

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

IMPLEMENTACIÓN DE UN PROTOTIPO DE KVM SOBRE IP

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN REDES Y TELECOMUNICACIONES**

BRYAN STEVEN GUACHAMIN PARREÑO

DIRECTOR: LEANDRO ANTONIO PAZMIÑO ORTIZ

DMQ, agosto 2022

CERTIFICACIONES

Yo, BRYAN STEVEN GUACHAMIN PARREÑO declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



BRYAN GUACHAMÍN

bryan.guachamin@epn.edu.ec

bsgp23msj@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por Bryan Steven Guachamín Parreño, bajo mi supervisión.



LEANDRO PAZMIÑO

DIRECTOR

leandro.pazmino@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

A handwritten signature in blue ink, appearing to be 'Bryan Steven Guachamín Parreño', written in a cursive style.

Bryan Steven Guachamín Parreño

1754300034

DEDICATORIA

El desarrollo, implementación e investigación del presente proyecto se lo dedico a mi hermana Mishella Guachamín quien a través de su carisma y espontaneidad ha sido un pilar fundamental en mi crecimiento académico y personal, llenándome de fortaleza para seguir adelante. Es así como mediante esta dedicatoria quiero dejar plasmada la importancia de esta persona en mi vida en conjunto con un mensaje de apoyo para que cumpla sus sueños y recuerde que todo lo que se proponga lo podrá alcanzar con dedicación y decisión. Este logro es de los dos mi querida hermana.

AGRADECIMIENTO

Agradezco a mis padres, Diana Parreño y Byron Guachamín quienes con su paciencia supieron guiarme por la travesía universitaria. Además, agradezco a mi tío Jimmy Parreño pues con su apoyo y consejo me enseñó a comprender, superar y afrontar diferentes adversidades. También quiero agradecer a Guissela Cárdenas, pues con su cariño y comprensión me ha acompañado desde el momento en que nos conocimos llenándome de fuerza, valor y alegría. Con la misma gratitud menciono a mi familia y compañeros de la universidad quienes estuvieron aportando constantemente sus conocimientos para ayudarme a mejorar. Finalmente agradezco a mi tutor de tesis, el ING. Leandro Pazmiño quien estuvo dispuesto a aportar todo su conocimiento para el desarrollo correcto de este proyecto de titulación. Mi gratitud hacia ustedes.

ÍNDICE DE CONTENIDOS

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
RESUMEN	VII
<i>ABSTRACT</i>	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
1.1 Objetivo general	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco Teórico	2
<i>Raspberry Pi 4b</i>	2
<i>Software Pi KVM sobre IP</i>	3
Convertor de <i>HDMI</i> a <i>CSI</i>	4
Cable <i>USB</i> tipo <i>C</i>	5
<i>Software Raspberry Pi Imager</i>	5
Software <i>Putty</i>	5
Archivos <i>.yaml</i>	5
2 METODOLOGÍA.....	6
3 RESULTADOS	7
3.1 Identificación los requerimientos para el diseño del prototipo de <i>KVM</i>	7
3.2 Selección del <i>hardware</i> y <i>software</i> requeridos para el prototipo.....	9
Hardware	9
Software	12

3.3	Diseño del prototipo de KVM.....	13
	Esquema general del prototipo	13
3.4	Implementar el prototipo de <i>KVM</i>	19
	Adaptación de la carcasa de TV Box.....	19
	Montaje de <i>Raspberry Pi</i> sobre la carcasa.....	19
	Conexión de <i>Raspberry</i> a la red local.....	20
3.5	Realizar pruebas de funcionamiento del prototipo.....	21
	Acceso al servidor <i>PI KVM</i>	21
	Comprobación de operabilidad	22
	Ingreso al <i>BIOS</i> del computador controlado.....	24
	Creación de carpeta y reubicación de esta	26
	Revisión de <i>IP</i> del equipo desde los símbolos del sistema	26
	Recorte de pantalla	27
	Reproducción de video	29
	Funcionamiento y mantenimiento del prototipo.....	29
	Costo del proyecto	29
4	CONCLUSIONES.....	30
5	RECOMENDACIONES	33
6	REFERENCIAS BIBLIOGRÁFICAS	34
7	ANEXOS.....	38
	ANEXO I: CERTIFICADO DE ORIGINALIDAD	i
	ANEXO II: Enlaces a videos explicativos.....	ii

