

# INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA  
UNIDAD CULHUACÁN

“DISEÑO E IMPLEMENTACIÓN DE UN DISPOSITIVO  
PORTATIL DE ALMACENAMIENTO DE ARCHIVOS Y  
MULTIMEDIA QUE GENERE RESPALDOS AUTOMÁTICOS  
CON DOS MODALIDADES, RED LOCAL ESPECÍFICA O  
DIRECCION WEB”

2023-ATC-ICE-49V

ALEJANDRO JOSHUA CASTILLO GÓMEZ

ASESORES

ING. ORLANDO BELTRÁN NAVARRO

M EN C EDITH GONZÁLEZ LEE

DESARROLLO PROSPECTIVO DE PROYECTOS

GRUPO 9EV21

# INDICE

I.	PLANTEAMIENTO DEL PROBLEMA .....	3
II.	JUSTIFICACIÓN .....	4
III.	OBJETIVO GENERAL.....	5
IV.	OBJETIVOS ESPECÍFICOS.....	5
1.	ESTADO DEL ARTE .....	7
2.	MARCO TEÓRICO.....	12
3.	DISEÑO E IMPLEMENTACIÓN .....	24
4.	PRUEBAS Y RESULTADOS .....	36
V.	CONCLUSIONES Y RECOMENDACIONES .....	45

## I. PLANTEAMIENTO DEL PROBLEMA

En la actualidad, el almacenamiento de archivos y multimedia personal y profesional como documentos importantes, fotos, videos, archivos de trabajo y configuraciones personalizadas en dispositivos electrónicos es esencial. Sin embargo, este aumento en la dependencia de los dispositivos electrónicos también ha llevado a un riesgo significativo de pérdida de datos debido a diversas razones, como errores humanos, fallas de hardware, ataques de malware o incluso robos.

El problema clave radica en la falta de una solución de respaldo efectiva y conveniente. A menudo, los usuarios se ven obligados a depender de prácticas manuales de respaldo, que son propensas a olvidos y retrasos, o bien, se ven limitados por la falta de automatización y recursos. Además, las soluciones de respaldo tradicionales, como los servicios en la nube, pueden plantear costos sostenidos a largo plazo y preocupaciones de privacidad.

Sumado a que según el sitio [Checkpoint.com](https://www.checkpoint.com) *“El 94 % de las organizaciones está moderada a extremadamente preocupada por la seguridad en la nube. Cuando se consultó sobre cuáles son las amenazas de seguridad más grandes que enfrentan las nubes públicas, las organizaciones calificaron la configuración incorrecta (68 %) como la principal, seguida del acceso no autorizado (58 %), las interfaces no seguras (52 %) y el secuestro de cuentas (50 %). Aquí analizaremos las principales amenazas y preocupaciones de seguridad en la nube en el mercado actual.”*

Y estudiado por el Instituto Federal de Telecomunicaciones en México (IFT) “ en su apartado 6.2, páginas 72 y 73 del estudio de Cloud Computing en México en Julio del 2020 *“Hoy en día las organizaciones del sector público siguen teniendo una preocupación legítima por la seguridad de sus datos, por lo que algunos gobiernos han llegado a establecer a los Proveedores de Servicios de Comunicaciones (PSC) como requisito de que todo contenido de los clientes procesado y almacenado en un sistema de TI, permanezca dentro de las fronteras del país, con el afán de brindar una capa adicional de seguridad. La residencia de los datos es una combinación de problemas asociados principalmente a riesgos de seguridad percibidos (y en algunos casos reales), relacionados con el acceso de terceros a la información, incluidas las autoridades competentes extranjeras. Los clientes del sector público desean tener la certeza de que sus datos están protegidos frente al acceso no deseado de atacantes maliciosos o de otros gobiernos. La preocupación general en torno a la ciberseguridad, así como la posible extralimitación de la vigilancia gubernamental por parte de algunos países ha contribuido a que el debate se centre continuamente en mantener los datos en el país. No obstante, esta restricción puede ser contraproducente para el objetivo de proteger de manera efectiva datos sensibles respecto a diversos ámbitos del sector público. Los niveles más altos de protección sólo se pueden garantizar en el entorno y hábitat donde se ubica la plataforma del proveedor de Cloud Computing ya que se puede mantener la soberanía reglamentaria de nación-estado.”*

## II. JUSTIFICACIÓN

Para abordar la necesidad de gestionar de manera eficiente los respaldos de datos en dispositivos portátiles, se propone la implementación de un dispositivo innovador que ofrezca soluciones integrales a los desafíos actuales.

Este proyecto busca desarrollar un dispositivo de almacenamiento portátil con respaldo automático que cumpla con los siguientes objetivos fundamentales:

### Automatización Eficiente:

El dispositivo realizará respaldos automáticos programados a intervalos regulares de 5 minutos, eliminando la necesidad de intervención manual por parte del usuario. Esta automatización garantizará que los archivos y multimedia estén siempre actualizados y protegidos.

### Accesibilidad:

Con el objetivo de proporcionar una solución verdaderamente accesible, el dispositivo será diseñado para ser fácilmente transportable y permitirá a los usuarios recuperar sus archivos y multimedia en cualquier momento y lugar.

### Seguridad de Datos:

El proyecto se centrará en implementar un sistema de respaldo confiable que proteja los archivos y multimedia contra pérdidas, ya sea debido a errores humanos, fallos de hardware, ciberataques u otros desastres digitales. Esta función crítica aborda la necesidad de mantener la integridad de los datos almacenados.

Desde una perspectiva social, este proyecto adquiere una relevancia destacada al empoderar a los usuarios mediante la entrega de una solución tecnológica avanzada y accesible. En el ámbito empresarial, la implementación de respaldos automáticos en dispositivos portátiles puede tener un impacto significativo en la productividad y la continuidad del negocio, ya que los datos críticos se respaldan de manera más eficiente y segura.

Este proyecto representa una innovación tecnológica al fusionar la conectividad inalámbrica, la detección de redes específicas y la automatización de respaldos. La detección contextual de redes permite que el dispositivo actúe de manera proactiva, marcando un avance significativo en comparación con las soluciones tradicionales. Esta convergencia tecnológica no solo mejora la eficiencia del respaldo, sino que también establece un precedente para el desarrollo futuro de dispositivos portátiles inteligentes y autónomos.

### **III. OBJETIVO GENERAL**

Diseñar e implementar un dispositivo portátil de almacenamiento de archivos multimedia basado en un script controlador en una Raspberry Pi 4B, que realice respaldos automáticos mediante un servidor web o una red local, al detectar una red específica, para automatizar y aumentar la seguridad de la información almacenada.

### **IV. OBJETIVOS ESPECÍFICOS**

- Diseñar el dispositivo de almacenamiento portátil con una unidad de almacenamiento de estado sólido y un controlador basado en un microprocesador Raspberry Pi 4b 8 GB.
- Desarrollar un script controlador en Python, para el dispositivo y su acceso a la red especificada.
- Implementar un control de acceso para cada usuario específico y un registro de uso dentro de un servidor WEB.
- Diseñar una interfaz gráfica para la gestión de los respaldos en el servidor WEB.
- Coordinar el hardware y el software garantizando la seguridad y la eficiencia del sistema en conjunto.

# **CAPÍTULO I.**

## **ESTADO DEL ARTE**

## **1. ESTADO DEL ARTE**

El dispositivo de almacenamiento forma parte de una evolución constante de la tecnología, fusionando la funcionalidad práctica con la ingeniería. A través de la historia del arte en el campo de la tecnología y dispositivos de almacenamiento, se evidencia una progresión continua hacia soluciones más eficientes y convenientes para el usuario.

### **1.1 Antecedentes Tecnológicos**

#### **1.1.1 Dispositivos Precursores**

Los primeros antecedentes de almacenamiento de datos se remontan a las décadas de 1950 y 1960, donde se utilizaban cintas y tambores magnéticos como principales medios de almacenamiento en sistemas informáticos primitivos. Estos dispositivos eran limitados en capacidad y velocidad.

#### **1.1.2 Nacimiento del Disco Duro**

El nacimiento del disco duro como lo conocemos hoy se remonta a la década de 1950. En 1956, IBM introdujo el primer disco duro comercial, el IBM 305 RAMAC, que tenía una capacidad asombrosa de 5 megabytes distribuidos en 50 discos magnéticos.

#### **1.1.3 Evolución de la Tecnología de Grabación**

A lo largo de los años 60 y 70, los discos duros experimentaron mejoras significativas en la densidad de grabación magnética. La transición de la grabación de película delgada a la grabación de película gruesa permitió aumentar la capacidad de almacenamiento.

#### **1.1.4 Introducción de la Interfaz IDE**

En la década de 1980, la introducción de la interfaz de disco de tipo IDE ("Integrated Drive Electronics") marcó un hito al permitir que el controlador del disco se integrara directamente en el propio disco duro. Esto simplificó la conexión y mejoró la compatibilidad con diferentes sistemas.

#### **1.1.5 Ascenso de las Unidades de Estado Sólido (SSD)**

A fines de la década de 2000 y principios de la de 2010, las unidades de estado sólido (SSD) emergieron como una alternativa a los discos duros tradicionales. Estas unidades, basadas en memoria flash, ofrecían velocidades de lectura y escritura significativamente más rápidas y mayor durabilidad, aunque inicialmente a un costo más elevado.

#### **1.1.6 Aumento Exponencial de la Capacidad**

En las últimas décadas, la capacidad de almacenamiento de los discos duros ha experimentado un crecimiento exponencial. Se ha pasado de discos duros con capacidades de megabytes y gigabytes en los primeros días a terabytes y, más recientemente, a discos duros de múltiples terabytes.

### 1.1.7 Tecnologías Emergentes

En la actualidad, se están explorando tecnologías emergentes para el futuro de los discos duros. Esto incluye avances en la grabación magnética asistida por calor (HAMR), que busca aumentar aún más la densidad de almacenamiento, y la grabación magnética asistida por corriente (CAMR).

### 1.1.8 Integración de Tecnologías en la Nube

La evolución de la tecnología de almacenamiento no solo se limita a dispositivos físicos. La integración de servicios en la nube ha cambiado la forma en que se accede y almacena la información, complementando y, en algunos casos, compitiendo con las soluciones de almacenamiento local.

### 1.1.9 Desarrollos Actuales y Futuros

En la actualidad, los discos duros continúan evolucionando con la introducción de tecnologías como la grabación magnética perpendicular (PMR) y el almacenamiento en capas (SMR). Además, se exploran enfoques más allá de los discos duros convencionales, como discos duros híbridos (HDD + SSD) y tecnologías de almacenamiento basadas en memorias no volátiles (NVM).

## 1.2 Conectividad Inalámbrica

El auge de la conectividad inalámbrica en computadoras y redes tiene sus raíces en la necesidad de superar las limitaciones de las conexiones por cable. Aunque las conexiones cableadas eran predominantes en los primeros días de las redes informáticas, surgieron las primeras formas de conectividad inalámbrica en la década de 1970 con experimentos que utilizaron microondas para transmitir datos.

La incorporación de la conectividad inalámbrica en dispositivos de almacenamiento ha sido una tendencia clave en la historia reciente. El proyecto se inspira en la demanda de soluciones que permitan la transferencia de datos de manera más flexible y sin las limitaciones de cables, aprovechando las redes Wi-Fi para facilitar los respaldos automáticos.

### 1.2.1 La Revolución Wi-Fi

La década de 1990 marcó un hito con el desarrollo y la estandarización de la tecnología Wi-Fi. El lanzamiento del estándar IEEE 802.11 en 1997 allanó el camino para la popularización de la conectividad inalámbrica en entornos domésticos y empresariales. La capacidad de acceder a la red sin depender de cables físicos cambió fundamentalmente la forma en que las personas interactúan con la tecnología.

### 1.2.2 Impacto en la Movilidad y la Flexibilidad

El auge de la conectividad inalámbrica ha transformado la forma en que las personas trabajan y se comunican. La movilidad y la flexibilidad se han vuelto esenciales en entornos laborales y personales. Los usuarios pueden acceder a la información y compartir archivos en tiempo real sin estar vinculados por cables.



### 1.3 Internet de las Cosas (IoT)

La integración de la detección de redes específicas para iniciar automáticamente respaldos se alinea con la filosofía de la Internet de las Cosas. El dispositivo no solo actúa como un medio de almacenamiento, sino como un elemento inteligente que responde de manera proactiva a su entorno, mejorando la experiencia del usuario.

#### 1.3.1 Precursor

El arte de la interconexión de dispositivos electrónicos y objetos cotidianos comenzó a vislumbrarse en la década de 1980. La idea de conectar objetos a la red para permitir la comunicación entre ellos fue conceptualizada en un laboratorio de investigación de Carnegie Mellon, donde se crearon los primeros dispositivos capaces de compartir información entre sí.

#### 1.3.2 Terminología y Conceptualización

La expresión "Internet de las Cosas" fue acuñada en 1999 por el investigador británico Kevin Ashton. Ashton propuso esta terminología para describir la conexión de objetos físicos a internet para recopilar y compartir datos. Esta conceptualización marcó el inicio formal del IoT como una disciplina en sí misma.

#### 1.3.3 Desarrollo Tecnológico

A medida que avanzaba la década de 2000, los avances tecnológicos en sensores, actuadores y sistemas de comunicación permitieron una mayor integración de objetos en la red. La miniaturización de componentes electrónicos y la mejora de las tecnologías inalámbricas facilitaron la conexión de una amplia gama de dispositivos.

#### 1.3.4 Crecimiento de Dispositivos Conectados

En la última década, hemos sido testigos de un rápido crecimiento en el número de dispositivos conectados. Desde electrodomésticos inteligentes y dispositivos de salud hasta sistemas de monitoreo industrial, la variedad de objetos integrados en el IoT ha ampliado enormemente, formando una red interconectada de "cosas" que recopilan y comparten datos.

### 1.4 Seguridad y Ciberseguridad

La gestión de archivos y datos en entornos en la nube demanda una atención especial para resguardar la información sensible ante amenazas tanto internas como externas. Para ello, se recurre a una variedad de medidas y tecnologías diseñadas para preservar la confidencialidad, integridad y disponibilidad de los datos.

Entre estas medidas se incluyen la autenticación multifactorial, la encriptación de extremo a extremo y la gestión de accesos basada en roles. Estas estrategias garantizan que únicamente usuarios autorizados accedan a la información y que los datos permanezcan protegidos tanto en reposo como en tránsito.

#### 1.4.1 Autenticación Multifactorial

La autenticación multifactorial es un método de seguridad que requiere la verificación de la identidad del usuario a través de múltiples factores. Estos factores suelen clasificarse en tres categorías: algo que sabe el usuario (como una contraseña), algo que tiene (como un token o tarjeta de seguridad) y algo que es (como una huella dactilar o reconocimiento facial). Los métodos comunes de autenticación multifactorial incluyen el uso de contraseñas junto con códigos de verificación enviados a dispositivos móviles, el uso de tarjetas de seguridad que generan códigos únicos y la biometría, como el escaneo de huellas dactilares o reconocimiento facial.

