

Universidad Técnica Federico Santa María

Departamento de Electrónica

Valparaíso Chile



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

AVM DOCKING STATION

Daniel González Figueroa

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELECTRÓNICO

PROFESOR GUÍA: Matías Zañartu

PROFESOR CORREFERENTE: Alejandro Weinstein

Resumen

En este trabajo se desarrollara la componente de software de un dispositivo conocido como AVM Docking Station. Este busca otorgar autonomía al dispositivo AVM, el cual se utilizara para registrar la actividad de las cuerdas vocales de un paciente.

Se soluciona en parte la falta de autonomía del AVM al permitir que este pueda liberar su memoria sin intervención del paciente al automatizar la transferencia de datos del AVM al Docking Station simplemente realizando la conexión del primero.

También se desarrolla un método para la comunicación entre el Docking Station y el usuario. El cual permite configurar la conexión WiFi del Docking Station, así como establecer el horario en que se realizara la transferencia de información al servidor externo.

Ademas se automatiza la conexión entre el Docking Station y un servidor externo, para transferir los datos descargados previamente desde el AVM.

Además se realiza un análisis del Docking Station respecto a lo establecido en la ley HIPAA, lo cual incluye establecer el flujo de ePHI del sistema completo, y analizar los riesgos y vulnerabilidades que este presenta, para finalmente tomar medidas de seguridad cuando sea posible.

Abstract

In this work, the software component of a device known as AVM Docking Station will be developed. This seeks to grant autonomy to the AVM device, which is used to record the activity of the vocal cords of a patient.

The lack of autonomy of the AVM is partly solved by allowing it to free its memory without patient intervention by automating the transfer of data from the AVM to the docking station simply by connecting the AVM to the first.

A method for communication between the docking station and the user is also developed. Which allows you to configure the WiFi connection of the Docking Station, as well as to establish the schedule in which the information will be transferred to the external server.

In addition, the connection between the Docking Station and an external server is automated, to transfer the data previously downloaded from the AVM.

In addition, an analysis of the Docking Station is carried out with respect to what is established in the HIPAA law, which includes establishing the ePHI flow of the complete system, and analyzing the risks and vulnerabilities that it presents, to finally take security measures when possible.

Glosario

ePhi: Información médica del paciente almacenada de forma electronica.

Llave: En criptografía, una llave es un valor que es aplicado mediante algún algoritmo a un bloque de información.

AVM : Dispositivo que permite registrar el uso de la voz en condiciones cotidianas

HIPAA Compliance: Se dice de una solución, entidad, empresa que cumpla con las normas HIPAA.

Enlace simbólico: Un acceso a un directorio similar a un acceso directo.

Script: Un programa que consiste en una secuencia de comandos, que es interpretado por otro programa.

Índice general

1. Introducción	1
1.1. Resumen y Objetivos Generales del Proyecto	1
1.2. Actividades a Desarrollar	2
1.3. Estado del Arte	3
2. Herramientas Utilizadas	5
2.1. Alternativas de Solución	5
2.2. Comunicación con el Usuario	8
2.3. Transferencia de archivos	9
2.4. Transferencia de información medica a través de Internet	11
3. Cumplimiento Reglas HIPAA	13
3.1. Regla de Seguridad HIPAA	13
3.2. Flujo de información	14
3.3. Vulnerabilidades, amenazas y riesgos	15
3.3.1. Vulnerabilidades	15
3.3.2. Amenazas y riesgos	16
3.4. Medidas de seguridad	17
4. Conexión con AVM	19
4.1. Trabajo Desarrollado	20
4.2. Pruebas	23

5. Comunicación con Usuario	26
5.1. Actividades Desarrolladas	27
5.2. Pruebas	31
6. Conexión con Servidor	38
6.1. Actividades Desarrolladas	38
6.2. Pruebas	43
7. Integración y Resultado Final	49
7.1. Integración	49
7.2. Resultado Final	52
7.2.1. Problemas del Sistema	53
A. Salidas de scripts	59
B. Código	64

Índice de figuras

1.1. Arquitectura del AVM Docking Station	3
3.1. Flujo de ePHI.	15
5.1. Utilización de la aplicación.	32
5.2. Conexión correcta a red WiFi desde la aplicación.	34
5.3. Conexión errónea a red Wifi desde la aplicación.	34
5.4. Modificación horario de subida, con inicio de transferencia.	36
5.5. Modificación horario de subida, sin inicio de transferencia.	36
7.1. Arquitectura interna de pipes	50
7.2. Mensaje enviado a la aplicación cuando el AVM transfiere datos. . .	52

Índice de cuadros

4.1. Requisitos funcionales para la conexión entre Docking Station y AVM.	19
5.1. Requisitos funcionales para la conexión entre Docking Station y teléfono.	27
6.1. Requisitos funcionales y pruebas para conexión Docking Station con servidor.	39

Listings

4.1. Verificacion que regla udev es lanzada	23
4.2. Estado del servicio 'backup-avm' cuando este termina su funcion.	24
4.3. Estado del servicio 'backup-avm' mientras opera.	24
4.4. Salida del servicio de transferencia de archivos entre AVM y Docking Station.	24
4.5. Registro de actividad del modulo de almacenamiento	25
4.6. Registro de actividad cuando el AVM es desconectado	25
5.1. Salida resumida del script que parea dispositivos	31
5.2. Archivo con credenciales WiFi agregadas.	33
5.3. Tiempos en los que se ejecuta las transferencias al servidor.	36
5.4. Registro de actividad del modulo de comunicacion.	37
6.1. Conexion SSH desde Docking Station al servidor	43
6.2. Registro de actividad cuando la subida de datos se realiza de forma ideal.	44
6.3. Registro de actividad cuando el Docking Station es iniciado y debe estar transfiriendo.	45
6.4. Registro de actividad cuan.	45
6.5. Prueba de funcionamiento con y sin internet.	46
6.6. Transferencia de archivos desde Docking Station al servidor	46
6.7. Salida herramienta iperf que entrega la velocidad de conexion.	47
6.8. Registro donde se muestra una conexion constante.	47

6.9. Registro de los archivos subidos al servidor	48
7.1. Ejemplo de registro de actividad en un uso diario.	52
A.1. Verificacion que regla udev es lanzada	59
A.2. Estado del servicio 'backup-avm' cuando este termina su funcion .	59
A.3. Estado del servicio 'backup-avm' mientras opera	59
A.4. Salida del servicio de transferencia de archivos entre AVM y Docking Station	60
A.5. Script que parea dispositivos	61
A.6. Timers activos del sistema.	61
A.7. Conexion ssh desde Docking Station al servidor	62
A.8. Salida herramienta iperf	62

Capítulo 1

Introducción

1.1. Resumen y Objetivos Generales del Proyecto

El monitoreo remoto de pacientes es una forma de atención que usa diversas tecnologías para recolectar la información de un paciente en un ambiente cotidiano, fuera de un ámbito clínico tradicional. Esto permite obtener una mayor y mejor información, además de entregar una mayor comodidad y autonomía al paciente tratado además de mejorar su adherencia a los tratamientos.

En este contexto existe el dispositivo conocido como Advance Voice Monitor o AVM [1], el cual permite realizar un registro del uso de las cuerdas vocales a lo largo del día. El problema que presenta este dispositivo es la falta de autonomía para usos prolongados de tiempo debido al tamaño limitado de su memoria.

Para lidiar con esto se desarrollará en este trabajo el software para el dispositivo que complemente el funcionamiento del AVM, el cual será llamado AVM Docking Station. Este dispositivo debe descargar los datos del AVM, subirlos a un servidor externo para que sean procesados, e implementar un procedimiento sencillo para que el usuario pueda realizar las configuraciones necesarias para realizar lo anterior.

Además, considerando que se estará trabajando con información medica electrónica,

también conocida como Electronic Protected Health Information (ePHI), se dará énfasis en el tratamiento de este tipo de información de acuerdo a lo establecido en el Health Insurance Portability and Accountability Act (HIPAA).

1.2. Actividades a Desarrollar

El trabajo a desarrollar consiste principalmente en la componente de software del dispositivo Docking Station. La arquitectura del sistema a desarrollar esta representada en la figura 1.1, donde se muestra de forma general las interacciones entre los diversos módulos del Docking Station, el usuario y el servidor externo. La función que debe cumplir cada uno de estos módulos y el objetivo general que se busca esta listado a continuación.

- Carga: Encargado de recargar la batería del AVM.
- Conexión AVM: Descargar los datos del AVM hacia el Docking Station.
- Comunicación Usuario: Encargado de recibir y enviar mensajes al usuario para configurar el Docking Station.
- Comunicación Servidor: Encargado de establecer la comunicación de forma segura con el servidor y de transferir los datos descargados del AVM hacia este.

Dentro de este trabajo se desarrollarán los últimos 3 módulos mencionados. El módulo de carga no será desarrollado debido a que este es mayormente hardware. En el capítulo 2 se describen las herramientas que se utilizarán para desarrollar este trabajo. En el capítulo 3 se tratará lo relacionado al trabajo con datos médicos, los resguardos que se deben tener y el proceso que se debe seguir para poder desarrollar una solución que cumpla con la ley HIPAA (HIPAA Compliance). En los capítulos 4,5 y 6 se definirán objetivos específicos para cada

uno de estos módulos y se explicará como fueron desarrollados y probados. En el capítulo 7 se integrarán los módulos desarrollados anteriormente y se analizará el resultado final.

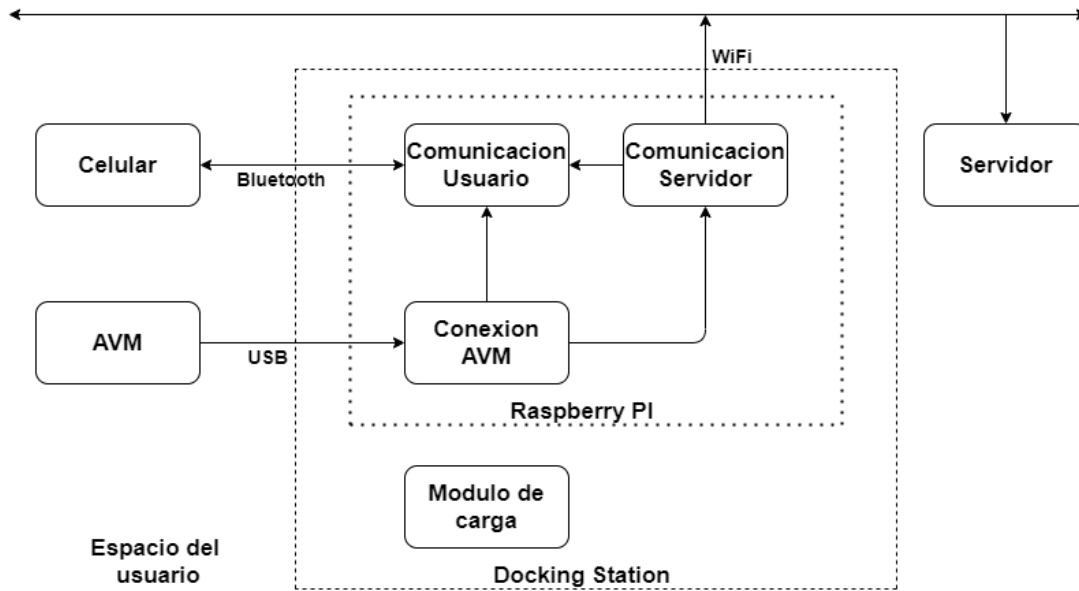


Figura 1.1: Arquitectura del AVM Docking Station

1.3. Estado del Arte

Debido al contexto actual, el monitoreo remoto de pacientes y sobre todo la telemedicina han cobrado una gran importancia en la vida de las personas. Esto sumado al uso cada vez más común de tecnologías IoT, ha generado diversos desarrollos de sistemas de monitoreo remoto, que llenan distintas necesidades y permiten enfrentar desafíos a futuro. Se investigaron diversos trabajos para ver el alcance que tiene actualmente el monitoreo remoto de pacientes, las diversas formas en que este es utilizado, y como fue implementado.

En [2] se muestra un sistema utilizado para monitorear el estado de pacientes luego de una operación. Para registrar el estado del paciente el sistema cuenta con varios sensores utilizados para medir distintos signos vitales, los cuales terminan

siendo almacenados en una Raspberry Pi, la cual genera una interfaz para mostrar la información en una tablet o algún dispositivo móvil.

En [3] se desarrolla un sistema que permite monitorear de forma constante el estado de un paciente y que en caso de emergencia puede enviar en tiempo real audio y video a un centro médico. Para obtener el audio se utilizan 3 Raspberry Pi y para el envío de video se utiliza una NVIDIA Jetson.

El trabajo realizado en [4] busca dar atención médica a gente de sectores rurales. Este sistema, al igual que los anteriores, utiliza diversos sensores para registrar signos vitales del paciente, y permite conectarse a diversos tipos de redes que otorgan internet para poder obtener atención médica. En [5] se realizó una revisión de distintos dispositivos médicos y las tecnologías que estos usan.

Existen también trabajos como el desarrollado en [6] que utilizan un microcontrolador para obtener la información de los diversos sensores y enviar esta directamente a una aplicación móvil.

Respecto a la transferencia de información médica a través de internet, trabajos como el mostrado en [7] utilizan encriptación simétrica (específicamente AES) para realizar la transferencia de imágenes médicas. En [8] también recomiendan encriptar la transferencia de datos utilizando algún tipo de encriptación asimétrica y finalmente en [9] se utiliza SSH para realizar transferencia de imágenes médicas, el cual es un protocolo que utiliza ambos tipos de encriptación.

