

Instituto Politécnico Nacional  
Escuela Superior de Ingeniería  
Mecánica y Eléctrica - ESIME.  
Unidad Zacatenco  
Ingeniería en comunicaciones y  
electrónica.

# PWNAGOTCHI

Alumna: Perez Aranda Daniela Lisset

Profesor: Ing. Galicia Galicia Roberto  
Materia; Microprocesadores

No. Boleta: 2015120928  
Grupo: 6CM4

Para entender lo que es un "Pwnagotchi" y trabajar con este dispositivo, es útil tener conocimientos previos en los siguientes campos:

- a) **Redes y Wi-Fi:** Comprender los conceptos básicos de redes y cómo funcionan los protocolos Wi-Fi es esencial. Esto incluye entender los conceptos como SSID, encriptación, autenticación, y protocolos de seguridad como WPA2 o WPA3.
- b) **Linux:** El Pwnagotchi se basa en sistemas operativos Linux. Tener conocimientos básicos de cómo funciona Linux, la línea de comandos (bash), y la administración del sistema sería beneficioso.
- c) **Programación y Scripting:** Puede ser útil tener habilidades de programación, especialmente en lenguajes como Python, ya que el Pwnagotchi utiliza Python para su configuración y personalización.
- d) **Seguridad Informática:** Tener una comprensión básica de los principios de la seguridad informática, incluyendo conceptos como hacking ético, captura de paquetes, y seguridad de redes, proporcionará una base sólida.
- e) **Hardware y Raspberry Pi:** El Pwnagotchi a menudo se implementa en un dispositivo de hardware como una Raspberry Pi. Tener conocimientos básicos de hardware, electrónica y cómo trabajar con dispositivos como la Raspberry Pi sería útil.
- f) **Gestión de Energía y Hardware de Pantalla:** Dependiendo de la configuración específica de tu Pwnagotchi, podrías necesitar entender cómo gestionar la energía y trabajar con pantallas y otros periféricos conectados al hardware.

Es importante destacar que el uso de herramientas de seguridad como el Pwnagotchi debe realizarse de manera ética y legal. Además, ten en cuenta que este tipo de proyectos a menudo están en constante evolución, por lo que la documentación y las comunidades en línea pueden ser recursos valiosos para aprender y mantenerse actualizado.

### **Conceptos básicos:**

#### **a) Redes:**

**1. Red:** Una red es un conjunto de dispositivos interconectados que pueden comunicarse entre sí. Las redes pueden ser pequeñas, como una red doméstica, o grandes, como la Internet.

- 1. **Nodos:** Los nodos son los dispositivos individuales en una red, como computadoras, impresoras, enrutadores, etc.
- 2. **Topología de Red:** La topología se refiere a la disposición física o lógica de los nodos en una red. Ejemplos de topologías incluyen estrella, bus, anillo y malla.
- 3. **Protocolo de Red:** Un protocolo es un conjunto de reglas que rige la comunicación entre nodos en una red. El Protocolo de Internet (IP) es un ejemplo común.

4. **Dirección IP:** Cada dispositivo en una red tiene una dirección IP única que lo identifica. Las direcciones IP pueden ser IPv4 (por ejemplo, 192.168.1.1) o IPv6.
5. **Subred:** Una subred es un rango de direcciones IP dentro de una red más grande. Se utiliza para dividir una red en segmentos más pequeños.
6. **Enrutador:** Un enrutador es un dispositivo que conecta diferentes redes y dirige el tráfico entre ellas.
7. **Switch:** Un switch es un dispositivo que conecta varios nodos en una red local y facilita la comunicación entre ellos.
8. **Gateway:** Un gateway es un nodo que actúa como punto de entrada o salida para otra red. Puede ser un enrutador o un dispositivo similar.
9. **Firewall:** Un firewall es una barrera de seguridad que controla el tráfico entre redes y puede bloquear o permitir ciertos tipos de comunicación.
10. **Ancho de Banda:** El ancho de banda es la capacidad de una conexión para transportar datos. Se mide en bits por segundo (bps).
11. **Protocolo de Transferencia de Hipertexto (HTTP) y Protocolo de Transferencia de Hipertexto Seguro (HTTPS):** Estos protocolos son fundamentales para la navegación web. HTTP se utiliza para la transferencia de datos no cifrados, mientras que HTTPS cifra la comunicación para mayor seguridad.
12. **DNS (Sistema de Nombres de Dominio):** El DNS traduce nombres de dominio (como [www.ejemplo.com](http://www.ejemplo.com)) en direcciones IP.
13. **LAN (Red de Área Local) y WAN (Red de Área Amplia):** Una LAN es una red limitada a una ubicación geográfica específica, como una casa u oficina, mientras que una WAN abarca áreas geográficas más extensas, como una red de sucursales empresariales interconectadas.

### **Protocolos Wi-Fi:**

**Los protocolos Wi-Fi son conjuntos de reglas y especificaciones que definen cómo los dispositivos se comunican y transmiten datos a través de una red inalámbrica. Aquí se explica cómo funcionan los protocolos Wi-Fi de manera general:**

1. **Modulación y Frecuencias:** Los protocolos Wi-Fi operan en bandas de frecuencia de radio, generalmente en la banda de 2.4 GHz y/o 5 GHz. La modulación de la señal convierte los datos en señales de radio que pueden ser transmitidas por el aire.
2. **Estándares Wi-Fi:** Los estándares Wi-Fi son especificaciones que definen cómo los dispositivos inalámbricos se comunican entre sí. Algunos de los estándares más comunes incluyen IEEE 802.11a, 802.11b, 802.11g, 802.11n, 802.11ac y 802.11ax. Cada estándar tiene diferentes velocidades máximas de transferencia y características.
3. **SSID (Service Set Identifier):** El SSID es el nombre de la red Wi-Fi y se utiliza para identificarla. Los dispositivos deben estar configurados con el mismo SSID para conectarse a la misma red.

4. **Canal:** En una banda de frecuencia determinada, se dividen en canales para evitar interferencias. Los dispositivos Wi-Fi pueden operar en diferentes canales dentro de una banda.
5. **Autenticación y Encriptación:** Los protocolos Wi-Fi utilizan métodos de autenticación para permitir o denegar el acceso a la red. La encriptación se utiliza para proteger la privacidad de los datos transmitidos. Los métodos comunes incluyen WEP, WPA, y WPA2, con WPA3 como la versión más reciente y segura.
6. **Asociación y Autenticación de Dispositivos:** Antes de que un dispositivo pueda comunicarse con una red Wi-Fi, debe asociarse y autenticarse. Esto implica la verificación de la identidad del dispositivo y la generación de claves de cifrado para asegurar la comunicación.
7. **MAC (Media Access Control):** Cada dispositivo Wi-Fi tiene una dirección MAC única que se utiliza para identificarlo en la red. El control de acceso a los medios (MAC) regula cómo los dispositivos comparten el medio de transmisión.
8. **Handshake de Cuatro Vías (Four-Way Handshake):** Es un proceso de autenticación y generación de claves de cifrado entre el dispositivo cliente y el punto de acceso Wi-Fi.
9. **Roaming:** Permite que un dispositivo se mueva de un área de cobertura a otra sin perder la conexión. El roaming efectivo requiere una infraestructura de red adecuada y es especialmente importante en entornos empresariales.

Estos son algunos de los conceptos clave que explican cómo funcionan los protocolos Wi-Fi. Es importante notar que la tecnología Wi-Fi está en constante evolución, y nuevos estándares y mejoras continúan siendo desarrollados para proporcionar conexiones más rápidas y seguras.

#### b) Linux:

##### **Sistema Operativo Linux:**

- **Kernel:** El núcleo del sistema operativo, que gestiona los recursos del hardware y proporciona servicios básicos.
  - **Distribuciones:** Variantes específicas de Linux que incluyen el kernel de Linux y herramientas de sistema adicionales. Ejemplos comunes son Ubuntu, Fedora, Debian, y CentOS.
2. **Línea de Comandos (bash):**
    - **Terminal:** La interfaz de línea de comandos donde puedes ingresar comandos.
    - **Comandos Básicos:** Conocer comandos fundamentales como `cd` para cambiar directorios, `ls` para listar archivos, `cp` para copiar, `mv` para mover, `rm` para eliminar, etc.
    - **Permisos de Archivos:** Comprender y manejar los permisos de archivos y directorios con comandos como `chmod` y `chown`.
  3. **Administración del Sistema:**
    - **Usuarios y Grupos:** Crear, modificar y eliminar usuarios y grupos con comandos como `useradd`, `passwd`, `groupadd`, etc.

- **Gestión de Procesos:** Monitorear y gestionar procesos en ejecución con comandos como ps, top, kill, y pkill.
- **Instalación de Software:** Utilizar herramientas de gestión de paquetes como apt, yum, o dnf para instalar, actualizar y eliminar software.
- **Configuración de Red:** Configurar interfaces de red, asignar direcciones IP, y gestionar conexiones con comandos como ifconfig, ip, y netstat.
- **Logs del Sistema:** Revisar y analizar registros del sistema con comandos como journalctl y examinar archivos de registro en /var/log/.

#### 4. **Scripting con Bash:**

- Comprender la sintaxis básica de scripts de shell (bash).
- Crear scripts simples para automatizar tareas repetitivas

#### c) Scripting:

Scripting se refiere a la escritura de scripts, que son conjuntos de comandos o instrucciones que se ejecutan en un entorno específico. Los scripts suelen ser interpretados, lo que significa que se ejecutan línea por línea por un intérprete en lugar de ser compilados en código de máquina antes de la ejecución. Los scripts son utilizados para automatizar tareas y realizar acciones específicas en un entorno determinado. Mientras que la programación a menudo se asocia con la creación de aplicaciones completas, el scripting se centra en tareas más pequeñas y específicas. Ejemplos de lenguajes de scripting incluyen Python, JavaScript, Bash, PowerShell, entre otros.

#### d) Seguridad Informática:

La seguridad informática se centra en proteger sistemas, redes y datos contra amenazas, ataques y accesos no autorizados.

##### 1. **Hacking Ético:**

- El hacking ético implica el uso de habilidades y conocimientos de hacking para identificar y corregir vulnerabilidades en sistemas de manera ética. Los hackers éticos son contratados para evaluar la seguridad de sistemas y redes con el objetivo de fortalecerlas.

##### 2. **Captura de Paquetes:**

- La captura de paquetes es el proceso de interceptar y analizar los paquetes de datos que se transmiten a través de una red. Herramientas como Wireshark permiten examinar el tráfico de red para detectar posibles problemas de seguridad o analizar el rendimiento de la red.

##### 3. **Firewall:**

- Un firewall es una barrera de seguridad que controla y monitorea el tráfico entre una red privada y el mundo exterior. Puede bloquear o permitir ciertos tipos de tráfico según reglas predefinidas.

##### 4. **Ataques y Amenazas:**

- Comprender diferentes tipos de ataques (por ejemplo, phishing, malware, ataques de fuerza bruta) y amenazas (virus, gusanos, troyanos) que pueden afectar la seguridad de sistemas y redes.
5. **Cifrado y Criptografía:**
- El cifrado se utiliza para proteger la confidencialidad de la información. La criptografía es la ciencia de codificar y decodificar información de manera segura. Protocolos como SSL/TLS en la web utilizan cifrado para asegurar la comunicación.
6. **Autenticación y Autorización:**
- La autenticación verifica la identidad de un usuario, mientras que la autorización determina los privilegios y el acceso que tiene un usuario una vez autenticado. Métodos como contraseñas, tokens y biometría son comunes en la autenticación.
7. **Seguridad de Redes:**
- Incluye medidas para proteger la integridad, confidencialidad y disponibilidad de los datos en una red. Esto puede involucrar configurar cortafuegos, implementar políticas de seguridad y realizar auditorías regulares.
8. **Parches y Actualizaciones:**
- Mantener el software actualizado mediante la aplicación de parches y actualizaciones es crucial para corregir vulnerabilidades conocidas y mejorar la seguridad.
9. **Políticas de Seguridad:**
- Desarrollar y aplicar políticas de seguridad que establezcan reglas y procedimientos para proteger la información y los recursos de la organización.

#### e) Hardware y Raspberry Pi:

##### Hardware:

- **CPU, RAM y Almacenamiento:** Comprender los conceptos básicos de la CPU (Unidad Central de Procesamiento), la memoria RAM y el almacenamiento, ya que son componentes fundamentales de cualquier sistema computacional.
  - **Placa Base:** Conocer los componentes principales de una placa base y cómo se conectan entre sí.
  - **Conectividad:** Comprender puertos y conectores comunes, como USB, HDMI, Ethernet, etc.
  - **Fuente de Alimentación:** Saber cómo alimentar y suministrar energía a los dispositivos de forma segura y eficiente.
2. **Electrónica:**
- **Circuitos y Componentes Electrónicos:** Entender cómo funcionan los circuitos eléctricos básicos y los componentes como resistencias, condensadores y transistores.
  - **Diodos y LED:** Conocer el funcionamiento de diodos y LED, componentes comunes en electrónica.

- **Sensores y Actuadores:** Familiarizarse con sensores que capturan datos del entorno y actuadores que realizan acciones en respuesta a señales.
- 3. **Trabajar con la Raspberry Pi:**
  - **La Raspberry Pi:** Conocer la Raspberry Pi como una computadora de placa única (SBC) y su versatilidad para proyectos embebidos.
  - **Sistema Operativo:** Instalar y trabajar con sistemas operativos en la Raspberry Pi, como Raspbian (basado en Linux).
  - **Puertos GPIO (General Purpose Input/Output):** Utilizar los pines GPIO para interactuar con componentes electrónicos y construir proyectos.
  - **Accesorios y Módulos:** Conectar y utilizar accesorios como cámaras, pantallas y módulos adicionales.
- 4. **Programación para Hardware:**
  - **Lenguajes de Programación:** Familiarizarse con lenguajes de programación utilizados para programar hardware, como Python, C o Java.
  - **APIs y Librerías:** Utilizar APIs y librerías específicas de hardware para facilitar la programación.
- 5. **Seguridad Eléctrica y Estática:**
  - **Descarga Electroestática (ESD):** Conocer las precauciones necesarias para evitar daños por descarga electroestática.
  - **Seguridad Eléctrica:** Entender la importancia de seguir buenas prácticas de seguridad al trabajar con dispositivos alimentados eléctricamente.
- 6. **Ensamblaje y Desmontaje:**
  - **Montaje y Desmontaje:** Saber cómo ensamblar y desmontar componentes de hardware de manera segura y adecuada.

f) **Gestión de Energía y Hardware de Pantalla:**

Gestionar la energía y trabajar con pantallas y otros periféricos conectados al hardware es esencial cuando trabajas con dispositivos embebidos como la Raspberry Pi. Aquí hay algunos conceptos clave:

1. **Gestión de Energía:**
  - **Fuente de Alimentación:**
    - Utilizar una fuente de alimentación adecuada para la Raspberry Pi.
    - Asegurarse de que la fuente de alimentación proporcione la corriente y voltaje correctos.
  - **Consumo de Energía:**
    - Conocer el consumo de energía de la Raspberry Pi y de los periféricos conectados.
    - Monitorear y optimizar el consumo de energía cuando sea necesario.
  - **Ahorro de Energía:**



- Configurar la Raspberry Pi para aprovechar opciones de ahorro de energía cuando no esté en uso.
- Apagar o desconectar periféricos no utilizados para reducir el consumo de energía.

## 2. Pantallas y Periféricos:

- **Conexión de Pantallas:**
  - Conectar pantallas correctamente a través de puertos como HDMI o GPIO (para pantallas específicas de Raspberry Pi).
  - Configurar la resolución y la orientación de la pantalla según sea necesario.
- **Periféricos USB:**
  - Conectar y configurar periféricos USB, como teclados, ratones y cámaras.
  - Asegurarse de que la Raspberry Pi tenga suficientes puertos USB y suministro de energía para los periféricos conectados.
- **Sensores y Actuadores:**
  - Conectar y programar sensores y actuadores según las necesidades del proyecto.
  - Asegurarse de que los pines GPIO y los recursos de la Raspberry Pi sean suficientes para los periféricos.

## 3. Control de Energía por Software:

- **Ajuste de Brillo de Pantalla:**
  - Configurar el brillo de la pantalla para ahorrar energía.
  - Utilizar software para ajustar automáticamente el brillo según las condiciones ambientales.
- **Suspensión y Reinicio:**
  - Utilizar comandos y configuraciones para poner la Raspberry Pi en modo de suspensión o reinicio cuando sea necesario.
  - Implementar lógica de suspensión o reinicio en scripts o programas según las condiciones del proyecto.
- **Temporizadores y Planificación:**
  - Programar tareas y procesos para ejecutarse en momentos específicos para aprovechar eficientemente la energía.

## 4. Monitoreo de Energía:

- **Herramientas de Monitoreo:**
  - Utilizar herramientas de monitoreo de energía para evaluar el consumo y la eficiencia energética.
  - Monitorear el rendimiento de la Raspberry Pi y ajustar la configuración según sea necesario.

## 5. Backup de Energía:

- **Baterías y UPS:**
  - Considerar el uso de baterías o sistemas de alimentación ininterrumpida (UPS) para garantizar un suministro de energía estable durante cortes de energía.