RESUMEN

La primera sección desarrollada describe la problemática presentada y por ende en que consiste el proyecto desarrollado, así mismo se detallan los objetivos específicos en base al objetivo general y se explican los conceptos fundamentales en conjunto con características propias del *Raspberry Pi*, el controlador *Pi – KVM*, el comportamiento de los *scripts* “.yaml” y el adaptador *HDMI a CSI*.

En la segunda sección se desarrolla la metodología aplicada en donde se explica el procedimiento correspondiente para que el proyecto mencionado cumpla con los objetivos planteados y solucione la problemática.

Dentro de la sección de resultados, que es el tercer apartado, se puede evidenciar de forma específica los requerimientos y especificaciones de los elementos necesarios para la implementación del proyecto, así mismo se indica la selección del microordenador *Raspberry Pi 4B* y el software libre de la plataforma *Pi – KVM* y su funcionamiento a través de la *web*. Además, se visualiza los pasos a seguir para el diseño y la implementación del prototipo junto con la modificación del *script* de nombre *override.yaml*. También se observa las pruebas realizadas con el prototipo en base a cada uno de los objetivos planteados con la finalidad de determinar su adecuado funcionamiento.

Posterior al desarrollo del proyecto se plasmaron las conclusiones y recomendaciones en la cuarta sección en base al desarrollo del prototipo.

Finalmente, se tienen las referencias bibliográficas de donde se obtuvo la información necesaria y los anexos del escrito, que contienen los certificados de originalidad y códigos *QR* con videos explicativos del proyecto.

PALABRAS CLAVE: *Pi – KVM, Raspberry Pi, web, script, Adaptador.*

ABSTRACT

The first section describes the problem and therefore what the project is about. On the same way, it details the specifics objectives based on the general objective, explains the fundamental concepts along with Raspberry Pi's own features, the Pi - KVM controller, the .yaml scripts function and the HDMI to CSI adapter.

The second section develops the methodology applied where the correspondent procedure is explained so that the mention project reaches the proposed objectives and solves the problem.

Within the results section, which is the third section, the requirements and specifications of the elements necessary for the implementation of the project can be specifically evidenced, as well as the selection of the Raspberry Pi 4B microcomputer and the free software of the Pi - KVM platform and its operation through the web. In addition, the steps to follow for the design and implementation of the prototype are visualized along with the modification of the script named override.yaml. It also shows the tests performed with the prototype based on each of the objectives set to determine its proper functioning.

After the development of the project, the conclusions and recommendations were embodied in the fourth section, based on the development of the prototype. Finally, there are the bibliographic references from which the necessary information was obtained and the annexes to the document, which contain the certificates of originality and QR codes with explanatory videos of the project.

KEYWORDS: *Pi – KVM, Raspberry Pi, web, script, Adapter.*

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El mundo evoluciona constantemente y cada cambio trae consigo una nueva forma de solucionar problemas mediante el desarrollo de aplicaciones, equipos o simplemente con variaciones en la rutina de las personas. La pandemia desatada por el Covid-19 es un gran ejemplo de este caso, puesto que mucha gente alrededor del mundo dejó de movilizarse a su lugar de trabajo o estudio, para realizar dichas actividades de manera virtual desde la comodidad de su hogar.

Sin embargo, en el caso de varias profesiones, no se pudo palpar esta realidad, como se evidenció en los médicos, quienes continuaron operando y atendiendo a sus pacientes en los casos necesarios. Varios técnicos eléctricos se desplazaban aún a las centrales eléctricas para mantener a la población con electricidad. Los técnicos y proveedores de *Internet* continuaron operando con la finalidad de mantener comunicados a quienes lo necesitaban, por mencionar algunos ejemplos.