Capítulo 2

Herramientas Utilizadas

En este capítulo se analizan las distintas alternativas que existen para implementar la solución, dando ventajas y desventajas de cada una de estas. Luego se describen diversas herramientas que también serán utilizadas a lo largo del desarrollo de este trabajo.

2.1. Alternativas de Solución

Lo primero que se debe realizar es seleccionar la tarjeta de desarrollo que se utilizará, ya que según la que se escoja, las herramientas que es posible utilizar serán diferentes y presentaran distintos problemas. En general cuando se busca desarrollar un dispositivo existen 3 tipos de hardware que se deben considerar; microcontroladores, single board computer y FPGA. Cada una de estas herramientas posee diversas características las cuales serán nombradas a continuación

Microcontroladores

Los microcontroladores son los más simples de los 3 dispositivos. El ejemplo más conocido en este tipo de dispositivos son los Arduino. Debido a esta simpleza,

su costo es mucho menor que el de los otros dos dispositivos

Generalmente el hardware de estos dispositivos es más limitado que en los otros casos; presentan procesadores de un solo núcleo, RAM limitada, memoria FLASH para guardar el programa a ejecutar y pocos pines de entrada y salida de propósito general. Debido a este limitado hardware, de los 3 dispositivos es el con menor consumo de potencia.

Estos dispositivos generalmente son programados con lenguajes de bajo nivel (Assembler, C, C++) lo que permite también realizar control de tareas en tiempo real, pero aumenta el tiempo de desarrollo. En general estos dispositivos son recomendados cuando se desea interactuar con el mundo real, pero no se requiere un trabajo computacional intensivo. No son recomendados si se requieren realizar tareas de alta complejidad o múltiples tareas a la vez.

Single Board Computer

Un Single Board Computer (o SBC) es similar a un computador de escritorio, pero con menores especificaciones (RAM, velocidad del procesador, espacio de almacenamiento y capacidad gráfica), y disponibles en una sola tarjeta. Uno de los dispositivos más conocidos de este tipo son las Raspberry PI. Estos dispositivos además incluyen generalmente conexiones USB(2.0), HDMI, WiFi, Bluetooth y ethernet.

En general estos dispositivos corren un sistema operativo (en general alguna distribución de Linux). Existen diversas ventajas al correr un SO: Permite realizar programación en alto nivel lo que acelera el tiempo de desarrollo. Existen diversas herramientas de código abierto que es posible utilizar para distintas tareas (conectarse a internet, conectarse a un servidor remoto, etc.). Permite correr diversos programas a la vez de forma sencilla, ya que el sistema operativo se encarga de repartir los recursos para este, y en general cualquier actividad que se pueda realizar con un computador de escritorio.

Como desventaja, tener un sistema operativo de uso general hace que tareas que requieren un control más fino no puedan ser realizadas de forma correcta, como por ejemplo manejar los pines de propósito general o correr algún programa en tiempo real.

FPGA

A diferencia de los 2 anteriores, una FPGA no se programa en el sentido tradicional. Esta utiliza un lenguaje de descripción de hardware (HDL), con el cual se configura el circuito interno de la FPGA para ejecutar alguna tarea.

Al igual que una SBC, una FPGA puede tener diversos conectores como WiFi, Bluetooth, HDMI, VGA, etc, pero a diferencia del SBC, utilizar estos requiere un mayor tiempo de implementación.

Este tipo de dispositivos tienen un costo mucho mayor a los otros dos y un tiempo de desarrollo también mucho mayor. Como ventaja permiten configurar todo lo que ocurre dentro del circuito y brilla especialmente cuando se debe manejar una gran cantidad de señales a la vez.

Alternativa Seleccionada

La alternativa seleccionada es utilizar un SBC, específicamente una Raspberry Pi 3B. Se decanta por esta opción debido a que al ser un primer acercamiento a la solución, se puede lograr un desarrollo más rápido en este tipo de dispositivos. También debido a la facilidad de conectarse a internet y utilizar herramientas externas y la inclusión de WiFi y Bluetooth de forma nativa en este dispositivo. Esto implica que el sistema operativo a utilizar será Linux, específicamente la distribución creada para una RPI, Raspbian OS Lite.

2.2. Comunicación con el Usuario

En este punto se busca un protocolo que permita intercambiar mensajes entre el teléfono del usuario y el Docking Station. Esto con el objetivo de que el usuario pueda conectar su teléfono al Docking Station de forma sencilla, para configurar una red WiFi y los horarios en que los datos serán subidos.

Existen diversas formas de realizar esta conexión, pero a continuación se detallarán 3 de estos junto a sus características, ventajas y desventajas.

MQTT

MQTT es un protocolo principalmente usado en IoT. Este se encuentra optimizado para ser usado con dispositivos de baja potencia por lo que optimiza el uso de batería y el ancho de banda utilizado para intercambiar mensajes.

Existen dos entidades en este protocolo, un broker que funciona como router y varios clientes los cuales se suscriben al broker y pueden enviar mensajes a otros clientes a través de este.

Es posible configurar este para interactuar con una aplicación del celular utilizando como broker la Raspberry Pi, pero para recibir mensajes es necesario que esta aplicación se encuentre abierta.

Para que exista comunicación tanto cliente como broker deben estar conectados a internet o dentro de la misma red WiFi.

Firestore Cloud Messaging

También abreviado como FCM, este es un servicio de mensajería multiplataforma gratuito, que soporta aplicaciones para Android e IOS. Este permite además de intercambiar mensajes, enviar notificaciones hacia estas aplicaciones. Requiere que un servidor este a cargo de enviar mensajes y estos pueden enviarse tan solo a

un dispositivo o a un conjunto de estos a la vez. Requiere una conexión a internet para funcionar.

Comunicación Serial por Bluetooth

No es un protocolo en sí, pero utilizar este conjunto de herramientas permite establecer una comunicación entre Docking Station y teléfono del usuario sin necesidad de una conexión a internet, además al ser una forma de comunicación más simple que las otras dos permite que el desarrollo sea más fácil y rápido.

Alternativa Seleccionada

Debido a que no se cuenta con una aplicación para celular para utilizar MQTT o FCM, la opción elegida será realizar la comunicación serial mediante bluetooth. Además esta entrega la ventaja que no requiere que los dispositivos que se desean comunicar se encuentren en la misma red o conectados a internet.

2.3. Transferencia de archivos

En esta sección se estudiarán diversas herramientas para transferir archivos presentes en Linux para evaluar cuál es la mejor en dos casos: transferir archivos desde AVM hacia Docking Station, y subir los archivos del Docking Station al servidor.

SCP

SCP o Secure Copy Protocol es una de las formas más sencillas para transferir información. Este protocolo simplemente realiza una copia lineal de los archivos, es decir, lee el archivo en el directorio fuente y lo escribe en el directorio de destino. Respecto a la seguridad este permite utilizarse por sobre SSH, por lo que se puede

esperar la misma seguridad que existe al establecer una conexión por medio de este protocolo.

Rsync

Rsync es una aplicación creada con el objetivo de sincronizar dos directorios en dos dispositivos remotos por lo que se encuentra optimizado para esto. Esto no significa que sea su única función, ya que también puede funcionar para transferir archivos de forma local.

Tiene como característica distintiva la capacidad que tiene de reanudar las transferencias entre dispositivos remotos que sean interrumpidas. Esto debido al algoritmo que utiliza el cual se puede encontrar explicado con detalle en [10]. Además, es altamente configurable permitiendo entre otras cosas conservar los permisos de los archivos transferidos, un modo que permite excluir o incluir archivos según su tamaño, fecha de creación, nombre, escoger la forma en que se realiza la conexión con el servidor remoto, el formato de la salida de la herramienta, etc.

Al igual que SCP puede funcionar a través de una sesión de SSH por lo que en el ámbito de seguridad ambos se comportan igual. Finalmente es importante mencionar que debido al algoritmo que utiliza, requiere un uso mayor de recursos de los dispositivos involucrados para realizar ciertas operaciones.

SFTP

SFTP o SSH file transfer protocol está diseñado para transferir archivos por medio de una conexión SSH. Además de transferir estos archivos permite realizar diferentes operaciones sobre estos.

También permite resumir la transferencia de archivos aunque su configuración para esto es más compleja que en el caso de rsync.

Alternativa Seleccionada

En este caso, se prefiere la robustez que entrega rsync al momento de continuar las transferencias de archivos, y la facilidad al momento de ser configurado. Esto también permite que el dispositivo funcione con conexiones WiFi poco robustas.

2.4. Transferencia de información medica a través de Internet

Otro de los temas relevantes es el tratamiento de la información medica cuando esta es movilizada a través de internet. En [9] se llega a la conclusión de que una buena forma de transmitir información a través de internet es utilizando el protocolo SSH, el cual permite establecer un canal de comunicación a través de internet a través del cual es seguro transferir información. En [8], también se recomienda utilizar alguna técnica de encriptación asimétrica (una de las dos utilizadas en SSH) para establecer canales de comunicación seguro en el contexto de dispositivos IoT que trabajen con información sensible. También en [7] se desarrolla un método para transferir información médica a través de internet, el cual utiliza AES, que es un protocolo de encriptación simétrica. En general se puede ver que esta combinación es una buena forma de transferir información de manera segura a través de internet. A continuación se explicara de forma más detallada como funcionan estos tipos de encriptación

Encriptacion Asimétrica

Esta técnica consiste en utilizar un par de **llaves** relacionadas de forma matemática. Una de ellas será utilizada para realizar la encriptación de los datos (llave pública) y la otra será utilizada para realizar la desencriptación (llave privada). Como sus nombres lo indican, la llave pública puede ser compartida.

Esta técnica funciona pidiendo la llave pública del usuario al que se desea enviar un mensaje. Este mensaje será encriptado utilizando esta llave pública y enviado al receptor. Luego este podrá descifrar el mensaje utilizando su llave privada. La desventaja que este tipo de técnica presenta es la lentitud con la que es ejecutada para grandes volúmenes de información, por lo que en general es utilizada para establecer un canal seguro por medio de otra técnica conocida como **encriptación simétrica**. Dentro de los algoritmos asimétricos, el más utilizado actualmente es conocido como RSA.

Encriptacion Simétrica

La encriptación simétrica consiste en utilizar la misma llave para encriptar y desencriptar el mensaje. En general se utiliza en conjunto con la encriptación asimétrica, utilizando esta para transferir de forma segura una llave simétrica común. El algoritmo que se utiliza como estándar en este caso es conocido como AES.

Capítulo 3

Cumplimiento Reglas HIPAA

Como se mencionó anteriormente, se debe trabajar con datos médicos (e-PHI). Esto requiere tener una serie de consideraciones técnicas dadas por la regla de seguridad de HIPAA. En este capítulo se tratará de forma general este conjunto de reglas y se realizará un análisis de las diversas vulnerabilidades que pudiesen existir en el sistema para en capítulos posteriores tratar estas de ser posibles.

3.1. Regla de Seguridad HIPAA

La ley HIPAA, es un conjunto de reglas que aseguran el correcto tratamiento de la información médica de un paciente. Esta cubre a cualquier tipo de entidad que trabaje con esta información de forma electrónica (no aplica para información transmitida de forma escrita u oral), y requiere que estas entidades mantengan una protección administrativa, técnica y física razonable respecto a la información que manipulan. De forma general estas entidades deben cumplir con lo siguiente

1. Asegurar la confiabilidad, integridad y disponibilidad de todo e-PHI que se cree, mantenga y transmita.

2. Identificar y proteger ante amenazas anticipadas y razonables que puedan comprometer e-PHI de alguna forma.
3. Proteger contra cualquier mal uso razonable y anticipado, que se le pueda dar a la información.
4. Que todo el personal que trabaje con este e-PHI se encuentre familiarizado con las normas HIPAA.

Debido a que este trabajo solo se enfoca en desarrollar el Docking Station, nos enfocaremos principalmente en los tres primeros puntos.

Para esto, lo primero es identificar el flujo de información médica en torno al Docking Station, luego identificar las vulnerabilidades que existen en este, las distintas amenazas que pueden surgir y los riesgos que se podrían correr al ocurrir alguno de estos tipos de ataques.

3.2. Flujo de información

El primer paso debe ser determinar por donde fluye la información médica, donde se genera, almacena y la forma en que esta se transfiere de un lugar a otro. La información médica entra al sistema en el AVM, en este dispositivo se generan archivos mediante la medición de la actividad de las cuerdas vocales de un paciente, la cual es almacenada en la memoria interna del AVM. Luego mediante una conexión física (conexión USB) entre el AVM y el Docking Station, los archivos almacenados en el AVM se transfieren al Docking Station y una vez transferidos se eliminan del AVM. Finalmente estos archivos se transfieren a través de internet hacia un servidor externo donde serán finalmente almacenados. Se puede ver un esquema de este flujo de ePHI en la figura 3.1.

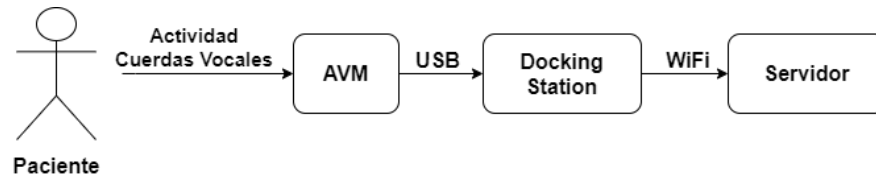


Figura 3.1: Flujo de ePHI.