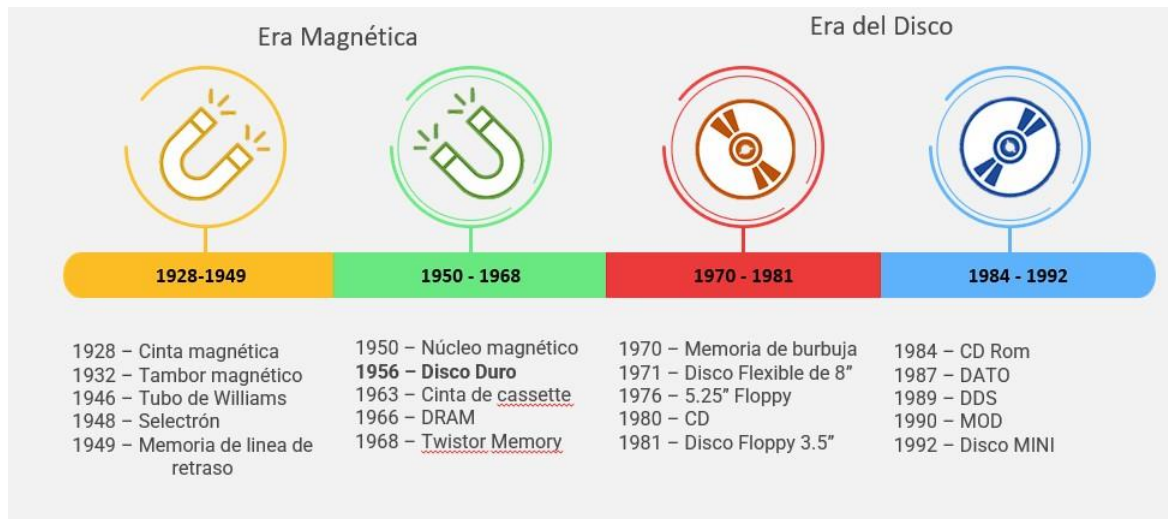
#### 1.4.2 Encriptación de Extremo a Extremo

La encriptación de extremo a extremo es un método de seguridad que protege los datos durante su transmisión y almacenamiento, garantizando que solo los remitentes y destinatarios autorizados puedan acceder a la información. Los algoritmos de encriptación comunes utilizados en la encriptación de extremo a extremo incluyen AES (Advanced Encryption Standard) y RSA (Rivest-Shamir-Adleman). Estos algoritmos son altamente seguros y utilizan claves de encriptación únicas para proteger los datos.

#### 1.4.3 Gestión de Accesos Basada en Roles

La gestión de accesos basada en roles (RBAC) es un enfoque para controlar el acceso a los recursos de la red según la función o el rol de un usuario dentro de una organización. La gestión de accesos basada en roles proporciona una manera eficiente y segura de administrar los privilegios de acceso, reduciendo el riesgo de accesos no autorizados y ayudando a garantizar la integridad y confidencialidad de los datos en la nube.

## 1.5 LÍNEA DEL TIEMPO



# **CAPÍTULO II.**

## **MARCO TEÓRICO**

## I. Elementos utilizados en el proyecto

- Raspberry Pi 4b
- SSD Sata - XrayDisc 240 Gb
- Display OLED 128x64, 0.96", SSD1306, I2C
- Baterías Li-ion 2500mAh Sony US18650GR (G6)
- Cargador de Baterías Li-Ion TP4056 Tipo C
- AMS1117-5 Regulador de voltaje 5 V
- Cable SATA a USB - USB 3.1 (10Gbps) – UASP

### 2.1 Raspberry Pi 4b

El Raspberry Pi 4 Model B (Pi4B) es el primero de una nueva generación de ordenadores compatibles con Raspberry Pi. Más RAM y con un rendimiento de CPU, GPU e E/S significativamente mejorado en un factor de forma similar, sobre de potencia y coste como la generación anterior Raspberry Pi 3B+. El Pi4B está disponible con 1, 2, 4 u 8 gigabytes de LPDDR4 SDRAM.

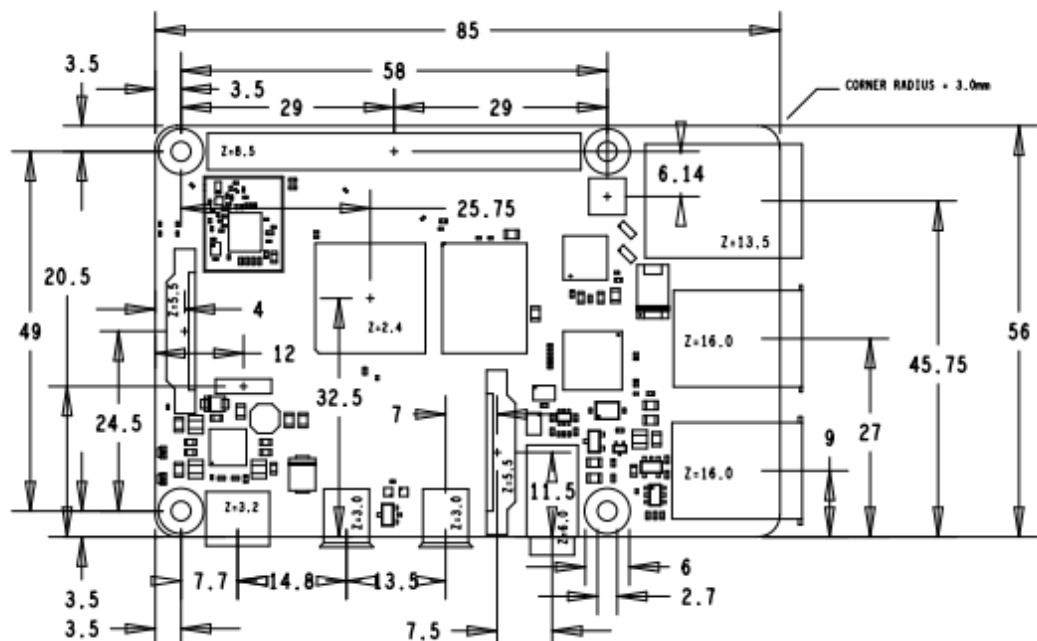
#### 2.1.1 Hardware

- Quad core ARM-Cortex A72 de 64 bits funcionando a 1,5 GHz
- Opciones de RAM LPDDR4 de 1, 2 y 4 gigabytes
- Decodificación de hardware H.265 (HEVC) (hasta 4Kp60)
- Decodificación de hardware H.264 (hasta 1080p60)
- Gráficos en 3D de VideoCore VI
- Admite doble salida de pantalla HDMI hasta 4Kp60

#### 2.1.2 Interfaces

1.5802.11 b/g/n/ac LAN inalámbrica

- Bluetooth 5.0 con BLE
- 1x tarjeta SD
- 2 puertos micro-HDMI que admiten pantallas duales de hasta 4Kp60 de resolución
- 2x puertos USB2
- 2x puertos USB3
- 1x puerto Gigabit Ethernet (soporta PoE con complemento PoE HAT)
- 1 puerto de cámara Raspberry Pi (MIPI CSI de 2 carriles)



### 2.1.5 Especificaciones Eléctricas

¡Precaución! Las tensiones mencionadas en la tabla 2 pueden causar daños permanentes al dispositivo. Esto es una clasificación de tensión únicamente; funcionamiento funcional del dispositivo en estas o en otras condiciones por encima de las no está implícito. Exposición al máximo absoluto las condiciones de habilitación durante períodos prolongados pueden afectar a la fiabilidad del dispositivo.

Símbolo	Parámetro	mínima	máxima	Unidad
VIN	Voltaje de entrada 5V	-0.5	6.0	V

Tabla 1: Calificaciones máximas absolutas

*Raspberry Pi 4 Modelo B Ficha de datos*  
*Copyright Raspberry Pi (Trading) Ltd. 2024*

Símbolo	Parámetro	Condiciones	Mínima	Típica	Máxima	Unidad
V <sub>IL</sub>	Entrada baja voltaje <sup>a</sup>	VDD IO = 3.3V	0		0.8	V
V <sub>IH</sub>	Entrada alta voltaje <sup>a</sup>	VDD IO = 3.3V	2.0		VDD IO	V
I <sub>IL</sub>	Corriente de fuga de entrada	TA = +85°C			10	μA
C <sub>IN</sub>	Capacitancia de entrada			5		pF
V <sub>OL</sub>	Salida baja de voltaje <sup>b</sup>	VDD IO = 3.3V IOL = -2mA			0.4	V
V <sub>OH</sub>	Salida alta de voltaje <sup>b</sup>	VDD IO = 3.3V IOH = 2mA	VDD IO - 0.4			V
I <sub>OL</sub>	Salida baja de corriente <sup>c</sup>	VDD IO = 3.3V VO = 0.4V	7			mA
I <sub>OH</sub>	Salida alta de corriente <sup>c</sup>	VDD IO = 3.3V VO = 2.3V	7			mA
R <sub>PU</sub>	Resistencia PullUp		18	47	73	kΩ
R <sub>PD</sub>	Resistencia PullDown		18	47	73	kΩ

- <sup>a</sup> Histéresis habilitada  
<sup>b</sup> Fuerza de accionamiento por defecto (8mA)  
<sup>c</sup> Fuerza máxima de accionamiento (16mA)

Tabla 2: Características de DC

Nombre del pin	Símbolo	Parámetro	Mínimo	Típico	Máximo	Unidad
Salida Digital	T <sub>rise</sub>	10-90% Tiempo de elevado <sup>a</sup>		TBD		ns
Salida Digital	T <sub>fall</sub>	90-10% Tiempo de caída <sup>a</sup>		TBD		ns

<sup>a</sup> Fuerza de accionamiento por defecto, CL = 5pF, VDD IO = 3.3V

Tabla 3: Características de los pines digitales I/O

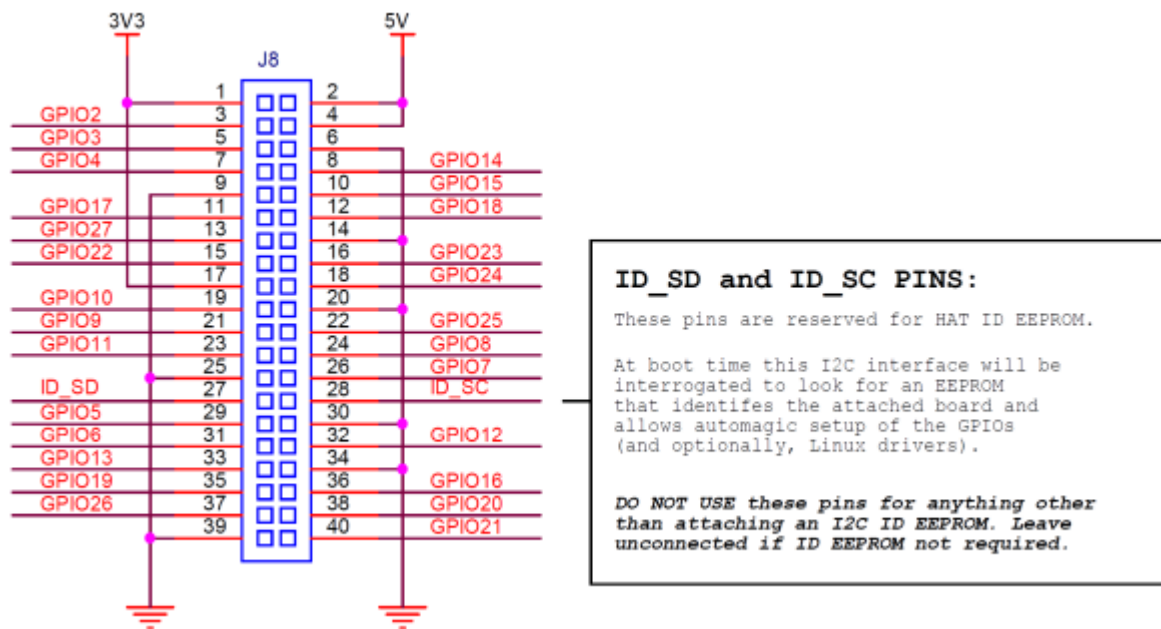


## 2.1.6 Requisitos de potencia

El Pi4B requiere una fuente de alimentación USB-C de buena calidad capaz de entregar 5V a 3A. Si los dispositivos USB conectados consumen menos de 500mA, se puede usar una fuente de 5V, 2.5A.

## 2.1.7 Periféricos

Interfaz GPIO: El Pi4B hace 28 BCM2711 GPIOs disponibles a través de un encabezado estándar de 40 pines Raspberry Pi. Este encabezado es compatible con todas las placas Raspberry Pi anteriores con un encabezado de 40 direcciones



Además de poder utilizarse como entrada y salida controladas por software (con tirones programables), los pines GPIO se pueden cambiar (multiplexados) a varios otros modos respaldados por bloques periféricos dedicados como I2C, UART y SPI.

Además de las opciones periféricas estándar que se encuentran en los Pis heredados, se han añadido periféricos adicionales I2C, UART y SPI al chip BCM2711 y están disponibles como opciones adicionales de “mux” en el Pi 4. Esto da a los usuarios mucha más flexibilidad al conectar complementos en hardware en comparación con los modelos más antiguos.

### 2.1.7.1 Pantalla de interfaz paralela (DPI)

Los GPIO disponen de una interfaz RGB (DPI) paralela estándar. Esta interfaz paralela de hasta 24 bits puede admitir una pantalla secundaria.

### 2.1.7.2 Interfaz SD/SDIO

El Pi4B tiene un zócalo de tarjeta SD dedicado que soporta 1.8V, modo DDR50 (en un ancho de banda máximo de 50 Megabytes / seg). Además, una interfaz



SDIO heredada está disponible en los pines GPIO.

#### 2.1.7.3 Interfaces de cámara y pantalla

El Pi4B tiene una cámara MIPI CSI de 2 carriles Raspberry Pi 1x y un conector MIPI DSI de 2 carriles Raspberry Pi. Estos conectores son retro compatibles con las antiguas placas Raspberry Pi, y admiten todos los periféricos de la cámara y la pantalla Raspberry Pi disponibles.

#### 2.1.7.4 USB

El Pi4B tiene 2x tomas USB2 y 2x USB3 tipo A. La corriente USB aguas abajo está limitada a aproximadamente 1.1A en conjunto sobre las cuatro tomas.

#### 2.1.7.5 HDMI

El Pi4B tiene 2 puertos micro-HDMI, ambos compatibles con CEC y HDMI 2.0 con resoluciones de hasta 4Kp60.

#### 2.1.7.6 Audio y composición (salida de TV)

El Pi4B es compatible con salida de audio analógica de calidad casi CD y salida de TV compuesta a través de un conector TRS 'A/ V' de 4 anillos.

La salida de audio analógico puede conducir auriculares de 32 ohmios directamente.

#### 2.1.7.7 Rango de temperatura y térmicas

El rango recomendado de temperatura ambiente de funcionamiento es de 0 a 50 grados Celsius.

Para reducir la salida térmica al ralentí o bajo carga ligera, el Pi4B reduce la velocidad del reloj de la CPU y el voltaje. Durante una carga más pesada la velocidad y el voltaje (y por lo tanto la salida térmica) se incrementan. El regulador interno acelerará tanto la velocidad de la CPU como el voltaje para asegurarse de que la temperatura de la CPU nunca exceda los 85 grados C.