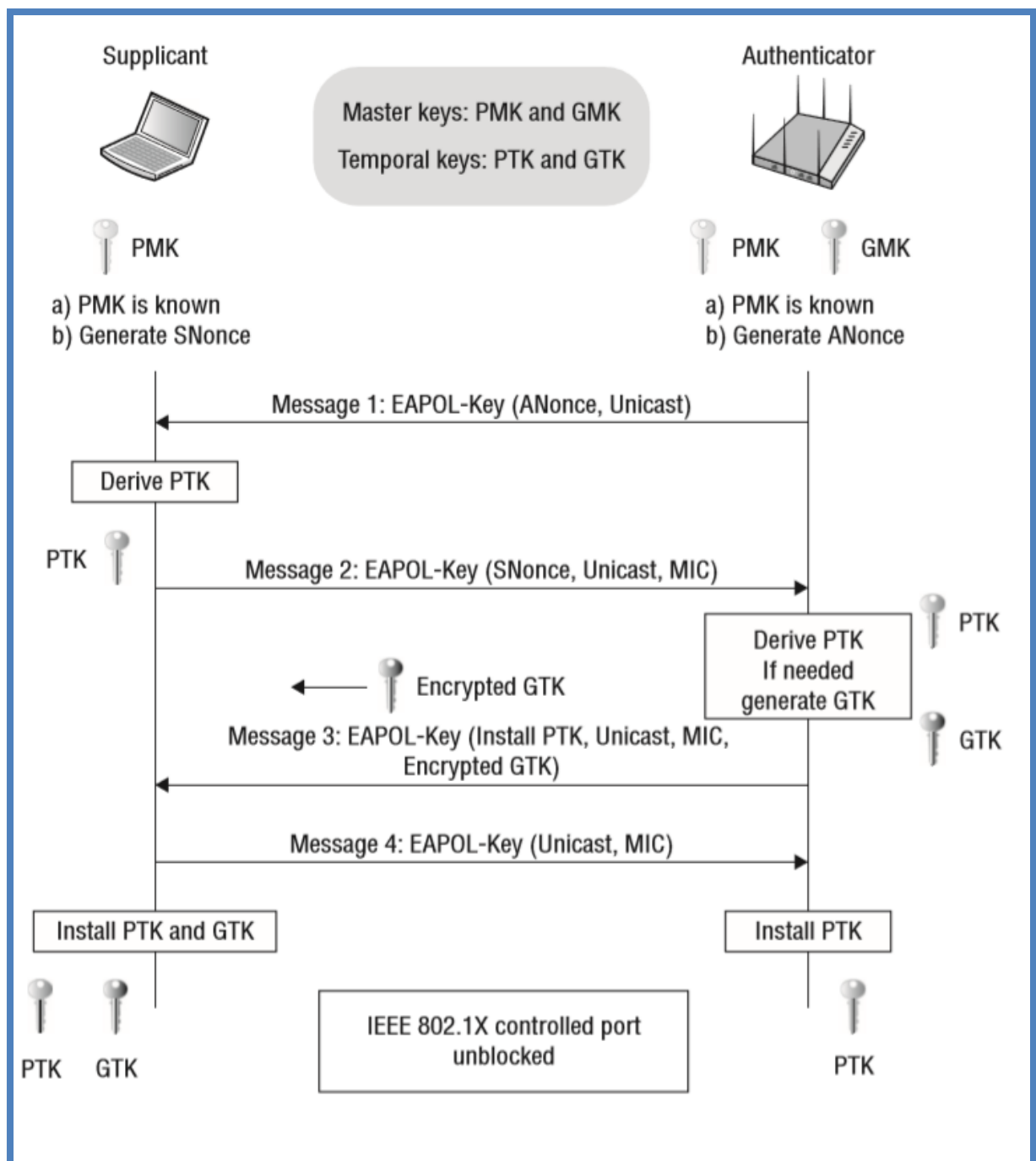
Un Pwnagotchi es un proyecto de hacking ético y de ciberseguridad que combina hardware y software para crear un dispositivo autónomo diseñado para mejorar las habilidades y conocimientos en seguridad informática, especialmente en el ámbito de redes inalámbricas. La palabra "Pwnagotchi" es una combinación de "pwn" (una jerga de la cultura hacker que significa "poseer" o "dominar") y "Tamagotchi" (un popular juguete virtual de los años 90).

Una breve explicación de lo que hace y con qué funciona:

1. **Captura de Handshakes WPA/WPA2:** Un Pwnagotchi puede estar equipado con una interfaz de red inalámbrica que le permite capturar handshakes, que son intercambios de mensajes necesarios para realizar ataques de fuerza bruta en redes Wi-Fi protegidas con WPA/WPA2.
2. **Aprendizaje automático:** El Pwnagotchi utiliza técnicas de aprendizaje automático para adaptarse a su entorno. Aprende patrones de comportamiento en las redes Wi-Fi a su alrededor y ajusta sus estrategias de ataque en consecuencia. Esto lo hace más efectivo y autónomo con el tiempo.
3. **Colaboración con otros Pwnagotchis:** Los Pwnagotchis pueden colaborar entre sí. Comparten información sobre redes Wi-Fi, estrategias de ataque y otros datos relevantes, lo que mejora la eficiencia y la efectividad del grupo.
4. **Visualización de información:** Muchos Pwnagotchis tienen una pantalla pequeña donde muestran información útil, como las redes Wi-Fi cercanas, el estado del aprendizaje automático, y más. Esto permite a los usuarios interactuar y obtener información sobre la actividad del dispositivo.
5. **Funciona con baterías y Raspberry Pi:** El Pwnagotchi típicamente se construye utilizando una Raspberry Pi (un miniordenador de una sola placa) y una batería para hacerlo portátil y autónomo. Puede llevarse en un bolsillo y ejecutarse sin necesidad de una conexión directa a una computadora.

Es importante destacar que el uso de un Pwnagotchi debe ser ético y legal. Los usuarios deben respetar las leyes y regulaciones locales y utilizar este tipo de herramientas exclusivamente con fines educativos, de investigación y de seguridad informática autorizados. Utilizar un Pwnagotchi para acceder a redes Wi-Fi sin autorización es ilegal y va en contra de las normativas de ética en hacking.

Antes de que un dispositivo cliente que se conecta a un punto de acceso inalámbrico (por ejemplo, su teléfono que se conecta a la red WiFi de su hogar) pueda transmitir y recibir datos de forma segura desde ese punto de acceso, es necesario que se produzca un proceso llamado protocolo de enlace de 4 vías para que el cifrado WPA



Una vez teniendo más contexto de nuestro proyecto podremos avanzar a la acción, para esto necesitaremos los siguientes materiales:

### Hardware requerido

- ♣ Cuerpo
- ♣ Tarjeta SD
- ♣ Batería (opcional)
- ♣ Complemento de indicador de nivel de batería UPS-Lite
- ♣ Reloj de hardware
- ♣ Pantalla

- ♣ Pantallas de tinta electrónica en color frente a en blanco y negro
- ♣ Recomendaciones
- ♣ Carcasa (opcional)
- o Flasheo de una imagen
- o Instalación en cualquier GNU/Linux
- ♣ 1. Bettercap
- ♣ 2. pwngrid
- ♣ 3. Pwnagotchi

A continuación, se presentará una detallada descripción de su instalación, en caso de dudas visite los sitios oficiales:

#### Official Links

Website: pwnagotchi.ai (<https://pwnagotchi.ai/>)

Forum: community.pwnagotchi.ai (<https://community.pwnagotchi.ai/>)

Slack: pwnagotchi.slack.com (<https://invite.pwnagotchi.ai/>)

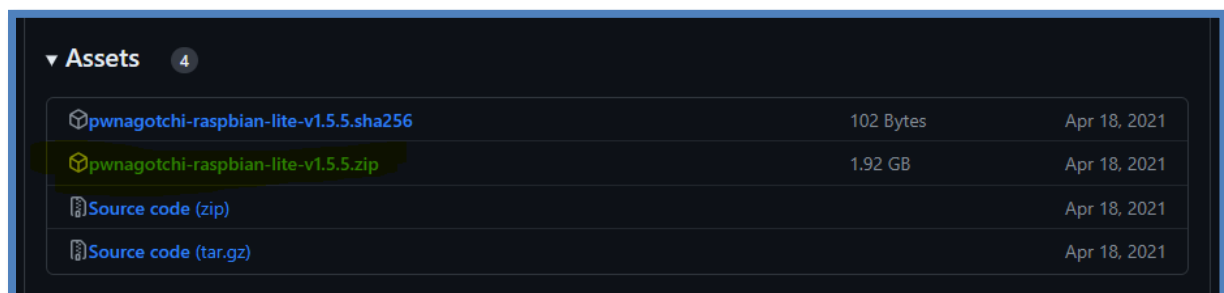
Subreddit: r/pwnagotchi (<https://www.reddit.com/r/pwnagotchi/>)





Twitter: @pwnagotchi (<https://twitter.com/pwnagotchi>)

Con lo anterior listo procedemos a grabar la microSD, en este caso se usará una de 32 GB, pero es importante aclarar que para dicho proyecto basta con una memoria de 8GB, así como lo indica su sitio web.

El primer paso es descargar el archivo .zip donde vendrá nuestro programa, para eso ingresamos a la página: [Releases · evilsocket/pwnagotchi \(github.com\)](https://github.com/evilsocket/pwnagotchi/releases)

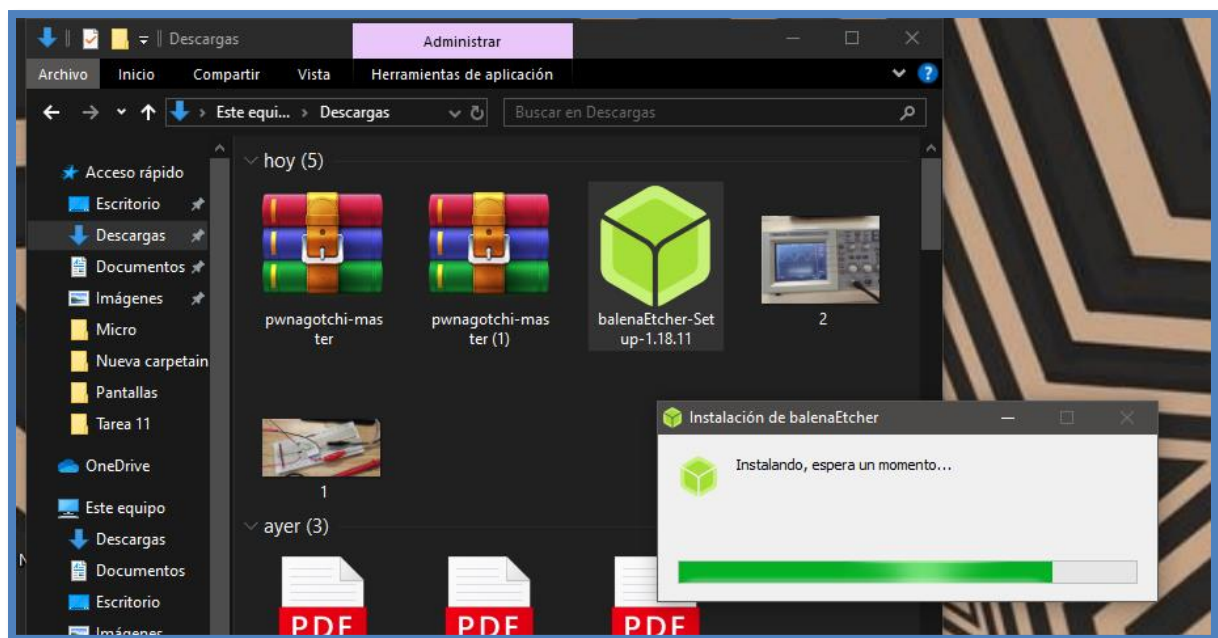
Específicamente buscaremos este archivo y lo descargamos:



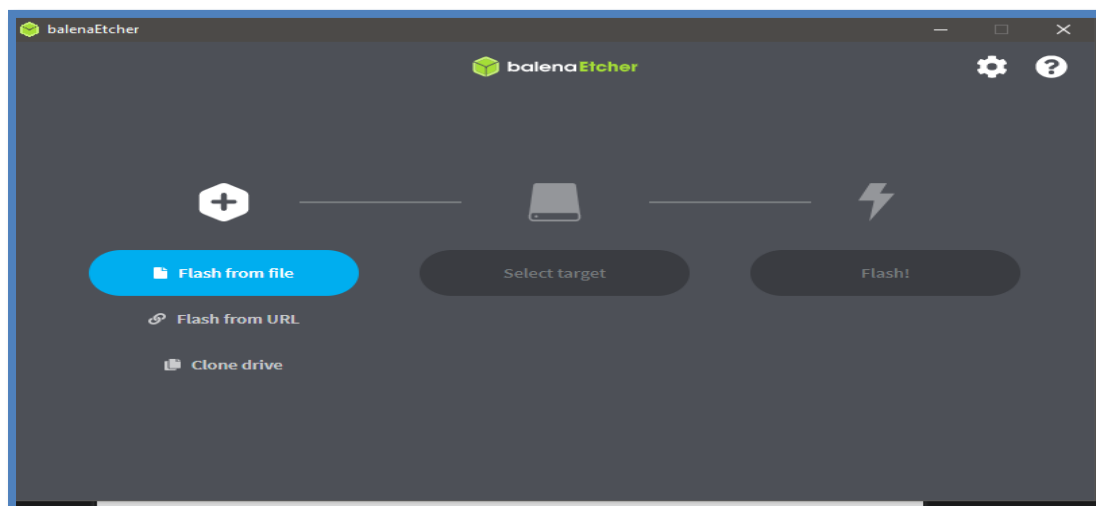
▼ Assets 4		
 pwnagotchi-raspbian-lite-v1.5.5.sha256	102 Bytes	Apr 18, 2021
 pwnagotchi-raspbian-lite-v1.5.5.zip	1.92 GB	Apr 18, 2021
 Source code (zip)		Apr 18, 2021
 Source code (tar.gz)		Apr 18, 2021

Una vez teniendo el archivo, lo que sigue es descargar el programa “balenaEtcher” de la página <https://etcher.balena.io>

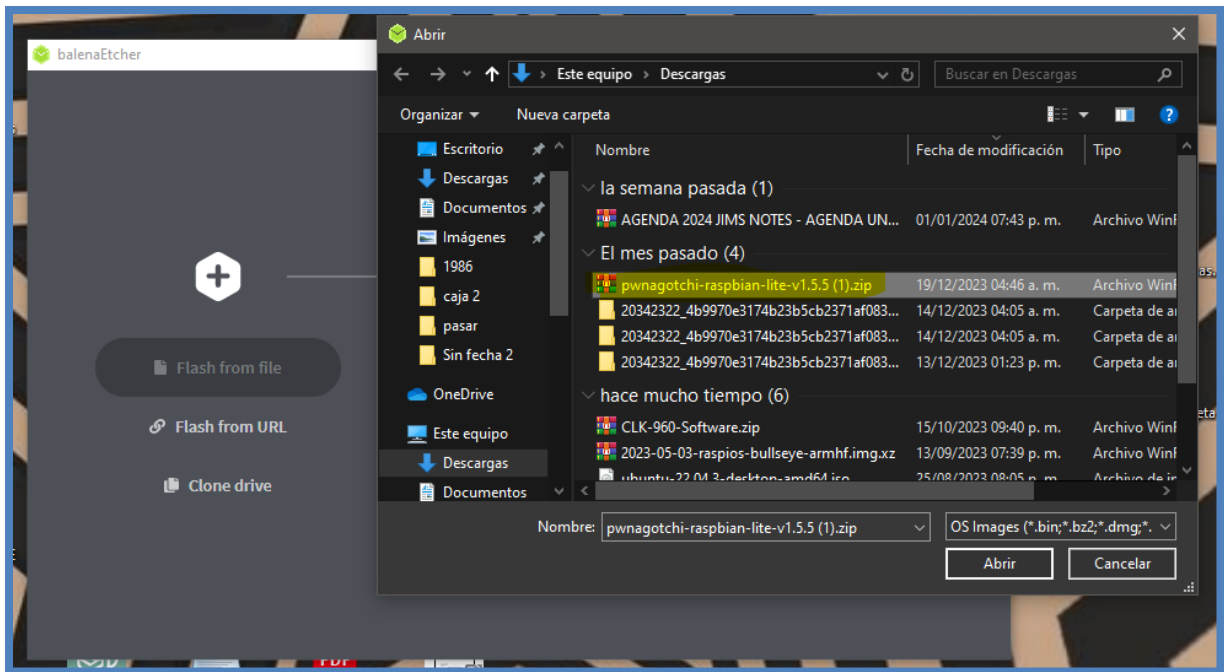
Ya descargado, se instala



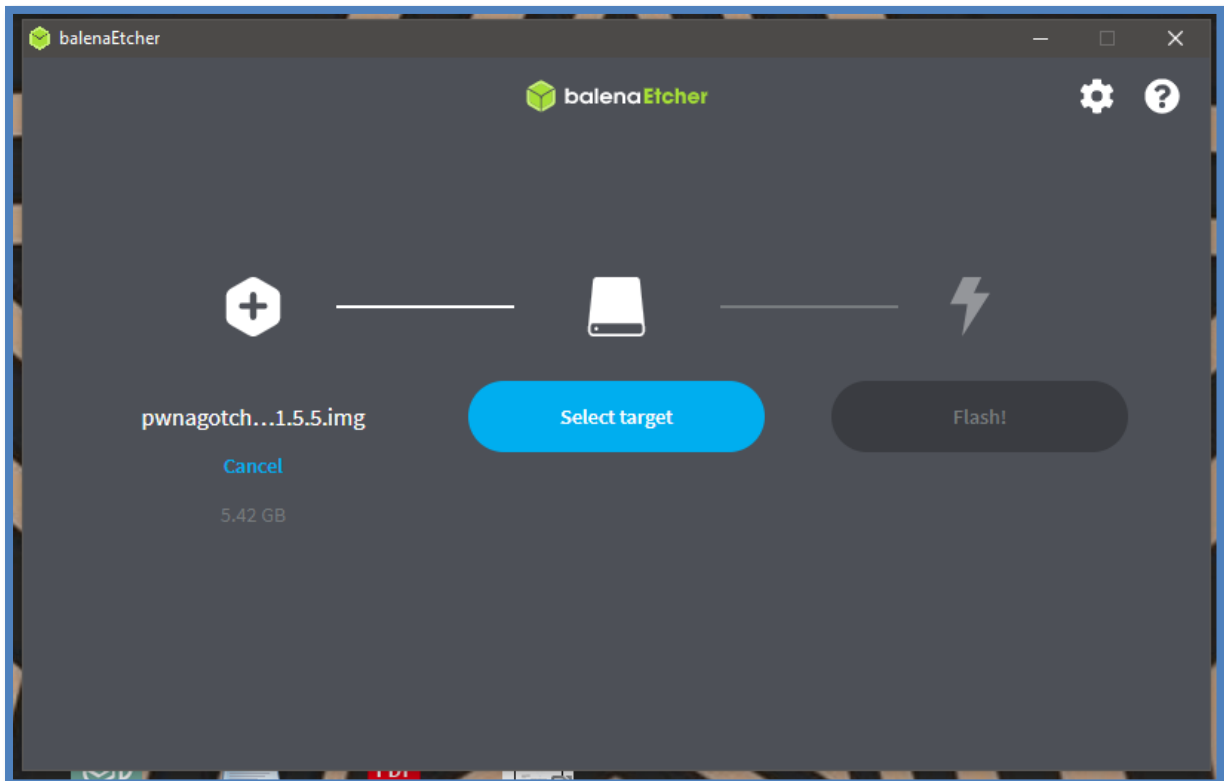
Una vez ya descargado procederemos a grabar la memoria como se indica a continuación:



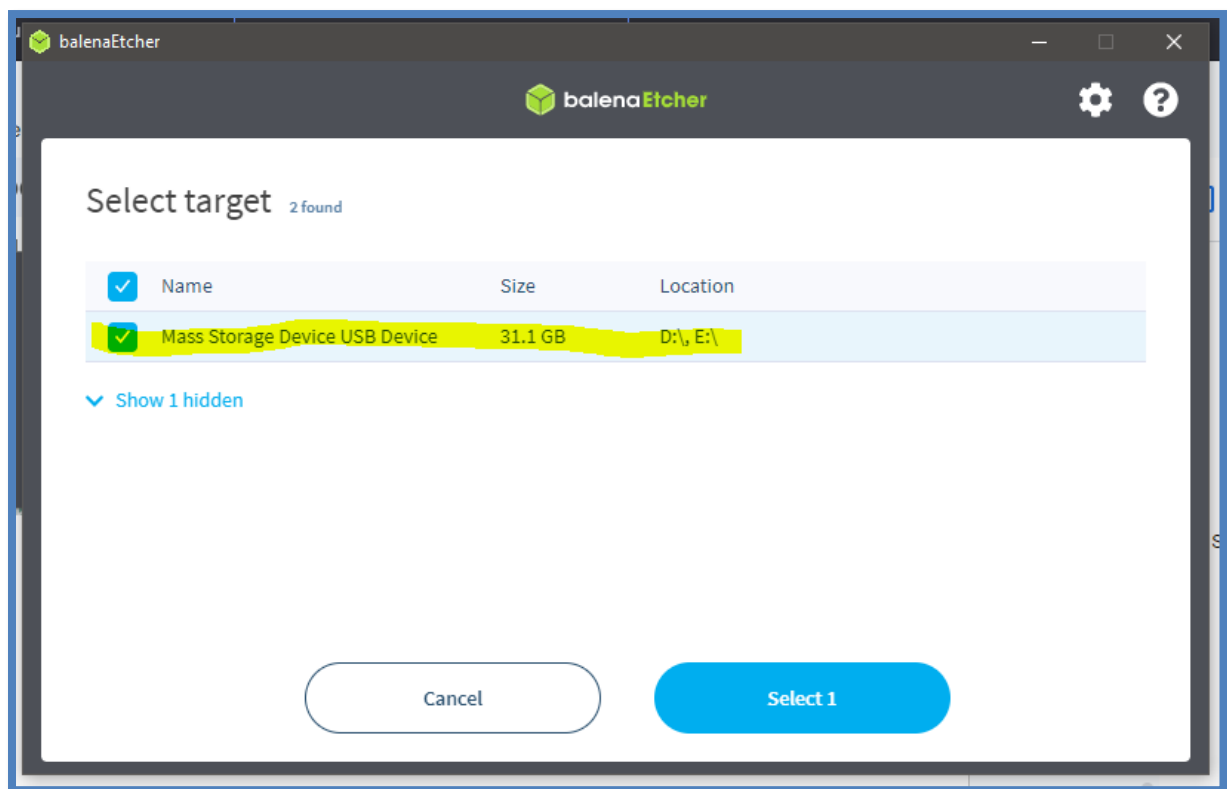
Seleccionamos **“Flash from file”** y el archivo que descargamos en el paso anterior



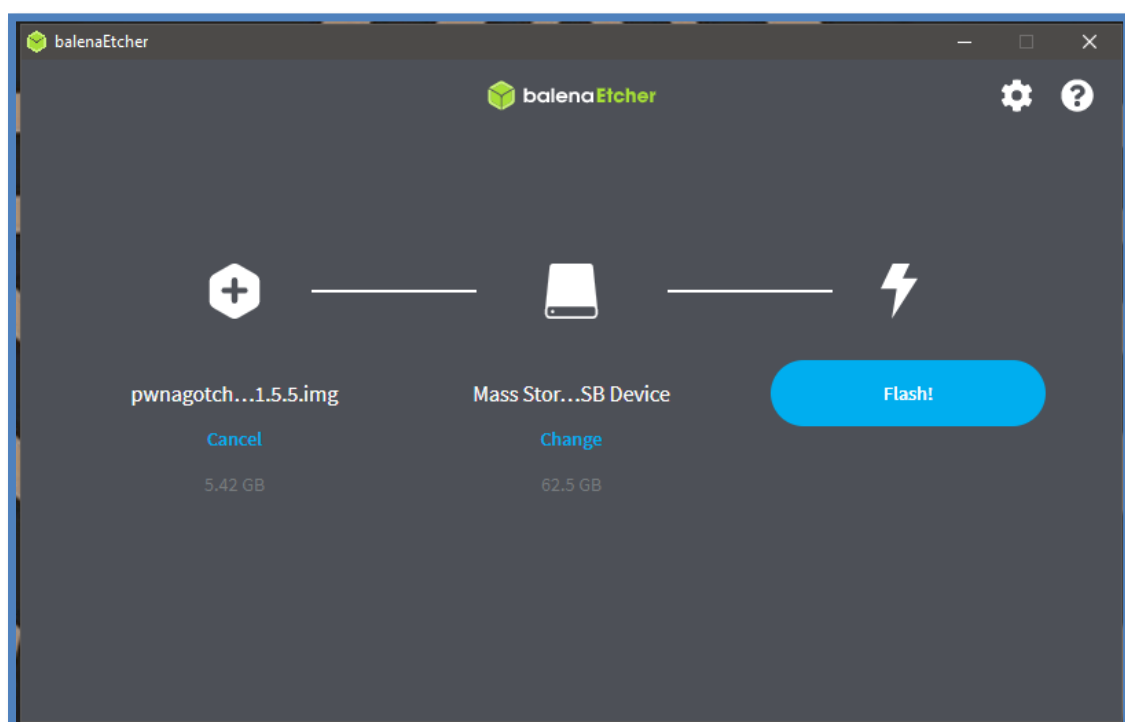
y damos aceptar  
Ahora seleccionamos **“Select target”**



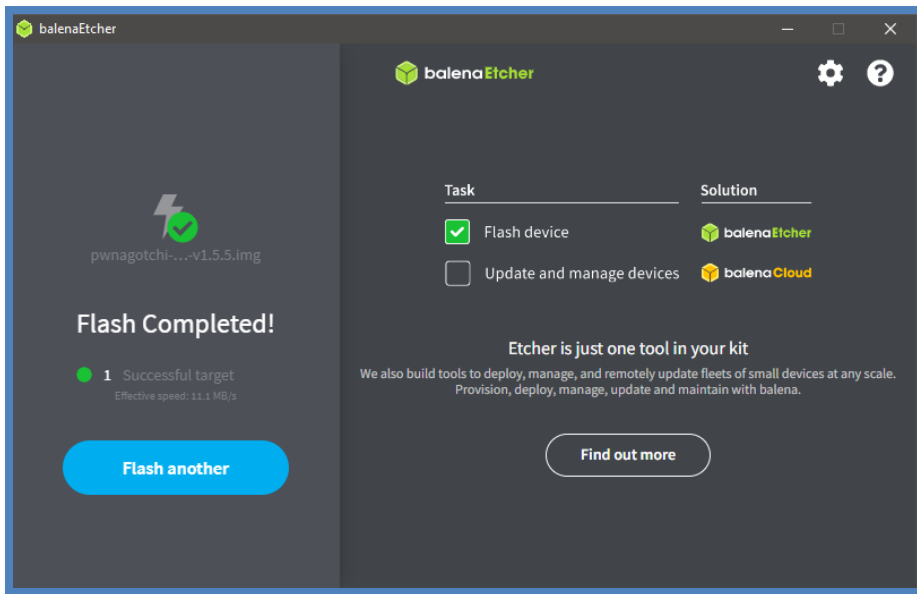
Y seleccionamos nuestra microSD y **“Select”**



Y procedemos a grabarla dando en “Flash!”



Este proceso puede tardar entre 15 a 20 minutos, hay que ser pacientes



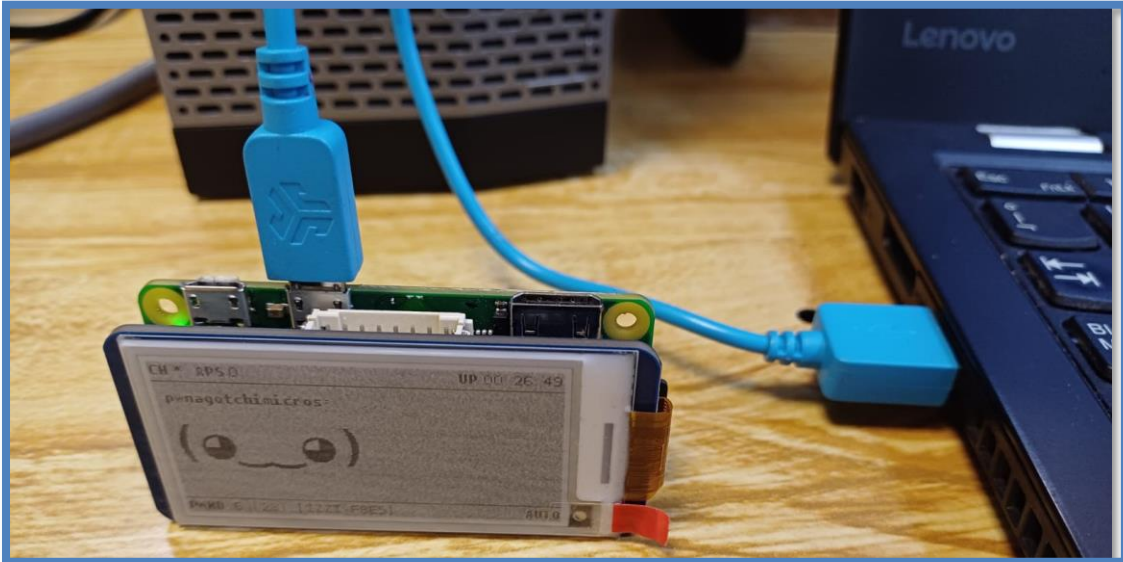
Puede que después de este proceso, te pide formatear la USB, **pero no debe formatearse.**

Una vez hecho esto podremos conectar nuestra Raspberry Pi de la siguiente forma:

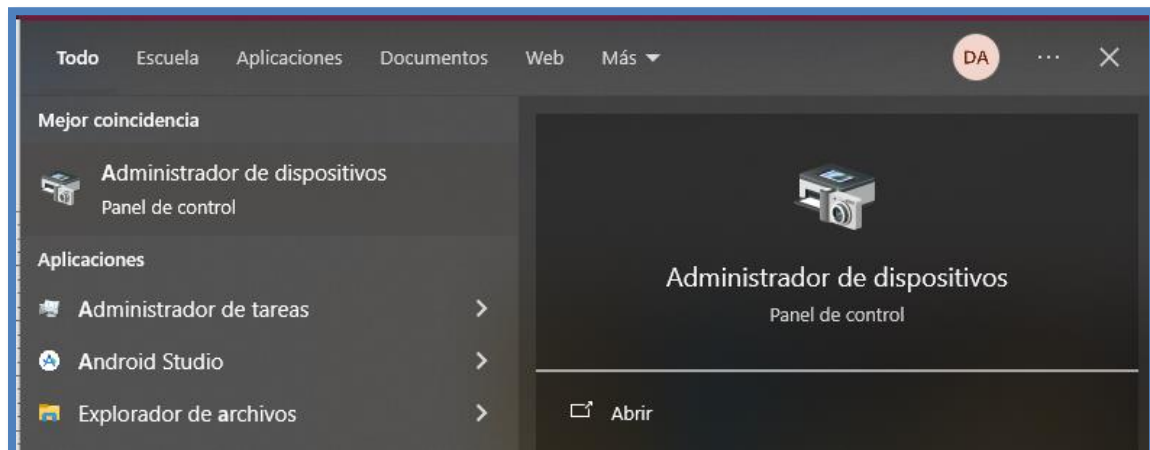




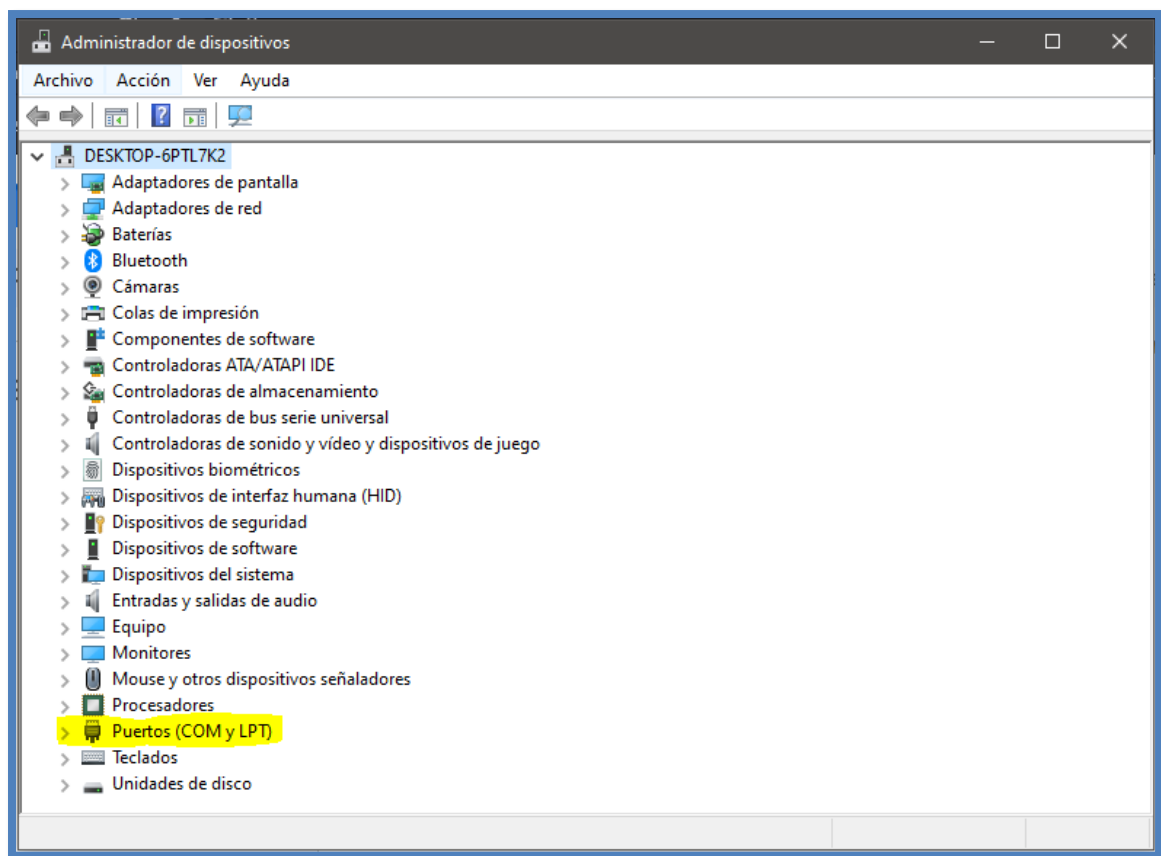
De las dos entradas, seleccionamos la segunda, ya que esta es específicamente para la transferencia de datos, y conectamos a la computadora.



Ahora vamos a **“Administrador de archivos”**



Una vez abierto, buscamos **“Puertos (COM y LPT)”**



y buscamos nuestro dispositivo como **“Dispositivo serie USB (COM3)”**



Esto puede tardar un momento, en lo que se actualizan los puertos, procederemos a descargar los controladores.

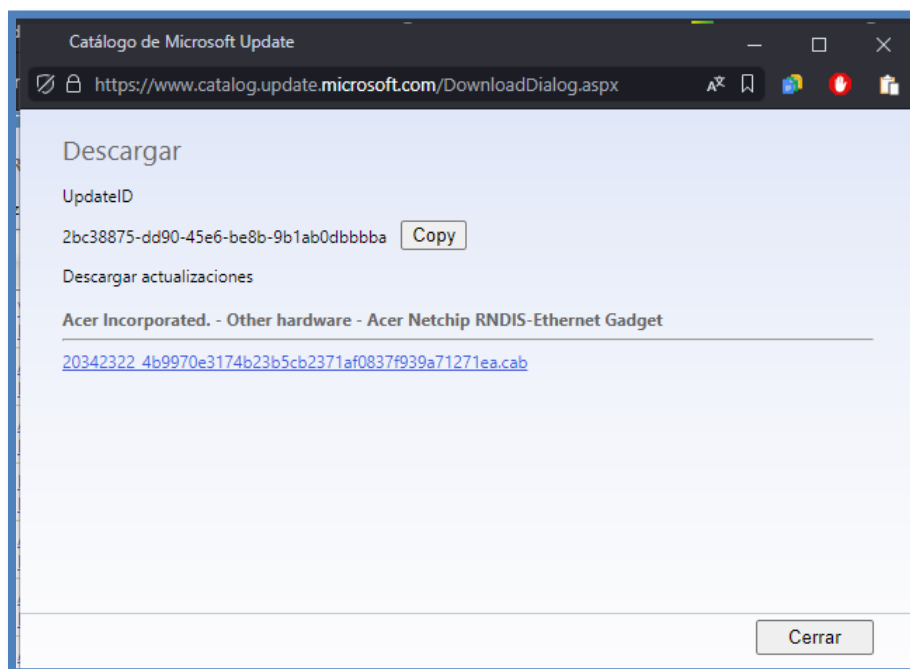
Para esto vamos a la página:

<https://www.catalog.update.microsoft.com/Search.aspx?q=USB+RNDIS%20Gadget>

Y damos en “descargar” en la penúltima opción:

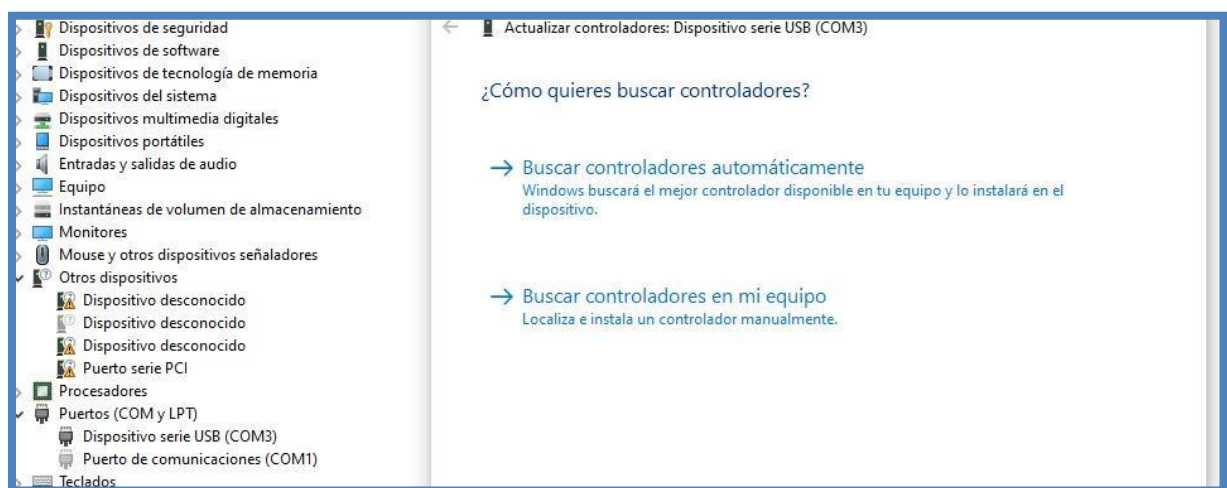
Título	Productos	Clasificación	Última actualización	Versión	Tamaño	Descargar
<a href="#">Vernier Software &amp; Technology - Other hardware - LabQuest2 USB Ethernet/RNDIS Gadget</a>	Windows XP, Windows Vista, Windows 7, Windows 8, Windows 8.1 and later drivers , Windows 10 Creators Update and Later Servicing Drivers	Drivers (Other Hardware)	11/02/2013	n/d	21 KB	<a href="#">Descargar</a>
<a href="#">Acer Incorporated - Other hardware - Acer USB Ethernet/RNDIS Gadget</a>	Windows XP, Windows Vista, Windows 7, Windows 8, Windows 8.1 and later drivers	Drivers (Other Hardware)	30/05/2010	n/d	104 KB	<a href="#">Descargar</a>
<a href="#">Acer Incorporated - Other hardware - Acer USB Ethernet/RNDIS Gadget</a>	Windows XP, Windows Vista, Windows 7, Windows 8, Windows 8.1 and later drivers	Drivers (Other Hardware)	30/05/2010	n/d	103 KB	<a href="#">Descargar</a>
<a href="#">INFOMARK Co. Ltd. - Other hardware - INFOMARK USB Ethernet/RNDIS Gadget</a>	Windows 8, Windows 8.1 and later drivers	Drivers (Other Hardware)	29/03/2010	n/d	22 KB	<a href="#">Descargar</a>
<a href="#">Acer Incorporated - Other hardware - USB Ethernet/RNDIS Gadget</a>	Windows XP	Drivers (Other Hardware)	12/01/2010	n/d	11 KB	<a href="#">Descargar</a>
<a href="#">Acer Incorporated - Other hardware - Acer Netchip RNDIS/Ethernet Gadget</a>	Windows Vista	Drivers (Other Hardware)	12/01/2010	n/d	11 KB	<a href="#">Descargar</a>
<a href="#">Acer Incorporated - Other hardware - Acer Netchip RNDIS/Ethernet Gadget</a>	Windows XP	Drivers (Other Hardware)	12/01/2010	n/d	11 KB	<a href="#">Descargar</a>
<a href="#">Acer Incorporated - Other hardware - Acer Netchip RNDIS/Ethernet Gadget</a>	Windows 7, Windows 8, Windows 8.1 and later drivers	Drivers (Other Hardware)	12/01/2010	n/d	21 KB	<a href="#">Descargar</a>
<a href="#">Acer Incorporated - Other hardware - USB Ethernet/RNDIS Gadget</a>	Windows 7, Windows 8, Windows 8.1 and later drivers	Drivers (Other Hardware)	12/01/2010	n/d	21 KB	<a href="#">Descargar</a>

Esto nos abrirá una ventana emergente:

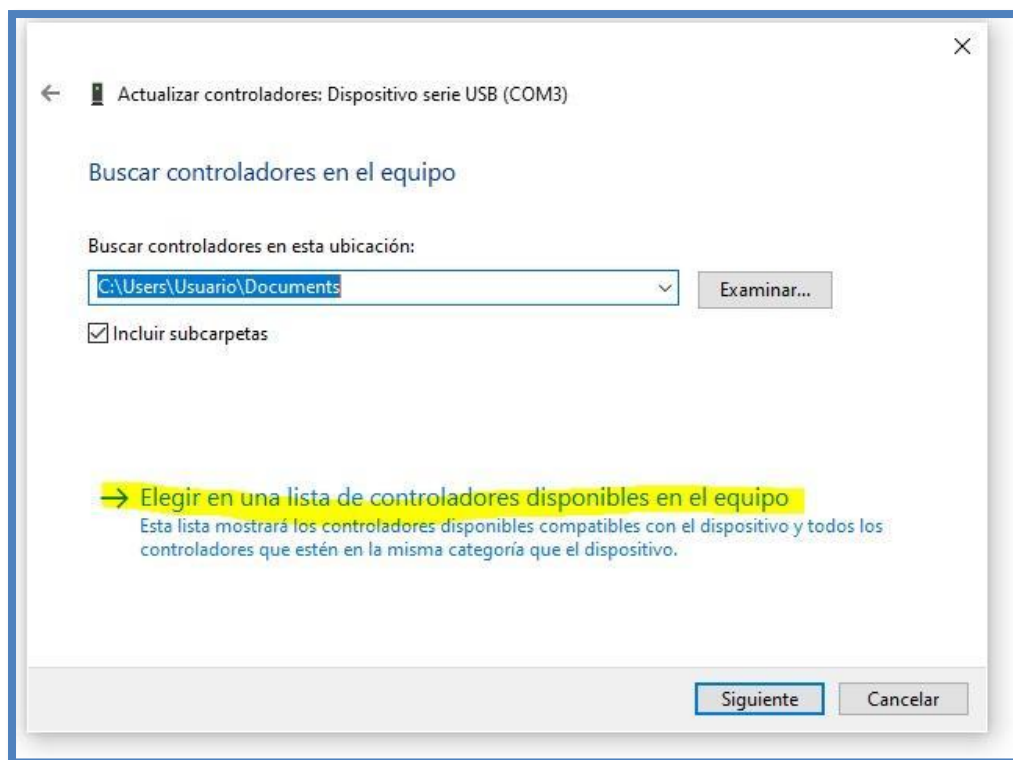


Le damos click al link en azul y se guardará el archivo que necesitamos.

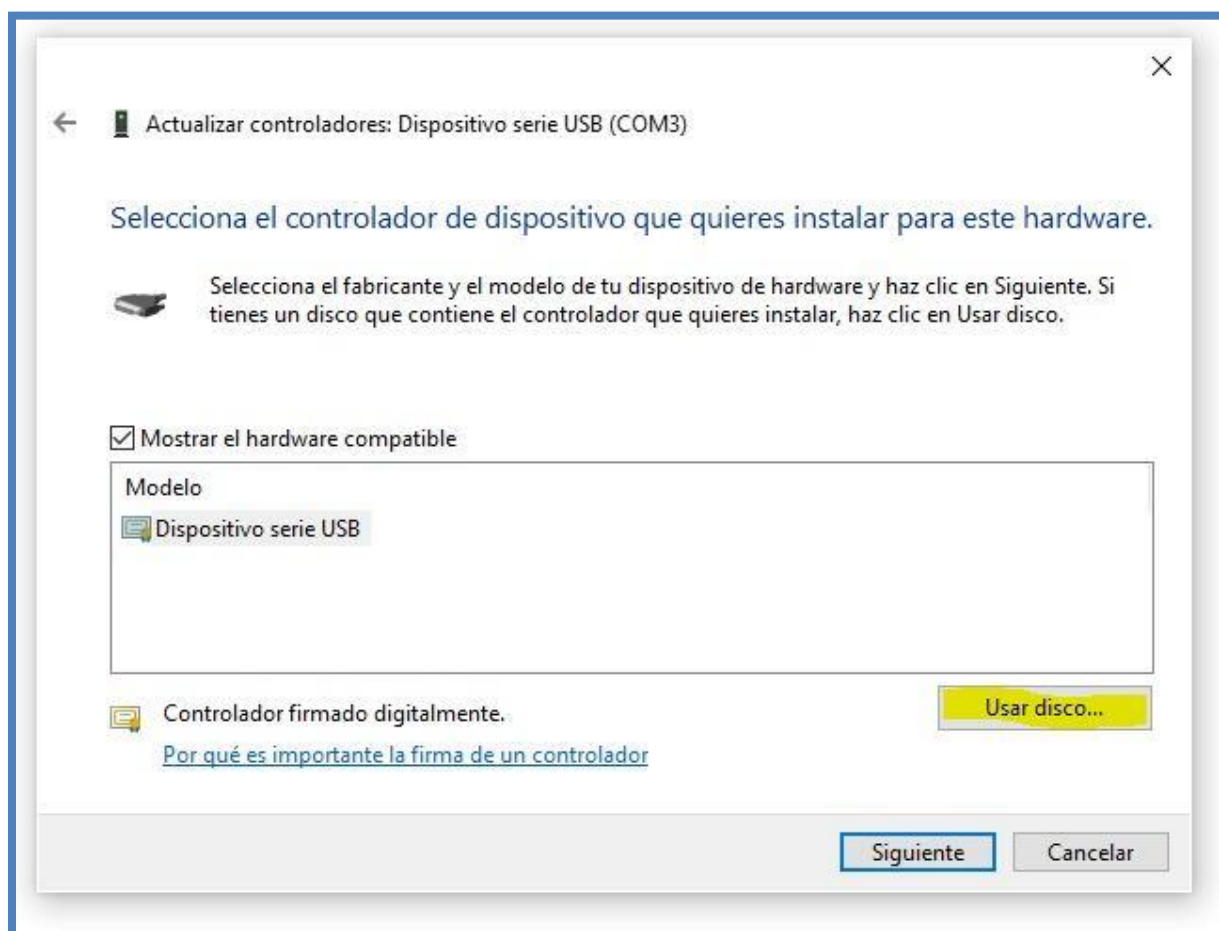
Volvemos al administrador de archivos y donde esta nuestra USB le damos click derecho, actualizar controlador



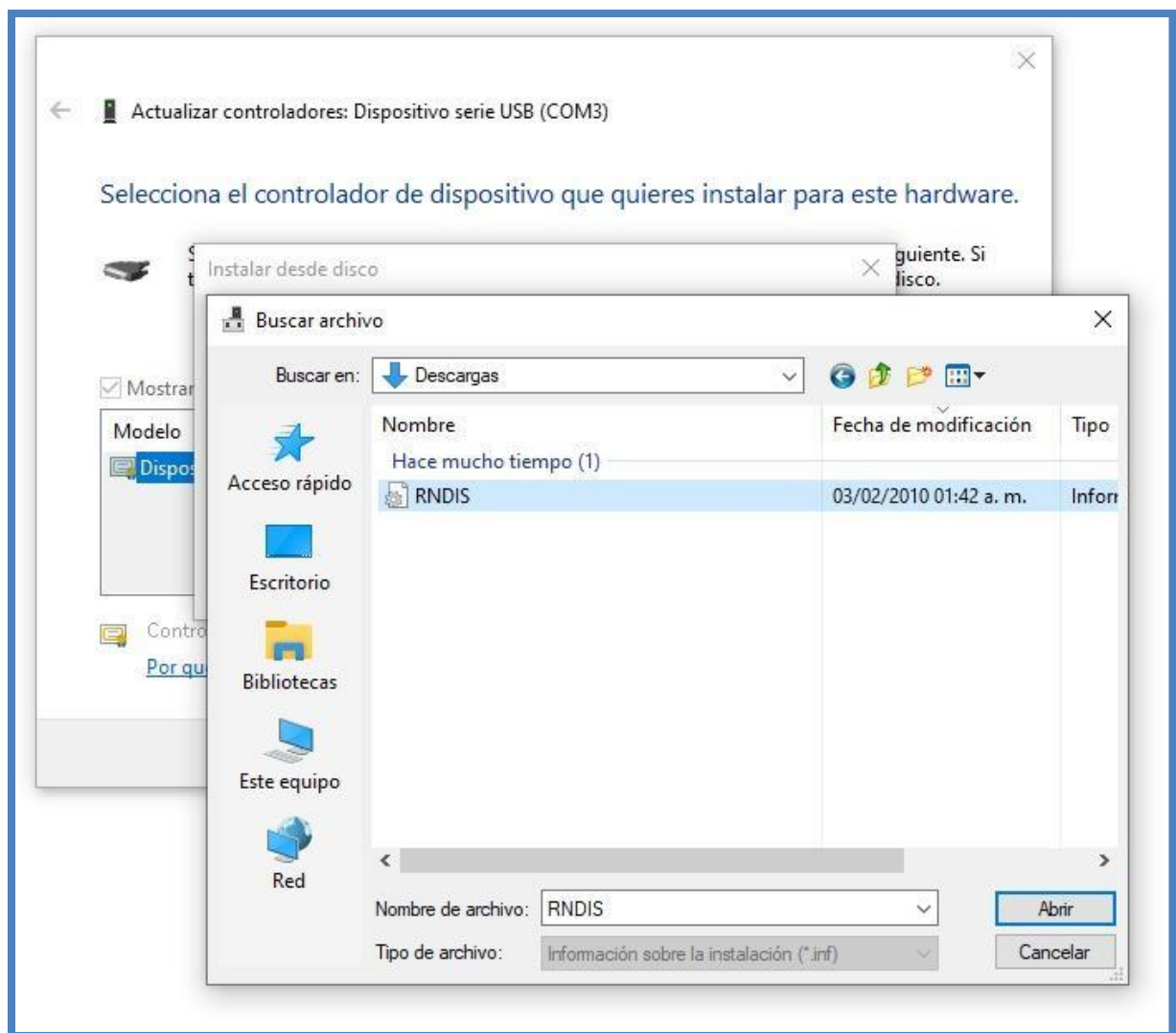
Y seleccionamos “Buscar controladores en mi equipo”



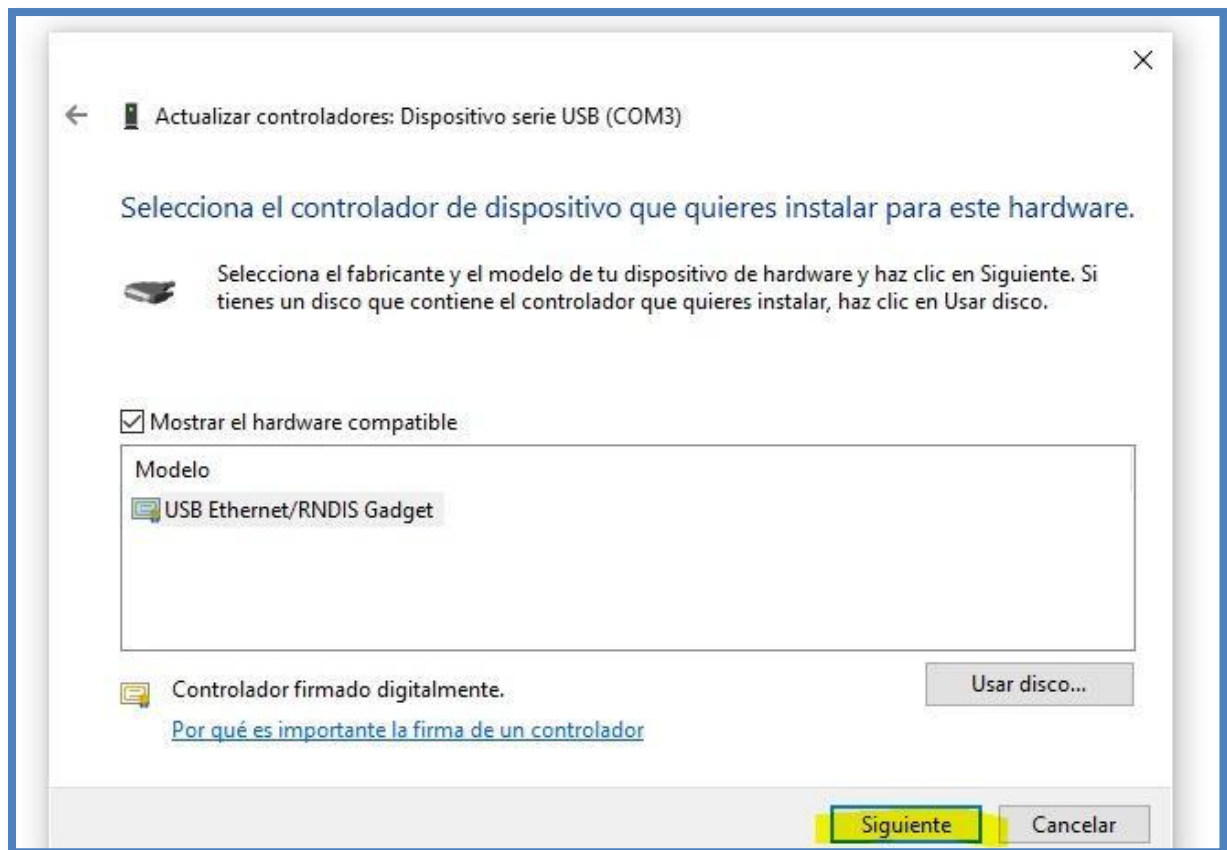
Seleccionamos **“Elegir en una lista de controladores disponibles en el equipo”**



Y le damos en **“Usar disco”**

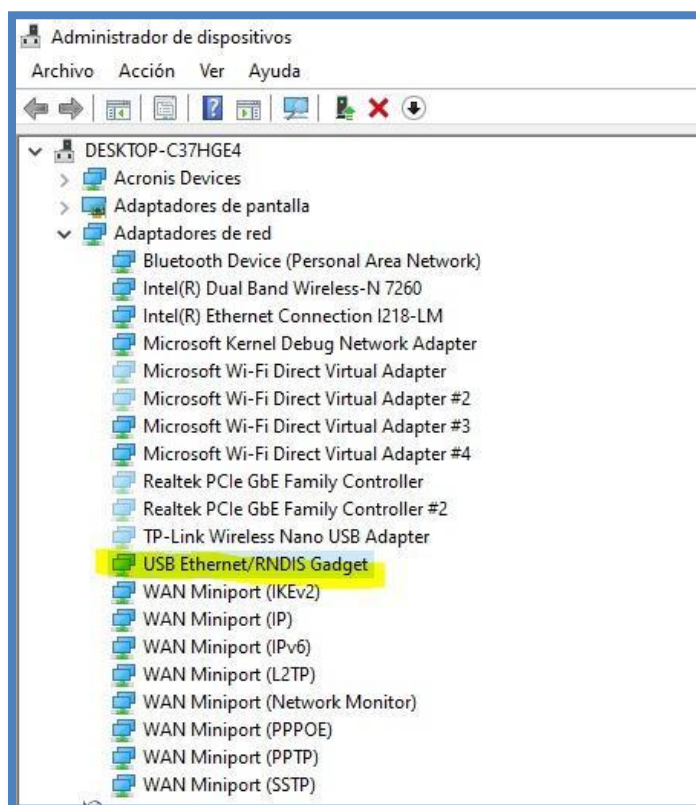


y seleccionamos el archivo que descargamos.



