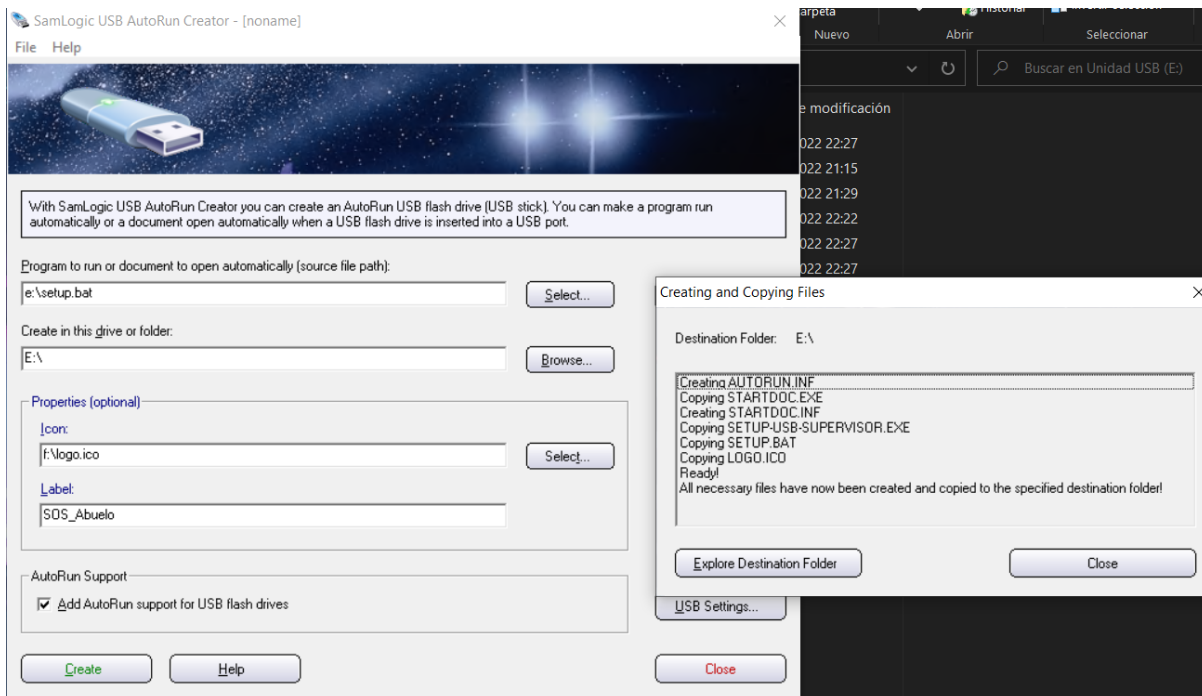
Tras tener todos los archivos necesarios, ahora sí vamos a configurar el pendrive para que auto-arranque:

SamLogic USB Autorun creator



En el **primer apartado** pones el fichero que va a arrancar el pendrive al detectarlo el pc, en este caso, **setup.bat** que inicia el script que va llamando a cada .exe y .bat con diferentes órdenes

En el **segundo apartado**, donde se creará el archivo autorun.inf y demás archivos que permiten el auto-arranque, aquí seleccionamos el pendrive. Lo demás es puramente opcional, aunque es aconsejable poner el "autorun support".