El Pi4B funcionará perfectamente sin ningún enfriamiento adicional y está diseñado para el rendimiento de sprint - esperando un caso de uso ligero en promedio y aumentando la velocidad de la CPU cuando sea necesario (p. ej., al cargar una página web). Si un usuario desea cargar el sistema continuamente u operarlo a alta temperatura a pleno rendimiento, puede ser necesario un enfriamiento adicional.

## 2.2 SSD Sata - XrayDisc 240 Gb

El XrayDisk SSD es una unidad de estado sólido en el factor de forma de 2.5". Con el resto del sistema, las interfaces SSD de XrayDisk utilizan una conexión SATA de 6 Gbps. El controlador SSD es el YS9082HP de Yeestor, una caché DRAM no está disponible. XrayDisk ha instalado el flash NAND TLC de 232 capas en el SSD, los chips flash son hechos por SpecTek. Tenga en cuenta que este SSD se vende en múltiples variantes con diferentes flash o controlador NAND, lo que podría afectar el rendimiento, la sección "Notas" al final de esta página tiene más información. Para mejorar las velocidades de escritura, se

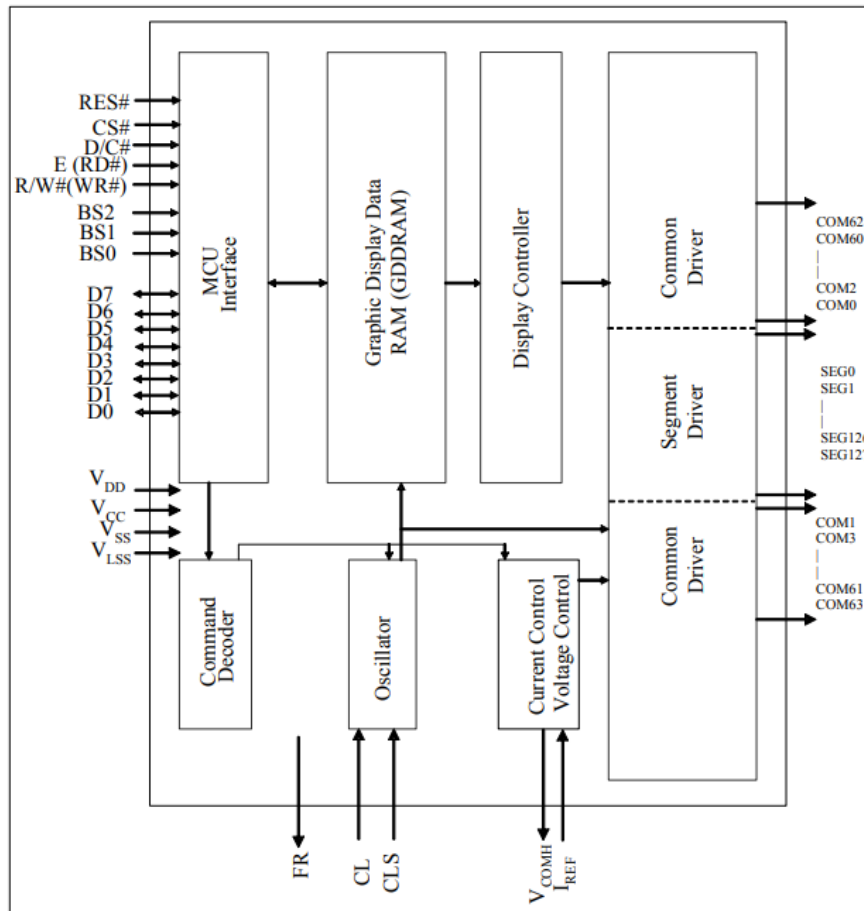
utiliza una caché pseudo-SLC, así ráfagas de escrituras entrantes se absorben más rápidamente. El SSD está clasificado para velocidades de lectura secuenciales de hasta 550 MB/s y escritura de 490 MB/s.

<b>Categoría</b>	<b>Información</b>
Marca registrada	XrayDisk
Modelo	XrayDisk 240GB SSD
Capacidad	231.9 GB
Interfaz	NVMe
Versión de NVMe	1.3
Firmware	V0926A3
Firmwares alternativos	11
Actualización de firmware	compatible
Consumo máximo de energía	6 W
Formato avanzado	no compatible
Unidades probadas	1
Veces probadas	1
Rendimiento	
Velocidad máxima de interfaz	3940 MB/s (PCIe 3.0 x4)
Velocidad máxima de lectura	1476 MB/s
Tiempo medio de acceso mínimo	0.16 milisegundos
TRIM	compatible
Búfer de memoria del host (HMB)	compatible (64 MB)
Control de temperatura de estrangulamiento	compatible
Fiabilidad	
Duración de la prueba completa	10 minutos
Temperatura máxima crítica	85°C (185°F)
Temperatura máxima recomendada	83°C (181°F)
Temperatura máxima	42°C (107°F)
Temperatura máxima promedio	42°C (107°F)
Seguridad	
Restricción de acceso al disco	compatible
Destrucción instantánea de datos	no compatible

## 2.2 Display OLED 128x64, 0.96", SSD1306, I2C

SSD1306 es un controlador OLED/PLED de un solo chip CMOS con controlador para el sistema de visualización gráfica de matriz de puntos de diodo emisor de luz/ polímero orgánico. Consta de 128 segmentos y 64commons. Este CI está diseñado para el panel OLED de tipo cátodo común.

El SSD1306 se integra con control de contraste, pantalla RAM y oscilador, lo que reduce el número de componentes externos y el consumo de energía. Tiene control de brillo de 256 pasos. Los datos/comandos se envían desde MCU general a través de la interfaz paralela compatible con la serie 6800/8000 seleccionable por hardware, Interfaz I2C o Serial Peripheral Interfaz. Es conveniente para muchas aplicaciones portátiles compactas, tales como teléfono móvil sub-display, reproductor de MP3 y calculadora, etc.



## 2.3 Baterías Li-ion 2500mAh Sony US18650GR (G6)

Esta especificación se aplicará a la batería recargable de iones de litio denominada US18650G6F. US18650G6F utilizó grafito especial diseñado por Sony como material de ánodo. El material de carbono se da cuenta de alta densidad de energía y alta seguridad.

También lleva excelente carga y temperatura característica.

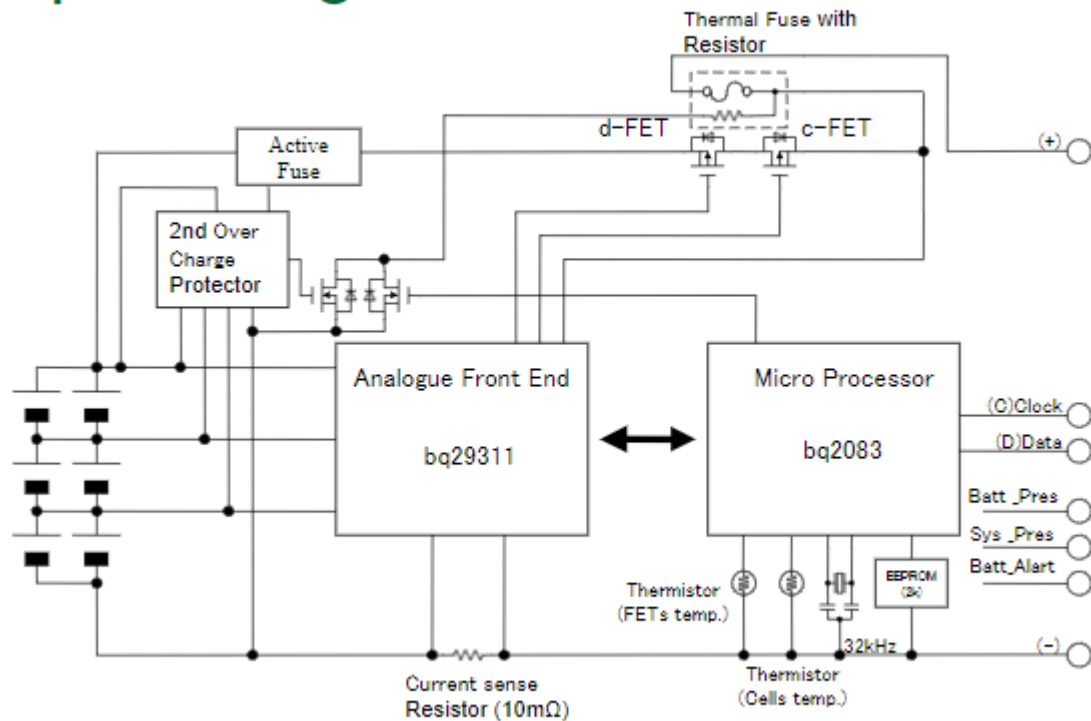
Además de eso, el cambio de estructura para aumentar el volumen de material activo trae

más aumento de la capacidad. Mediante la aplicación de la tecnología de dos cabezas, 2200mAh

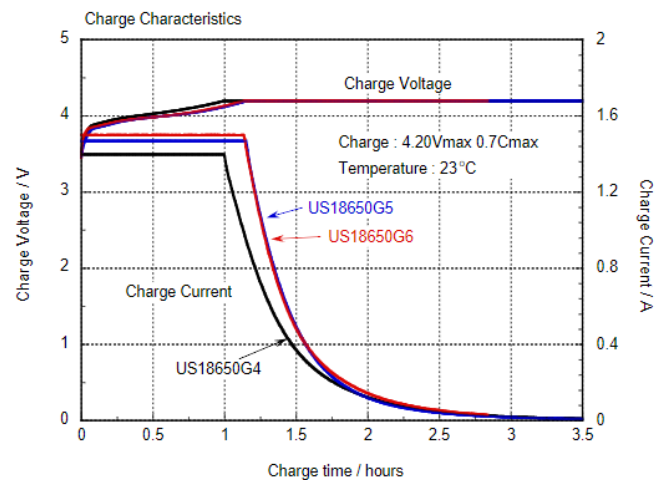
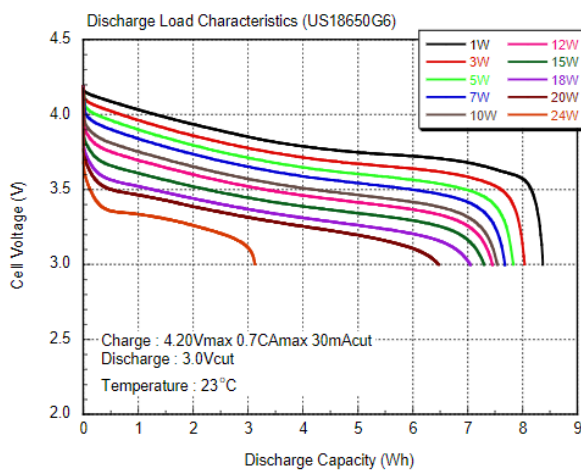
capacidad se ha logrado en tamaño 18650.

Categoría	Información
Marca	Sony
Modelo	US18650GR (G6)
Capacidad	2100mAh (Nominal)
Voltaje	3.70V (Nominal)
Carga	4.20V (Máximo), 1000mA (Estándar), --- mA (Máximo)
Descarga	3.00V (Corte), 420mA (Estándar), --- mA (Máximo)
Descripción	Envoltura de celda verde, Anillo aislante negro, Factor de forma 18650

# bq-uP Diagram



\*Battery\_pres and Sys\_pres are not necessary.



## 2.3 Cargador de Baterías Li-Ion TP4056 Tipo C

TP4056 Cargador de Baterías es un pequeño módulo que permite cargar baterías de Li-Ion o LiPo de una celda 1S. Disponible con puerto de Micro-USB y Tipo C, se venden por separado, de acuerdo con los requerimientos de tu proyecto podrás elegir el que más se ajuste a tus necesidades, ambos módulos tienen protección dual, su voltaje de entrada mediante los puertos USB es de 5V DC y corriente de carga de 1A.

El TP4056 es un controlador de carga y descarga de batería de iones de litio que se utiliza comúnmente en dispositivos portátiles y dispositivos de alimentación móvil. Es un componente electrónico que se encarga de controlar el proceso de carga y descarga de una batería de iones

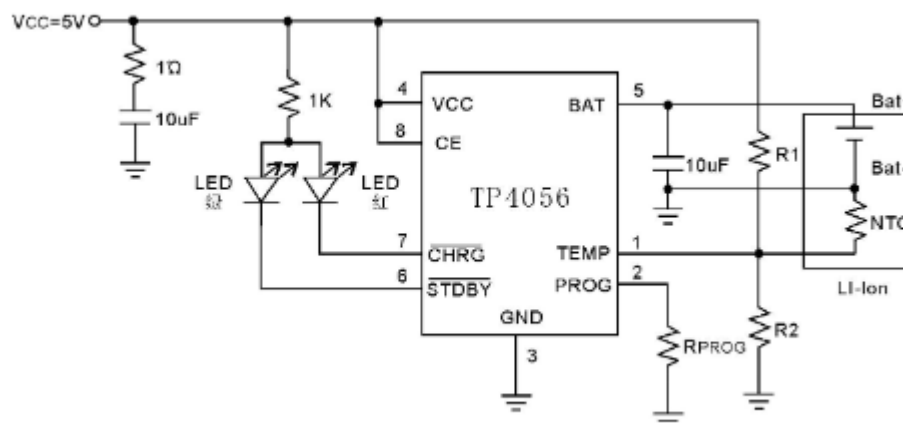
de litio, protegiéndola de daños y prolongando su vida útil. El TP4056 incluye una protección contra sobretensión, sobre temperatura y cortocircuitos, así como una función de carga constante y una función de descarga constante. También tiene un indicador LED para mostrar el estado de carga de la batería. El TP4056 es un componente compacto y eficiente que se utiliza ampliamente en aplicaciones de alimentación móvil, como cargadores de teléfonos móviles y dispositivos de alimentación portátil.

## ELECTRICAL CHARACTERISTICS

The ● denotes specifications which apply over the full operating temperature range, otherwise specifications are at  $T_A=25^{\circ}\text{C}$ ,  $V_{CC}=5\text{V}$ , unless otherwise noted.