Por esta razón, el proyecto en curso trata de resolver la limitante dada por la necesidad de que una persona esté físicamente atendiendo una situación fuera de lo común provocada por factores aleatorios, siempre y cuando sea posible hacerlo. Por lo que, se montará sobre la placa de *Raspberry Pi* un prototipo de control remoto de teclado y mouse de un dispositivo a otro que pertenezcan a una misma red.

Con ayuda del *software Pi Kernel Virtual Machine (KVM)* será posible mantener el control del equipo *Personal Computer (PC)* o servidor incluso cuando este ha sido reiniciado. Hecho que, por consiguiente, facilitará también la administración del *Basic Input Output System (BIOS)*. En otras palabras, se instaurará un prototipo de *KVM* para controlar de manera remota el teclado y *mouse* de un dispositivo (servidor, *PC*) por medio del protocolo *Internet Protocol (IP)*.

La herramienta será implementada con la finalidad de gestionar, administrar y solucionar problemas con computadoras o servidores, reflejando así un claro ejemplo de cómo el ser humano busca generar sus propias soluciones para dar respuesta a problemáticas que pueden afectar al desempeño de actividades cotidianas que requieran de la tecnología para ser realizadas [1].

1.1 Objetivo general

Implementar un prototipo de *KVM* sobre *IP*.

1.2 Objetivos específicos

- Identificar los requerimientos para el diseño del prototipo de *KVM*.
- Seleccionar el *hardware* y *software* requeridos para el prototipo.
- Diseñar el prototipo de *KVM*.
- Implementar el prototipo de *KVM*.
- Realizar pruebas de funcionamiento del prototipo.

1.3 Alcance

Por medio del presente proyecto se busca implementar un prototipo que permita tener acceso remoto a un dispositivo (servidor, *PC*) emulando que los cambios se realizan de manera presencial por el operador, sin requerir la instalación de ningún programa en los dispositivos. En resumen, permitirá realizar lo siguiente:

- Ejecutar tareas que no se pueden realizar a través de un programa de acceso remoto como *Anydesk*, *Logmein*, etc., por ejemplo: reiniciar el dispositivo y mantener el control del dispositivo, además de controlar y realizar cambios incluso a nivel del *BIOS* del dispositivo.
- Controlar de manera remota el *mouse* y teclado del dispositivo por medio de un navegador *web*.

Además, el prototipo podrá ser empleado en cualquier dispositivo que se encuentre en la misma red, para lo cual se requerirá simplemente conectar los periféricos (*mouse*, teclado, video) al prototipo.

1.4 Marco Teórico

Raspberry Pi 4b

La placa madre *Raspberry Pi 4* (Ver Figura 1.1) en su modelo b es un microordenador o microcomputador que permite realizar actividades básicas y comunes de un ordenador personal [3].

El *Raspberry Pi 4b* cuenta con las siguientes características:

- Un procesador *Broadcom BCM2711B0* de cuatro núcleos [2].
- Memoria *Random Access Memory (RAM)* de 2 (GB) de capacidad [2].
- 40 pines *General Purpose Input/Output (GPIO)* para conexiones generales [4].
- 2 puertos *USB 2.0* y 2 puertos *USB 3.0* [2].
- 2 ranuras micro *High-Definition Multimedia Interface (HDMI)* de salida [2].
- 1 puerto *Computer Supported instrucción (CSI)* para la cámara de *Raspberry PI* [5] [2].
- 1 puerto *Delivered Source Instructions (DSI)* para salida a la pantalla de *Raspberry PI* [6].
- Salida auxiliar de audio [2].
- Ranura para tarjeta *Micro Secure Digital (SD)* para montaje del sistema operativo y almacenamiento [7] [2].
- Puerto *USB* tipo *C* para alimentación y transmisión de datos [2].
- 1 ranura *GPIO* para aplicaciones *POE (Power-Over-Ethernet)* [2].