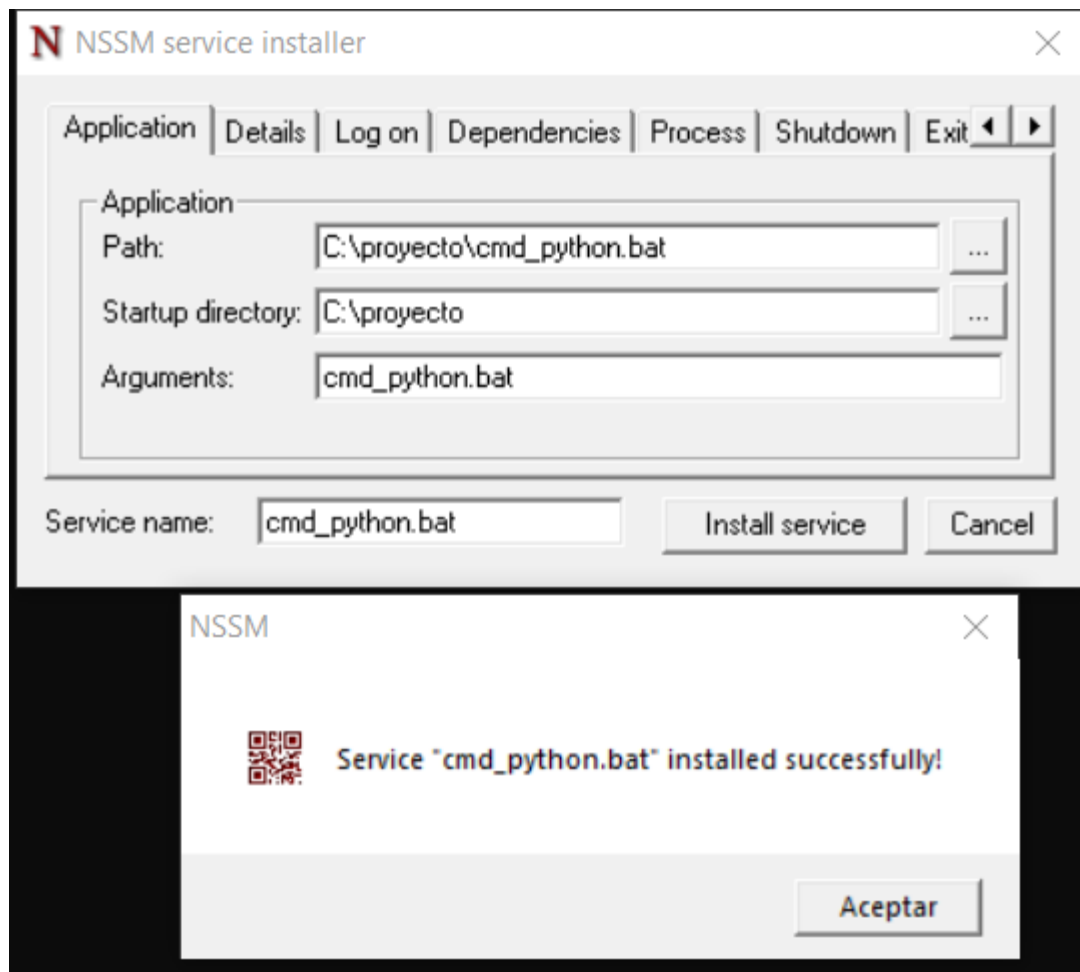
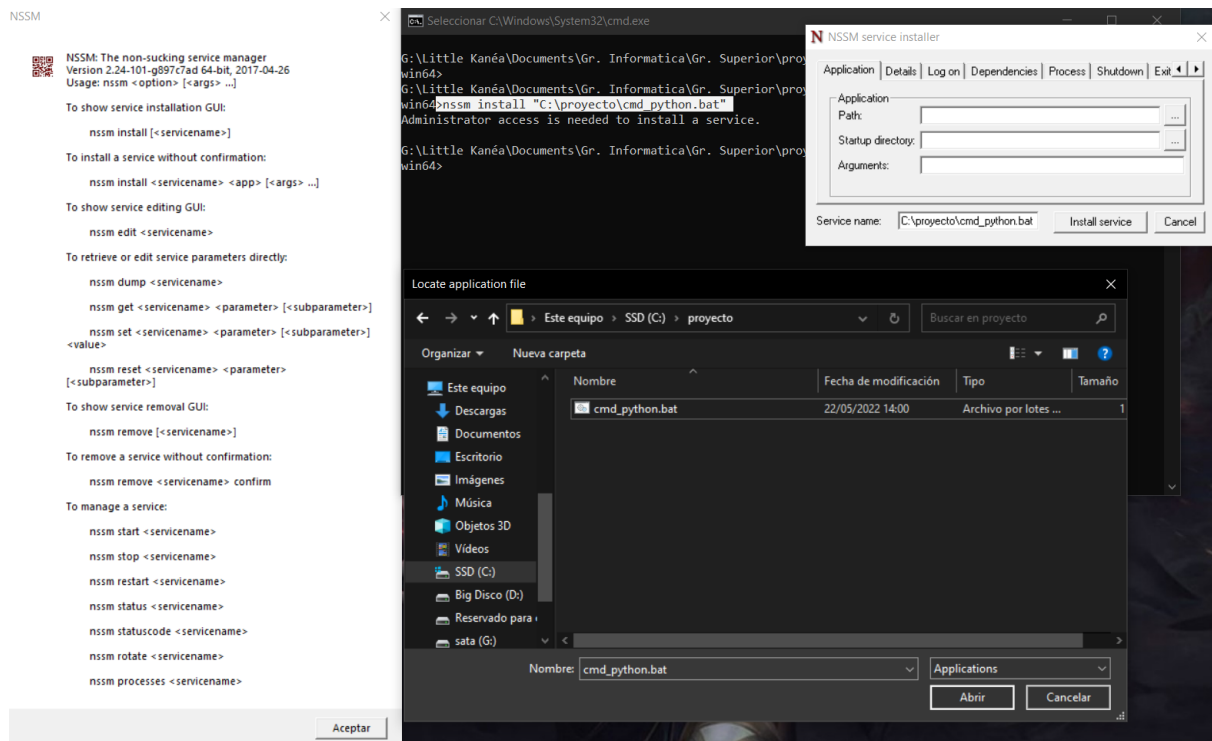