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
$V_{CC}$	Input Supply Voltage		● 4.0	5	8.0	V
$I_{CC}$	Input Supply Current	Charge Mode, $R_{PROG} = 1.2\text{k}$	●	150	500	$\mu\text{A}$
		Standby Mode (Charge Terminated)	●	55	100	$\mu\text{A}$
		Shutdown Mode ( $R_{PROG}$ Not Connected, $V_{CC} < V_{BAT}$ , or $V_{CC} < V_{UV}$ )	●	55	100	$\mu\text{A}$
$V_{FLOAL}$	Regulated Output (Float) Voltage	$0^{\circ}\text{C} \leq T_A \leq 85^{\circ}\text{C}$ , $I_{BAT}=40\text{mA}$	4.137	4.2	4.263	V
$I_{BAT}$	BAT Pin Current Text condition: $V_{BAT}=4.0\text{V}$	$R_{PROG} = 2.4\text{k}$ , Current Mode	● 450	500	550	mA
		$R_{PROG} = 1.2\text{k}$ , Current Mode	● 950	1000	1050	mA
		Standby Mode, $V_{BAT} = 4.2\text{V}$	● 0	-2.5	-6	$\mu\text{A}$
$I_{TRIKL}$	Trickle Charge Current	$V_{BAT} < V_{TRIKL}$ , $R_{PROG}=1.2\text{K}$	● 120	130	140	mA
$V_{TRIKL}$	Trickle Charge Threshold Voltage	$R_{PROG}=1.2\text{K}$ , $V_{BAT}$ Rising	2.8	2.9	3.0	V
$V_{TRHYS}$	Trickle Charge Hysteresis Voltage	$R_{PROG}=1.2\text{K}$	60	80	100	mV
$T_{LIM}$	Junction Temperature in Constant Temperature Mode			145		$^{\circ}\text{C}$

## 典型应用电路



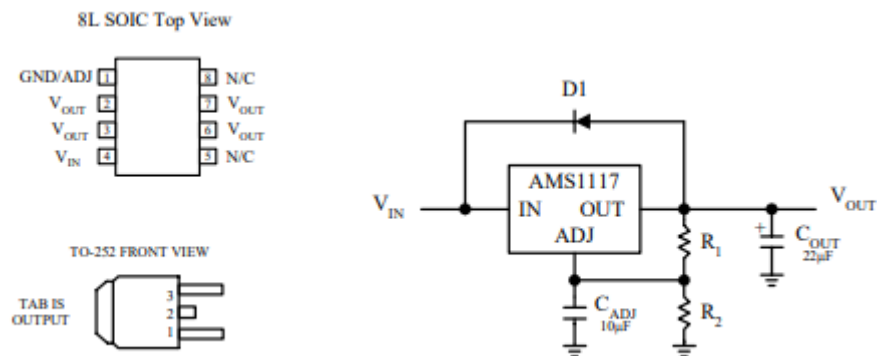
## 2.4 AMS1117-5 Regulador de voltaje 5 V

Los AMS1117 son una serie de reguladores de tensión regulables y fijos está diseñada para proporcionar corriente de salida 1A y operar hasta un diferencial de entrada a salida de 1V. El voltaje de caída del dispositivo se garantiza máximo 1.3V en la corriente de salida máxima, disminuyendo en corrientes de carga más bajas.

El recorte en el chip ajusta el voltaje de referencia al 1%. El límite de corriente también se recorta, minimizando la tensión en condiciones de sobrecarga tanto en el regulador como en los circuitos de la fuente de alimentación.

Los dispositivos AMS1117 son compatibles con otros reguladores SCSi de tres terminales y se ofrecen en el paquete de montaje superficial de bajo perfil SOT-223, en el paquete 8L SOIC y en el paquete de plástico TO-252 (DPAK).

PACKAGE TYPE			OPERATING JUNCTION TEMPERATURE RANGE
TO-252	SOT-223	8L SOIC	
AMS1117CD	AMS1117	AMS1117CS	-40 to 125° C
AMS1117CD-1.5	AMS1117-1.5	AMS1117CS-1.5	-40 to 125° C
AMS1117CD-1.8	AMS1117-1.8	AMS1117CS-1.8	-40 to 125° C
AMS1117CD-2.5	AMS1117-2.5	AMS1117CS-2.5	-40 to 125° C
AMS1117CD-2.85	AMS1117-2.85	AMS1117CS-2.85	-40 to 125° C
AMS1117CD-3.3	AMS1117-3.3	AMS1117CS-3.3	-40 to 125° C
AMS1117CD-5.0	AMS1117-5.0	AMS1117CS-5.0	-40 to 125° C



## 2.5 Cable SATA a USB - USB 3.1 (10Gbps) – UASP

Este cable SATA a USB es un adaptador de unidad externa que le ofrece acceso rápido y fácil a una unidad SATA, a través del puerto USB-A de su ordenador portátil o de sobremesa equipado con USB 3.1(10Gbps). El adaptador USB 3.1 a SATA ofrece mayor ancho de banda y velocidad, con tasas de transferencia de datos de hasta 10Gbps. Es el doble de la velocidad de USB 3.0 y 20 veces más rápido que USB 2.0. Le permite aprovechar al máximo el alto rendimiento de los modelos más recientes de unidades SSD y DD, lo cual evita los cuellos de botella en sus transferencias de datos. Además, le permite alcanzar velocidades de transferencia 70% más rápidas que USB 3.1 convencional, cuando se conecta a un ordenador compatible con UASP.

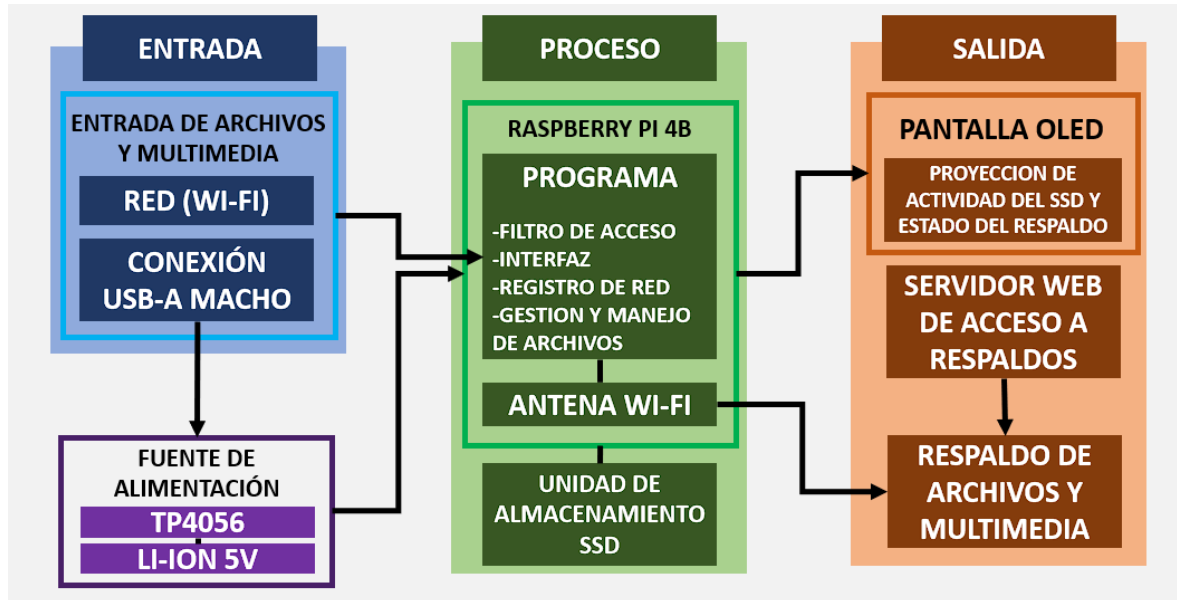
	Interfaz	USB 3.1 Gen 2
	Tamaño de la Unidad	2.5in & 3.5in
	Tipo de Bus	USB 3.1 Gen 2
	Unidades Compatibles	SATA
<b>Rendimiento</b>	4Kn Support	Sí
	Capacidad Máxima de la Unidad	Probado con discos duros de hasta 6TB
	Compatibilidad con UASP	Sí
	Soporta Hot Swap	Sí
	Tasa de Transferencia de Datos Máxima	10 Gbps
	Tipo y Velocidad	USB 3.1 Gen 2 - 10 Gbit/s SATA III (6 Gbps)
	TRIM Support	Sí
<b>Conector(es)</b>	Conectores de la Unidad	1 - Combinación de alimentación y datos por SATA (7+15 pines) Receptáculo (toma)
	Conectores del Host	1 - USB 3.1 USB Type-A ( 9 pines, Gen 2, 10Gbps) Macho
<b>Software</b>	Compatibilidad OS	Independiente del Sistema Operativo; No requiere de software o controladores
<b>Indicadores</b>	Indicadores LED	1 - Alimentación y actividad
<b>Requisitos de Energía</b>	Adaptador de Corriente Incluido	Adaptador de alimentación de CA incluido
	Alimentación de Salida	2.0 A
	Consumo de Energía	24
	Corriente de Entrada	0.6 A
	Polaridad Central	Positivo
	Tensión de Entrada	110V-240V AC
	Tensión de Salida	12 DC

# **CAPÍTULO III.**

## **DISEÑO E IMPLEMENTACIÓN**



### 3.1 DIAGRAMA DE BLOQUES



*Ilustración 1: Diagrama a Bloques*

En el diagrama a bloques se ven representadas tres etapas primordiales del proyecto, que son entrada, proceso y salida, así como un anexo para la etapa de alimentación.

Para la etapa de entrada se tienen descritos los dos procesos de entrada de datos, dónde se empezará la transferencia de estos, iniciando por la conexión de red WI-FI y la conexión física, USB-A Macho.

La etapa de alimentación nos permite tener en cuenta la manera en la que el dispositivo nos permite manejar sus pertinentes funciones de manera portátil, es decir, sin la necesidad de un cableado externo para su función, dicha etapa está conformada por un arreglo mixto “serie-paralelo” de baterías 18650 controlando su salida a los 5v de trabajo del dispositivo por medio de un TP5056 que funciona como controlador de carga para baterías LI-ION.

Pasando por el procesos, dónde nuestro núcleo de dicho sector es el microprocesador Raspberry pi 4B, que centraliza la operación del proyecto, concentrando en sí, el código pertinente para el desarrollo del filtro de acceso que permite saber y obtener el registro de los usuarios de los respaldos y archivos, la interfaz que permite conectar de manera más directa y amigable el dispositivo a la red pertinente, así como a la dirección IP dónde los respaldos serán depositados de manera fija, el registro de la red dónde se estaría conectando el dispositivo y la gestión de archivos a respaldar.

Por último, en la etapa de salida, a modo de proyección del proceso funciona un display oled SSD1306 que nos permite saber en tiempo real los procedimientos llevados a cabo por el dispositivo, incluyendo también una proyección de informes para errores, tanto de conexión, como de error de respaldo.

### 3.2 DIAGRAMA DE FLUJO

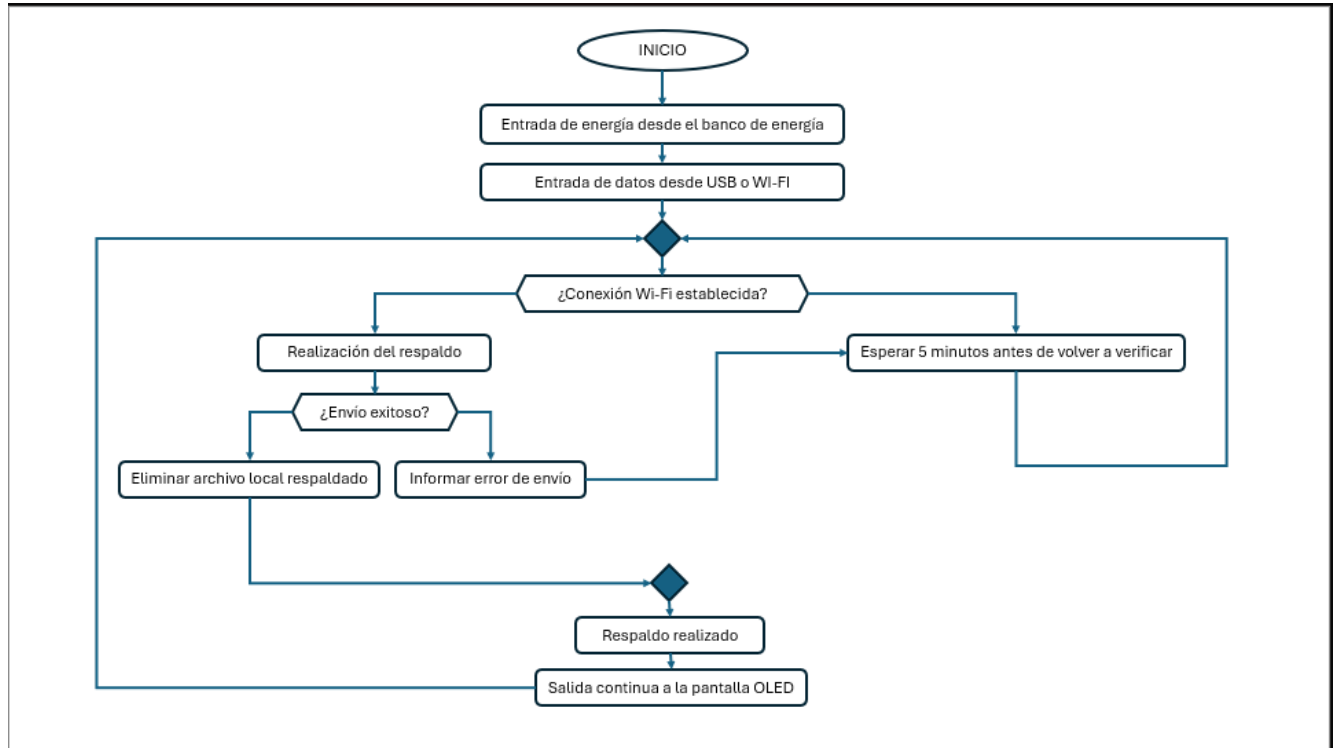
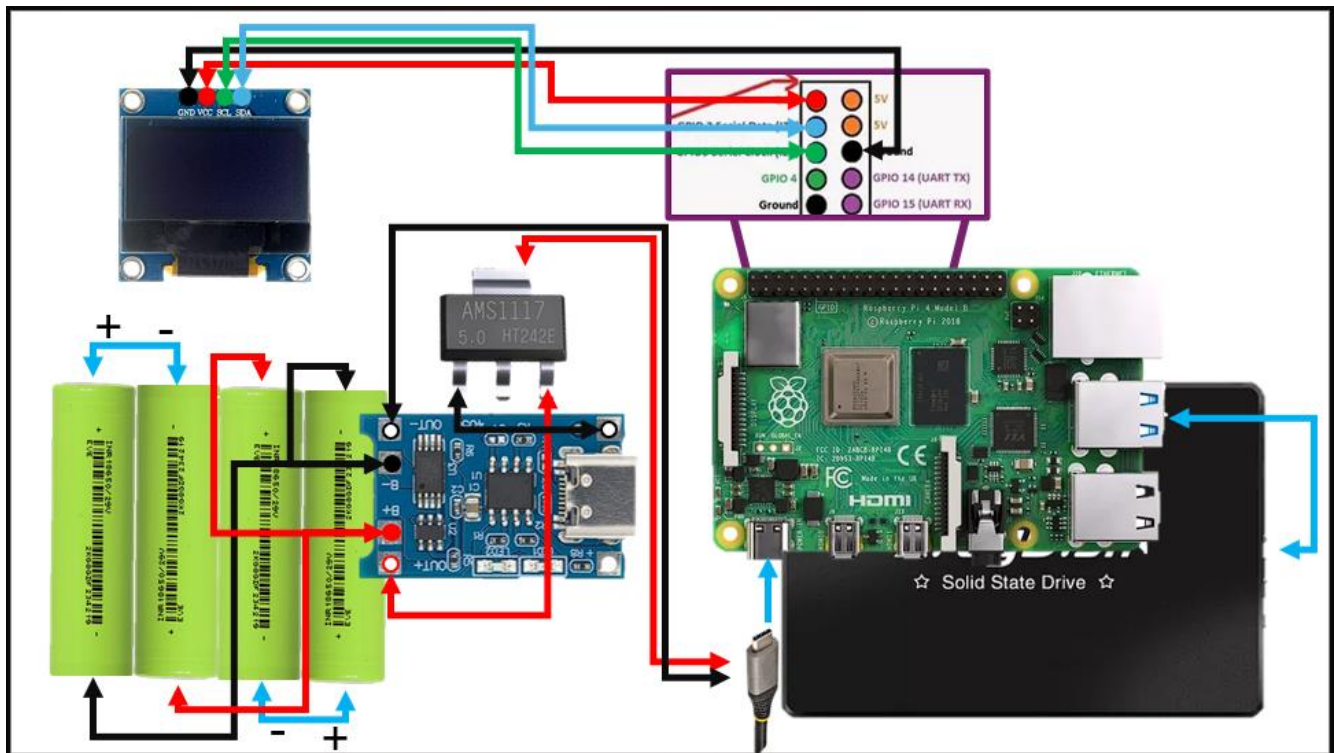