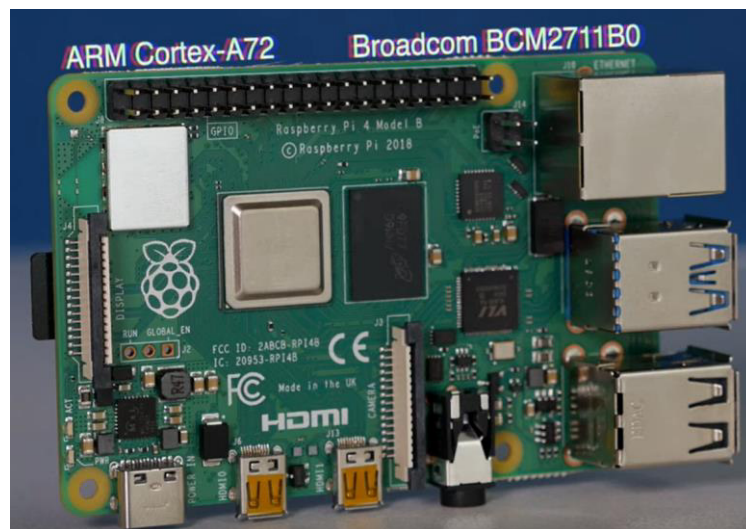


Figura 1.1 Placa *Raspberry Pi 4b* [2]

Software Pi KVM sobre IP

Es un hipervisor dedicado a la virtualización y fue desarrollado en base al código abierto de Linux [8]. Es así como ofrece una virtualización completa de los componentes físicos típicos de un computador (*Hardware* principal del equipo) en un entorno virtual aislado que llega a simular al dispositivo casi en su totalidad [9].

Es de fácil implementación y permite administrar estaciones de trabajo o servidores de manera remota vía *web* dentro de una misma *Local Área Network (LAN)* como muestra la Figura 1.2.

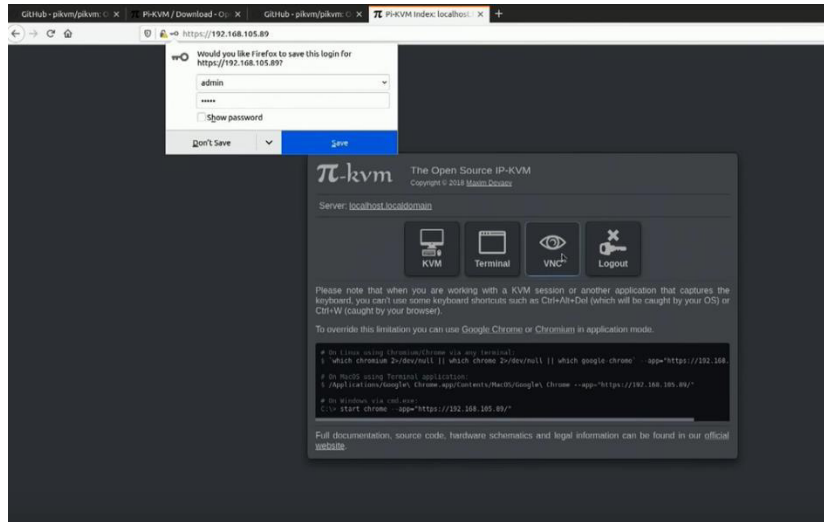


Figura 1.2 Interfaz *Pi KVM* [10]

Algunos aspectos adicionales que se tiene por parte de *Pi KVM* es que, como se mencionó con anterioridad, facilita la revisión del *BIOS* de una computadora, solucionar problemas del sistema operativo e incluso en algunos casos facilita la reinstalación de este mediante un *Compact Disc (CD)* o *USB* gracias a que administra completamente el *hardware* simulado de los equipos controlados, así como el teclado y *mouse* de estos [11].