3.3. Vulnerabilidades, amenazas y riesgos

Para conocer ante que tipo de ataques debe protegerse el Docking Station es importante analizar la forma en que este está expuesto a estas amenazas. Primero es importante distinguir entre vulnerabilidad, amenaza y riesgo; la vulnerabilidad es el grado de exposición que tiene el sistema, fallas y omisiones en su seguridad; se dice amenaza a la forma en que pueden explotarse estas vulnerabilidades y riesgo son las posibles consecuencias que existirían de concretarse una amenaza. Por lo que lo primero es conocer de que forma se encuentra expuesto el Docking Station, luego a partir de eso ver que amenazas y la probabilidad de que estas ocurran y finalmente analizar los riesgos y el impacto que estos tendrían.

3.3.1. Vulnerabilidades

Lo primero es considerar que este dispositivo será entregado a un paciente, por lo que se encontrará expuesto a cualquier mal uso que este pueda darle. Dentro de estas vulnerabilidades ambientales se puede considerar cualquier evento de fuerza mayor que pueda poner en riesgo la integridad del Docking Station (incendios, subidas de voltaje, robos, etc.), además cualquier persona con acceso al Docking Station se considera una potencial vulnerabilidad.

También según el flujo de e-PHI descrito en la figura 3.1, se pueden identificar otras dos vulnerabilidades. La primera vulnerabilidad del dispositivo es la conexión USB que está disponible en el Docking Station para conectarse con el AVM. La segunda vulnerabilidad es la conexión a internet del Docking Station la que lo hace

vulnerable a todo tipo de ataques provenientes de esta fuente (hackers, denial of service, etc.)

Finalmente existe una vulnerabilidad que no se ve en el flujo de ePHI, la cual es la conexión entre el teléfono del usuario y el Docking Station.

3.3.2. Amenazas y riesgos

Según lo anterior, podemos separar las distintas amenazas con sus respectivos riesgos según la vulnerabilidad que estas atacan. En primer lugar consideremos que el dispositivo no se encuentra en un lugar monitoreado, el cual es el hogar del paciente. Esto puede traer distintos tipos de amenazas tales como eventos de fuerza mayor (incendios, subidas de voltajes, etc.) que dejen inutilizable el Docking Station, Estos eventos no son muy probables, pero representan un riesgo, ya que de ocurrir es muy probable que se pierda la información médica almacenada en el Docking Station. También existen amenazas externas como podría ser un robo del dispositivo, la cual a pesar de ser un evento poco probable representa un alto riesgo debido a que no solo el atacante gana acceso a la información médica, sino que obtiene un Docking Station para poder probar otro tipo de ataques.

Existe también la amenaza de que se utilice el puerto USB para realizar algún ataque. La principal amenaza es un tipo de ataque conocido como inyección de código, el cual permite ejecutar código en una máquina actuando como administrador. Para realizar esto se requiere conocer el funcionamiento interno del Docking Station (lo que este realiza al conectar un dispositivo) y además se debe tener acceso al Docking Station, por lo que la probabilidad de que ocurra es bastante baja, aunque posee un gran riesgo, ya que de realizarse se puede obtener acceso como administrador al Docking Station, viéndose comprometidos los datos del paciente, así como el funcionamiento interno del Docking Station.

Otro punto vulnerable es que al estar expuesto a internet, el Docking Station puede sufrir los mismos ataques que cualquier otro equipo (ataques a SSH, DOS,

etc.). La probabilidad de que estos ataques se realicen es bastante alta, ya que solo se requiere un dispositivo con internet para realizarlos. Dentro de los riesgos que se pueden correr existe la posibilidad de una simple denegación de servicios, en cuyo caso la información no se podría transmitir correctamente al servidor externo aunque esta no se vería comprometida, un acceso remoto por SSH, que permitirá al atacante obtener información sobre el funcionamiento del Docking Station y la información médica del paciente, y también alguien que esté interceptando la comunicación en cuyo caso se podría poner en riesgo la información médica del paciente. Por lo que el riesgo que se corre depende en gran medida del ataque realizado.

Finalmente, un punto vulnerable por el que está expuesto el Docking Station es la conexión con el celular del usuario. La amenaza que esta presenta es que se puedan ejecutar comandos como administrador lo que podría darle control del Docking Station al atacante. Para realizar esto se requiere acceso al Docking Station por lo que tiene una probabilidad baja de ocurrir, aunque un alto riesgo de ser ejecutadas.

3.4. Medidas de seguridad

Para decidir que medida de seguridad debe tomarse, se debe tomar en cuenta principalmente dos factores: La probabilidad de que se cumpla una amenaza, y la gravedad del riesgo que esta conlleva. También deben considerarse otros factores como la complejidad, el software y el hardware requerido, y el costo de implementar una solución. Debido a que en la plataforma que se trabaja es una Raspberry Pi, el hardware no se puede modificar. Además como se dijo anteriormente, los ataques a través de Internet son los más comunes y presentan un grave riesgo, por lo que la prioridad es protegerse ante este tipo de ataques.

Respecto a las amenazas externas y físicas (robos, eventos de fuerza mayor), debido a la baja probabilidad de que estas ocurran y a la dificultad de implementar alguna protección ante estas, se dejarán sin tratar.

Las amenazas que pudiesen surgir utilizando el puerto USB del Docking Station, a pesar de ser difíciles de realizar, no es difícil agregar una capa extra de protección por lo que se agregará.

Para tratar la vulnerabilidad de la conexión con el usuario, se requiere un trabajo en las dos partes (teléfono y Docking Station), por lo que esto se dejara sin tratar. Finalmente, las amenazas que existan utilizando internet deben ser tratadas, debido a lo comunes que son y al peligro que representan para la información médica del paciente.

Todas las medidas que se tomarán se indicarán, junto a como son implementadas, serán indicadas en los capítulos siguientes.

Capítulo 4

Conexión con AVM

En este capítulo se tratara la conexión entre Docking Station y AVM mediante USB, la funcionalidad que el Docking Station debe ofrecer en este aspecto y como se tratara la vulnerabilidad presente en esta conexión.

Existen diversas cosas que se deben tener en cuenta para este módulo. Primero recordar que su objetivo principal es realizar la descarga de datos desde el AVM, y que esta descarga de datos implica movilizar ePhi, por lo que de acuerdo a las normas HIPAA se debe generar algún registro de estos movimientos, y que este puerto al estar expuesto presenta una vulnerabilidad que es posible explotar. Con esto se pueden fijar los requisitos mostrados en la tabla 4.1 , junto con las pruebas para probar su funcionamiento.

Requisitos funcionales	Pruebas
Identificar si el dispositivo conectado es un AVM u otro dispositivo.	Conectar el AVM al Docking Station. y chequear que se lanza la regla correspondiente.
Transferencia automática de AVM a Docking Station	Con el AVM conectado, verificar que el script de respaldo, efectivamente copia los archivos.
Se debe tener un registro de las transferencias de archivos entre AVM, Docking Station.	Al realizar una transferencia esta debe quedar registrada en un archivo de registro.

Cuadro 4.1: Requisitos funcionales para la conexión entre Docking Station y AVM.

Se debe considerar que al momento de realizar este trabajo no es posible contar con un AVM por lo que el primer requisito no sera tomado en cuenta. En el anexo B se encuentra el enlace donde se detalla como es posible lograr esto. Para realizar las pruebas se trabajará con dispositivo de almacenamiento USB común y corriente.

4.1. Trabajo Desarrollado

En esta sección se detallara el trabajo desarrollado para cumplir cada uno de los objetivos.

Identificar dispositivo

Identificar el dispositivo conectado permite agregar una capa de seguridad a esta interfaz con el usuario, esto debido a que solo se tomarán acciones con el dispositivo si este es reconocido como un AVM.

La herramienta utilizada para detectar el dispositivo conectado será 'udev', el gestor de dispositivos integrado en el kernel de Linux y que permite gestionar todos los dispositivos conectados al sistema. Esto se realiza mediante reglas que permiten identificar un dispositivo o familia de dispositivos según los atributos que estos presenten.

Como no se cuenta con un AVM, no es posible realizar las configuraciones para identificar este. Por lo que el funcionamiento de este punto se probara utilizando un dispositivo de almacenamiento USB cualquiera. Se indicara en el anexo B, como es posible identificar un dispositivo en especifico o familia de dispositivos y ejecutar una acción cuando este sea conectado o removido.

Para identificar un dispositivo USB cualquiera, la regla que se escribe tiene los siguientes parámetros.

- ACTION==add : Cuando un dispositivo con las otras características sea agregado.

- SUBSYSTEM=block : Todos los dispositivos de almacenamiento cumplen con pertenecer a este subsistema.
- KERNEL=sd* : El dispositivo es accesible en el sistema Linux en una carpeta con nombre sd* donde * es un número.
- SYMLINK=avm_link: Permite generar un enlace simbólico al dispositivo identificado.
- RUN+= : Entrega la acción a realizar cuando se agregue un dispositivo que cumpla con las reglas anteriores.

Los 3 primeros parámetros se utilizarán para identificar el dispositivo. El parámetro SYMLINK como se mencionó antes genera un enlace simbólico al dispositivo, lo que permite que este sea montado en una dirección a elección. Finalmente el parámetro RUN, entregará el comando a ejecutar cuando se identifique un dispositivo que cumpla con las reglas anteriores, el cual en este caso será un servicio.

De la misma manera, se agregará una regla que identifique cuando el dispositivo es desconectado. Esto se logra cambiando el parámetro ACTION que identifica cuando un dispositivo es agregado, a 'remove', lo que indica que el dispositivo fue desconectado, y se cambiara la acción a ejecutar por un script que permita registrar esta acción.

Descarga de datos

La descarga de datos estará a cargo del servicio lanzado previamente por la regla udev. Este servicio estará encargado de lanzar un script para cumplir el funcionamiento de este módulo.

Lo primero que realizará este script es montar el enlace simbólico creado con anterioridad en un directorio predeterminado, lo cual permite ver los contenidos

del dispositivo conectado al acceder a este directorio.

Una vez realizado lo anterior, comenzará la transferencia de datos utilizando la herramienta rsync. Esta se configura de forma que a medida que los archivos sean transferidos de forma exitosa, se eliminen del AVM. Es importante mencionar que cuando rsync es utilizado de forma local como en este caso, no utilizara el mismo algoritmo que cuando se transfieren datos de forma remota y simplemente realizara una copia lineal, es decir, lee los datos desde el AVM y los escribe en el Docking Station. Además cabe mencionar que para agregar una capa adicional de seguridad, rsync estará configurado para solo transferir archivos con cierto patrón en su nombre, lo que solo permite que tengan un nombre correspondiente a este formato sean transferidos. En este caso, el patrón es que el nombre de estos archivos comience con AVM.

Finalmente desmontará el dispositivo conectado para permitir una desconexión segura y sin corrupción en la información. Esta es la manera correcta de desconectar el dispositivo, y permite asegurar que no queden archivos corruptos en el Docking Station. A pesar de que en la mayoría de los casos retirar el dispositivo USB a la mitad de la transferencia no tiene efectos negativos mas que cortar la transferencia, esto no es recomendable. Por lo mismo sistemas operativos como Windows dan la opción de remover un dispositivo con seguridad, lo que equivale al ultimo paso de desmontar el dispositivo para asegurarse que no este siendo utilizado.

Registro de actividad

El registro de actividad de este modulo se encargara de registrar las transferencias de archivos que se realicen, y de registrar cuando es conectado y desconectado el AVM.

El script lanzado cuando es conectado el dispositivo se encargara de registrar la conexión de este, mientras que la regla encargada de detectar la desconexión lanzara un script que registre la desconexión del dispositivo.

Para registrar las transferencias de archivos, se utilizara la salida del comando `rsync` (la cual es mostrada a partir de la segunda linea en 4.5). En esta salida se buscarán los archivos cuya transferencia se haya completado, y agregará esta transferencia al registro.

4.2. Pruebas

Identificar dispositivo

El mismo `udev` entrega herramientas para verificar que un dispositivo conectado, es efectivamente identificado. En 4.1 se muestra una salida resumida al realizar este proceso, en donde se puede ver que cuando el dispositivo es conectado se ejecuta lo descrito en el parámetro `run`, lo cual es el inicio del servicio. En el anexo B, se entrega documentación para utilizar esta herramienta.

También es importante verificar que el servicio es lanzado de forma correcta, para lo cual es posible verificar su estado. Si es lanzado correctamente se puede obtener 4.2 donde se ve que el servicio ha terminado de ser ejecutado correctamente, y también se puede obtener 4.3 cuando el servicio continúa operando.

```
pi@raspberrypi:~ $ udevadm test DEVPATH
...
run:  '/bin/systemctl --no-block start backup_avm.service'
...
```

Listing 4.1: Verificacion que regla udev es lanzada

Descarga de datos

Para verificar que la descarga de datos se realiza correctamente se deben verificar dos cosas. Que el servicio es lanzado de forma correcta y que efectivamente `rsync` está copiando los datos.