Diagrama 1: Diagrama de Flujo

### 3.3 DIAGRAMA DE CONEXIÓN



*Diagrama 2: Diagrama de conexión*

### 3.4 DIAGRAMA ESQUEMÁTICO 3D

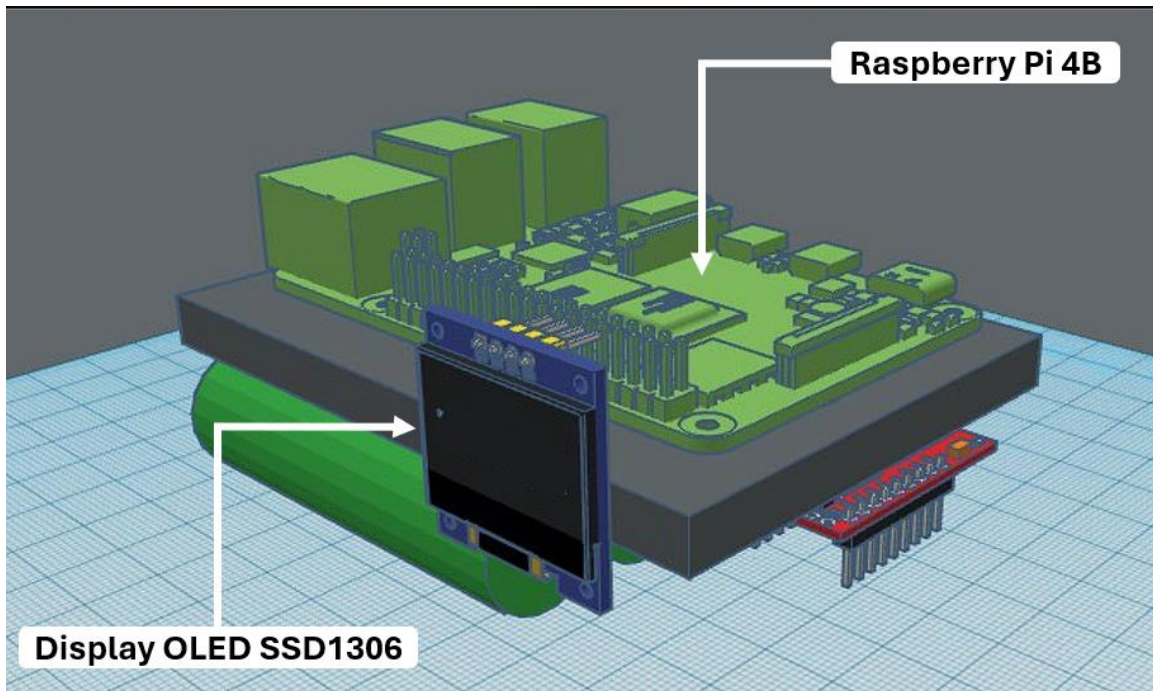


Diagrama 3: Vista 1 del Diagrama esquemático 3D

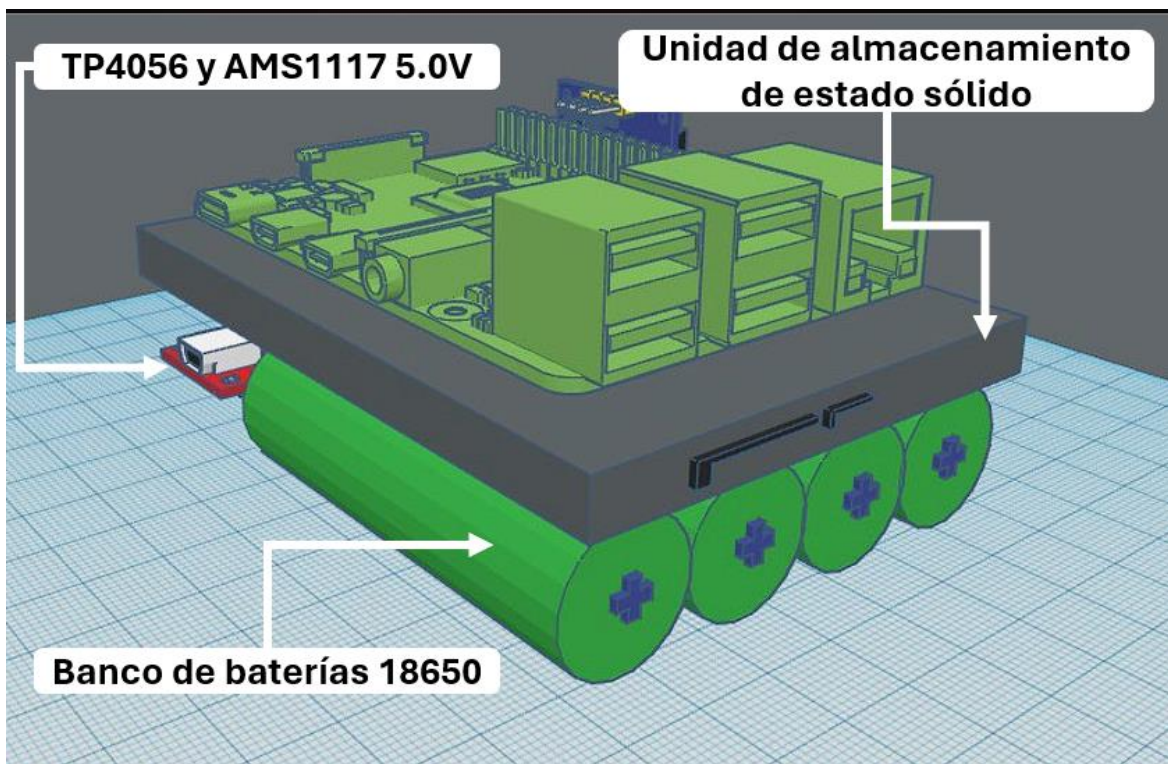


Diagrama 4: Vista 2 del Diagrama esquemático 3D

### 3.5 CÁLCULOS DE DISEÑO

Se mostrarán los cálculos pertinentes para el funcionamiento específico del banco de batería, que permitirá la portabilidad del dispositivo.

Primero, necesitaremos estimar el consumo total de energía por hora de todos los dispositivos. Asumiendo que la Raspberry Pi 4B consume alrededor de 3-5W, el SSD y el display OLED también tienen sus propios consumos de energía, pero para simplificar, tomemos un consumo total de 10W para todos los dispositivos juntos.

El cálculo para el consumo de energía por hora sería:

$$\text{Consumo total} = 10 \text{ W}$$

Para convertir este consumo de energía a miliamperios (mA), usaremos la fórmula:

*Fórmula 1: Cálculo para el consumo total en mA (Miliamperes)*

$$\text{Consumo total (mA)} = \frac{\text{Consumo total (W)}}{\text{Tensión de las baterías (V)}}$$

La Raspberry Pi 4B generalmente funciona con 5V, por lo que podemos usar 5V como la tensión de las baterías para este cálculo.

*Fórmula 2: Sustitución para el consumo total en mA (Miliamperes)*

$$\text{Consumo total (mA)} = \frac{10 \text{ W}}{5 \text{ V}} = 2000 \text{ mA}$$

Ahora que sabemos que el consumo total de energía por hora es de aproximadamente 2000mAh, podemos calcular una estimación del tiempo de funcionamiento dividiendo la capacidad total de las baterías por el consumo total de energía por hora:

*Fórmula 3: Cálculo para el tiempo de funcionamiento en Horas*

$$\text{Tiempo de funcionamiento (Horas)} = \frac{2100 \text{ mAh}}{2000 \text{ mAh}} = 1.05 \text{ Horas}$$

Al conectar dos baterías en serie y luego ese par en paralelo con otro par en serie, estás aumentando tanto la capacidad total como el voltaje total suministrado.

En el arreglo serie-paralelo, el voltaje total sería la suma del voltaje de las dos baterías en serie. Sin embargo, la capacidad total sería la suma de las capacidades de las dos baterías en serie, multiplicada por dos (debido a la conexión en paralelo).

Si cada batería tiene una capacidad de 2100mAh, en el arreglo serie-paralelo tendríamos una capacidad total de 4200mAh debido a la conexión en paralelo.

Entonces, usando la misma fórmula utilizada anteriormente para calcular el tiempo de funcionamiento con la nueva capacidad total de las baterías (4200mAh), el tiempo de funcionamiento sería:

*Fórmula 4: Sustitución para cálculo del tiempo de funcionamiento*

$$\text{Tiempo de funcionamiento (Horas)} = \frac{4200 \text{ mAh}}{2000 \text{ mAh}} = 2.1 \text{ Horas}$$



### 3.6 CÓDIGO DEL PROGRAMA

```
1 import os
2 import subprocess
3 import time
4 from luma.core.interface.serial import i2c
5 from luma.core.render import canvas
6 from luma.oled.device import ssd1306
7
8 # Direccion IP especifica a la que se enviaron los respaldos
9 ip_destino = "192.168.1.253"
10
11 # Ruta del directorio que deseas respaldar
12 directorio_respaldar = "/home/pi/media/joshux07/Rasp/Respaldos"
13
14 # Inicialización de la pantalla OLED SSD1306
15 serial = i2c(port=1, address=0x3C)
16 device = ssd1306(serial)
17
18 # Comprobacion de la conexion a la red WiFi
19 def esta_conectado():
20     try:
21         output = subprocess.check_output(["iwgetid"]).decode("utf-8")
22         if "ESSID" in output:
23             return True
24         else:
25             return False
26     except Exception as e:
27         print("Error al comprobar la conexion WiFi:", e)
28         return False
29
30 # Realizacion del respaldo
31 def realizar_respaldo():
32     try:
33         # Nombre del archivo de respaldo con la fecha y hora actual
34         nombre_archivo = time.strftime("%Y%m%d-%H%M%S") + "_respaldo.tar.gz"
35
36         # Comando para crear el archivo de respaldo
37         comando_respaldo = "tar -czf {} {}".format(nombre_archivo, directorio_respaldar)
38
39         # Ejecutar el comando de respaldo
40         os.system(comando_respaldo)
41
42         # Enviar el archivo de respaldo a la IP de destino
43         comando_envio = "scp {} usuario@{}:/ruta/destino".format(nombre_archivo, ip_destino)
44         os.system(comando_envio)
45
46         # Eliminar el archivo de respaldo localmente
47         os.remove(nombre_archivo)
48
49         # Mostrar mensaje en la pantalla OLED
50         with canvas(device) as draw:
51             draw.text((0, 0), "Respaldo hecho", fill="white")
52     
```

```

53     print("Respaldo realizado y enviado exitosamente a", ip_destino)
54 except Exception as e:
55     # Mostrar mensaje de error en la pantalla OLED
56     with canvas(device) as draw:
57         draw.text((0, 0), "Error en el respaldo", fill="white")
58
59     print("Error al realizar el respaldo:", e)
60
61 # Bucle principal
62 while True:
63     if esta_conectado():
64         print("Conectado a la red WiFi.")
65         realizar_respaldo()
66     else:
67         print("No se detecto conexion a la red WiFi.")
68     # Esperar 5 minutos antes de volver a comprobar
69     time.sleep(300) # 300 segundos = 5 minutos
70

```

### 3.6.1 Bloque de importaciones:

```

1  import os
2  import subprocess
3  import time
4  from luma.core.interface.serial import i2c
5  from luma.core.render import canvas
6  from luma.oled.device import ssd1306

```

En este bloque, importamos los módulos necesarios para nuestro programa:

*os*: Proporciona funciones para interactuar con el sistema operativo, como ejecutar comandos del sistema y manipular archivos.

*subprocess*: Permite crear procesos, ejecutar comandos y comunicarse con ellos.

*time*: Proporciona funciones relacionadas con el tiempo, como obtener la hora actual y pausar la ejecución del programa.

*i2c*, *canvas* y *ssd1306* son clases y métodos proporcionados por la biblioteca *luma.oled*, que nos permitirá interactuar con la pantalla OLED SSD1306.

### 3.6.2 Definición de variables:

```

8  # Direccion IP especifica a la que se enviaron los respaldos
9  ip_destino = "192.168.1.253"
10
11 # Ruta del directorio que deseas respaldar
12 directorio_respaldo = "/home/pi/media/joshux07/Rasp/Respaldos"

```

Aquí establecemos la dirección IP a la que se enviarán los respaldos (*ip\_destino*) y la ruta del directorio que se respaldará (*directorio\_respaldo*).

### 3.6.3 Inicialización de la pantalla OLED:

```
14 # Inicialización de la pantalla OLED SSD1306
15 serial = i2c(port=1, address=0x3C)
16 device = ssd1306(serial)
```

Creamos una instancia de la clase `i2c` para establecer la comunicación a través del bus I2C en el puerto 1 y con la dirección 0x3C (dirección predeterminada para la pantalla OLED SSD1306). Luego, creamos un objeto `device` de la clase `ssd1306` para interactuar con la pantalla.

### Función para comprobar la conexión WiFi:

```
19 def esta_conectado():
20     try:
21         output = subprocess.check_output(["iwgetid"]).decode("utf-8")
22         if "ESSID" in output:
23             return True
24         else:
25             return False
26     except Exception as e:
27         print("Error al comprobar la conexión WiFi:", e)
28     return False
```

Esta función (`esta_conectado`) verifica si la Raspberry Pi está conectada a una red WiFi. Utiliza el comando `iwgetid` para obtener información sobre la conexión WiFi. Si la conexión está activa, devuelve `True`; de lo contrario, devuelve `False`.

`subprocess.check_output(["iwgetid"])`: Esta parte ejecuta el comando `iwgetid`, que normalmente se utiliza para obtener información sobre la red inalámbrica a la que está conectado el dispositivo. Este comando puede proporcionar información como el SSID (nombre de la red) a la que está conectado el dispositivo.

`.decode("utf-8")`: La función `.decode()` se utiliza para decodificar una secuencia de bytes en una cadena Unicode utilizando el formato especificado. En este caso, estamos decodificando la salida del comando `iwgetid` utilizando UTF-8, que es un formato de codificación de caracteres ampliamente utilizado y compatible con una amplia gama de caracteres y símbolos.