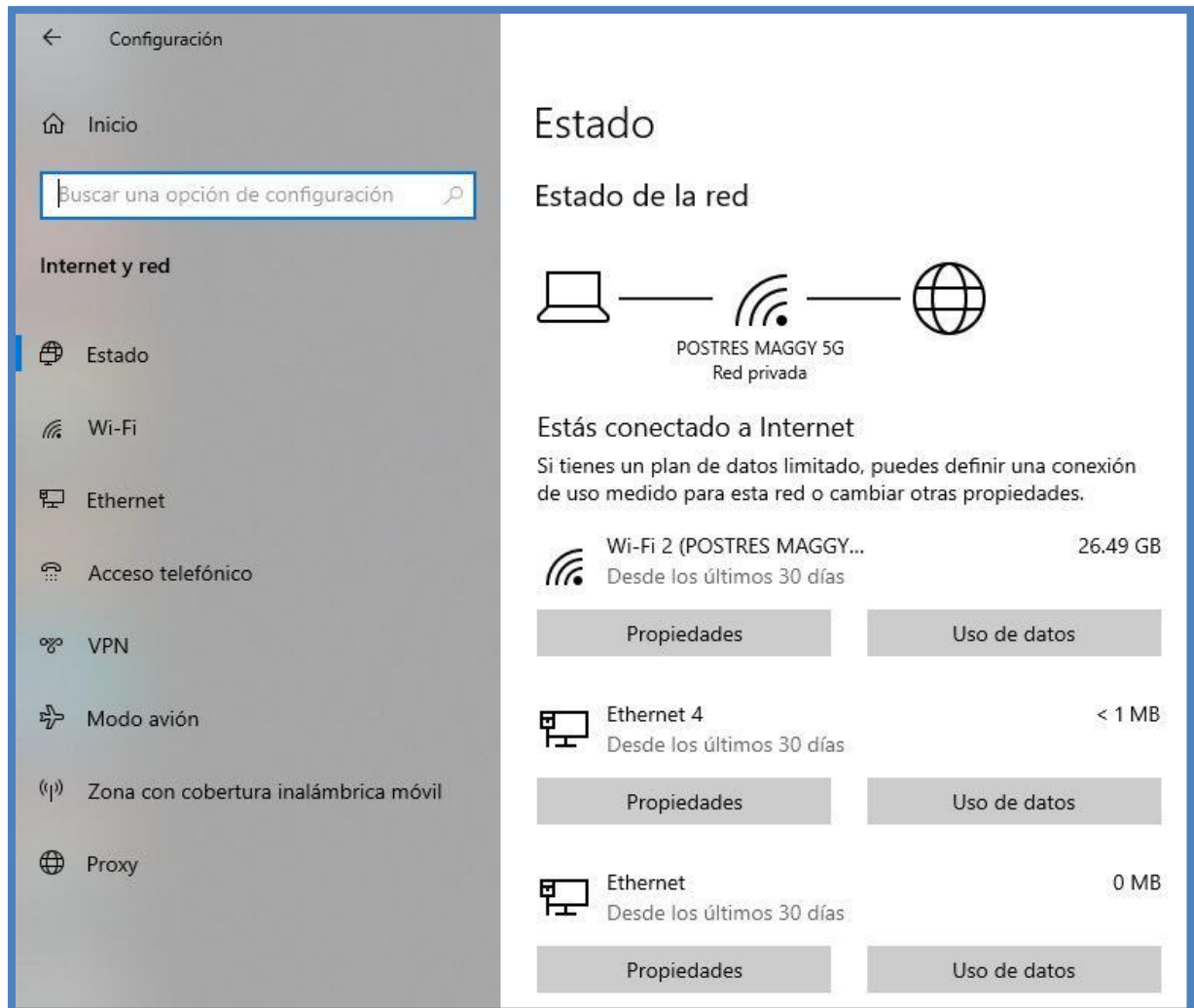
y le damos en **“siguiente”**

Sabemos que hicimos lo anterior bien porque debe de aparecer nuestro dispositivo en las listas de **“Adaptadores de red”**





Ahora abrimos la “**Configuración de red**”



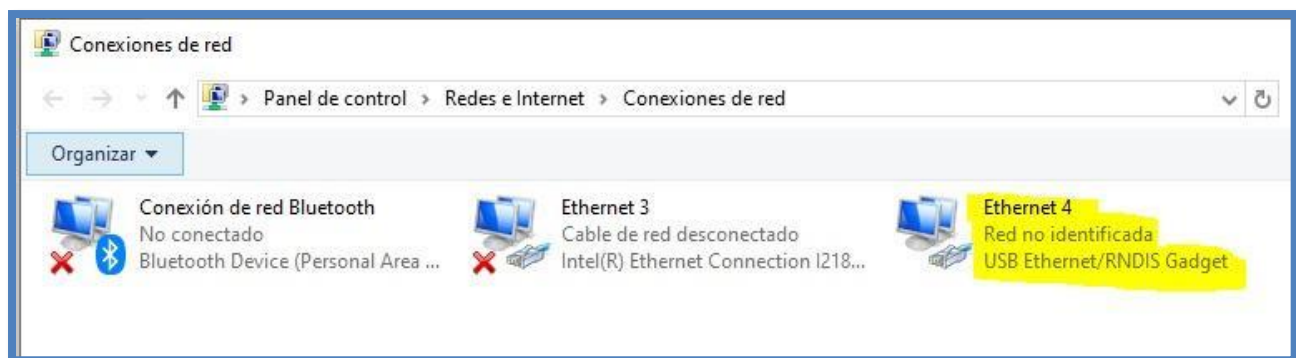
Nos vamos a configuración avanzada de red



y le damos en “Cambiar opciones de adaptador”

Buscamos nuestra red con el nombre de “**USB Ethernet/RNDIS Gadget**”

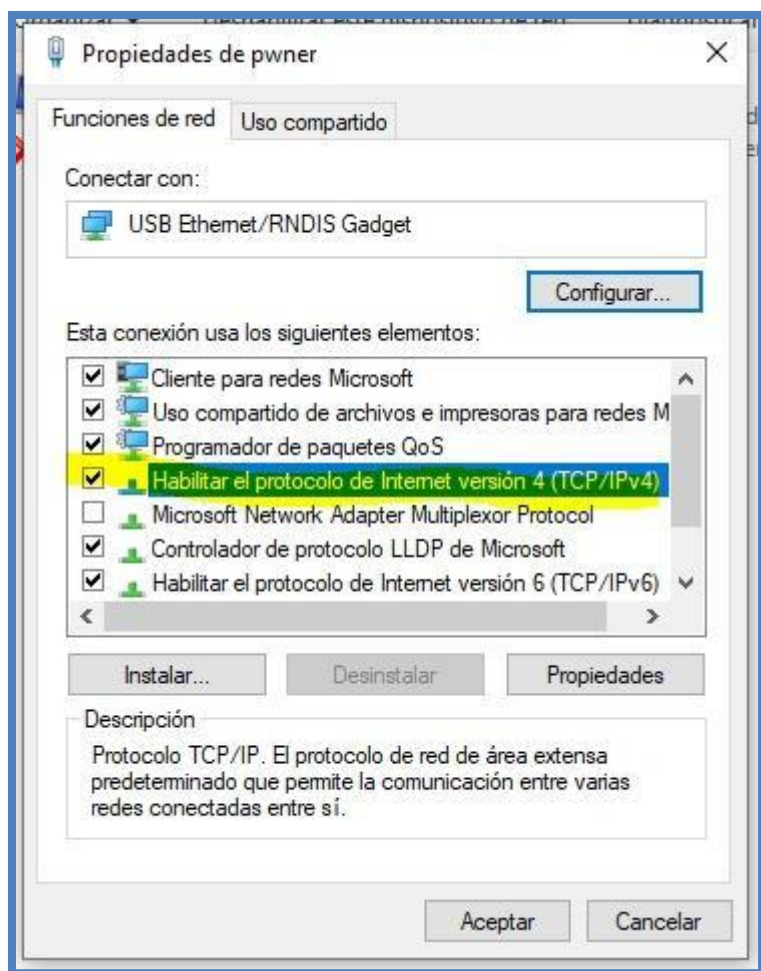




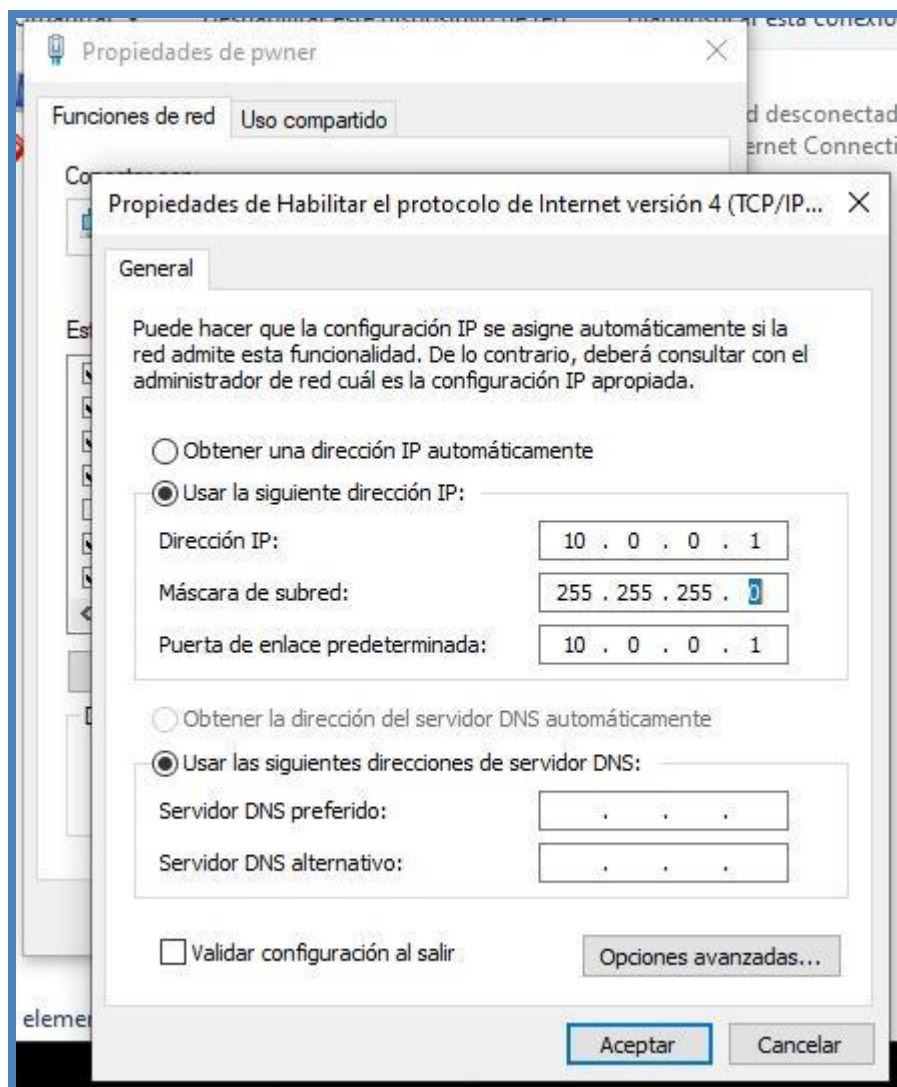
Damos clic derecho y le cambiamos el nombre



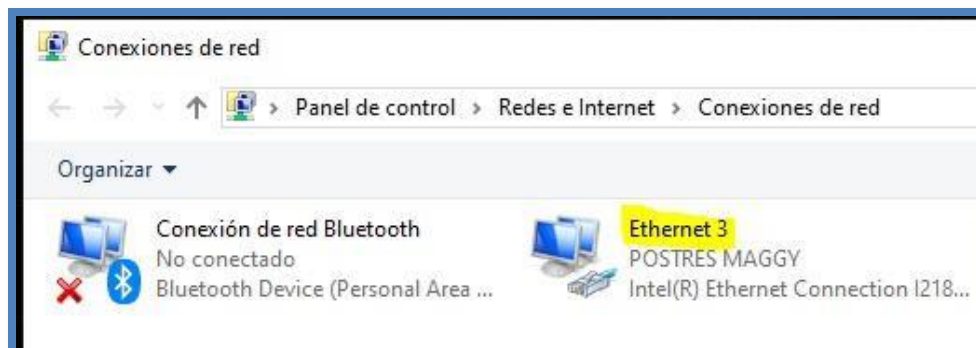
Le damos clic derecho y le damos a **propiedades**



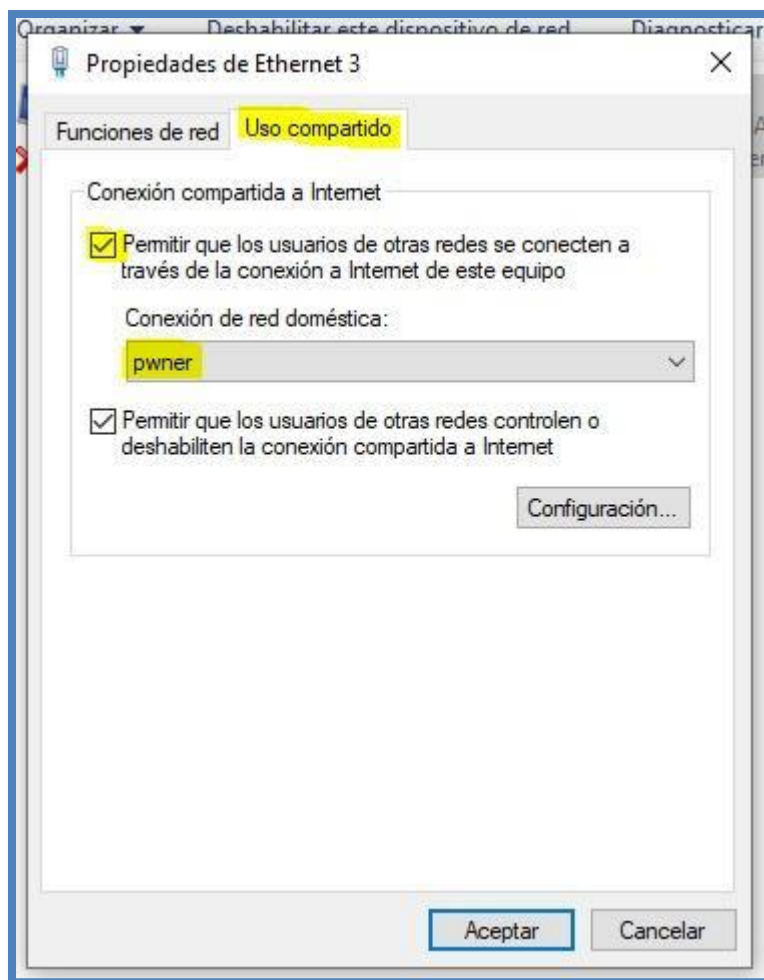
Le damos a **“Habilitar el protocolo de Internet”** y nos abrirá otra ventana donde pondremos las siguientes especificaciones:



y le damos en aceptar  
ahora nos vamos a nuestra red de Ethernet y le damos en propiedades

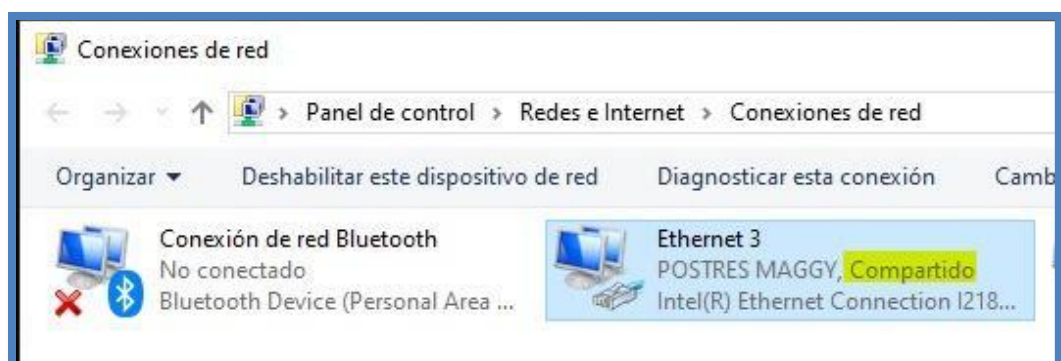


y le ponemos las siguientes especificaciones:

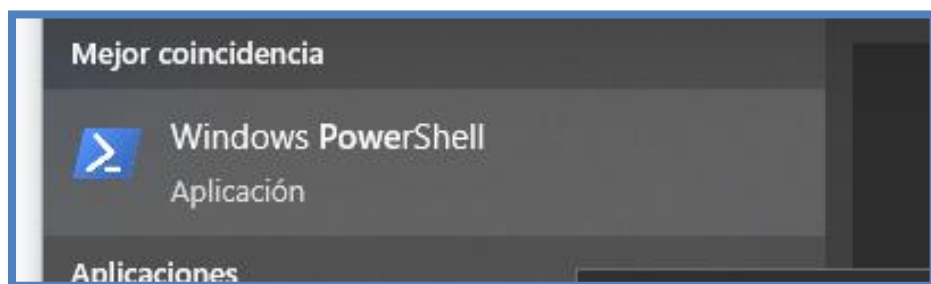


le damos en **“aceptar”**

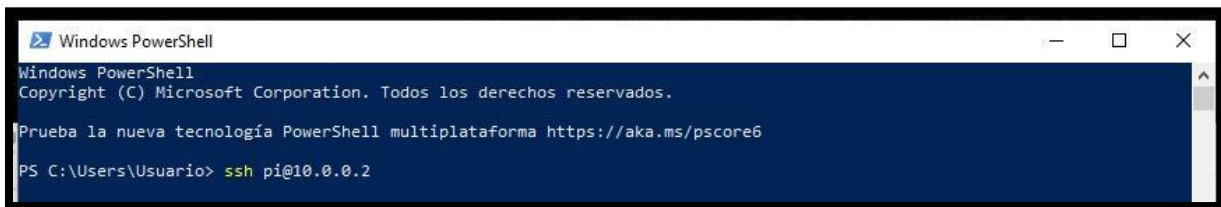
Ahora nos debe aparecer la red como **“Compartido”**



Ahora procederemos a lo más divertido, necesitamos abrir la **“PowerShell”**



y aquí seremos más cuidadosos, le damos la siguiente sintaxis: **ssh pi@10.0.0.2**

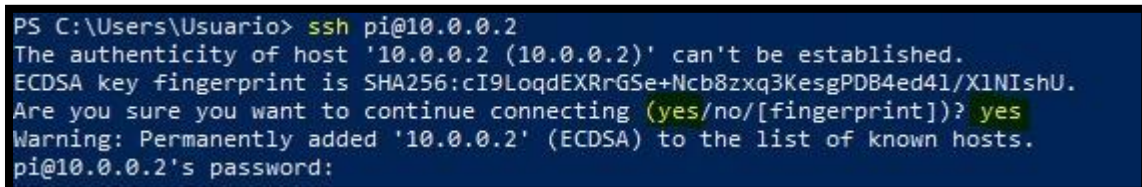


```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\Usuario> ssh pi@10.0.0.2
```