Convertor de *HDMI* a *CSI*

Es utilizado para adaptar una entrada *HDMI* al puerto *Computer Supported Instruction (CSI)* de *Raspberry Pi*. La señal *HDMI* es convertida a señales *CSI* de entrada de video para el *Raspberry* como se ve en la Figura 1.3 [12].



Figura 1.3 Funcionamiento del adaptador *HDMI* a *CSI* [12]

Cable *USB* tipo C

Es un cable de transferencia de datos y de paso de energía hacia un dispositivo electrónico de uso común. La Figura 1.4 muestra que en una de sus puntas posee un conector macho de tipo *USB* 2.0 y en la otra punta un conector macho *USB* tipo C. Cuenta con una velocidad de transmisión de datos de 480 (*Mbps*) y una capacidad de carga de 5 (V_{DC}) y 500 (*mA*) con una potencia de 2.5 (*W*) [13].

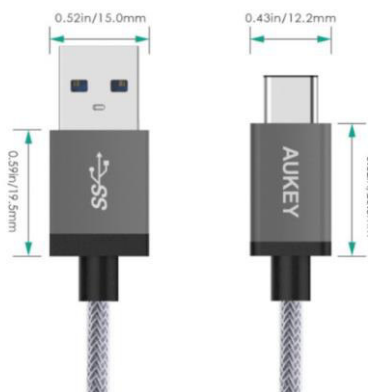


Figura 1.4 Cable USB a USB tipo C [13]

Software *Raspberry Pi Imager*

Es un programa que permite crear la imagen del controlador de *Pi KVM* para que el microordenador pueda arrancar desde dicho programa al encenderse e instalarlo correctamente sobre la placa para su uso [14].

Software *Putty*

Putty es un terminal de simulación de protocolos como por ejemplo *Transport Control Protocol (TCP)*, *Telecommunication Teletype Network (Telnet)* o *Secure Shell (SSH)* que permite realizar conexiones remotas mediante una IP con las políticas de seguridad establecidas en cada uno de los protocolos mencionados con anterioridad. Por lo general es utilizado para la administración de redes gracias a que su interfaz gráfica puede ser instalada en cualquier dispositivo y mediante ella establecer las conexiones [15].

Archivos *.yaml*

Se lo comprende como un archivo que maneje el lenguaje de serialización de datos *Yet Another Markup Language (YAML)*. Este lenguaje es útil con varios lenguajes de programación (*C*, *XML*, *HTML*, *JSON*, *Python*, entre otros) puesto que integra varias

funciones de sus semejantes, lo que hace que su principal aplicación sea en el almacenamiento y diseño de archivos de configuración [16].

YAML es un lenguaje muy popular puesto que por lo general se lo escribe a manera de texto listado y con pocas etiquetas (Ver Figura 1.5) a diferencia del resto de lenguajes que operan a nivel de línea de comandos [17]. Finalmente, gracias a todo lo mencionado con anterioridad *YAML* presenta una infraestructura de alto rendimiento en los dispositivos que lo utilizan [18].

```
nombre: Henrique Marques Fernandes #Texto
edad: 28 #Número
interesado: #Matriz
  - javascript
  - docker
  - aleteo
dirección: #Objeto
padres: Brasil
Estado: Sao Paulo
```

Figura 1.5 Estructura *YAML* [17]

2 METODOLOGÍA

Se realizó un análisis en base a los requisitos detallados en la página oficial de *KVM* en donde se identificaron los requerimientos físicos del prototipo, lo cual determinó las características generales de *hardware* tanto para el levantamiento del servidor de *KVM* (Elementos de procesamiento de datos) así como para su implementación y almacenamiento (Carcasa protectora y periféricos) en los posibles entornos en donde se lo usará. También se determinó el *software* necesario para el diseño del prototipo de *KVM* relacionado a la creación de imágenes de arranque o de controladores especiales.