### 3.6.4 Función para realizar el respaldo:

```

31 def realizar_respaldo():
32     try:
33         # Nombre del archivo de respaldo con la fecha y hora actual
34         nombre_archivo = time.strftime("%Y%m%d-%H%M%S") + "_respaldo.tar.gz"
35
36         # Comando para crear el archivo de respaldo
37         comando_respaldo = "tar -czf {} {}".format(nombre_archivo, directorio_respaldar)
38
39         # Ejecutar el comando de respaldo
40         os.system(comando_respaldo)
41
42         # Enviar el archivo de respaldo a la IP de destino
43         comando_envio = "scp {} usuario@{}:/ruta/destino".format(nombre_archivo, ip_destino)
44         os.system(comando_envio)
45
46         # Eliminar el archivo de respaldo localmente
47         os.remove(nombre_archivo)
48
49         # Mostrar mensaje en la pantalla OLED
50         with canvas(device) as draw:
51             draw.text((0, 0), "Respaldo hecho", fill="white")
52
53         print("Respaldo realizado y enviado exitosamente a", ip_destino)
54     except Exception as e:
55         # Mostrar mensaje de error en la pantalla OLED
56         with canvas(device) as draw:
57             draw.text((0, 0), "Error en el respaldo", fill="white")
58
59         print("Error al realizar el respaldo:", e)

```

Esta función (realizar\_respaldo) realiza el respaldo del directorio especificado (directorio\_respaldar). Primero, crea un nombre de archivo único para el respaldo basado en la fecha y hora actual. Luego, utiliza el comando tar para crear un archivo comprimido del directorio. A continuación, utiliza scp para enviar el archivo de respaldo a la dirección IP especificada (ip\_destino). Si el respaldo se realiza correctamente, muestra "Respaldo hecho" en la pantalla OLED y lo imprime en la consola. En caso de error, muestra "Error en el respaldo" en la pantalla OLED y lo imprime en la consola.

nombre\_archivo = time.strftime("%Y%m%d-%H%M%S") + "\_respaldo.tar.gz":

time.strftime("%Y%m%d-%H%M%S"): Utiliza la función strftime del módulo time para formatear la hora actual según el formato especificado. %Y representa el año con cuatro dígitos, %m representa el mes, %d representa el día, %H representa la hora en formato 24 horas, %M representa los minutos y %S representa los segundos. Por ejemplo, si la hora actual es 14:30:15 del 1 de enero de 2024, esta parte del código produciría la cadena "20240101-143015".

"\_respaldo.tar.gz": Es una cadena constante que se concatena con la cadena formateada anteriormente para crear el nombre completo del archivo de respaldo. En este caso, el nombre del archivo de respaldo será algo como "20240101-143015\_respaldo.tar.gz". Esta convención de nombres es común para incluir la marca de tiempo en el nombre del archivo para que sea único y fácil de identificar.

comando\_respaldo = "tar -czf {} {}".format(nombre\_archivo, directorio\_respaldar)":

"tar -czf {} {}".format(nombre\_archivo, directorio\_respaldar)": Este es un comando de shell que utiliza el programa tar para crear un archivo comprimido (gzip) que contiene los archivos del directorio especificado.



{}}: En la cadena de formato, {} son marcadores de posición que se llenan con los valores proporcionados en el método `format()`.

`nombre_archivo`: Este es el primer argumento pasado a `format()`, por lo que se sustituye en el primer {} en la cadena de formato. Esto proporciona el nombre del archivo que se creará.

`directorio_respaldo`: Este es el segundo argumento pasado a `format()`, por lo que se sustituye en el segundo {} en la cadena de formato. Esto proporciona la ruta del directorio que se comprimirá en el archivo de respaldo.

En resumen, esta línea de código construye un comando de shell que utiliza `tar` para comprimir el contenido del directorio `directorio_respaldo` en un archivo con nombre `nombre_archivo`.

### 3.6.5 Bucle principal:

```
62 while True:|
63     if esta_conectado():
64         print("Conectado a la red WiFi.")
65         realizar_respaldo()
66     else:
67         print("No se detecto conexion a la red WiFi.")
68     # Esperar 5 minutos antes de volver a comprobar
69     time.sleep(300) # 300 segundos = 5 minutos
```

Este bucle principal se ejecuta continuamente. En cada iteración, verifica si la Raspberry Pi está conectada a la red WiFi. Si está conectada, realiza el respaldo llamando a la función `realizar_respaldo()`. Si no está conectada, muestra un mensaje indicando que no se detectó conexión a la red WiFi. Después de cada iteración, el programa espera 5 minutos antes de volver a verificar la conexión WiFi.

## 3.7 CÓDIGO DE LA PÁGINA.

En este código se crea una página de inicio de sesión que valida los campos de usuario y contraseña ingresados y redirige al usuario a otra página si la información es correcta, creando así, un filtro de seguridad básico pero funcional para la consulta de datos respaldados.

### 3.7.1 Declaración del HTML

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Iniciar Sesión</title>
7     <style>
8         body {
9             font-family: Arial, sans-serif;
10            margin: 0;
11            padding: 0;
12            background-color: #f4f4f4;
13        }
14        .container {
15            width: 300px;
16            margin: 100px auto;
17            background-color: #fff;
18            padding: 20px;
19            border-radius: 5px;
20            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
21        }
22        h1 {
23            text-align: center;
```

```

24     }
25     input[type="text"],
26     input[type="password"] {
27         width: calc(100% - 20px);
28         padding: 10px;
29         margin: 10px 0;
30         border: 1px solid #ccc;
31         border-radius: 3px;
32     }
33     input[type="submit"] {
34         width: 100%;
35         background-color: #007bff;
36         color: #fff;
37         border: none;
38         padding: 10px;
39         border-radius: 3px;
40         cursor: pointer;
41     }
42     input[type="submit"]:hover {
43         background-color: #0056b3;
44     }

```

Aquí, es dónde se inicia un documento HTML con la declaración Doctype, la etiqueta <html> con el atributo lang establecido en "es" (español) se abre para indicar el idioma principal del contenido. Se define la etiqueta <meta> para especificar la escala inicial del contenido y el ancho de la ventana de visualización estableciendo el título de la página como "Iniciar Sesión". Se incluyen los estilos CSS dentro de la etiqueta <style>.

### 3.7.2 Interfaz y declaración de seguridad.

```

46 </head>
47 <body>
48     <div class="container">
49         <h1>Disco Duro de Respaldo Automático</h1>
50         <form onsubmit="return validarFormulario()" method="POST">
51             <input type="text" id="usuario" placeholder="Usuario" required>
52             <input type="password" id="contrasena" placeholder="Contraseña" required>
53             <input type="submit" value="Iniciar Sesión">
54         </form>
55     </div>
56

```

Se crea un contenedor con la clase "container" agregando un encabezado <h1> con el texto "Disco Duro de Respaldo Automático", posterior se inicia un formulario con el atributo *onsubmit* que llama a la función *validarFormulario()* cuando se envía el formulario y el método POST en el que se incluyen dos campos de entrada (<input>): uno para el usuario y otro para la contraseña, ambos con la etiqueta *required* para hacerlos obligatorios y por último, se agrega un botón de envío (<input type="submit">) con el valor "Iniciar Sesión".

### 3.7.3 Script en JavaScript para la función de seguridad

```

57     <script>
58         function validarFormulario() {
59             var usuario = document.getElementById("usuario").value;
60             var contrasena = document.getElementById("contrasena").value;
61
62             // Comparar usuario y contraseña con valores esperados
63             if (usuario === "Admin" && contrasena === "0000ABC") {
64                 alert("Inicio de sesión exitoso como Admin");
65                 window.location.href = "index3.html"; // Redirigir a index3.html
66                 return false; // Detener el envío del formulario
67             } else if (usuario === "JoshuaCast" && contrasena === "1234567") {
68                 alert("Inicio de sesión exitoso como JoshuaCast");
69                 window.location.href = "index3.html"; // Redirigir a index3.html
70                 return false; // Detener el envío del formulario
71             } else if (usuario === "Usuario" && contrasena === "Contraseña") {
72                 alert("Inicio de sesión exitoso como Usuario");
73                 window.location.href = "index3.html"; // Redirigir a index3.html
74                 return false; // Detener el envío del formulario
75             } else {
76                 alert("Usuario o contraseña incorrectos");
77                 return false; // Detener el envío del formulario
78             }
79         }
80     </script>
81 </body>

```

Se define una función llamada *validarFormulario()* que se activa cuando el formulario es enviado, dentro de esta función, se obtienen los valores ingresados en los campos de usuario y contraseña comparando estos valores con algunos valores predefinidos.

Si los valores coinciden con alguna combinación predefinida, se muestra un mensaje de alerta indicando un inicio de sesión exitoso y se redirige al usuario a "*index3.html*": si no hay coincidencias, se muestra un mensaje de alerta indicando que el usuario o la contraseña son incorrectos y se detiene el envío del formulario.

### 3.7.4 Pie de página

```

82     <footer>
83         &copy;Copyright <strong> <span>IPN; ESIME CULHUACÁN; JOSHUA CASTILLO, 2024</span> </strong>
84     </footer>
85 </html>

```

Se cierra la etiqueta *<body>* y *<html>* y se incluye un pie de página (*<footer>*) con información de derechos de autor y atribución.

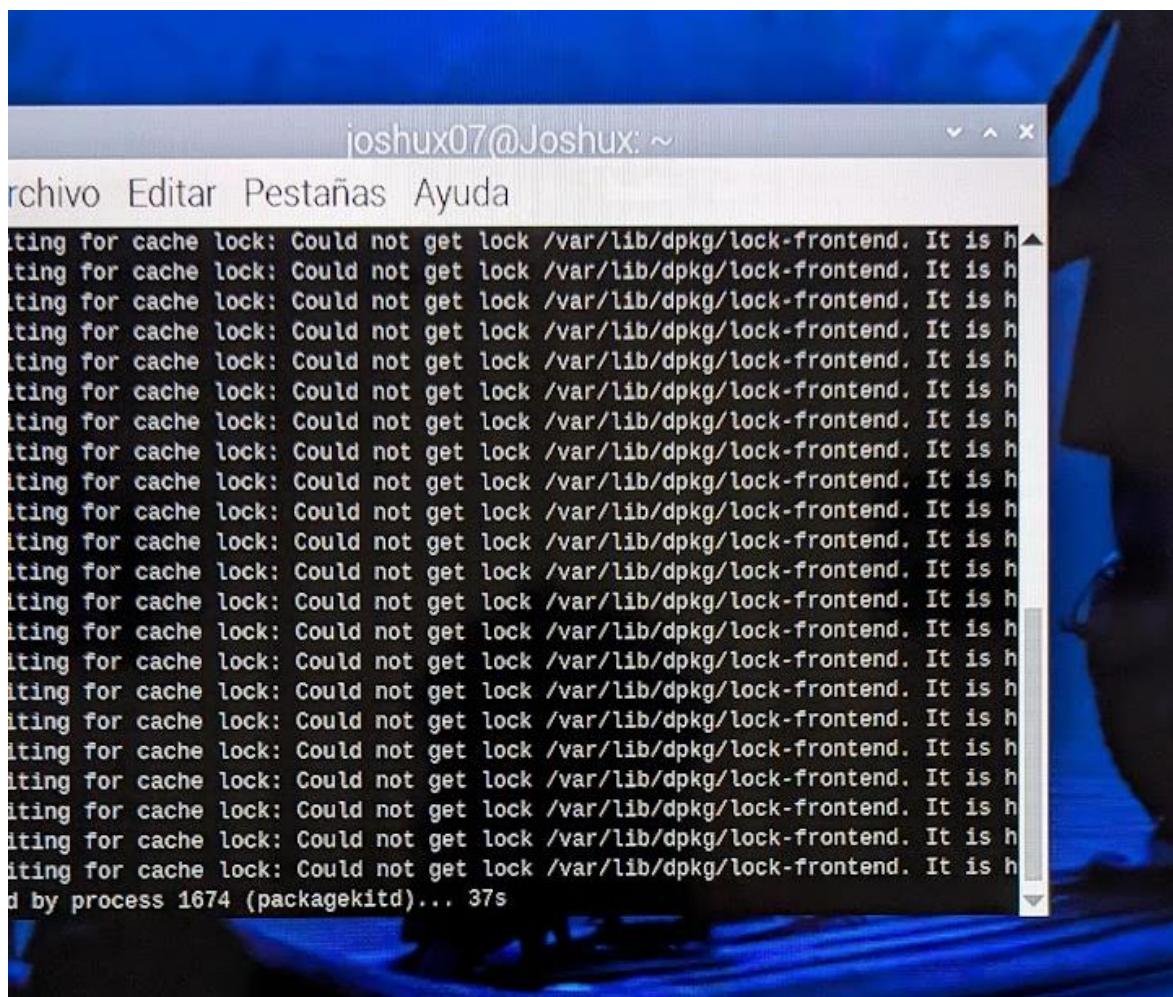
# **CAPÍTULO IV.**

## **PRUEBAS Y RESULTADOS**

## 4.1 PRUEBAS EN EL CÓDIGO

En el desarrollo del proyecto, nos encontramos con el primer obstáculo al interactuar con el monitor de la Raspberry Pi, la plataforma sobre la cual se ejecuta el código. RaspberryOS, siendo un sistema operativo relativamente novedoso en comparación con otros más establecidos, nos presentó un desafío inesperado y significativo. Es crucial destacar este punto dado que las soluciones disponibles para sistemas tan recientes son limitadas y, en muchos casos, se encuentran en un estado de continua evolución y desarrollo.

### 4.1.1 PRUEBAS Y ERRORES EN LA INTERFÁZ



*Ilustración 2: Marca de error de Raspbian:*

El problema inicial surgió al tratar de integrar nuestro código con los controladores equivalentes disponibles para el sistema. Este proceso se reveló como un verdadero dolor de cabeza, ya que los controladores existentes no eran compatibles con el nuevo código que estábamos implementando para el arranque. Este inconveniente marcó únicamente el inicio de una serie de dificultades que tuvimos que enfrentar durante el desarrollo de nuestro proyecto.

Una solución temporal que encontramos para sortear este problema fue reiniciar forzosamente la Raspberry Pi. Si bien esta medida permitía continuar con el trabajo, era una solución poco práctica y, en última instancia, insostenible a largo plazo. Nos enfrentamos a la necesidad de encontrar una solución más definitiva y efectiva para garantizar el funcionamiento fluido de nuestro proyecto en la Raspberry Pi.



```
Seleccionando...
Preparando para desempaque...
Desempaquetando libjpeg62-turbo-dev:arm64 (1:2.1.5-2) ...
Seleccionando el paquete libjpeg-dev:arm64 previamente no seleccionado
Preparando para desempaquetar .../5-libjpeg-dev_1%3a2.1.5-2_arm64.deb ...
Desempaquetando libjpeg-dev:arm64 (1:2.1.5-2) ...
Seleccionando el paquete libpng-tools previamente no seleccionado.
Preparando para desempaquetar .../6-libpng-tools_1.6.39-2_arm64.deb ...
Desempaquetando libpng-tools (1.6.39-2) ...
Configurando libpng-tools (1.6.39-2) ...
Configurando libjpeg62-turbo-dev:arm64 (1:2.1.5-2) ...
Configurando libjpeg-dev:arm64 (1:2.1.5-2) ...
Configurando libbrotli-dev:arm64 (1.0.9-2+b6) ...
Configurando libjpeg-dev:arm64 (1:2.1.5-2) ...
Configurando libfreetype-dev:arm64 (2.12.1+dfsg-5) ...
Configurando libfreetype6-dev:arm64 (2.12.1+dfsg-5) ...
Procesando disparadores para man-db (2.11.2-2) ...
joshux07@Joshux:~$ sudo pip3 install adafruit-circuitpython-ssd1306 adafruit-blinka
error: externally-managed-environment

× This environment is externally managed
╵ To install Python packages system-wide, try apt install
  python3-xyz, where xyz is the package you are trying to
  install.