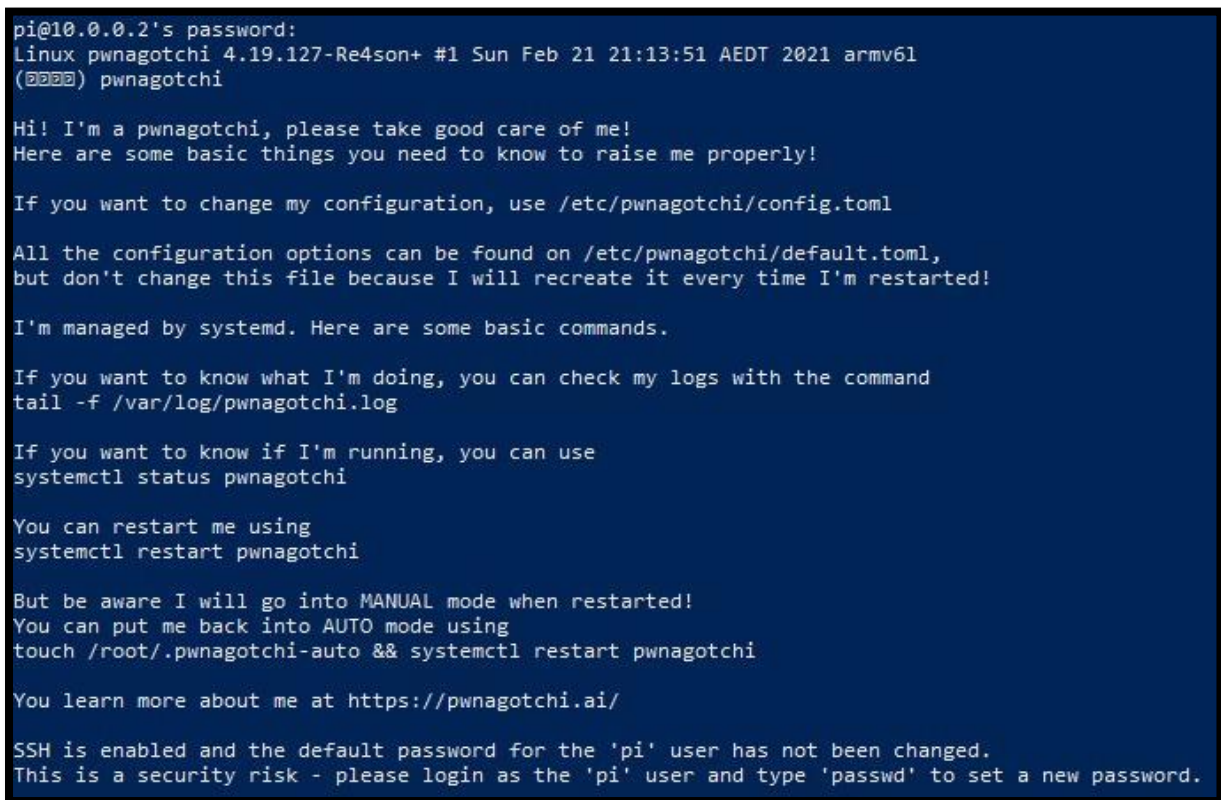
y damos enter, nos saldra el siguiente mensaje:



```
PS C:\Users\Usuario> ssh pi@10.0.0.2
The authenticity of host '10.0.0.2 (10.0.0.2)' can't be established.
ECDSA key fingerprint is SHA256:cI9LoqdEXRrGSe+Ncb8zxq3KesgPDB4ed4l/XlNIshU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.0.2' (ECDSA) to the list of known hosts.
pi@10.0.0.2's password:
```

escribimos “yes”

para esto nos pedirá una contraseña, la cual es **raspberry**, le daremos enter



```
pi@10.0.0.2's password:
Linux pwnagotchi 4.19.127-Re4son+ #1 Sun Feb 21 21:13:51 AEDT 2021 armv6l
(0000) pwnagotchi

Hi! I'm a pwnagotchi, please take good care of me!
Here are some basic things you need to know to raise me properly!

If you want to change my configuration, use /etc/pwnagotchi/config.toml

All the configuration options can be found on /etc/pwnagotchi/default.toml,
but don't change this file because I will recreate it every time I'm restarted!

I'm managed by systemd. Here are some basic commands.

If you want to know what I'm doing, you can check my logs with the command
tail -f /var/log/pwnagotchi.log

If you want to know if I'm running, you can use
systemctl status pwnagotchi

You can restart me using
systemctl restart pwnagotchi

But be aware I will go into MANUAL mode when restarted!
You can put me back into AUTO mode using
touch /root/.pwnagotchi-auto && systemctl restart pwnagotchi

You learn more about me at https://pwnagotchi.ai/

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.
```

nos pedirá una nueva contraseña la cual será “**contrasena**”  
sabemos que se configuró bien porque sale el siguiente mensaje:



```
pi@pwnagotchi: ~$ passwd
Changing password for pi.
Current password:
New password:
Retype new password:
Bad: new password is too simple
New password:
Retype new password:
Bad: new password is too simple
New password:
Retype new password:
passwd: password updated successfully
pi@pwnagotchi: ~$
```

después de todo esto, escribimos: **sudo su**

```
passwd: password updated successfully
pi@pwnagotchi: ~$ sudo su
root@pwnagotchi: /home/pi#
```

y enter y la sintaxis: **passwd root**

Donde repetiremos las contraseñas, la cual es “**contrasena**”

```
pi@pwnagotchi: ~$ sudo su
root@pwnagotchi: /home/pi# passwd root
New password:
Retype new password:
passwd: password updated successfully
root@pwnagotchi: /home/pi#
```

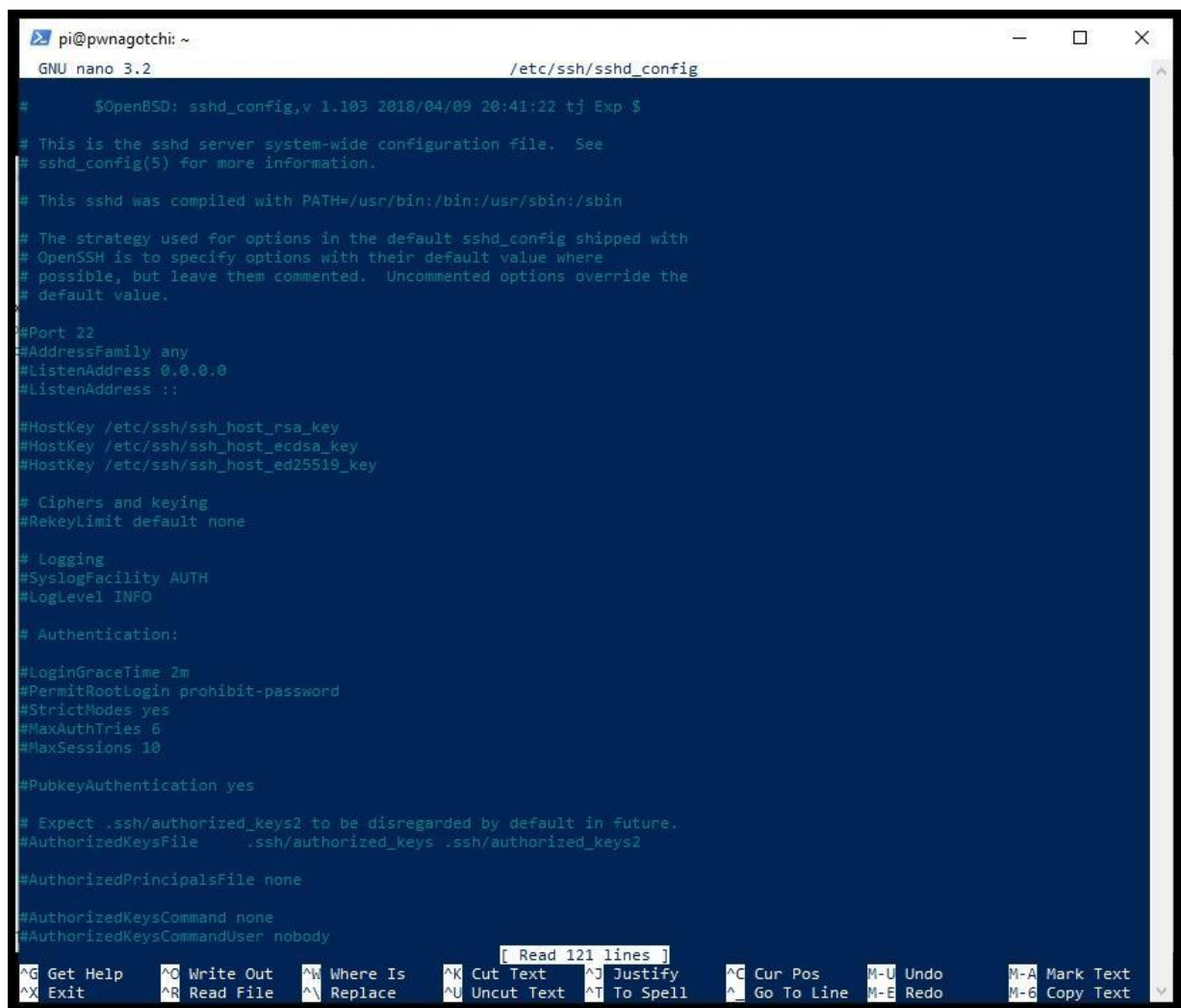
nos debe decir “**successfully**”

Después de eso escribimos: **sudo nano /etc/ssh/sshd\_config**

```
Retype new password:
passwd: password updated successfully
root@pwnagotchi: /home/pi# sudo nano /etc/ssh/sshd_config
```

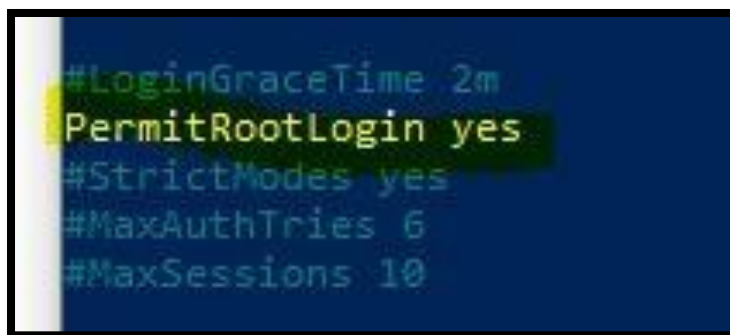
y damos enter

Nos debe de aparecer la siguiente pantalla:



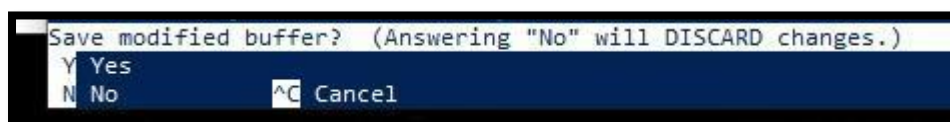
```
pi@pwnagotchi: ~  
GNU nano 3.2 /etc/ssh/sshd_config  
# $OpenBSD: sshd_config,v 1.103 2018/04/09 20:41:22 tj Exp $  
  
# This is the sshd server system-wide configuration file. See  
# sshd_config(5) for more information.  
  
# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin  
  
# The strategy used for options in the default sshd_config shipped with  
# OpenSSH is to specify options with their default value where  
# possible, but leave them commented. Uncommented options override the  
# default value.  
  
#Port 22  
#AddressFamily any  
#ListenAddress 0.0.0.0  
#ListenAddress ::  
  
#HostKey /etc/ssh/ssh_host_rsa_key  
#HostKey /etc/ssh/ssh_host_ecdsa_key  
#HostKey /etc/ssh/ssh_host_ed25519_key  
  
# Ciphers and keying  
#RekeyLimit default none  
  
# Logging  
#SyslogFacility AUTH  
#LogLevel INFO  
  
# Authentication:  
  
#LoginGraceTime 2m  
#PermitRootLogin prohibit-password  
#StrictModes yes  
#MaxAuthTries 6  
#MaxSessions 10  
  
#PubkeyAuthentication yes  
  
# Expect .ssh/authorized_keys2 to be disregarded by default in future.  
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2  
  
#AuthorizedPrincipalsFile none  
  
#AuthorizedKeysCommand none  
#AuthorizedKeysCommandUser nobody  
  
[ Read 121 lines ]  
^G Get Help ^O Write Out ^H Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo M-A Mark Text  
^X Exit ^R Read File ^N Replace ^U Uncut Text ^T To Spell ^_ Go To Line M-E Redo M-G Copy Text
```

Vamos a modificar la parte de “**PermitRootLogin**” de la siguiente manera y guardamos



```
#LoginGraceTime 2m  
PermitRootLogin yes  
#StrictModes yes  
#MaxAuthTries 6  
#MaxSessions 10
```

Repito, LO GUARDAMOS



```
Save modified buffer? (Answering "No" will DISCARD changes.)  
Y Yes  
N No ^C Cancel
```

una vez realizado esto, escribimos la siguiente sintaxis “**service ssh restart**”

```

passwd: password updated successfully
root@pwnagotchi:/home/pi# sudo nano /etc/ssh/sshd_config
root@pwnagotchi:/home/pi# service ssh restart
root@pwnagotchi:/home/pi#

```

y damos enter

Ahora haremos uso de servidores, para ello haremos uso del programa “**FileZilla**” el cual se puede descargar en: <https://filezilla-project.org>

Una vez descargado e instalado, procederemos a abrir el programa, donde tendremos la siguiente interfaz:



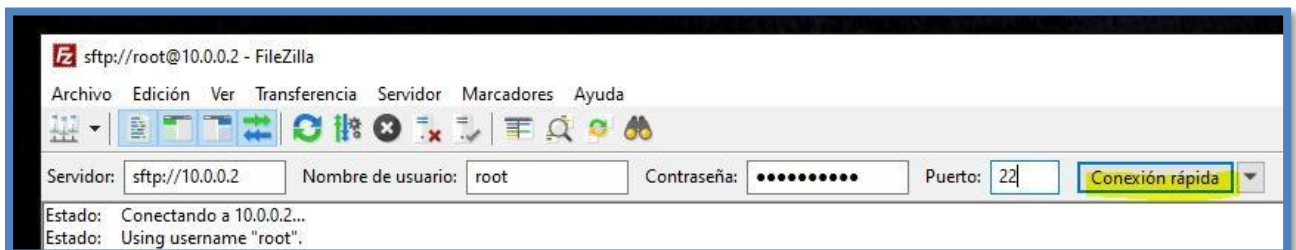
En la cual pondremos la información de nuestro pwnagotchi:

Servidor: **10.0.0.2**

Nombre de usuario: **root**

Contraseña: **contrasena**

Puerto: **22**



Una vez estos campos llenados, le damos clic en “conexión rápida”

Volviendo a nuestra PowerShell, ingresamos el siguiente comando:

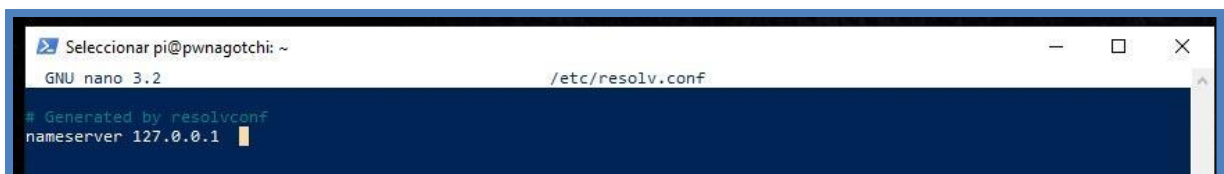
**sudo nano /etc/resolv.conf** y le damos enter

```

root@pwnagotchi:/home/pi# sudo nano /etc/ssh/sshd_config
root@pwnagotchi:/home/pi# service ssh restart
root@pwnagotchi:/home/pi# sudo nano /etc/resolv.conf
root@pwnagotchi:/home/pi#

```

Nos debe de abrir una ventana que se debe ver así:



En el cual editaremos el nombre del server a: **8.8.8.8**





```
pi@pwnagotchi: ~  
GNU nano 3.2 /etc/resolv.conf Modified  
# Generated by resolvconf  
nameserver 8.8.8.8
```