Para verificar que el servicio está operando, se monitorea el estado de este. De ser lanzado de forma exitosa se pueden dar dos casos, que el servicio se encuentra

activo o que este ya fue lanzado. El primer caso se puede ver en 4.3 donde se indica en la última línea mostrada que el servicio está corriendo. El segundo caso se muestra en 4.2 donde se ve que el servicio se encuentra inactivo, pero ya fue lanzado con anterioridad de manera satisfactoria.

```
backup_avm.service — Backup from AVM
Loaded: loaded (/etc/systemd/system/backup_avm.service; static; vendor
       preset: enabled)
Active: inactive (dead)
...
Sep 19 14:08:54 raspberrypi systemd[1]: backup_avm.service: Succeeded.
```

Listing 4.2: Estado del servicio 'backup-avm' cuando este termina su funcion.

```
pi@raspberrypi:~ $ sudo systemctl status backup_avm.service
backup_avm.service — Backup from AVM
Loaded: loaded (/etc/systemd/system/backup_avm.service; static; vendor
       preset: enabled)
Active: active (running) since Sat 2020-09-19 14:08:49 -03; 1s ago
...
```

Listing 4.3: Estado del servicio 'backup-avm' mientras opera.

Luego para verificar que la transferencia es realizada de forma correcta se debe ver la salida del script cuando el dispositivo se encuentra conectado. Esto se muestra en 4.5 donde se ve que el dispositivo es montado correctamente y luego los 3 archivos que contenía se transfieren de forma exitosa.

```
mount: /dev/sdcl mounted on /mnt/AVM.
sending incremental file list
./
AVM1-18-09-1
 100,000,000 100% 11.89MB/s 0:00:08 (xfr#1, to-chk=2/4)
AVM1-18-09-2
 100,000,000 100% 7.91MB/s 0:00:12 (xfr#2, to-chk=1/4)
AVM1-18-09-3
 100,000,000 100% 9.08MB/s 0:00:10 (xfr#3, to-chk=0/4)

sent 300,073,436 bytes received 76 bytes 10,171,983.46 bytes/sec
total size is 300,000,000 speedup is 1.00
```

Listing 4.4: Salida del servicio de transferencia de archivos entre AVM y Docking Station.

Ademas, de la penúltima línea de 4.5 se puede obtener una velocidad de transferencia aproximada entre Docking Station y dispositivo USB, la cual es 10MB/s.

Registro de actividad

Acá se muestra el funcionamiento que tiene el registro de actividad, junto con los posibles mensajes que tiene este modulo. En 4.5 se ve cuando es conectado el AVM, seguido de los 3 archivos que se suben. Finalmente se ve cuando el dispositivo es desconectado.

Se puede ver que en el registro de actividad el tiempo en que se conecta el AVM y el tiempo en que se realiza la primera transferencia son el mismo. Esto se debe a que rsync solo informa la hora de inicio de cada una de las transferencias de archivos.

```
2020/12/07 05:38:23 AVM conectado
2020/12/07 05:38:23 Transferencia AVM->Docking Station AVM1-18-09-1
2020/12/07 05:38:31 Transferencia AVM->Docking Station AVM1-18-09-2
2020/12/07 05:38:40 Transferencia AVM->Docking Station AVM1-18-09-3
2020/12/07 05:39:40 AVM removido
```

Listing 4.5: Registro de actividad del modulo de almacenamiento

En caso de que el dispositivo sea desconectado el registro de actividad sera similar a lo mostrado en 4.6. En este caso se puede ver que primero aparece el dispositivo desconectado y luego la transferencia, aunque el horario en que aparecen es correcto ya que la transferencia ocurre antes que el dispositivo sea removido. Esto ocurre simplemente porque el script que detecta que el dispositivo fue removido escribe antes el el log que el script que se encuentra transfiriendo archivos.

```
2020/12/07 05:57:03 AVM conectado
2020/12/07 05:57:08 AVM removido
2020/12/07 05:57:03 Transferencia AVM->Docking Station AVM1-18-09-1
```

Listing 4.6: Registro de actividad cuando el AVM es desconectado

Es importante mencionar que en caso de un corte abrupto en la alimentación del Docking Station, este no podrá registrar lo que ocurrió con los archivos ya transferidos, esto debido a que la escritura en el registro ocurre después de terminar la transferencia y no a medida que se transfiere.

Capítulo 5

Comunicación con Usuario

En esta sección se trabajara en la conexión entre el teléfono del usuario y el Docking Station, y en como se realizará la comunicación de forma que se pueda configurar el Docking Station de manera sencilla. Como se dijo en el segundo capítulo la forma de comunicación que se implementará será una comunicación serial a través de Bluetooth. Para utilizar Bluetooth la RPI y por lo tanto Linux en su defecto cuenta con un conjunto de herramientas bajo el nombre de Bluez, el cual permite interactuar con diversos dispositivos Bluetooth a través de una consola.

En forma general se busca establecer una comunicación para poder configurar específicamente la red Wifi a la que se conectara el Docking Station, y el horario en que los datos serán enviados al servidor. Esto puede traducirse en los requisitos mostrados en la tabla 5.1.

Requisitos funcionales	Pruebas
Pareo bluetooth entre Docking Station y teléfono de forma automática.	Mediante la pulsación de un botón, se debe poder descubrir el Docking Station y poder parear con este de forma automática.
Docking Station puede conectarse a una red WiFi mediante el teléfono del usuario.	La función utilizada para realizar esta conexión debe poder modificar el archivo con las redes disponibles.
Modificar horario de transferencia de información al servidor	La función utilizada para realizar la modificación de horarios, debe ser capaz de modificar los archivos encargados de subir y de terminar la transferencia.

Cuadro 5.1: Requisitos funcionales para la conexión entre Docking Station y teléfono.

5.1. Actividades Desarrolladas

Pareo Automático

Lo primero es establecer el pareo de ambos dispositivos. Debido a que el bluetooth de la Raspberry Pi funciona a través de una consola se utilizará una herramienta adicional conocida como expect, la que permite automatizar por medio de un script las interacciones con herramientas que se utilizan mediante consola.

Para comenzar el pareo se configuró un botón utilizando los pines de propósito general el cual iniciará el script descrito anteriormente al ser presionado por un par de segundos.

Para desarrollar este script se deben primero verificar los pasos necesarios para realizar el pareo de forma manual, los cuales deben ser transcritos al script para que otorguen la misma funcionalidad.

Por lo tanto, este script lo primero que hará será iniciar la herramienta que controla los dispositivos bluetooth, luego hará que el Docking Station sea detectable por otros dispositivos y esperará la conexión de algún dispositivo. Una vez que esta conexión ocurra el script tomará la dirección MAC de este dispositivo y ejecutara

los respectivos comandos para aceptar el pareo.

Para agregar seguridad al Docking Station, cada vez que se realice el pareo de un dispositivo, si existe un dispositivo pareado anteriormente este sera eliminado. De esta forma solo un usuario puede tener control y comunicación con el Docking Station a través de su celular.

Establecer Comunicación

Antes de establecer un canal de comunicación serial se debe configurar el servicio que inicia todo lo relacionado con las herramientas que usan Bluetooth. Se configurará para que cada vez que se inicie agregue un puerto serial el cual será utilizado para realizar esta conexión. Para establecer este canal de comunicación serial se utilizarán dos servicios. El primero se encargará de escuchar constantemente si se intenta establecer una conexión serial, y el segundo iniciará el programa que permite intercambiar mensajes cuando se detecte una conexión. El servicio encargado de escuchar comunicaciones estará configurado para iniciarse después del servicio que habilita y configura Bluetooth en el Docking Station. Cuando se inicia una conexión, este servicio crea un archivo, correspondiente al dispositivo conectado bajo un nombre específico el cual está relacionado con el puerto serial utilizado el cual siempre será el mismo.

El servicio encargado del intercambio de mensajes estará configurado para iniciarse cuando se detecte el archivo creado cuando se inicia la comunicación. Este servicio se encargara de correr un script en Python el cual enviará un mensaje al usuario indicando que la conexión fue realizada de manera exitosa, y permitirá la interacción del usuario con el Docking Station.

Establecer Conexión WiFi

Para poder conectar el Docking Station a internet es necesario que el usuario pueda entregar el nombre de su red y su contraseña de alguna forma. En este caso

se utilizará el script mencionado anteriormente para transmitir esta información. Cuando se selecciona esta opción, el Docking Station identifica las redes disponibles y las envía al usuario para que este pueda escoger entre estas. Una vez escogida la red se pide su contraseña. Se debe mencionar que entregar una contraseña de un largo no aceptado, es decir que contenga menos de 8 caracteres, provoca que esta sea pedida de forma indefinida al usuario hasta que este entregue una contraseña de largo valido. Una vez obtenidas estas, se procede a agregar la red.

Para configurar una red Wifi con esta información es necesario modificar el archivo `wpa_supplicant`, el cual almacena la información de las redes WiFi a las que se puede conectar el Docking Station. Por lo que una vez recibida la información de la red a conectarse, se verifica que es posible conectarse a la red y de ser posible, se escribe la información de esta red en el archivo mencionado anteriormente.

Es importante mencionar que el sistema operativo se encargará de buscar todas las redes disponibles en este archivo y se conectara de forma automática a la que presente una mejor conexión. Por lo que agregar una red no significa que el dispositivo quedará conectado a esta necesariamente. Por lo anterior, y para verificar que la red agregada funciona de forma correcta, se eliminan temporalmente las redes guardadas anteriormente y solo se utiliza la nueva red hasta poder verificar si es posible conectarse a internet con ésta.

Finalmente se verifica si existe conexión a internet con esta red. Para esto se realizara un ping a un servidor de Google. Es decir, se probará si es posible conectarse a un servidor de Google. De ser posible, se le comunicará al usuario que existe una conexión a internet y se volverán a escribir las redes que fueron removidas anteriormente.

Horarios de subida de información

Para cambiar los horarios en que se sube la información es necesario modificar dos servicios, el que comienza la subida de información y el que la termina. Para

esto al usuario se le preguntará de forma separada el horario de subida y el de bajada, los cuales deben ser entregados en un formato HH:MM. Una vez que estos sean entregados se modificara los respectivos servicios con los horarios entregados y finalmente se reiniciara la herramienta systemd para realizar la actualización de estos cambios.

Una vez realizados estos cambios, se verificara si la transferencia al servidor debiese estar ocurriendo en este momento. De ser este el caso, se lanzara el servicio encargado de realizar esta transferencia. Para lograr esto, se lanzara un script que compara la hora actual con las horas a las que están configurados estos servicios. Si la hora actual esta entre la hora que se inicia la subida y la hora en que finaliza, se lanzara el servicio encargado de realizar la transferencia.

Es importante mencionar que reiniciar esta herramienta interrumpe las transferencias entre Docking Station, AVM y servidor que se estén llevando a cabo. Se tomo la decisión de esperar a que las transferencias del AVM al Docking Station terminen antes de realizar el reinicio. Esto debido a que al ser una transferencia local, es más rápida y no puede ser resumida en caso de ser interrumpida, por lo que se deben enviar nuevamente los archivos cuya transferencia se vea interrumpida. Esto no ocurre con la transferencia a través de internet, la cual permite reanudar archivos en caso de ser interrumpida.

También se debe mencionar que todo lo anteriormente descrito solo se permite si se cuenta con conexión a internet. Esta decisión fue tomada ya que la Raspberry Pi sobre la que se desarrolla el Docking Station, puede ajustarse al horario actual solo al estar conectada a internet. Para evitar que el servicio encargado de subir los datos sea lanzado cuando no corresponde debido a que el Docking Station no se encuentra configurado con el horario correcto, se tomo la decisión de solo permitir modificar estos servicios si se cuenta con conexión a internet.

Registro de Actividad

En este registro de actividad se registrarán las actividades que se realicen en la aplicación. Para esto se escribirán en el registro diversos mensajes solo cuando una operación sea exitosa. Es decir, cuando se conecte a una red de forma satisfactoria, cuando se cambien los horarios de subida y bajada de la información y cuando se eliminen todas las redes WiFi.

5.2. Pruebas

Pareo Automático

Para probar esto se lanzará el script que realiza el pareo y se verificará su salida. En 5.1 se muestra lo más importante de la salida de este en donde se puede ver varios de los pasos realizados.

La primera línea inicia la herramienta (spawn bluetoothctl), luego se ve como el dispositivo se hace detectable y pareable para otros dispositivos. Luego de realizar esto se espera hasta que un dispositivo intente realizar el pareo, lo cual ocurre en la línea que comienza con [NEW]. Finalmente se ve que el dispositivo es pareado de forma correcta y no se aceptan más pareos de otros dispositivos.

```
spawn sudo bluetoothctl
...
[bluetooth]# discoverable on
Changing discoverable on succeeded
[bluetooth]# pairable on
[CHG] Controller B8:27:EB:B7:EE:F3 Discoverable: yes
Changing pairable on succeeded
[NEW] Device 7C:03:AB:37:86:A0 Danas
...
[Danas]# Device paired correctly
pairable off
...
```

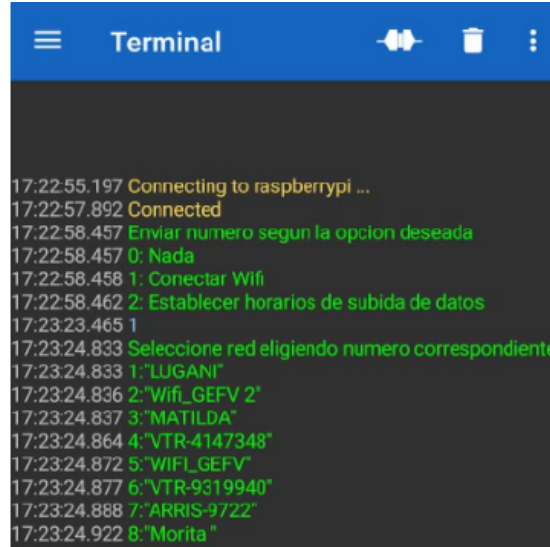
Listing 5.1: Salida resumida del script que para dispositivos

Establecer Comunicación

Para probar que la comunicación es establecida correctamente se utilizará la aplicación Serial Bluetooth Terminal. Además se asume que los dispositivos ya se encuentran pareados al realizar esto.