  If you wish to install a non-Debian-packaged Python package,
  create a virtual environment using python3 -m venv path/to/venv.
  Then use path/to/venv/bin/python and path/to/venv/bin/pip. Make
  sure you have python3-full installed.

  For more information visit http://rptl.io/venv

note: If you believe this is a mistake, please contact your Python installation or OS distribution
hint: See PEP 668 for the detailed specification.
joshux07@Joshux:~$ sudo apt update
sudo apt install python3-adafruit-circuitpython-ssd1306 python3-adafruit-blinka
Obj:1 http://deb.debian.org/debian bookworm InRelease
Obj:2 http://deb.debian.org/debian-security bookworm-security InRelease
Obj:3 http://deb.debian.org/debian bookworm-updates InRelease
Obj:4 http://archive.raspberrypi.com/debian bookworm InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Todos los paquetes están actualizados.
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
E: No se ha podido localizar el paquete python3-adafruit-circuitpython-ssd1306
joshux07@Joshux:~$ sudo apt install python3-adafruit-blinka
E: No se ha podido localizar el paquete python3-adafruit-blinka
```

Ilustración 3: Etapa 2 del error

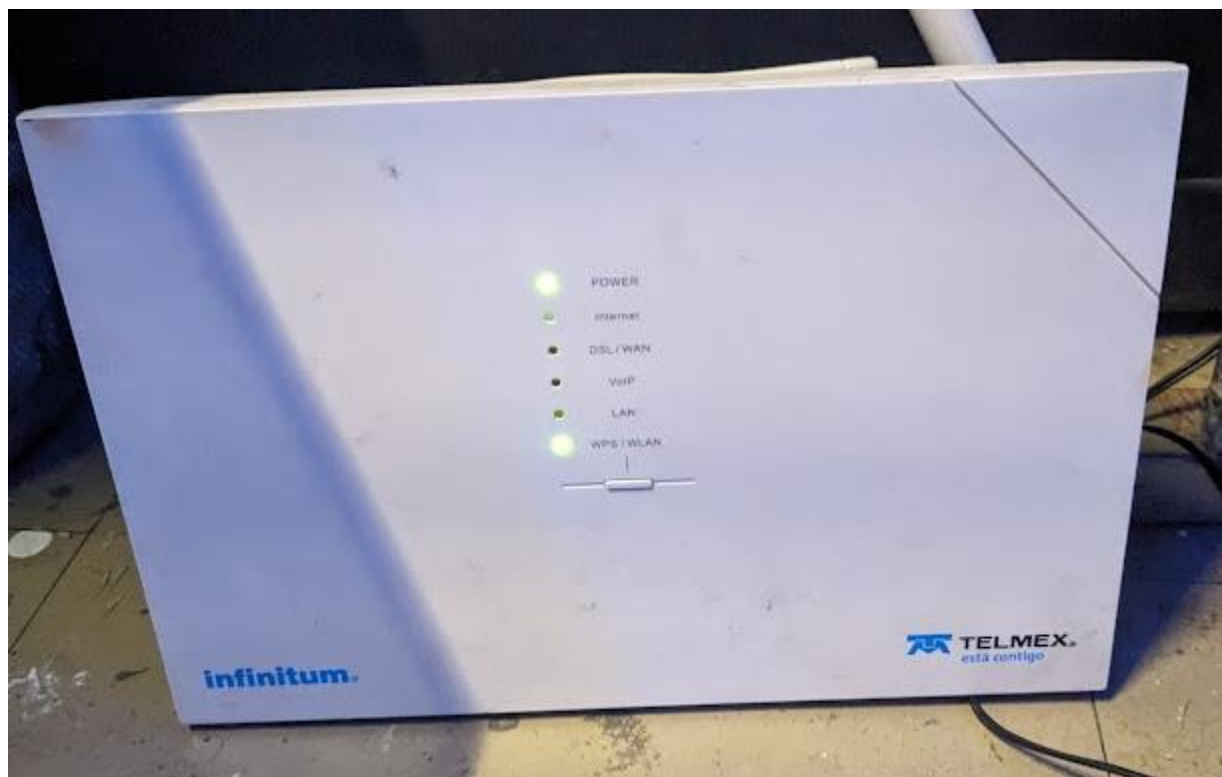
A medida que avanzábamos en nuestro proyecto, nos dimos cuenta de que el problema inicial no era un incidente aislado, sino más bien el preludio de un desafío mucho mayor. Con el tiempo, el error inicial comenzó a manifestarse de manera más pronunciada, hasta que finalmente se volvió completamente imposible iniciar la Raspberry Pi de manera tradicional.

Una vez más, el corazón del problema residía en la falta de soporte para el nuevo sistema operativo de Raspberry, ahora conocido como Raspberry OS (anteriormente Raspbian). Esta carencia de compatibilidad representaba una seria amenaza para el progreso de nuestro proyecto, ya que impedía que nuestra aplicación se ejecutara de manera adecuada en la plataforma designada.

Ante esta situación crítica, nos vimos obligados a tomar medidas drásticas. Decidimos crear una nueva partición en el disco de la Raspberry Pi en un intento desesperado por recuperar el archivo del código, que lamentablemente no habíamos respaldado adecuadamente. Este archivo era esencial para el funcionamiento de nuestra aplicación y su pérdida comprometía gravemente todo el trabajo realizado hasta el momento.

El proceso de recuperación del código fue arduo y desafiante, pero finalmente logramos recuperarlo, aunque no sin sacrificio. Una vez que aseguramos el archivo del código, el siguiente paso fue reinstalar completamente el sistema operativo. Esta tarea no solo consumió una cantidad considerable de tiempo y recursos, sino que también nos recordó la importancia de implementar procedimientos de respaldo robustos y efectivos en futuros proyectos para evitar la pérdida de datos críticos. En resumen, el incidente nos enseñó valiosas lecciones sobre la importancia de la planificación y la preparación meticulosa en el desarrollo de proyectos tecnológicos.

#### 4.1.2 ERRORES EN LA CONEXIÓN WIFI



*Ilustración 4: Modem de segunda mano*

Para el apartado de pruebas, decidí adquirir un módem de segunda mano, específicamente uno proporcionado por TELMEX. Mi objetivo era configurarlo con un nombre de red y contraseña personalizados para adaptarlo a mis necesidades de respaldo de datos. Sin embargo, me encontré con una serie de inconvenientes que, en un principio, pasé por alto debido a que mi prioridad estaba en otras áreas del proyecto.

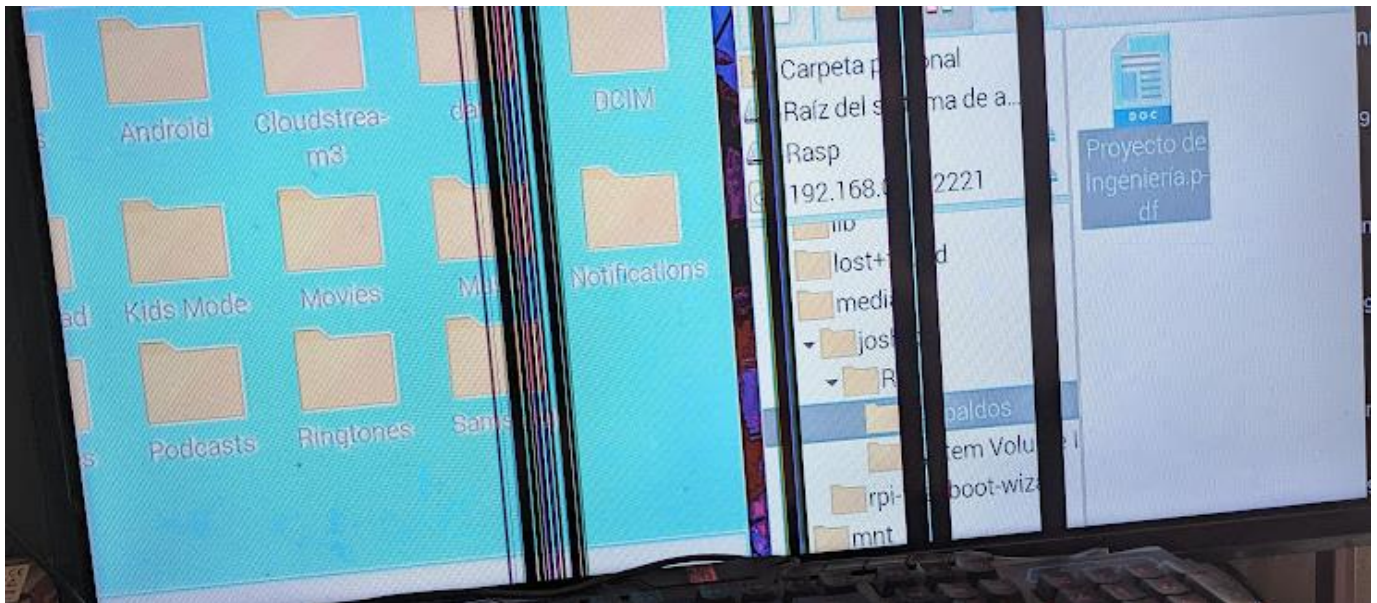
En un principio, ignoré las fallas de estabilidad que experimenté con el módem, ya que no las consideraba un problema crítico para el respaldo de datos en ese momento. Sin embargo, conforme avancé en el desarrollo y comencé a realizar pruebas más exhaustivas, se hizo evidente que estos problemas de estabilidad estaban teniendo un impacto directo en el funcionamiento de mi Raspberry Pi al ejecutar mi código.

Una de las manifestaciones más notorias de este problema fue la ocurrencia de un bucle infinito de transferencia de datos al intentar ejecutar mi código en la Raspberry Pi. Resultó que, debido a la naturaleza del código que estaba ejecutando, el cual funcionaba como un puerto de transferencia de datos en una red de área local, el módem no estaba recibiendo correctamente los datos transmitidos desde la Raspberry Pi.

Este malentendido en la transmisión de datos causó un efecto de "rebote" en los archivos que estaba intentando respaldar. Es decir, los datos que salían de la Raspberry Pi no lograban ingresar al puerto USB del módem de manera adecuada y, en lugar de ser almacenados, quedaban "en el aire", causando una interferencia significativa en la conexión Wi-Fi del dispositivo.

Este problema no solo afectó la estabilidad de la conexión Wi-Fi de mi Raspberry Pi, sino que también comprometió la integridad de los datos que estaba intentando respaldar. Fue un recordatorio importante de la necesidad de prestar atención a todos los componentes y aspectos del sistema durante las pruebas, incluso aquellos que inicialmente pueden parecer menos críticos. En conclusión, esta experiencia me enseñó valiosas lecciones sobre la importancia de abordar y solucionar los problemas de manera proactiva durante el proceso de desarrollo de un proyecto tecnológico.

#### 4.1.3 PRUEBAS SATISFACTORIAS



*Ilustración 5: Fase 1 de la prueba satisfactoria*

Para abordar el problema del módem, opté por una solución creativa: la creación de un servidor privado FTP en un dispositivo Android antiguo. Este dispositivo estaba conectado a la misma red Wi-Fi que mi Raspberry Pi, lo que me permitía replicar la funcionalidad que tenía la unidad USB conectada al módem. En la ilustración 5, se puede apreciar con claridad el apartado de archivos del servidor FTP Android en el lado izquierdo, que serviría como el destino final para los archivos respaldados. En el lado derecho, visualizo el archivo específico de la Raspberry Pi, denominado "Proyectos de Ingeniería", que deseaba transferir al servidor creado en el dispositivo Android.



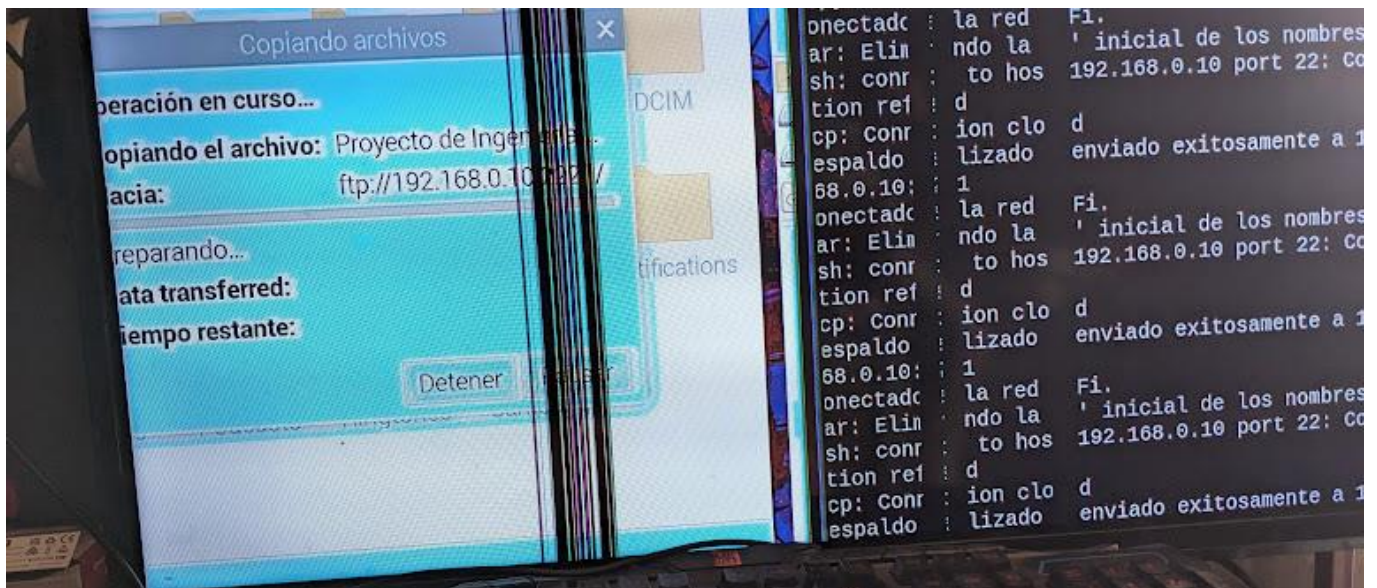


Ilustración 6: Fase 2 de la prueba satisfactoria

La ilustración 6 proporciona detalle del proceso en acción. En el lado derecho, observo cómo el código se está ejecutando y generando el respaldo de manera satisfactoria, lo cual es una señal alentadora de que la solución implementada está funcionando como se esperaba. Al mismo tiempo, en el lado izquierdo, puedo ver el proceso de copia del archivo desde su ubicación original en la Raspberry Pi hacia el servidor FTP que he configurado en el dispositivo Android. La dirección del servidor, `ftp://192.168.0.10:2221`, proporciona la ruta para la transferencia de archivos.

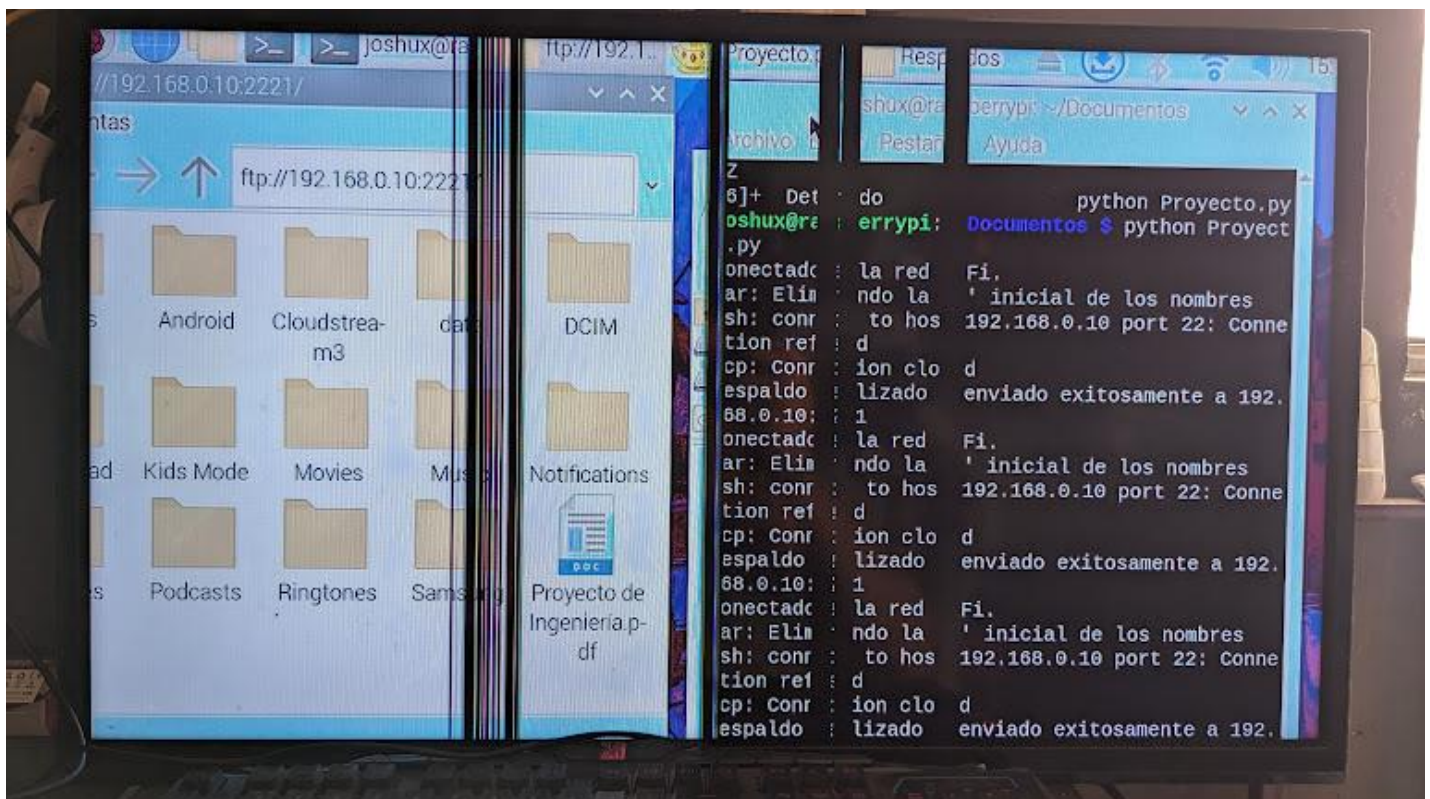


Ilustración 7: Fase final de la prueba satisfactoria

La ilustración presenta el resultado final. En el lado derecho, se muestra un mensaje que confirma que el respaldo se ha realizado exitosamente desde el código. Esta notificación representa un hito importante, indicando que la solución implementada ha sido efectiva en la tarea de respaldo de