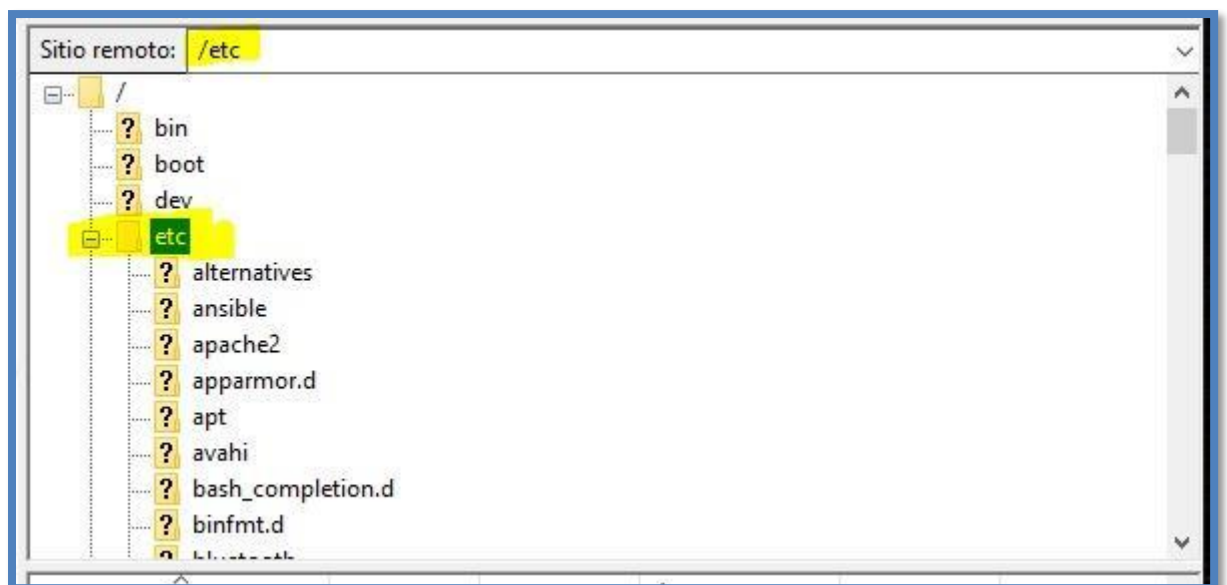
y guardamos



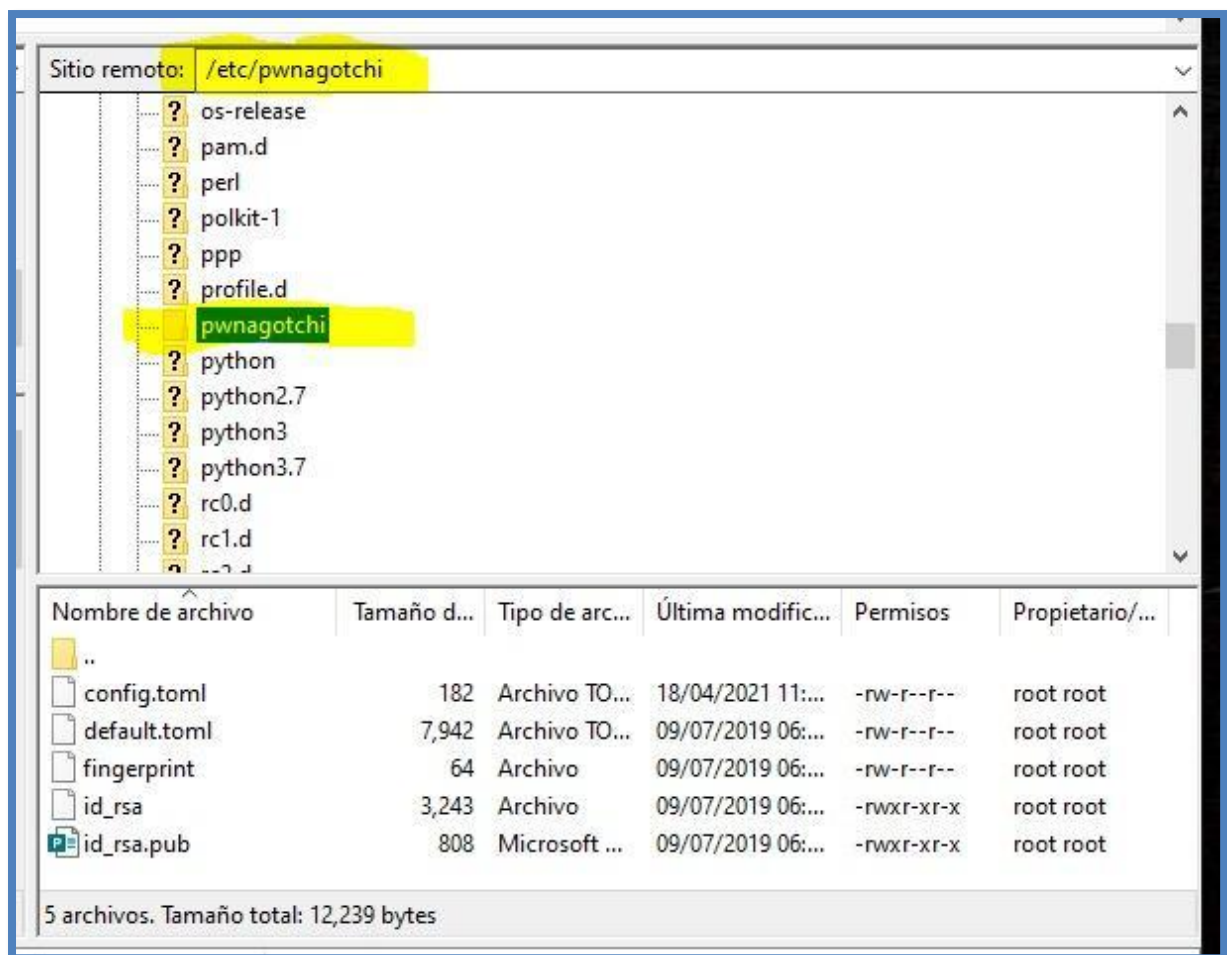
Una vez terminado esto, regresaremos a nuestro programa “FileZilla”  
Nos iremos a la parte inferior derecha donde nos ubicaremos en la carpeta “/”



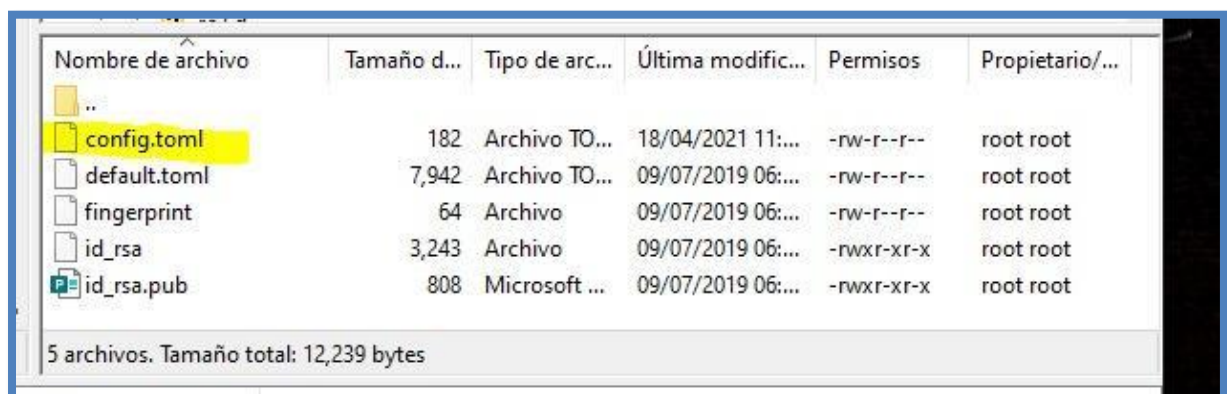
Tal como se muestra en la imagen anterior  
Ahora buscaremos la carpeta “etc”



Y de ahí buscamos la carpeta “pwnagotchi”



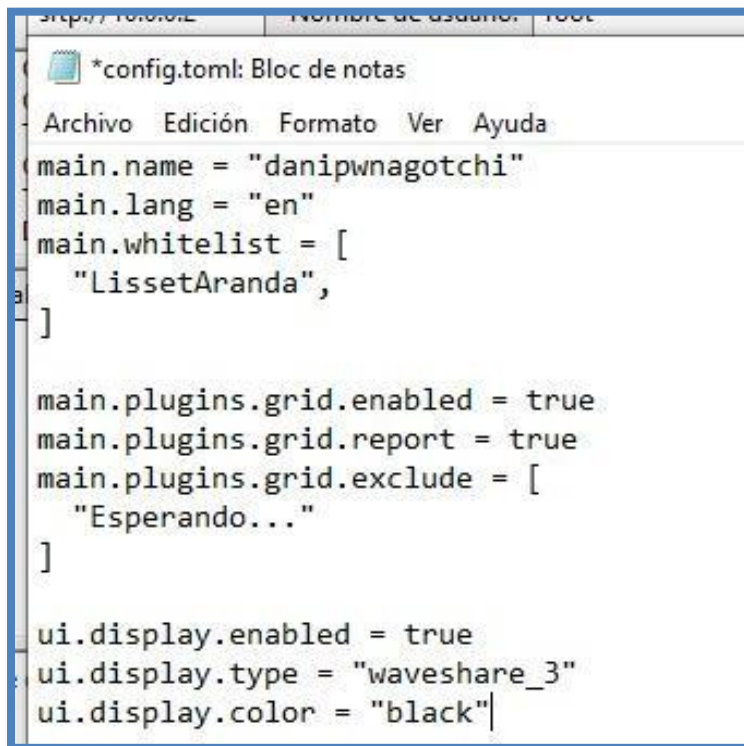
De ahí, en la parte inferior, nos iremos al archivo llamado “**config.toml**”



El cual abriremos, es importante editarlo con una aplicación de texto, en este caso usaremos el “**Bloc de notas**” y le ponemos la siguiente configuración:

```
main.name = "danipwnagotchi"
main.lang = "en"
main.whitelist = [
    "lissi-hacker",
]
main.plugins.grid.enabled = true
```

```
main.plugins.grid.report = true
main.plugins.grid.exclude = [
  "Esperando..."
]
ui.display.enabled = true
ui.display.type = "waveshare_3"
ui.display.color = "black"
```



Una vez con esa configuración, podemos guardarlo  
Ahora necesitaremos volver a la PowerShell para ingresar el siguiente comando:  
**#!/bin/bash**



de esta forma, solo esperamos a que termine de ejecutarse el comando

```
pi@pwnagotchi ~$ fast-forward
1
bin/pwnagotchi
2
builder/data/usr/bin/pwnlib
3
pwnagotchi/pwnagotchi.py
4
pwnagotchi/defaults.toml
5
pwnagotchi/locale/hr/LC_MESSAGES/voice.mo
6
pwnagotchi/locale/hr/LC_MESSAGES/voice.po
7
pwnagotchi/locale/tr/LC_MESSAGES/voice.mo
8
pwnagotchi/locale/tr/LC_MESSAGES/voice.po
9
pwnagotchi/log.py
10
pwnagotchi/plugins/default/gps.py
11
pwnagotchi/plugins/default/paw-gps.py
12
pwnagotchi/plugins/default/wigle.py
13
pwnagotchi/ui/display.py
14
pwnagotchi/ui/hw/_init_.py
15
pwnagotchi/ui/hw/lib/waveshare/v2131nb_v4/epd2in13b_v4.py
16
pwnagotchi/ui/hw/lib/waveshare/v2131nb_v4/epdconfig.py
17
pwnagotchi/ui/hw/lib/waveshare/v3/epd2in13_v3.py
18
pwnagotchi/ui/hw/lib/waveshare/v3/epdconfig.py
19
pwnagotchi/ui/hw/olcshat.py
20
pwnagotchi/ui/hw/waveshare2131nb_v4.py
21
pwnagotchi/ui/hw/waveshare3.py
22
pwnagotchi/ui/hw/waveshare351cd.py
23
pwnagotchi/ui/web/static/images/pwnagotchi.png
24
pwnagotchi/ui/web/templates/base.html
25
pwnagotchi/utlis.py
26
requirements.txt
27
scripts/backup.sh
28
scripts/openbsd_connection_share.sh
29
29 files changed, 1790 insertions(+), 34 deletions(-)
create mode 100644 pwnagotchi/locale/hr/LC_MESSAGES/voice.mo
create mode 100644 pwnagotchi/locale/hr/LC_MESSAGES/voice.po
create mode 100644 pwnagotchi/locale/tr/LC_MESSAGES/voice.mo
create mode 100644 pwnagotchi/locale/tr/LC_MESSAGES/voice.po
create mode 100644 pwnagotchi/ui/hw/lib/waveshare/v2131nb_v4/epd2in13b_v4.py
create mode 100644 pwnagotchi/ui/hw/lib/waveshare/v2131nb_v4/epdconfig.py
create mode 100644 pwnagotchi/ui/hw/lib/waveshare/v3/epd2in13_v3.py
create mode 100644 pwnagotchi/ui/hw/lib/waveshare/v3/epdconfig.py
create mode 100644 pwnagotchi/ui/hw/waveshare2131nb_v4.py
create mode 100644 pwnagotchi/ui/hw/waveshare3.py
create mode 100644 pwnagotchi/ui/hw/waveshare351cd.py
create mode 100644 pwnagotchi/ui/web/static/images/pwnagotchi.png
create mode 100644 scripts/backup.sh
create mode 100644 scripts/openbsd_connection_share.sh
root@pwnagotchi:~# cd /usr/local/src/pwnagotchi
root@pwnagotchi:~# cp /usr/local/src/pwnagotchi/pwnagotchi/ui/hw/lib/waveshare/v3/
root@pwnagotchi:~# cp /usr/local/src/pwnagotchi/pwnagotchi/ui/display.py /usr/local/lib/python3.7/dist-packages/pwnagotchi/ui/
root@pwnagotchi:~# cp /usr/local/src/pwnagotchi/pwnagotchi/ui/hw/_init_.py /usr/local/lib/python3.7/dist-packages/pwnagotchi/ui/hw/
root@pwnagotchi:~# cp /usr/local/src/pwnagotchi/pwnagotchi/ui/hw/lib/waveshare/v3/epd2in13_v3.py /usr/local/lib/python3.7/dist-packages/pwnagotchi/ui/hw/lib/waveshare/v3/
root@pwnagotchi:~# cp /usr/local/src/pwnagotchi/pwnagotchi/ui/hw/lib/waveshare/v3/epdconfig.py /usr/local/lib/python3.7/dist-packages/pwnagotchi/ui/hw/lib/waveshare/v3/
root@pwnagotchi:~# cp /usr/local/src/pwnagotchi/pwnagotchi/ui/hw/waveshare3.py /usr/local/lib/python3.7/dist-packages/pwnagotchi/ui/hw/
root@pwnagotchi:~# cp /usr/local/src/pwnagotchi/pwnagotchi/utlis.py /usr/local/lib/python3.7/dist-packages/pwnagotchi/
root@pwnagotchi:~# cp /usr/local/src/pwnagotchi/pwnagotchi/ui/hw/waveshare351cd.py /usr/local/lib/python3.7/dist-packages/pwnagotchi/ui/hw/
root@pwnagotchi:~#
```

Esperamos y esperamos

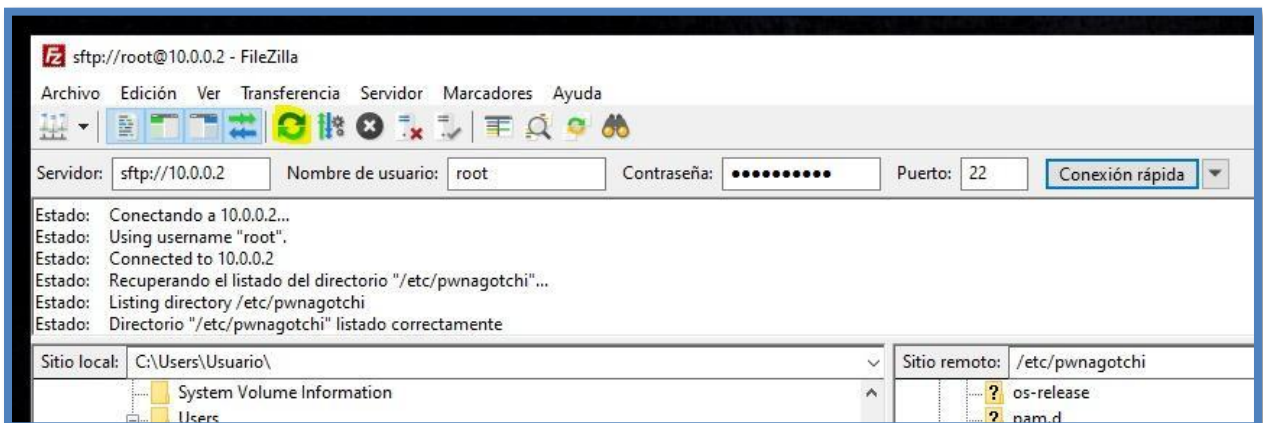
Al finalizar, le daremos enter y escribimos el siguiente comando:

**systemctl restart pwnagotchi**

```
root@pwnagotchi:~# cd /usr/local/src/pwnagotchi
root@pwnagotchi:~# sudo cp /usr/local/src/pwnagotchi/pwnagotchi/ui/hw/waveshare351cd.py /usr/local/lib/python3.7/dist-packages/pwnagotchi/ui/hw/
root@pwnagotchi:~# systemctl restart pwnagotchi
```

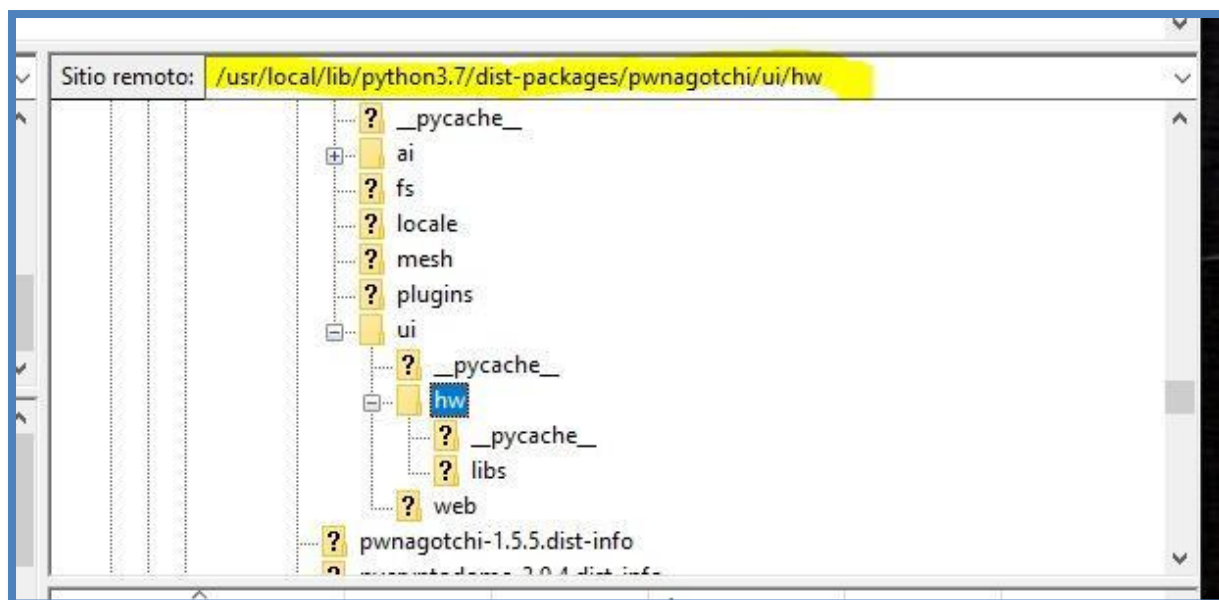
y le damos enter. por ahora, es lo único que tendremos que modificar en la PowerShell

Procederemos a volver al programa “**FileZilla**” le podemos dar en refrescar, ya que ya tenía nuestro servidor conectado