Se utilizará la aplicación para intentar establecer la comunicación. De realizarse de forma correcta se mostrará lo que se ve en la figura 5.1. Donde se ve en verde los mensajes que envía el Docking Station y en azul los mensajes enviados desde la aplicación.

Se puede ver que una vez establecida la conexión se envía automáticamente un mensaje con lo que se puede verificar que el script que realiza la comunicación se está ejecutando correctamente. Además se puede ver la forma en que es posible interactuar con la aplicación.



```
17:22:55.197 Connecting to raspberrypi ...
17:22:57.892 Connected
17:22:58.457 Enviar numero segun la opcion deseada
17:22:58.457 0: Nada
17:22:58.458 1: Conectar Wifi
17:22:58.462 2: Establecer horarios de subida de datos
17:23:23.465 1
17:23:24.833 Seleccione red eligiendo numero correspondiente
17:23:24.833 1:"LUGANI"
17:23:24.836 2:"Wifi_GEFV 2"
17:23:24.837 3:"MATILDA"
17:23:24.864 4:"VTR-4147348"
17:23:24.872 5:"WIFI_GEFV"
17:23:24.877 6:"VTR-9319940"
17:23:24.888 7:"ARRIS-9722"
17:23:24.922 8:"Morita "
```

Figura 5.1: Utilización de la aplicación.

Configuración WiFi

Para probar esto, se busca que el script sea capaz de modificar el archivo `wpa_supplicant` con la información que le es entregada, siempre y cuando la red

entregada pueda conectarse a internet.

En la figura 5.2 se puede ver que la red seleccionada tiene nombre 'Wifi_GEFV 2' y la contraseña que se entrega '80829671'. Se ve que en este proceso aparece dos veces el mensaje 'Configurando red WiFi'. Esto es debido a que primero se configura para conectarse a la red entregada, y luego con el archivo con todas las redes agregadas hasta el momento.

En la figura 5.3 se puede ver como se entrega una clave errónea para la misma red. En este caso también se configura dos veces la conexión a redes WiFi, pero la segunda vez no se encuentra explícita. En este caso es reemplazada por el mensaje 'Conexión con la red no disponible...'.

Finalmente se debe verificar que la red fue agregada correctamente al archivo antes mencionado. En el listado 5.2 se ve que tan solo se agrego el caso mostrado en la figura 5.2.

```
pi@raspberrypi:~ $ sudo cat /etc/wpa_supplicant/wpa_supplicant.conf
country=cl
update_config=1
ctrl_interface=/var/run/wpa_supplicant

network={
    ssid="Wifi_GEFV 2"
    #psk="80829671"
    psk=44afe115eb040f6b6547b734c6fb906b1f6c699d54f31f9bd2cc8ec3d18125ae
}
```

Listing 5.2: Archivo con credenciales WiFi agregadas.

Configuración horario de subida

Finalmente queda por verificar que utilizando la aplicación de celular, es posible cambiar los horarios en que se ejecutan los servicios encargados de realizar la subida de información. Para realizar esta prueba se verificarán los servicios timers encargados de lanzar los servicios mencionados anteriormente. Estos presentan un parámetro 'OnCalendar' que indica la hora a la que serán ejecutados. Por lo que este es el parámetro que se busca cambiar a través del uso

```

00:56:20.580 Seleccionar red eligiendo el numero correspondiente
00:56:20.588 1:"Wifi_GEFV 2"
00:56:20.595 2:"LUGANI"
00:56:20.596 3:"ARRIS-9722"
00:56:20.626 4:"MATILDA"
00:56:20.633 5:"WIFLGEFV"
00:56:20.641 6:"VTR-9319940"
00:56:20.643 7:"VTR-4147348"
00:56:23.346 1
00:56:23.361 Entregue la contraseña de la red
00:56:28.115 80829671
00:56:28.131 Chequeando nueva red
00:56:28.213
00:56:28.213 Configurando red wifi, por favor espere
00:56:44.791 Intentando conectar a internet
00:56:44.791 Conectado a la red :Wifi_GEFV 2
00:56:52.917
00:56:52.922 Configurando red wifi, por favor espere
00:57:09.086 Red agregada de forma exitosa

```

Figura 5.2: Conexión correcta a red WiFi desde la aplicación.

```

00:25:51.762 Seleccionar red eligiendo el numero correspondiente
00:25:51.796 1:"Wifi_GEFV 2"
00:25:52.007 2:"LUGANI"
00:25:52.112 3:"VTR-4147348"
00:25:52.115 4:"ARRIS-9722"
00:25:52.120 5:"VTR-9319940"
00:25:52.133 6:"MATILDA"
00:25:55.061 1
00:25:55.101 Entregue la contraseña de la red
00:26:00.097 asdf1234
00:26:00.671 Chequeando nueva red
00:26:00.674
00:26:00.674 Configurando red wifi, por favor espere
00:26:16.886 Intentando conectar a internet
00:26:34.878 Conexión con la red no disponible, por favor intente nuevamente

```

Figura 5.3: Conexión errónea a red Wifi desde la aplicación.

de la aplicación.

En la figura 5.4 se puede ver que se modifica el horario en que empieza la transferencia a las 1:00AM y termine a las 7:00AM. En la primera línea del listado 5.3 se puede ver que la hora a la que inicia es efectivamente las 01:00 (1:00AM), y en la segunda línea se ve que la hora a la que termina la transferencia son las 07:00 (7:00AM). Adicionalmente al final de la figura 5.4 se ve un mensaje indicando que la transferencia hacia el servidor inició. Esto concuerda con la hora a la que esto fue realizado, la cual es mostrada al comienzo de cada mensaje, en color blanco. De la misma forma en la figura 5.5 se puede ver un nuevo cambio en el horario, el cual se ve reflejado nuevamente en el listado 5.3 en los últimos dos comandos ejecutados. Adicionalmente se ve en esta figura que la transferencia de archivos finalizo, lo que nuevamente concuerda con el horario en que se realizo lo anterior. La forma en que los mensajes que informan de la transferencia al servidor son enviados a la aplicación, se detallara en el capítulo 7, encargado de integrar todos los servicios del Docking Station.

```
04:48:41.628 2
04:48:41.664 Intentando conectar a internet
04:48:41.903 A que hora desea comenzar a subir la informac
n?
04:48:41.904 Formato 24 hrs(HH:MM):04:48:55.892 01:00
04:48:56.804 A que hora desea detener la subida de datos?
04:48:56.805 Formato 24 hrs(0 - 23):04:49:04.574 07:00
04:49:06.415 Enviar numero segun la opcion deseada
04:49:06.420 0: Nada
04:49:06.420 1: Conectar Wifi
04:49:06.420 2: Establecer horarios de subida de datos
04:49:06.420 3: Eliminar todas las redes Wifi
04:49:06.546 Comenzando transferencias de archivos al servid
or
```

Figura 5.4: Modificación horario de subida, con inicio de transferencia.

```
pi@raspberrypi:~ $ sudo cat .../upload-server.timer | grep "OnCalendar"
OnCalendar=*-*-* 01:00
pi@raspberrypi:~ $ sudo cat .../stop-upload.timer | grep "OnCalendar"
OnCalendar=*-*-* 07:00
pi@raspberrypi:~ $ sudo cat .../upload-server.timer | grep "OnCalendar"
OnCalendar=*-*-* 12:00
pi@raspberrypi:~ $ sudo cat .../stop-upload.timer | grep "OnCalendar"
OnCalendar=*-*-* 23:00
```

Listing 5.3: Tiempos en los que se ejecuta las transferencias al servidor.

```

04:56:45.715 2
04:56:46.606 Intentando conectar a internet
04:56:46.608 A que hora desea comenzar a subir la informacio
n?
04:56:46.609 Formato 24 hrs(HH:MM):04:56:54.014 12:00
04:56:54.705 A que hora desea detener la subida de datos?
04:56:54.709 Formato 24 hrs(0 - 23):04:56:58.516 23:00
04:57:00.657 Enviar numero segun la opcion deseada
04:57:00.700 0: Nada
04:57:00.719 1: Conectar Wifi
04:57:00.756 2: Establecer horarios de subida de datos
04:57:00.792 3: Eliminar todas las redes Wifi
04:57:00.793 Transferencias de archivos al servidor terminada

```

Figura 5.5: Modificación horario de subida, sin inicio de transferencia.

Registro de Actividad

Para probar el funcionamiento del registro de actividad, se detallarán las acciones tomadas en las pruebas anteriores. Estas se pueden ver en el listado 5.4.

Lo primero que se ve es la red agregada en la figura 5.2. Luego se ve el cambio de horario realizado en la figura 5.4. Después de realizar esto se ven dos mensajes indicando el inicio de transferencia al servidor y un error al conectarse a este, los cuales serán explicados en el siguiente capítulo.

Finalmente puede verse el cambio de horario realizado en 5.5 junto con un mensaje indicando que la transferencia de archivos ha terminado.

```

2020/12/06 01:48:11 Agregada red :Wifi_GEFV 2
2020/12/06 04:49:07 Cambio de horarios transferencia al servidor. Horario de
    inicio: 01:00 Horario de termino 07:00
2020/12/06 04:49:07 Comenzando horario de transferencia al servidor
2020/12/06 04:49:10 Error al conectarse al servidor
2020/12/06 04:57:00 Cambio de horarios transferencia al servidor. Horario de
    inicio: 12:00 Horario de termino 23:00
2020/12/06 04:57:01 Horario de transferencia terminado

```

Listing 5.4: Registro de actividad del módulo de comunicación.

Capítulo 6

Conexión con Servidor

En este capítulo se tratará todo lo relacionado a la conexión con el servidor desde el Docking Station. Se tratará también los temas relacionados con la transferencia de información médica mediante esta conexión y como se realiza esta de forma segura.

El objetivo principal de este capítulo es desarrollar una transferencia de archivos automatizada de forma segura y robusta, además como se transfiere información medica al igual que para la conexión con el AVM se debe mantener un registro de los archivos transferidos. Esto se puede traducir en los siguientes requisitos.

6.1. Actividades Desarrolladas

Conexión con el Servidor

El servicio encargado de realizar la subida de datos ejecutara un script que utilizara rsync sobre SSH para conectarse al servidor. Esta conexión mediante SSH presenta dos problemas al ser automatizada. El primero es que en general se pide un usuario y una contraseña para conectarse servidor. El segundo, es que la primera vez que se realice la conexión el Docking Station preguntara si confiamos

Requisitos funcionales	Pruebas
Transferencia de archivos al servidor en horario correspondiente.	Se debe ver que los servicios encargados de iniciar y de terminar la transferencia son lanzados a las horas correspondientes.
Transferencia al servidor automatizada.	Se debe poder transferir archivos al servidor ejecutando el script encargado de esta transferencia, sin otra intervención.
Transferencia de archivos al servidor de forma robusta.	Verificar que la transferencia se intenta de forma constante aunque no exista conexión con el servidor.
Se debe tener un registro de las transferencias de archivos entre, Docking Station y servidor.	Al realizar una transferencia esta debe quedar registrada en el archivo correspondiente.

Cuadro 6.1: Requisitos funcionales y pruebas para conexión Docking Station con servidor.

en el servidor al que se desea conectar.

Para evitar que SSH pida un usuario y una contraseña se debe copiar la llave pública a utilizar generada por el Docking Station al servidor. Esto provocará que el servidor confíe en el host(en este caso Docking Station) y le permita conectarse sin pedir contraseña. Esto también sirve para el cumplimiento de las reglas HIPAA, ya que según estas, cada paciente debe tener un identificador único, por lo que en este caso, se puede utilizar la llave pública del Docking Station.

Luego para evitar que el Docking Station pida una confirmación la primera vez que se realiza la conexión al servidor, la llave pública que utiliza el servidor debe ser copiada a la carpeta 'known_hosts' del Docking Station, donde se guardan los servidores en los que se confía que son seguros.

Transferencia Automática

En este punto se trata de como es posible realizar la transferencia de manera automatizada en los horarios elegidos por el usuario.

Para realizar la transferencia de forma automática, se crearon dos servicios. El primero encargado de iniciar la transferencia de archivos al servidor y el segundo

de terminarla.

Ambos estarán configurados para ejecutarse a cierta hora y para terminarse en caso de que el otro sea iniciado. De esta forma se puede asegurar que cuando se inicie uno de los servicios el otro no continúe activo.

El principal problema que presenta este esquema es que si el Docking Station no se encuentra encendido a la hora en que estos servicios son iniciados, la transferencia no iniciara. Para poder iniciar la transferencia en estos casos, se lanzara un servicio al momento de encender el Docking Station, el cual verificara si la hora actual corresponde a una hora en que se deba estar corriendo el servicio. De ser este el caso, lanzara el servicio que inicia las transferencias.

Esto hace que surja otro problema, el cual es que el horario del Docking Station se ajusta en base a una conexión a internet. Por lo que si no hay conexión a internet no se puede asegurar que el horario que tiene el Docking Station sea el correcto. Esto sera solucionado restringiendo el servicio encargado de subir los datos a que solo inicie si existe internet.