Dentro de este pendrive, he copiado todos los archivos necesarios de instalación, en el caso de que el "Ayuda_abuelo.exe" no se ejecutara correctamente.

Y así tendríamos creado el pendrive autoarrancable.

2.2. Problemas encontrados

- Había muchos lenguajes de programación que te daban la opción de crear el servidor http, pero el más sencillo y rápido era en python por lo que la primera traba era que yo no sabía python, así que tuve que dedicarle tiempo para entenderlo y montarlo.
- Otra problema fue que pensé solo en que arrancase dándole solamente desde el icono de escritorio, pero entonces sería igual que un acceso directo de una página desde el escritorio, así que, había que automatizar el arranque de ayuda_abuelo.py
- También pensé ponerlo como tarea programada y es viable realmente, incluso se puede poner la orden por script de batch, pero en algunos equipos no se crea bien la tarea o no se ejecuta, mientras que poniendo el archivo "Ayuda_abuelo.py" en *"%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup"* nunca me ha dado problemas, por eso me decanté por este. En otros equipos donde he probado también funciona en *"C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp"*
- Un día el puerto de USB del arduino Nodemcu decidió no funcionar, y me dejaron prestado un **wemos lolin32 esp32**, y lo hice funciona con su código para realizar la misma función que el Nodemcu pero sin botón, si no con el botón RST que trae la placa, ya que estaba en el punto del proyecto que no lo tenía conectado. Al final decidí comprar otro Nodemcu V2 que ya sabía donde colocar y cambiar el código para poner el botón.
- Los pendrive auto-arrancables, probé 2 programas (Rufus y ultraISO) y se han estropeado 2 pendrives sin conseguir lo que quería. Al final usando SamLogic USB y un tercer pendrive ya lo conseguí.
- Se pensó en ponerlo como servicio y lo intenté, pero nunca llegó a ejecutarse aunque lo reiniciara.



2.3. Modificaciones sobre el proyecto planteado inicialmente

- Al principio pensé en montar el servidor web dentro del arduino que enlazará con el de amazon pero claro, no había forma de automatizar la búsqueda o el arranque del dispositivo así que la mejor forma era montar el dispositivo y que, tras pulsar un botón, un programa que escuchara alguna petición.
- Había muchos lenguajes de programación que te daban la opción de crear el servidor http, pero el más sencillo y rápido era en python por lo que la primera traba era que yo no sabía python, así que tuve que dedicarle tiempo para entenderlo y montarlo.
- La creación de un pendrive autoarrancable surgió al ver que hacía falta instalar algunos archivos para su automatización.

2.4. Posibles mejoras al proyecto

- La página web evidentemente es una primera visualización a la que se le puede añadir un chat de ayuda, una zona de comentarios debajo de donde se visualizan los tutoriales, más tutoriales, videos orientativos, etc
- Al wemos lolin32 lite que me prestaron tiene bluetooth se le podría poner para tener conectividad con teléfonos móviles a través de una app que fuera un servidor http, haciendo la misma función que el de python.

2.5. Bibliografía

- ☐ <http://robots-argentina.com.ar/didactica/servidor-web-basico-nodemcu-con-ide-de-arduino/>
- ☐ <https://www.luisllamas.es/como-emplear-el-esp8266-como-cliente-http/>
- ☐ [Python Silent Installs – Lab Core | The Lab of MrNetTek \(eddiejackson.net\)](#)
- ☐ [Features - NSIS \(sourceforge.io\)](#)
- ☐ [How to create scheduled tasks with PowerShell on Windows 10 | Windows Central](#)
- ☐ [¿Cómo ocultar la ventana de la consola en python? - python, consola, hide \(living-sun.com\)](#)