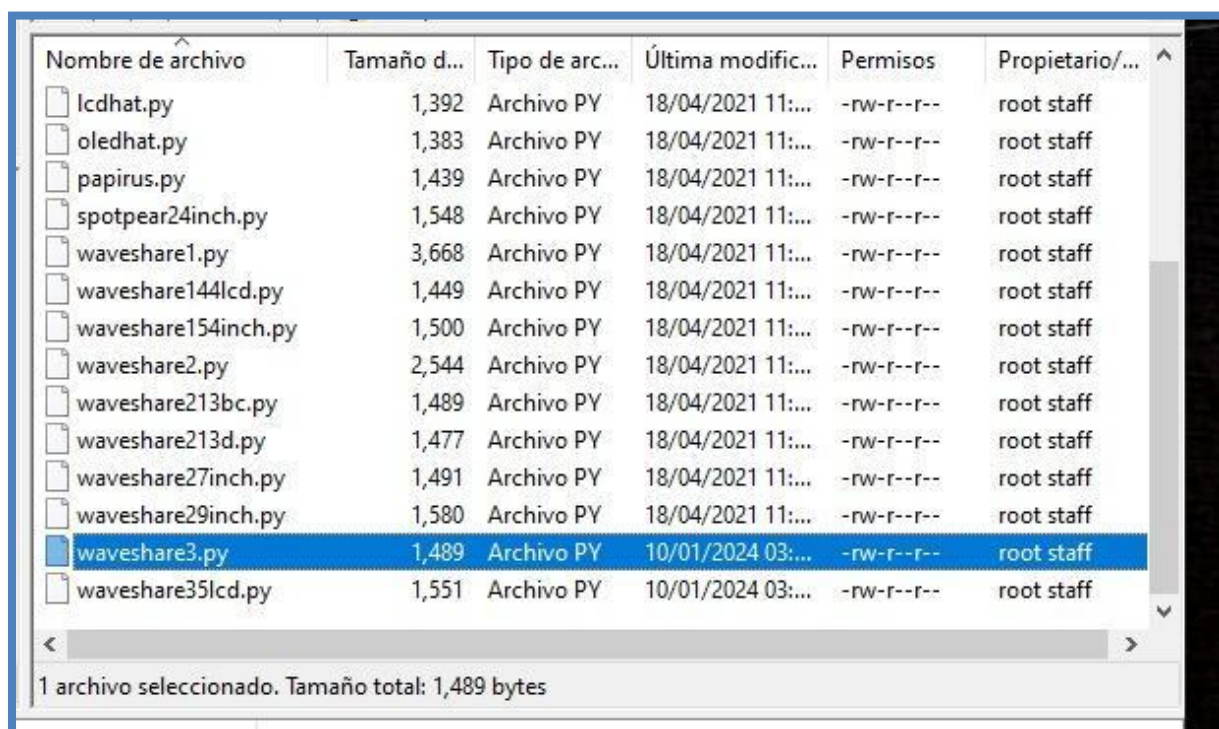


Como lo hicimos anteriormente, nos ubicamos en la carpeta: **/usr/local/lib/python3.7/dist-packages/pwnagotchi/ui/hw**





Donde en la parte inferior buscaremos al archivo: **waveshare3.py**



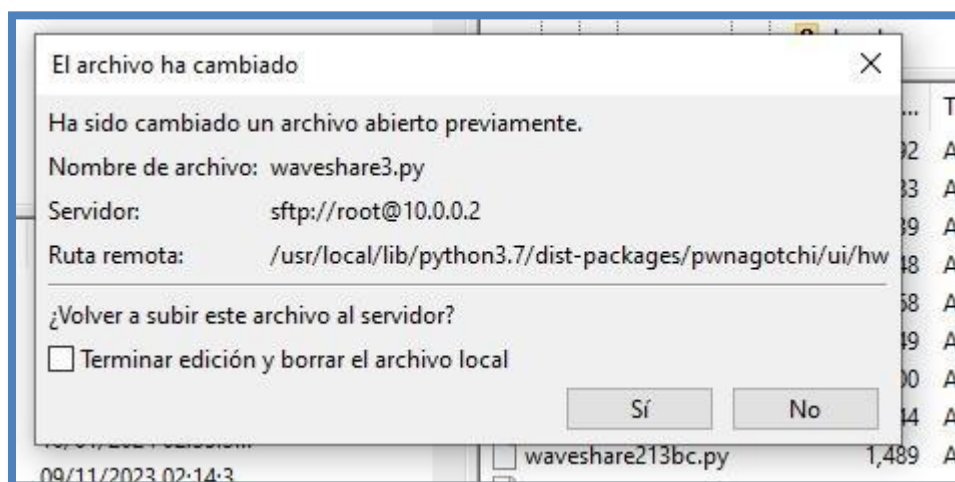
El cual vamos a editar en algún programa donde se pueda editar python3  
Tenemos por defecto la siguiente configuración:

```
waveshare3.py - ...\Temp\fb3temp-2
waveshare3.py x
1 import logging
2
3 import pwnagotchi.ui.fonts as fonts
4 from pwnagotchi.ui.hw.base import DisplayImpl
5
6
7 class WaveshareV3(DisplayImpl):
8     def __init__(self, config):
9         super(WaveshareV3, self).__init__(config, 'waveshare_3')
10        self._display = None
11
12    def layout(self):
13        fonts.setup(10, 8, 10, 35, 25, 9)
14        self._layout['width'] = 250
```

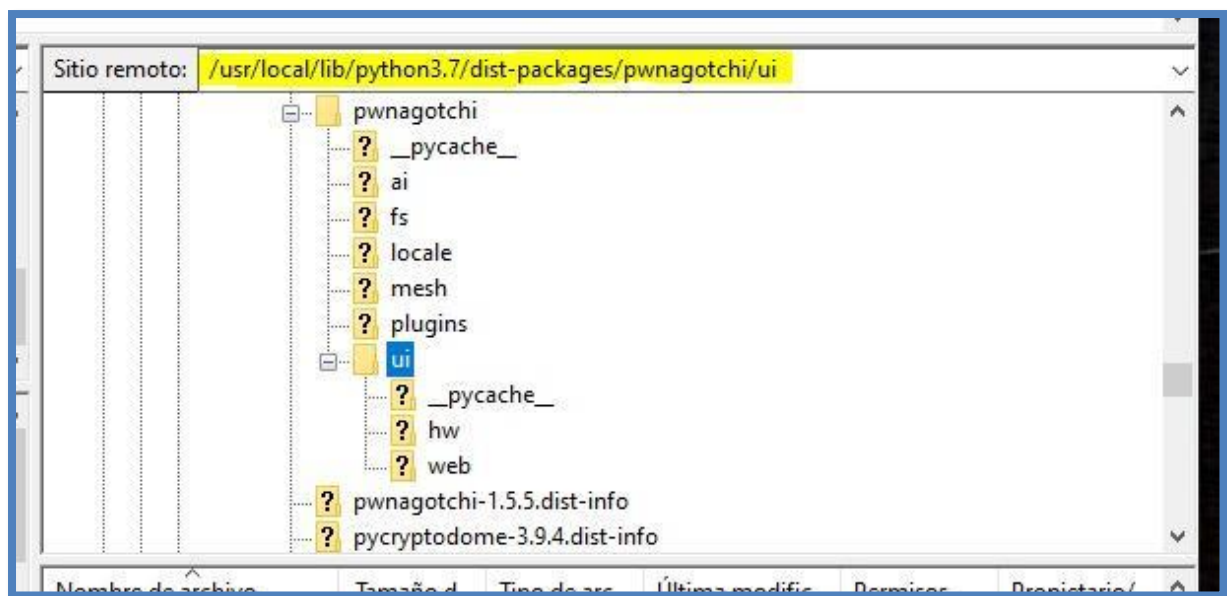
Tendremos que modificar la línea 13 de la siguiente manera  
**fonts.setup(10, 8, 10, 35, 25, 9)** para que quede así:

```
waveshare3.py •
1 import logging
2
3 import pwnagotchi.ui.fonts as fonts
4 from pwnagotchi.ui.hw.base import DisplayImpl
5
6
7 class WaveshareV3(DisplayImpl):
8     def __init__(self, config):
9         super(WaveshareV3, self).__init__(config, 'waveshare_3')
10        self._display = None
11
12    def layout(self):
13        fonts.setup(10, 8, 10, 35, 35, 9)
14        self._layout['width'] = 250
```

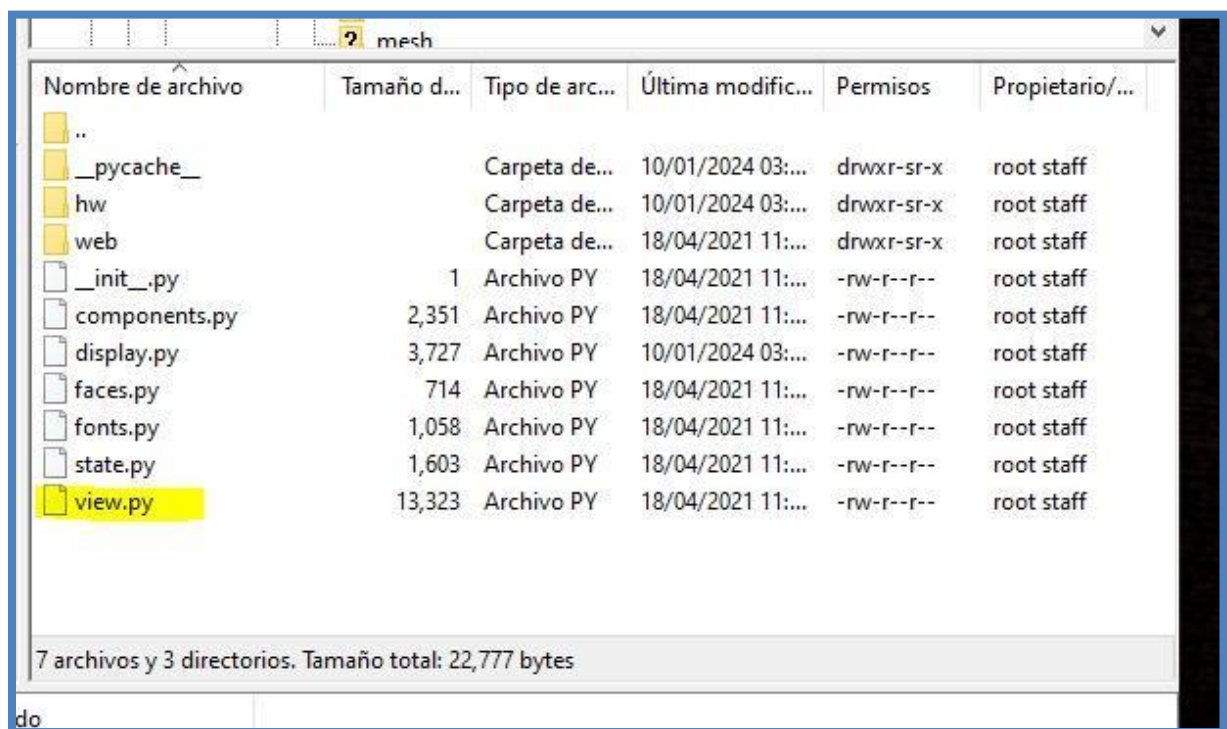
y guardamos:



Una vez terminado esto, nos iremos una carpeta atrás, es decir:  
**/usr/local/lib/python3.7/dist-packages/pwnagotchi/ui**

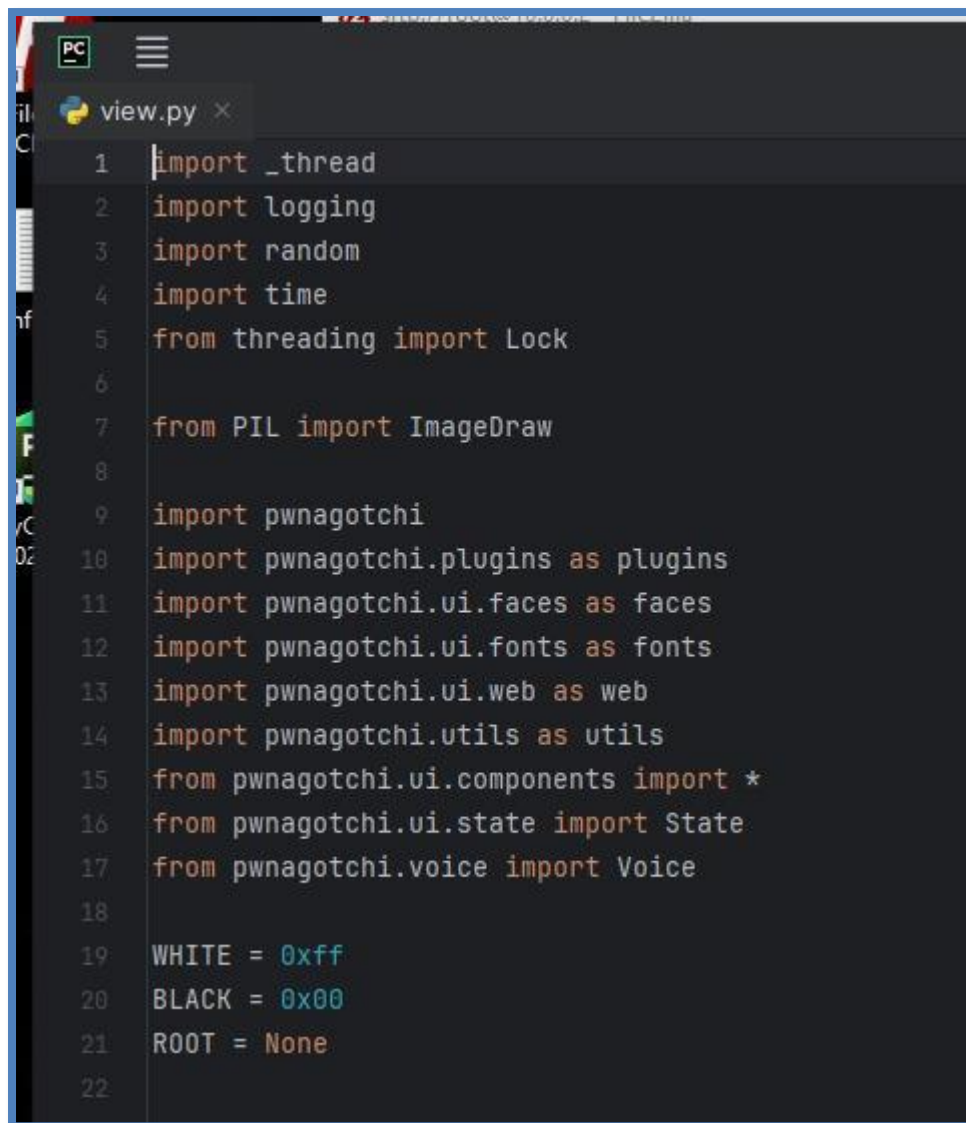


y aquí en la parte posterior buscaremos el archivo: view.py



El cual editaremos en un programa que maneje Python  
Tenemos la configuración siguiente por defecto:





```
1 import _thread
2 import logging
3 import random
4 import time
5 from threading import Lock
6
7 from PIL import ImageDraw
8
9 import pwnagotchi
10 import pwnagotchi.plugins as plugins
11 import pwnagotchi.ui.faces as faces
12 import pwnagotchi.ui.fonts as fonts
13 import pwnagotchi.ui.web as web
14 import pwnagotchi.utils as utils
15 from pwnagotchi.ui.components import *
16 from pwnagotchi.ui.state import State
17 from pwnagotchi.voice import Voice
18
19 WHITE = 0xff
20 BLACK = 0x00
21 ROOT = None
22
```

aquí editaremos las líneas 19 y 20 de la siguiente manera:

**WHITE = 0x00**

**BLACK = 0Xff**

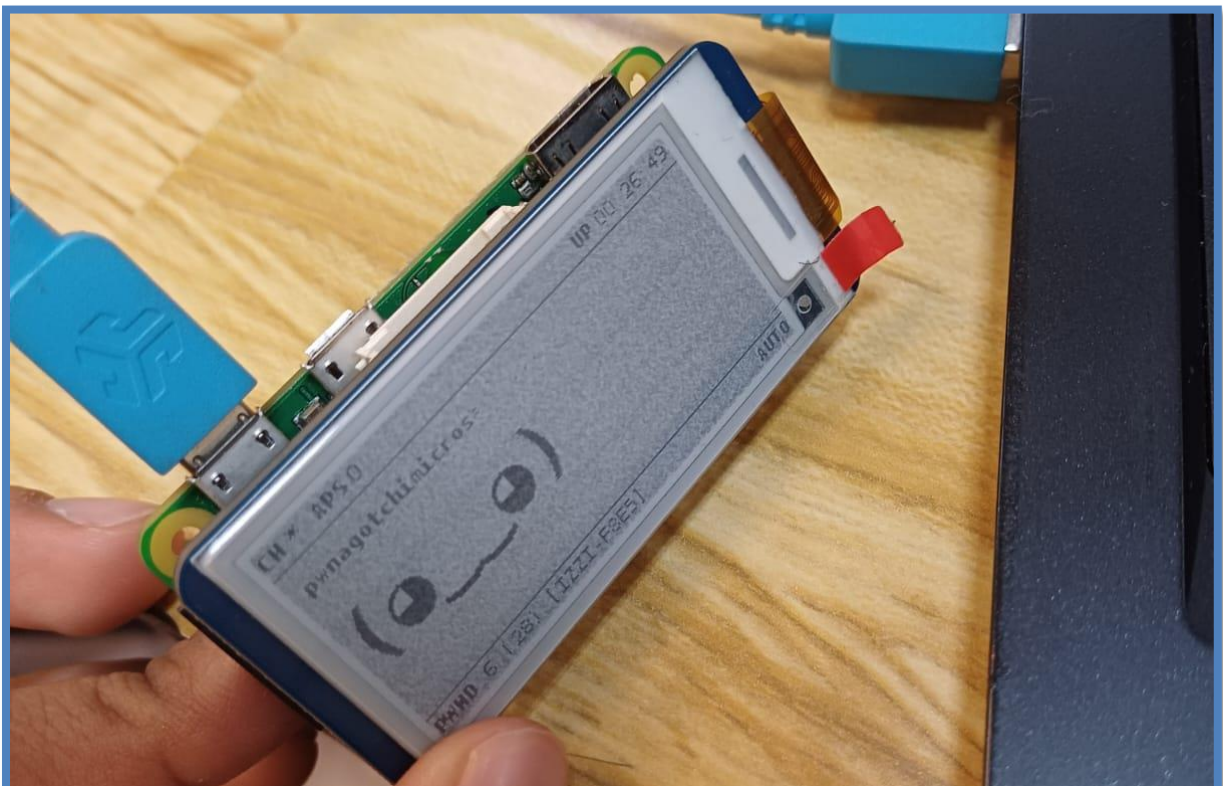
```

9  import pwnagotchi
10 import pwnagotchi.plugins as plugins
11 import pwnagotchi.ui.faces as faces
12 import pwnagotchi.ui.fonts as fonts
13 import pwnagotchi.ui.web as web
14 import pwnagotchi.utils as utils
15 from pwnagotchi.ui.components import *
16 from pwnagotchi.ui.state import State
17 from pwnagotchi.voice import Voice
18
19 WHITE = 0x00
20 BLACK = 0xff
21 ROOT = None
22

```

Una vez que tenemos esto, procedemos a guardar

Una vez que esta todo esto preparado, nuestro pwnagotchi ya debería está listo para funcionar, recuerda que puede tardar unos minutos, para poner ponerlo a funcionar basta con conectarlo de la siguiente manera:



Paso extra:

Para que podamos ver a nuestro pwnagotchi, tenemos que ingresar en nuestro navegador e ingresar 10.0.0.2:8080, saldrá un cuadro donde te pida ingresar user y psw, ingresas con la que cambiaste o por default tiene.