Esta restricción solo se agrega al comienzo del servicio, ya que con esto se asume que fue iniciado a la hora correcta. Luego si la conexión a internet es cortada durante el tiempo que el servicio se encuentra online, el estado del servicio no cambiara.

Una desconexión de internet durante el servicio de subida sera tratada de la misma forma que un fallo en la conexión al servidor, ya que ambos arrojarán un error mientras rsync se este ejecutando. La diferencia sera que el fallo de conexión al servidor sera escrito en el registro de actividad, mientras que la falta de internet se dejara pasar en este sentido.

Tambien existe el caso que el Docking Station se inicie sin internet y luego pueda conectarse. Como es complicado saber cuando exista conexión a internet, se lanzara el mismo servicio que se lanza al momento de iniciar el Docking Station cada cierto tiempo arbitrario (en este caso se escogio 1 minuto). Esto significa que

cada un minuto se chequeara si el servicio de transferencia debiese estar corriendo o no.

Transferencia a través de un canal seguro

Como se dijo anteriormente la transferencia de archivos será realizada utilizando rsync sobre SSH. Para aumentar la seguridad que proporciona SSH se deben tomar ciertas medidas las cuales serán nombradas a continuación.

El primer paso a tomar es cambiar el puerto que utiliza SSH (por defecto el 22), esto de por sí no es una medida que prevenga de un ataque exhaustivo, pero permite evitar gran parte del tráfico entrante proveniente de atacantes automatizados que intentan buscar vulnerabilidades en algún dispositivo.

Aprovechando lo hecho en el punto anterior, poseer de forma anticipada la llave (sin haber intentado una conexión con el servidor por primera vez) permite dejar sin efectividad un tipo de ataque conocido como man-in-the-middle, que consisten como su nombre lo indica en alguien que se encuentre en medio de la conexión viendo los paquetes que se envían y se reciben. Esto debido a la forma en que funciona la encriptación asimétrica, donde la única forma de leer un mensaje encriptado con la llave pública es utilizando la llave privada, por lo que de existir alguien haciéndose pasar por el servidor no podría descifrar los mensajes.

Otras consideraciones necesarias son desactivar el acceso mediante contraseña al Docking Station para que solo puedan realizarse comunicaciones contando con las respectivas llaves. Desactivar el protocolo SSH1 y solo utilizar SSH2 el cual es una versión actualizada, y además desactivar ciertas formas de encriptación consideradas no seguras.

Finalmente se configurará el firewall del Docking Station para que solo acepte conexiones de ciertas ip como la del servidor, reduciendo un posible trafico entrante.

Realizando estas configuraciones es posible aumentar la seguridad de la conexión.

Reanudar Transferencias

Como se ha mencionado, se utilizara rsync para transferir los archivos, este tiene la particularidad de resumir transferencias interrumpidas. Estas interrupciones pueden ser debido a diversas situaciones, como por ejemplo cortes de luz, internet y caídas del servidor, entre otras. Además el poder reanudar transferencias ayuda en casos de conexiones débiles o inestables.

Debido a que esta opción viene por defecto utilizando rsync, el desarrollo en este punto va en como asegurar que la transferencia sea realizada cuando corresponde. Si el problema es de alimentación del Docking Station la solución es lanzar el servicio cuando se inicie el Docking Station, lo cual fue tratado en la sección de transferencia automática.

Para tratar cortes en la conexión a internet y al servidor, se asume que el servicio fue iniciado a la hora correcta. Luego se programa el script que lanza este servicio para que siga intentando cada cierto tiempo realizar la transferencia de archivos al servidor. De esta forma se asegura que mientras el servicio se encuentre activo y sea necesario realizar alguna transferencia, se mantendrá intentando realizar esta.

Registro de Actividad

En el registro de actividad se anotarán las actividades más relevantes que realice este modulo, entre las que se incluye: Indicar cuando el servicio de transferencia se inicia, indicar cuando termina, anotar los archivos transferidos e informar cuando exista un error en la comunicación con el servidor.

6.2. Pruebas

Para realizar las pruebas que requieran conexión con el servidor se utilizará una maquina virtual con sistema operativo CentOS 7 y SSH configurado de la misma forma que el Docking Station.

Conexión con el Servidor

Lo primero es probar que la configuración del servidor y el Docking Station se encuentra realizada de forma correcta. Para esto se debe verificar que es posible establecer una comunicación SSH sin mas que entrar el comando SSH, usuario y dirección ip en la consola. Esto se puede realizar de manera sencilla, tal como se muestra en el listado 6.1, donde se ve el login al servidor solo con su ip y el usuario.

```
pi@raspberrypi:~ $ ssh asdf@192.168.1.25
Last login: Fri Nov 20 22:30:59 2020
[asdf@localhost ~]$
```

Listing 6.1: Conexion SSH desde Docking Station al servidor

Transferencia automática

Existen diversas cosas que se deben probar en este punto. Entre estas se incluyen las siguientes:

- La transferencia debe iniciar y terminar a la hora correspondiente.
- Si el Docking Station es iniciado durante el horario de transferencia, esta debe ser iniciada.
- Si el horario de transferencia es cambiado y se esta dentro del horario de transferencia, debe iniciar la transferencia.

- Si no hay conexión a internet, no se debe iniciar la transferencia.

En este caso las tres primeras pruebas se realizan para verificar el funcionamiento de lo desarrollado. La cuarta se realiza debido a que existen problemas al configurar el horario del Docking Station cuando no existe conexión a internet.

Para realizar las distintas pruebas, se utilizara lo desarrollado en el capítulo 5 para poder controlar los horarios de transferencia al servidor, eliminar la conexión a internet y para agregarla nuevamente. También se utilizará el registro de actividad de esta sección el cual será explicado más adelante.

Lo primero es probar que el servicio inicia y termina a la hora correspondiente. Para esto se utilizara la aplicación de celular para modificar el horario. En 6.2 se puede ver que la hora a la que se inicia el servicio fue modificada a las 4:45 y la de término a las 4:47. Luego se ve que a esa hora el servicio es iniciado, y que a las 4:47, la transferencia termina. En este caso se ve que existió un error de conexión con el servidor, ya que este no se encuentra online al momento de realizar estas pruebas.

2020/12/08 04:43:42	Cambio de horarios transferencia al servidor. Horario de inicio: 04:45 Horario de termino 04:47
2020/12/08 04:45:10	Comenzando horario de transferencia al servidor
2020/12/08 04:45:16	Error al conectarse al servidor
2020/12/08 04:47:08	Horario de transferencia terminado

Listing 6.2: Registro de actividad cuando la subida de datos se realiza de forma ideal.

Luego se verificara que el Docking Station inicia la transferencia si cuando es iniciado se encuentra dentro del horario de transferencia. En este caso la transferencia se programo para iniciarse entre 4:00 y 6:00. Luego se procede a apagar el Docking Station y encenderlo nuevamente. En el listado 6.3 se puede ver que un minuto después de que se registro que el Docking Station fue iniciado, se comenzó la transferencia de archivos. Esta diferencia de tiempos se debe a que el registro del inicio del Docking Station ocurre de forma inmediata, mientras que para iniciar la transferencia se debe esperar a que se configure la red WiFi después de iniciar, por lo que se le da un tiempo para que esto ocurra.

```
2020/12/08 05:29:26 Docking Station iniciado
2020/12/08 05:30:23 Comenzando horario de transferencia al servidor
2020/12/08 05:30:29 Error al conectarse al servidor
```

Listing 6.3: Registro de actividad cuando el Docking Station es iniciado y debe estar transfiriendo.

Lo tercero es verificar que si el horario de transferencia es cambiado el servicio cambiara su estado según corresponda. Para esto se utilizara la aplicación de celular para cambiar el estado del servicio. En el listado 6.4 se puede ver que se cambia el horario de transferencia para que esta ocurra entre 12:00 y 23:00. Debido que al momento de realizar esta transferencia son las 17:47, se inicia la transferencia.

Luego el horario vuelve a cambiarse para que la transferencia inicie a las 12:00 y termine a las 23:00, y posteriormente se ve que la transferencia fue terminada.

```
2020/12/08 17:47:21 Cambio de horarios transferencia al servidor. Horario de
    inicio: 12:00 Horario de termino 23:00
2020/12/08 17:47:23 Comenzando horario de transferencia al servidor
2020/12/08 17:47:28 Error al conectarse al servidor
2020/12/08 17:47:42 Cambio de horarios transferencia al servidor. Horario de
    inicio: 23:00 Horario de termino 12:00
2020/12/08 17:47:42 Horario de transferencia terminado
```

Listing 6.4: Registro de actividad cuan.

Finalmente queda verificar que el servicio no es iniciado sin conexión a internet, y que de estar corriendo, no corta su funcionamiento si se corta internet. Para cortar el internet se utilizara la aplicación de celular para eliminar las redes WiFi. Se describirá el proceso mostrado en el listado 6.5. Lo primero que se realiza es cambiar el horario de inicio y luego eliminar las redes y la conexión a internet. Se puede ver luego que solo se envía el mensaje "Comenzando horario de transferencia" luego de que se agrega nuevamente una red con conexión a internet. Con esto se comprueba que si no existe internet, el servicio no sera iniciado. Luego se ve el mensaje de error, debido a que existe conexión a internet. Recordando que si no es posible establecer una conexión al servidor, esta se intentara nuevamente en 10 minutos, se elimina la conexión a internet, y se procede a verificar que esta vez no se escriba el mensaje de error de conexion al

servidor, recordando que este mensaje solo se escribe si existe conexión a internet.

Finalmente se ve el mensaje de que el horario de transferencia ha terminado

```
2020/12/08 14:10:18 Cambio de horarios transferencia al servidor. Horario de
                inicio: 14:11 Horario de termino 14:30
2020/12/08 14:10:31 Eliminadas todas las redes wifi
2020/12/08 14:15:35 Agregada red :Wifi_GEFV 2
2020/12/08 14:16:05 Comenzando horario de transferencia al servidor
2020/12/08 14:16:11 Error al conectarse al servidor
2020/12/08 14:22:03 Eliminadas todas las redes wifi
2020/12/08 14:30:31 Horario de transferencia terminado
```

Listing 6.5: Prueba de funcionamiento con y sin internet.

Reanudar transferencias

En el punto anterior se mostró que el servicio corre siempre a la hora deseada. Por lo que solo queda probar que rsync, permite reanudar las transferencias interrumpidas al ser ejecutado cuando esto ocurrio.

Para probar que es posible reanudar transferencias primero se debe interrumpir una, para lo cual se lanzará y terminar el servicio encargado de subir archivos mientras este se encuentra activo. Luego se debe relanzar este servicio y de hacerse de forma correcta se debe ver algo similar a lo visto en 6.6, en donde el archivo 'AVM01-20-10-1' de un tamaño aproximadamente de 3GB fue subido en tan solo 2 min. Esto es posible debido a que el archivo ya se encontraba parcialmente en el servidor por lo que no debe subirse completo.

Para mostrar que no es posible alcanzar tal velocidad de subida se utilizó iperf, una herramienta que permite medir la velocidad de conexión entre dos dispositivos. Esta herramienta entrega la información mostrada en 6.7 que muestra una tasa máxima de transferencia de aproximadamente 2.7MB por segundo.

```
opening connection using: ssh -l asdf 192.168.1.25 rsync --server
...
sending incremental file list
delta-transmission enabled
AVM01-20-10-1
3,000,000,000 100% 23.47MB/s 0:02:01 (xfr#1, to-chk=3/5)
...
sent 300,655,481 bytes received 147,442 bytes 855,769.34 bytes/sec
total size is 3,300,000,000 speedup is 10.97
```

Listing 6.6: Transferencia de archivos desde Docking Station al servidor

```

pi@raspberrypi:~ $ iperf3 -c 192.168.1.25
Connecting to host 192.168.1.25, port 5201
[ 5] local 192.168.1.17 port 36492 connected to 192.168.1.25 port 5201
...
-- -- -- -- --
[ ID] Interval          Transfer      Bitrate      Retr          sender
[ 5] 0.00-10.00 sec 27.3 MBytes 22.9 Mbits/sec 1
[ 5] 0.00-10.00 sec 26.1 MBytes 21.9 Mbits/sec
receiver
iperf Done.

```

Listing 6.7: Salida herramienta iperf que entrega la velocidad de conexion.

Respecto a probar que la transferencia se intenta de forma constante, esto se puede ver en el listado 6.8, donde se muestra que cada 10 minutos desde que se inicia la transferencia, se intenta realizar esta nuevamente. En este caso y al igual que en los anteriores, se muestran mensajes de error de conexión al servidor, debido a que esta prueba se realiza con el servidor apagado.

```

2020/12/08 05:29:26 Docking Station iniciado
2020/12/08 05:30:23 Comenzando horario de transferencia al servidor
2020/12/08 05:30:29 Error al conectarse al servidor
2020/12/08 05:40:34 Error al conectarse al servidor
...

```

Listing 6.8: Registro donde se muestra una conexion constante.

Conexión Segura

Probar que el protocolo SSH es seguro escapa del alcance de este trabajo. Existen diversos trabajos como los mostrados en [11] y [12] que muestran de forma más detallada el funcionamiento de este protocolo y la razón por la que es seguro.

Registro de transferencias

El registro de transferencia fue utilizado con anterioridad, para realizar pruebas. El unico mensaje que falta por mostrar es el de un archivo transferido al servidor.