datos. Por otro lado, en el lado izquierdo, puedo ver el archivo respaldado en el repositorio, con la carpeta abierta para su visualización. Este paso final confirma que el archivo deseado ha sido transferido con éxito al servidor FTP en el dispositivo Android, completando así el proceso de respaldo de datos de manera satisfactoria.

## 4.2 PRUEBAS EN LA PÁGINA

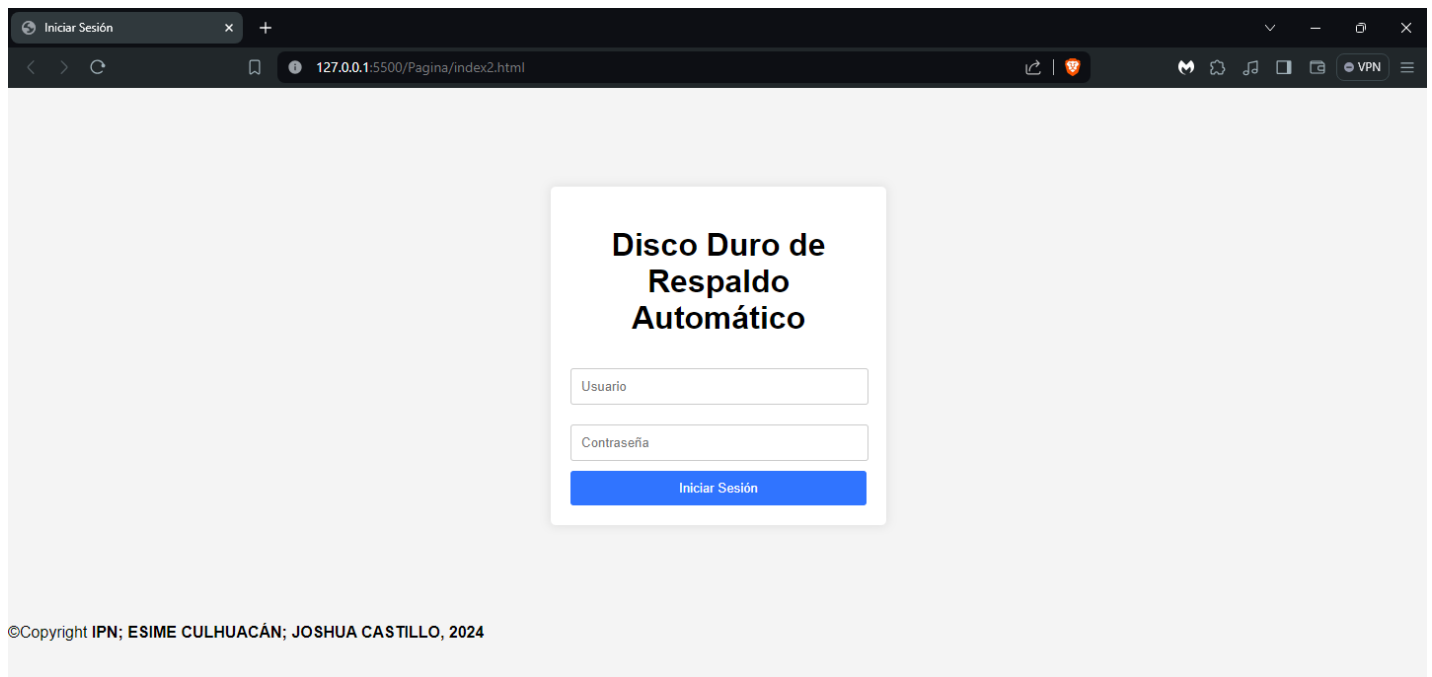


### Subir Respaldo

Seleccione el archivo de respaldo:  Sin archivos seleccionados

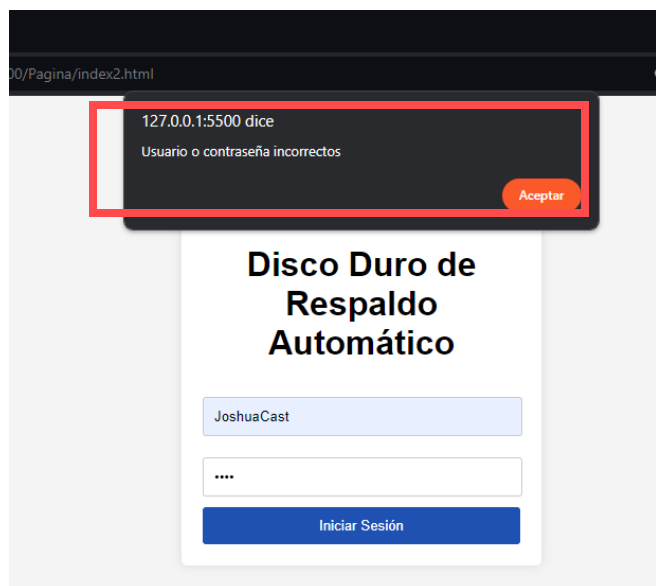
*Ilustración 8: Interfaz exclusiva de uso para la Raspberry*

En este primer código para la página, tenemos una interfaz exclusiva para la Raspberry, es decir, el usuario no va a tener acceso a esta dirección web, puesto que forma parte del código de respaldos implementado dentro de la Raspberry Pi 4b y su única función es servir como vínculo entre el microprocesador y el repositorio web en dónde se almacenarán los archivos.

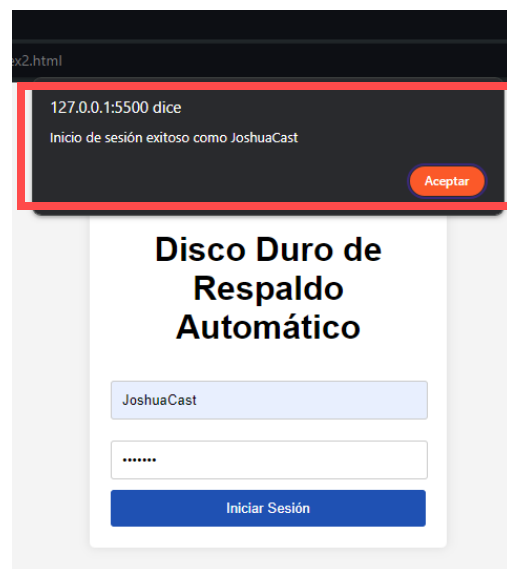


*Ilustración 9: Prueba de uso del filtro de seguridad de acceso a la página de los respaldos*

Este segundo código sirve para tener un control de acceso de los usuarios que entran a la página del repositorio para la consulta de los respaldos, sirve como un filtro de seguridad básico, en dónde si no se dan ambas condiciones en conjunto (Usuario y contraseña) simplemente no se le permitirá al usuario el acceso a la página del repositorio.



*Ilustración 10: Usuario incorrecto al que no se le permite el acceso*



*Ilustración 11: Prueba con Usuario admitido*

Como se nota en las imágenes, si el usuario no cuenta con su contraseña de acceso o sus credenciales son incorrectas, se le negará el acceso a la página de consulta de archivos, al contrario del usuario con su respectivo registro



*Ilustración 12: Página de consulta del repositorio de archivos*

Este es el *index* que forma parte primordial en el proceso, ya que es el código para la página del repositorio de archivos, dónde el usuario hace la consulta de los datos respaldados por la Raspberry, subidos a la página y posterior al filtrado de acceso, depositados en este repositorio de forma segura.

## **V. CONCLUSIONES Y RECOMENDACIONES**

### **V.I CONCLUSIONES**

El proyecto tiene una gran utilidad en el mundo real debido a su servicio de respaldo de información de forma segura, así como la consulta eficiente de esta, garantizando la accesibilidad a la información privada desde cualquier lugar que el usuario lo requiera.

Si bien representa un gran reto el desarrollo correcto en la vida diaria de este proyecto, por contratiempos como su diseño físico, su independencia energética en cuestiones de potencia, así como la garantía de conexión de red en una zona amplia, me gustaría pensar que sirve como precedente o antecedente para futuros proyectos, ya sea dentro de la ESIME Culhuacán o, ¿por qué no?, dentro de la industria, desarrollado por ingenieros de gran capacidad y acceso a grandes recursos que necesitan para lograr un proyecto aún más grande.

Los retos más grandes por vencer en este proyecto, sin duda alguna, se podrían resumir en cuatro. El número uno, la limitación de recursos significó un pequeño bache dentro del desarrollo del proyecto, dado que no se utilizó únicamente lo que se ve al final del proyecto en el resultado, sino que se utilizaron recursos que sirvieron como una prueba, es decir, fueron desechados al final del proyecto al no ser útiles, al no cumplir con los requerimientos o simplemente al no haber cabida para estos elementos dentro del proyecto. El número dos, la limitación del tiempo, así como la presión de desarrollar el proyecto en conjunto con otras clases esenciales para la formación de mi carrera como ingeniero en comunicaciones y electrónica. El número tres, las limitaciones de mi conocimiento dentro del formato educativo que tenemos hoy en día dentro de la carrera. Los planes de estudio nos ofrecen únicamente lo básico para el desarrollo del ingeniero en el mundo actual y de manera un poco contradictoria se nos solicita el desarrollo de un proyecto que rompa con estos paradigmas y salga justamente de lo básico, queriendo ir un paso más arriba sin darnos realmente las herramientas y conocimientos necesarios para este desarrollo, viéndonos en la necesidad de investigación y estudios autodidactas. Y, por último, el número cuatro, el conjunto de todos estos elementos se vio significativamente marcado en una etapa de complicaciones de salud mental de mi parte. Las limitaciones que yo sólo me puse al querer hacer este proyecto tan grande y de manera solitaria, he de confesar que no fueron de la mayor ayuda posible. Sin embargo, la presión tanto social como escolar, así como familiar, al desarrollar este proyecto, no fueron las mejores herramientas para ayudar a superar este tipo de problemas.

Después de todos los contratiempos, el proyecto, a mi parecer, es exitoso porque, repito una vez más, sirve como precedente para el futuro desarrollo de proyectos universitarios o incluso industriales que sirvan para el ahorro de tiempo en los trabajos en los hogares y en las diferentes modalidades que incluyen estos, es decir, el home office, poniendo así a mayor disponibilidad de tiempo a los jefes de familia que tienen una casa o una familia que atender, facilitando el acceso a este tiempo.

### **V.II RECOMENDACIONES**

Para darle al proyecto un impulso mucho más grande que el simple precedente del desarrollo del código, se proponen mejoras para que el proyecto cobre mucha más relevancia tanto educativa como dentro del mundo del desarrollo tecnológico.

#### **V.II.I Display Touch**

Con la finalidad de hacer de este proyecto lo más autosustentable posible en términos de conectividad con otros dispositivos, se recomienda utilizar un “480 x 320, 3.5inch Touch Screen TFT, 125MHz High-Speed SPI.” Diseñado para la Raspberry, para aprovechar de manera concreta la potencia del microprocesador.



Esta propuesta mejoraría el proyecto no solo en las salidas de información, sería posible también, intervenir la automatización de los respaldos, asignando de manera manual el tiempo de espera para la acción, las redes a las que se conecta, modos de pausa, exclusión de archivos, etc. Haciendo así, del dispositivo un HUB controlador de archivos y respaldos con respaldos automáticos.

### **V.II.II MEJORAS DE SEGURIDAD**

Se recomienda también ampliar los parámetros de seguridad de la consulta de datos, no solo recurriendo al servicio del hub en el que se almacena la página y por ende los archivos respaldados, sino implementar nuevos diseños en el código para aumentar los filtros de seguridad.

### **V.II.III MEJORAS DE AUTONOMÍA ENERGICA**

Y por último se sugiere también mejorar la autonomía de las baterías que respaldan a la Raspberry para hacerla portátil, los cálculos del proyecto nos dejan ver que en el actual diseño se cuentan únicamente con 2.1 horas de uso inalámbrico, por lo que, a ser un dispositivo de uso diario, se sugiere el cambio pertinente tanto de controladores, de ahorro y de baterías en cuestión.