Una vez identificados los requerimientos, se establecieron las características específicas del *hardware* y el *software* requerido con lo que se seleccionó de los elementos de *hardware* y el *software* adecuados para que el prototipo cumpla con las funciones establecidas. Para el proceso de selección se cumplió con un análisis en base a parámetros como características técnicas, precio, facilidad de uso, compatibilidad, etc.

Es así como en base al *hardware* y *software* seleccionados se procedió a diseñar el prototipo de *KVM* para lo cual a nivel de *hardware* se escogió el microordenador y los periféricos adecuados para el prototipo (adaptador de video y disipador de calor) en

conjunto con la carcasa, la cual fue elaborada de materiales reciclados con la finalidad de disminuir la producción de basura tecnológica y que sirvió de contenedor para el prototipo completo.

Además, se instaló del sistema operativo del prototipo de *KVM* y también se cambió las líneas correspondientes en los archivos necesarios para su buen funcionamiento. Todo esto fue hecho mediante *Putty*, un programa que facilitó la conexión remota al microordenador. El cual fue instalado en la computadora que está dedicada al control del segundo equipo de manera remota.

Con el sistema operativo del prototipo instalado se procedió a hacer uso de los aplicativos necesarios para el despliegue del prototipo de *KVM*. Además, se configuró los *scripts* dentro del sistema operativo del controlador de *KVM* para levantar el servicio del prototipo de *KVM*. Como se mencionó con anterioridad, se modificaron los archivos y ficheros del *software* de *KVM* con la finalidad de asegurar que el prototipo cumpla con los requisitos y objetivos planteados, junto con la compatibilidad adecuada para los diferentes navegadores web desde donde se lo usó.

Finalizada la implementación del prototipo de *KVM* se procedió a realizar pruebas de funcionamiento que permitieron validar el correcto desempeño del prototipo. Los resultados obtenidos de las pruebas fueron usados para subsanar errores en los casos que existieron y de esta manera se garantizó que el prototipo cumplió con todos los requerimientos establecidos.

3 RESULTADOS

A continuación, se detallan los parámetros de selección de los componentes utilizados en cuanto a *hardware* y *software*. Del mismo modo se explica paso a paso el montaje y creación del sistema operativo utilizado sobre el *Raspberry Pi* y la forma en la que opera como control remoto de dispositivos.

3.1 Identificación los requerimientos para el diseño del prototipo de KVM

En base al análisis previo se determinó que el prototipo diseñado debe suplir la necesidad de que un técnico esté físicamente presente para realizar cambios dentro del *BIOS* o con programas que requieran del reinicio de los servidores o computadoras que lo requieran. A continuación, se detalla cada uno de los requisitos.

Equipo de procesamiento

Dentro de los requerimientos físicos del prototipo una de las características importantes son las dimensiones de este, puesto que debe estar en *racks* y no debe ocupar un espacio muy grande para no interferir con el resto de los equipos. En conclusión, el hardware más adecuado es una placa de desarrollo o microordenador gracias a su tamaño, portabilidad y capacidad de procesamiento.

Refrigeración y almacenamiento del microordenador

Mantener la temperatura del microordenador es necesario dado a los grandes periodos de tiempo que estará encendido para manejar los equipos, así como un contenedor que evite el ingreso del polvo en el microordenador. En base a ello, es necesario un disipador de calor atado a la carcasa protectora, pero que no ocupen mayor espacio para evitar interferir en los entornos en los que se aplica.

Ingreso de datos de video

Se requiere que el prototipo maneje datos de video del equipo que está controlando y los transmita por medio de la red. Entonces es necesario que se tenga un ingreso de video al microordenador, por lo que es necesario que exista compatibilidad entre los puertos del microordenador y de los equipos que se va a manejar.

Software controlador

El *software* aplicado tiene la capacidad de manejar el dispositivo de manera remota y al mismo tiempo está transmitiendo en la red el video del equipo controlado para poder manipularlo desde su encendido, a diferencia de aquellos programas que pierden el control del dispositivo cuando este es apagado.