Estos se registran de la misma forma en que se transfieren archivos del Docking

Station al servidor. Es decir, se toma la salida de rsync, se buscan los archivos transferidos al 100 % y se anotan como se ve en 6.9 en el archivo de registro de actividad.

```
2020/11/20 22:26:27  Transferencia: Docking Station->Servidor
                    AVM1-19-09-1
2020/11/20 22:27:41  Transferencia: Docking Station->Servidor
                    AVM1-19-09-2
2020/11/20 22:28:53  Transferencia: Docking Station->Servidor
                    AVM1-19-09-3
```

Listing 6.9: Registro de los archivos subidos al servidor

Capítulo 7

Integración y Resultado Final

En esta sección se detalla como fue integrado los módulos desarrollados en los capítulos anteriores, se mostrara el resultado final y se analizara las falencias que tiene el sistema y como es posible mejorarlas.

7.1. Integración

El objetivo de integrar todos los módulos es realizar una comunicación entre estos, lo que permitirá mejorar la funcionalidad de cada uno. Para realizar esta comunicación se utilizara una herramienta llamada 'named pipes', la cual permite una comunicación entre procesos utilizando una estructura FIFO (First in first out) para enviar y recibir mensajes.

Para integrar los módulos se crearán dos pipes, resultando en un esquema como el mostrado en la figura 7.1, donde se ven las conexiones entre los modulos y la dirección de escritura en los pipes.

Pipe de comunicación

El objetivo de este pipe es recibir información de los otros módulos para notificar al usuario de ciertas actividades del Docking Station, entre las que se

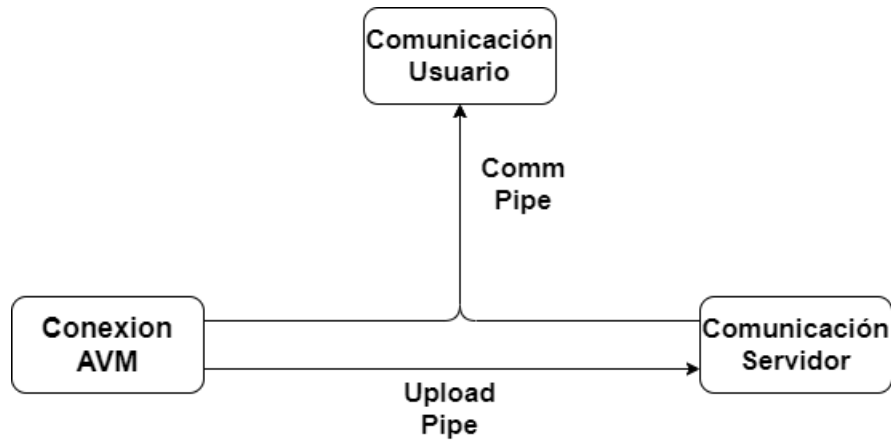


Figura 7.1: Arquitectura interna de pipes

incluyen: notificar al usuario cuando comiencen y terminen las transferencias hacia el servidor, y notificar cuando el AVM se encuentre descargando datos al Docking Station y cuando termine de realizar lo anterior.

Para implementar esto se creo un servicio que sera iniciado junto al servicio principal encargado de la comunicación desarrollado en el capítulo 5. Este servicio lanzara un script de forma paralela al utilizado para interactuar con el usuario. Por lo que este servicio se mantendrá corriendo mientras el teléfono del usuario este conectado tras lo cual sera terminado.

Lo primero que realiza este script es crear el pipe para que los otros procesos puedan comunicarse con este. Luego cuando se escriba algo en este pipe, el script leera lo escrito y según lo que se lea, enviara distintos mensajes al teléfono del usuario.

Finalmente cuando el usuario se desconecte, cerrara y eliminara el pipe, por lo cual este solo existirá mientras el teléfono del usuario se encuentre conectado.

Para escribir en el pipe, los otros dos módulos (conexión con AVM y comunicación servidor) seguirán el mismo proceso. Antes de enviar un mensaje verificarán que existe el pipe, y posteriormente escribirán en este. Esto es importante debido a que escribir en un pipe no existe o que no esta siendo leído, hará que el servicio

que envía el mensaje se quede detenido en este punto.

Este proceso de comunicación funciona de forma paralela al realizado anteriormente, por lo que ambos pueden enviar mensajes a la vez, aunque solo el descrito en el capítulo 5 puede recibir estos.

Pipe de subida

El pipe de subida tendrá el objetivo de informar al modulo encargado de la comunicación con el servidor que pueden existir nuevos datos en el Docking Station esperando a subirse.

Este pipe sera creado cuando el servicio encargado de subir los archivos al servidor comience a ejecutarse y se mantendrá abierto y leyendo lo que se escriba en este, luego de que intente una primera transferencia, y mientras el servicio no sea cerrado.

El encargado de escribir en este pipe sera el modulo de conexión con el AVM. Cuando se conecte el AVM y se terminen de transferir los datos al Docking Station, se enviara un mensaje para iniciar la subida de datos al servidor. Es importante notar que si el pipe existe esto significa que se esta en el horario establecido por el usuario para realizar esta transferencia, ya que el pipe junto al servicio correspondiente a la subida de información es borrado una vez termina este horario.

En caso de que el AVM sea removido, se considerara que la transferencia de archivos ya fue realizada y se indicara el inicio de transmisiones de igual forma, lo que solo transfiere los archivos que se hayan subido completamente.

Mezclando ambos componentes se puede entregar información adicional al usuario que pueden servir como notificaciones, tales como los mensajes mostrados en las figuras 5.4 y 5.5 donde se ve que se le informa al usuario cuando la transferencia al servidor comienza y termina.

7.2. Resultado Final

Aplicando lo anterior, es posible notificar al usuario de distintas actividades del Docking Station. En la figura 7.2 se puede ver como el usuario es notificado cuando el Docking Station se encuentra respaldando archivos desde el AVM.

También como se vio en el capítulo 5, en las figuras 5.4 y 5.5, se envían mensajes al usuario cuando el Docking Station comienza a transferir archivos al servidor y cuando deja de hacer esto.

```
17:52:56.366 Enviar numero segun la opcion deseada
17:52:56.367 0: Nada
17:52:56.368 1: Conectar Wifi
17:52:56.368 2: Establecer horarios de subida de datos
17:52:56.368 3: Eliminar todas las redes Wifi
17:53:18.142 Transfiriendo archivos AVM->Docking Station
17:53:18.176 Transferencia AVM->Docking Station terminada
```

Figura 7.2: Mensaje enviado a la aplicación cuando el AVM transfiere datos.

En el listado 7.1 se da un ejemplo de lo que podría ser el funcionamiento normal en un día de uso del Docking Station. Se ve que una vez que el AVM es conectado, transfiere todos los archivos presentes en el AVM al Docking Station, y una vez que esto terminan de transferirse, se transfieren desde el Docking Station al servidor.

```
2020/12/08 18:24:43 AVM conectado
2020/12/08 18:24:43 Transferencia AVM->Docking Station AVM1-18-10-1
2020/12/08 18:25:11 Transferencia AVM->Docking Station AVM1-18-10-2
2020/12/08 18:28:11 Transferencia AVM->Docking Station AVM1-18-10-3
2020/12/08 18:28:49 Transferencia Docking Station->servidor AVM1-18-10-1
2020/12/08 18:30:37 Transferencia Docking Station->servidor AVM1-18-10-2
2020/12/08 18:32:19 Transferencia Docking Station->servidor AVM1-18-10-3
2020/12/08 18:48:01 AVM removido
```

Listing 7.1: Ejemplo de registro de actividad en un uso diario.

7.2.1. Problemas del Sistema

En esta sección se describirán diversos inconvenientes que presenta el sistema y que no es posible solucionar completamente por software.

Problemas de Alimentación

En general el problema más grave que puede presentar el sistema, es un corte de alimentación repentino, ya que este no permite cerrar los procesos que se estén ejecutando.

Si ocurriese durante la transferencia de archivos del AVM al Docking Station, podrían quedar archivos corruptos en el Docking Station. Además debido a que termina el proceso de forma abrupta, no se podrá registrar las transferencias de archivos que se hayan completado. Esto se contrarresta en cierta medida utilizando rsync, ya que este chequea que los archivos sean correctamente transferidos a costa de una velocidad de transferencia menor.

Respecto a la comunicación con el usuario, el principal problema de cortarse la alimentación es si esto ocurre mientras se configura una red WiFi, ya que puede cortarse mientras las redes WiFi agregadas se encuentran en un archivo temporal y no en el que corresponde. Esto provoca que se pierdan las redes WiFi, lo cual no es un problema crítico ya que pueden agregarse nuevamente desde la aplicación. El proceso realizado para transferir archivos al servidor, el único problema que presenta es que no quedarán registrados los archivos que se hayan transferido. Tal como se vio en la sección en la que se desarrolló esto, es posible reanudar una transferencia interrumpida.

Una posible solución a esto es agregar al Docking Station un circuito para un apagado seguro. El cual permita alimentar por un periodo corto de tiempo el Docking Station, cuando sea cortada la alimentación y así poder cerrar de manera correcta los procesos que se estén corriendo.

Problemas de Comunicación con el Usuario

El problema que presenta la comunicación con el usuario, es la falta de una aplicación para celular. Debido a esto, la información viaja sin encriptar, lo que no es recomendable ya que es una vulnerabilidad que presenta el sistema. Además utilizar un método de comunicación serial requiere estar constantemente escuchando si la otra parte envía algo, por lo que no es una forma eficiente de realizar la comunicación.

En este punto es recomendable desarrollar una aplicación, que utilice un protocolo de comunicación distinto (o conjunto de protocolos), como por ejemplo MQTT y FCM.

Ajuste de Hora del Docking Station

El ultimo problema que se presenta es la falta de un reloj de tiempo real en el Docking Station, el cual permite conocer la hora sin necesidad de conexión a internet. Lo que evita chequear la conexión cada vez que se inicie la transferencia. Por lo que ahorra recursos en el sistema.

Conclusiones

En este trabajo se desarrollo la componente de software de un sistema complementario al dispositivo AVM, el cual entrega una mayor autonomía al funcionamiento de este.

Lo primero que se realizo fue un análisis de acuerdo a la ley HIPAA, de las componentes presentes en el sistema desarrollado. Se realizo un análisis de las distintas vulnerabilidades que este presenta para que estas pudiesen ser tratadas a lo largo de este trabajo.

Se desarrollo un sistema que permite respaldar los datos de un dispositivo de almacenamiento conectándolo al Docking Station. El cual cumple la función de transferir la información del AVM al Docking Station. En este punto no se pudo agregar mayor seguridad a esta conexión, debido a que no es posible trabajar con un AVM, por lo que existe un trabajo pendiente respecto a esto.

También se desarrollo una forma de conectarse entre el Docking Station y el telefono del usuario mediante el uso de Bluetooth. En este punto, lo más importante son las funciones que permiten configurar las redes WiFi a las que se puede conectar el Docking Station, y el horario en que el Docking Station subirá los datos del AVM a un servidor externo. Un trabajo pendiente en este punto es realizar una aplicación para el teléfono, que permita mejorar la comunicación con el usuario, y otorgar mayor comodidad para este.

Finalmente, se automatizo la transferencia de los datos descargados desde el AVM, la cual permite reanudar transferencias cuando estas son interrumpidas, y

es realizada de acuerdo a la ley HIPAA.

Finalmente se puede concluir que el sistema desarrollado cumple con respecto al tratamiento de la información medica de un paciente al momento de movilizar esta.

Bibliografía

- [1] Tecnología liderada por académico usm recibe premio emprendimiento ingeniería 2030 de corfo. <https://bit.ly/3nDBvbE>, Jul 2019.
- [2] Alexandru Archip, Nicolae Botezatu, Elena Serban, Paul-Corneliu Herghelegiu, and Andrei Zala. An iot based system for remote patient monitoring. *2016 17th International Carpathian Control Conference (ICCC)*, 2016.
- [3] M. Jayson Baucas and P. Spachos. Fog and iot-based remote patient monitoring architecture using speech recognition. In *2020 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6, 2020.
- [4] H. Bhojwani, G. K. Sain, and G. P. Sharma. A hybrid connectivity oriented telemedician system for indian landscape using raspberry pi sbc iot. In *2018 3rd Technology Innovation Management and Engineering Science International Conference (TIMES-iCON)*, pages 1–5, 2018.
- [5] Ashikur Rahaman, Md Islam, Rashedul Islam, Muhammad Sadi, and Sheikh Nooruddin. Developing iot based smart health monitoring systems: A review. *Revue d intelligence artificielle*, 33:435–440, 12 2019.
- [6] D. S. R. Krishnan, S. C. Gupta, and T. Choudhury. An iot based patient monitoring system and temperature controlling system. *International*

Journal of Innovative Technology and Exploring Engineering Regular Issue, 9(7):1172–1175, 2020.

- [7] S. Rajagopalan, S. Janakiraman, A. Rengarajan, S. Rethinam, S. Arumugham, and G. Saravanan. Iot framework for secure medical image transmission. In *2018 International Conference on Computer Communication and Informatics (ICCCI)*, pages 1–5, 2018.
- [8] Mythili Thirugnanam and Somasundaram Ragupathy. Internet of things (iot) technologies, applications, challenges, and solutions. 05 2019.
- [9] Felipe Basile and F. Amate. Secure transmission of medical images by ssh tunneling. volume 173, pages 486–490, 07 2011.
- [10] T. Wang, J. Zhou, A. Liu, M. Z. A. Bhuiyan, G. Wang, and W. Jia. Fog-based computing and storage offloading for data synchronization in iot. *IEEE Internet of Things Journal*, 6(3):4272–4282, 2019.
- [11] Burt Kaliski. The mathematics of the rsa public-key cryptosystem.
- [12] NIDHI KANDHIL and D. A. Kumar. A study on secure shell (ssh) protocol. 2012.

Apéndice A

Salidas de scripts

En este apéndice se entregan copias más completas de todos los scripts utilizados.

4.1

```
pi@raspberrypi:~ $ udevadm test DEVPATH
...
run:  '/bin/systemctl --no-block start backup_avm.service'
...
```

Listing A.1: Verificación que regla udev es lanzada

4.2

```
backup_avm.service - Backup from AVM
Loaded: loaded (/etc/systemd/system/backup_avm.service; static; vendor
       preset: enabled)
Active: inactive (dead)

Sep 19 14:08:49 raspberrypi systemd[1]: Started Backup from AVM.
Sep 19 14:08:49 raspberrypi sudo[4875]:      root : TTY=unknown ; PWD=/ ;
       USER=root ; COMMAND=/bin/device_added.sh
Sep 19 14:08:49 raspberrypi sudo[4875]: pam_unix(sudo:session): session
       opened for user root by (uid=0)
Sep 19 14:08:54 raspberrypi sudo[4875]: pam_unix(sudo:session): session
       closed for user root
Sep 19 14:08:54 raspberrypi systemd[1]: backup_avm.service: Succeeded.
```

Listing A.2: Estado del servicio 'backup-avm' cuando este termina su función

4.3

```
backup_avm.service - Backup from AVM
Loaded: loaded (/etc/systemd/system/backup_avm.service; static; vendor
       preset: enabled)
Active: active (running) since Sat 2020-09-19 14:08:49 -03; 1s ago
Main PID: 4875 (sudo)
```

```

Tasks: 3 (limit: 2200)
Memory: 1020.0K
CGroup: /system.slice/backup_avm.service
4875 /usr/bin/sudo /bin/device_added.sh
4880 /bin/bash /bin/device_added.sh
4881 sleep 5

Sep 19 14:08:49 raspberrypi systemd[1]: Started Backup from AVM.
Sep 19 14:08:49 raspberrypi sudo[4875]:      root : TTY=unknown ; PWD=/ ;
USER=root ; COMMAND=/bin/device_added.sh
Sep 19 14:08:49 raspberrypi sudo[4875]: pam_unix(sudo:session): session
opened for user root by (uid=0)
pi@raspberrypi:~ $ sudo systemctl status backup_avm.service

```

Listing A.3: Estado del servicio 'backup-avm' mientras opera

4.5

```

mount: /dev/sdc1 mounted on /mnt/AVM.
sending incremental file list
./
AVM1-18-09-1
32,768 0% 0.00kB/s 0:00:00
13,762,560 13% 13.09MB/s 0:00:06
23,625,728 23% 11.22MB/s 0:00:06
39,354,368 39% 12.45MB/s 0:00:04
56,393,728 56% 13.39MB/s 0:00:03
73,433,088 73% 14.17MB/s 0:00:01
73,465,856 73% 10.79MB/s 0:00:02
90,734,592 90% 11.12MB/s 0:00:00
100,000,000 100% 11.89MB/s 0:00:08 (xfr#1, to-chk=2/4)
AVM1-18-09-2
32,768 0% 51.61kB/s 0:32:16
6,324,224 6% 5.98MB/s 0:00:15
23,101,440 23% 10.93MB/s 0:00:06
31,817,728 31% 4.93MB/s 0:00:13
48,562,176 48% 6.48MB/s 0:00:07
54,886,400 54% 5.58MB/s 0:00:07
71,335,936 71% 5.54MB/s 0:00:05
87,851,008 87% 10.33MB/s 0:00:01
100,000,000 100% 7.91MB/s 0:00:12 (xfr#2, to-chk=1/4)
AVM1-18-09-3
32,768 0% 43.01kB/s 0:38:44
4,489,216 4% 4.27MB/s 0:00:21
21,266,432 21% 10.12MB/s 0:00:07
30,441,472 30% 7.43MB/s 0:00:09
35,684,352 35% 6.92MB/s 0:00:09
52,985,856 52% 9.41MB/s 0:00:04
58,032,128 58% 5.79MB/s 0:00:07
75,268,096 75% 8.28MB/s 0:00:02
92,372,992 92% 10.50MB/s 0:00:00
100,000,000 100% 9.08MB/s 0:00:10 (xfr#3, to-chk=0/4)

sent 300,073,436 bytes received 76 bytes 10,171,983.46 bytes/sec
total size is 300,000,000 speedup is 1.00
12 100% 0.01kB/s 0:00:00 (xfr#5, to-chk=0/7)

sent 300,073,745 bytes received 123 bytes 12,769,100.77 bytes/sec
total size is 300,000,088 speedup is 1.00

```

Listing A.4: Salida del servicio de transferencia de archivos entre AVM y Docking Station

5.1

```

spawn sudo bluetoothctl
Agent registered
[bluetooth]# power on
Changing power on succeeded
[bluetooth]# agent on
Agent is already registered
[bluetooth]# discoverable on
Changing discoverable on succeeded
[bluetooth]# pairable on
[CHG] Controller B8:27:EB:B7:EE:F3 Discoverable: yes
Changing pairable on succeeded
[NEW] Device 7C:03:AB:37:86:A0 Danas
[CHG] Device 7C:03:AB:37:86:A0 Modalias: bluetooth:v000Fp1200d1436
[CHG] Device 7C:03:AB:37:86:A0 UUIDs: 00001105
      -0000-1000-8000-00805f9b34fb
[CHG] Device 7C:03:AB:37:86:A0 UUIDs: 00001110
      a-0000-1000-8000-00805f9b34fb
[CHG] Device 7C:03:AB:37:86:A0 UUIDs: 00001110
      c-0000-1000-8000-00805f9b34fb
[CHG] Device 7C:03:AB:37:86:A0 UUIDs: 00001110
      e-0000-1000-8000-00805f9b34fb
[CHG] Device 7C:03:AB:37:86:A0 UUIDs: 00001112
      -0000-1000-8000-00805f9b34fb
[CHG] Device 7C:03:AB:37:86:A0 UUIDs: 00001115
      -0000-1000-8000-00805f9b34fb
[CHG] Device 7C:03:AB:37:86:A0 UUIDs: 00001116
      -0000-1000-8000-00805f9b34fb
[CHG] Device 7C:03:AB:37:86:A0 UUIDs: 00001111
      f-0000-1000-8000-00805f9b34fb
[CHG] Device 7C:03:AB:37:86:A0 UUIDs: 00001112
      d-0000-1000-8000-00805f9b34fb
[CHG] Device 7C:03:AB:37:86:A0 UUIDs: 00001112
      f-0000-1000-8000-00805f9b34fb
[CHG] Device 7C:03:AB:37:86:A0 UUIDs: 00001132
      -0000-1000-8000-00805f9b34fb
[CHG] Device 7C:03:AB:37:86:A0 UUIDs: 00001200
      -0000-1000-8000-00805f9b34fb
[CHG] Device 7C:03:AB:37:86:A0 UUIDs: 00001800
      -0000-1000-8000-00805f9b34fb
[CHG] Device 7C:03:AB:37:86:A0 UUIDs: 00001801
      -0000-1000-8000-00805f9b34fb
[CHG] Device 7C:03:AB:37:86:A0 ServicesResolved: yes
[CHG] Device 7C:03:AB:37:86:A0 Paired: yes
[Danas]# paired-devices
Device 7C:03:AB:37:86:A0 Danas
[Danas]# 7C:03:AB:37:86:A0trust 7C:03:AB:37:86:A0
[Danas]# Device paired correctly

pairable off

```

Listing A.5: Script que para dispositivos

5.3

```

pi@raspberrypi:~ $ sudo systemctl list-timers
NEXT                LEFT                LAST                ACTIVATES
PASSED             UNIT
Thu 2020-11-12 14:38:45 -03 4min 7s left  n/a
/a systemd-tmpfiles-clean.timer systemd-tmpfiles-clean.service n
Thu 2020-11-12 20:00:00 -03 5h 25min left n/a
/a stop-upload.timer stop-upload.service
Fri 2020-11-13 00:00:00 -03 9h left Thu 2020-11-12 00:00:16 -03
14h ago logrotate.timer logrotate.service

```

```

Fri 2020-11-13 00:00:00 -03 9h left      Thu 2020-11-12 00:00:16 -03
14h ago man-db.timer                  man-db.service
Fri 2020-11-13 03:50:41 -03 13h left    Thu 2020-11-12 14:24:23 -03
10min ago apt-daily.timer              apt-daily.service
Fri 2020-11-13 06:40:10 -03 16h left    Thu 2020-11-12 14:24:23 -03
10min ago apt-daily-upgrade.timer      apt-daily-upgrade.service
Fri 2020-11-13 13:00:00 -03 22h left    Thu 2020-11-12 14:24:23 -03
10min ago upload-server.timer          upload-server.service

7 timers listed.
Pass --all to see loaded but inactive timers, too.
pi@raspberrypi:~ $ sudo systemctl list-timers
NEXT                                LAST
PASSED    UNIT                                ACTIVATES
Thu 2020-11-12 14:38:45 -03 2min 28s left n/a
/a systemd-tmpfiles-clean.timer systemd-tmpfiles-clean.service
Thu 2020-11-12 17:00:00 -03 2h 23min left n/a
/a stop-upload.timer stop-upload.service
Thu 2020-11-12 18:40:04 -03 4h 3min left Thu 2020-11-12 14:24:23 -03
11min ago apt-daily.timer apt-daily.service
Fri 2020-11-13 00:00:00 -03 9h left      Thu 2020-11-12 00:00:16 -03
14h ago logrotate.timer logrotate.service
Fri 2020-11-13 00:00:00 -03 9h left      Thu 2020-11-12 00:00:16 -03
14h ago man-db.timer man-db.service
Fri 2020-11-13 06:40:40 -03 16h left    Thu 2020-11-12 14:24:23 -03
11min ago apt-daily-upgrade.timer apt-daily-upgrade.service
Fri 2020-11-13 11:00:00 -03 20h left    Thu 2020-11-12 14:24:23 -03
11min ago upload-server.timer upload-server.service

7 timers listed.
Pass --all to see loaded but inactive timers, too.

```

Listing A.6: Timers activos del sistema.

6.6

```

opening connection using: ssh -l asdf 192.168.1.25 rsync --server
-vvlogDtpre.iL

sfxC --partial . backup/ (10 args)
sending incremental file list
delta-transmission enabled
AVM01-20-10-1
3,000,000,000 100% 23.47MB/s 0:02:01 (xfr#1, to-chk=3/5)
AVM1-19-09-1
100,000,000 100% 1.21MB/s 0:01:18 (xfr#2, to-chk=2/5)
AVM1-19-09-2
100,000,000 100% 1.27MB/s 0:01:14 (xfr#3, to-chk=1/5)
AVM1-19-09-3
100,000,000 100% 1.27MB/s 0:01:14 (xfr#4, to-chk=0/5)
total: matches=142585 hash_hits=142585 false_alarms=0 data=300011600

sent 300,655,481 bytes received 147,442 bytes 855,769.34 bytes/sec
total size is 3,300,000,000 speedup is 10.97

```

Listing A.7: Conexion ssh desde Docking Station al servidor

A.8

```

pi@raspberrypi:~ $ iperf3 -c 192.168.1.25
Connecting to host 192.168.1.25, port 5201
[ 5] local 192.168.1.17 port 36492 connected to 192.168.1.25 port 5201
[ ID] Interval Transfer Bitrate Retr Cwnd
[ 5] 0.00-1.00 sec 3.14 MBytes 26.4 Mbits/sec 0 239 KBytes
[ 5] 1.00-2.00 sec 2.67 MBytes 22.4 Mbits/sec 0 283 KBytes
[ 5] 2.00-3.00 sec 2.73 MBytes 22.9 Mbits/sec 0 301 KBytes

```

[	5]	3.00–4.00	sec	2.80	MBytes	23.5	Mbits/sec	0	337	KBytes
[	5]	4.00–5.00	sec	2.55	MBytes	21.4	Mbits/sec	0	358	KBytes
[	5]	5.00–6.00	sec	2.73	MBytes	22.9	Mbits/sec	0	393	KBytes
[	5]	6.00–7.00	sec	2.30	MBytes	19.3	Mbits/sec	0	393	KBytes
[	5]	7.00–8.00	sec	2.80	MBytes	23.5	Mbits/sec	1	273	KBytes
[	5]	8.00–9.00	sec	2.73	MBytes	22.9	Mbits/sec	0	308	KBytes
[	5]	9.00–10.00	sec	2.80	MBytes	23.5	Mbits/sec	0	341	KBytes
[	ID]	Interval			Transfer		Bit rate		Retr	
[	5]	0.00–10.00	sec	27.3	MBytes	22.9	Mbits/sec		1	
	sender									
[	5]	0.00–10.00	sec	26.1	MBytes	21.9	Mbits/sec			
	receiver									
iperf Done.										

Listing A.8: Salida herramienta iperf

Apéndice B

Código

En este apéndice se entrega el enlace con los códigos descritos en este trabajo.

En este mismo se encuentran las configuraciones para identificar un AVM.

<https://github.com/Danas1/avm-docking-station-entrega/tree/main>