Software para SSH

Del mismo modo se determinó que el programa que brindará la conexión remota hacia los dispositivos de la red deberá brindar la opción de acceder a ellos mediante el protocolo *Security Shell (SSH)* y de esta manera asegurar una conexión segura entre el computador y el microordenador al momento de realizar cambios en su configuración.

Conexión a internet

La conexión del microordenador a la red local está dada por medio de un cable *Unshielded Twisted Pair (UTP)* hacia un *switch* en donde está conectada también la computadora o servidor que actúa como control remoto. El switch utilizado debe permitir manejar una cantidad moderada de datos para evitar retardos en la transmisión de los datos y el manejo remoto de los dispositivos a través del software aplicado.

3.2 Selección del *hardware* y *software* requeridos para el prototipo

Cada uno de los elementos escogidos eliminan problemas de compatibilidad de *hardware* y *software* y además mantienen soporte técnico por parte de las plataformas oficiales. En conjunto, se optimizó el costo de inversión para que pueda ser accesible a varios usuarios de cualquier estatus social. Por lo tanto, en los siguientes apartados se explica con detalle los parámetros de selección.

Hardware

Los componentes físicos como el *Raspberry Pi* y las computadoras que sirven como dispositivo controlador y dispositivo controlado, así como los periféricos fueron seleccionados considerando su año de elaboración para asegurar que las características de *hardware* y *software* sean recientes.

Proceso de selección del Microordenador

Para determinar el equipo más adecuado para la implementación del prototipo de *KVM* se empezó tomando en cuenta los requerimientos especificados por parte de los desarrolladores del controlador *Pi KVM*, los cuales fueron:

- 1 (GB) en *RAM* [19].
- 16 (GB) de almacenamiento [19].
- Entrada *CSI* [19].
- Puertos *USB* 3.0 [19].
- Procesador central de al menos 2 núcleos [19].
- Puerto *Ethernet* [19].
- Entrada/Salida *USB* tipo C.

En base a estos requerimientos se procedió a comparar 3 diferentes microordenadores de la marca *Raspberry* para determinar el más adecuado. En la Tabla **3.1** se detallan las consideraciones y características de los microordenadores mencionados en sus distintas versiones. Se muestra también su precio el cual fue obtenido de tiendas físicas de la ciudad de Quito.

Tabla 3.1 Comparación de equipos *Raspberry* [20], [21], [2]

Modelo	Raspberry Pi 3b	Raspberry Pi 3b+	Raspberry Pi 4b
Soporte de <i>Pi KVM</i>	No	No	Si
Puertos <i>USB</i> 2.0	Si	Si	Si

Puertos USB 3.0	No	No	Si
Puerto USB tipo C	No	No	Si
Memoria RAM 1(GB)	Si	Si	Si
Puerto ethernet	Si	Si	Si
Soporte de Raspberry	Descontinuado	Si	Si
Puerto CSI	Si	Si	Si
Costo (\$)	145.00	120.00	110.00

Una vez completo el análisis, se concluyó que el mejor microordenador para utilizar es el *Raspberry Pi 4b* puesto que como se detalla en la Tabla **3.1** este es el que mejor se ajusta con los requerimientos mínimos establecidos por el fabricante gracias a sus características mostradas en el Marco Teórico. El costo de adquisición también es muy accesible con relación a las versiones anteriores y su desenvolvimiento es adecuado.

Se consideró también la necesidad de mantener constante la temperatura del microordenador puesto que su aplicación demanda que esté encendido por grandes periodos de tiempo, lo que hizo necesario contar también con un disipador de calor para solucionar el inconveniente mencionado. El disipador de calor es de aluminio y cuenta con dos ventiladores que refrigeran el *Raspberry* como muestra la Figura **3.1**. Los ventiladores se conectan a los puertos *GPIO* número 4 que es una salida de 5 (V_{DC}) y número 6 que es la conexión a tierra del microordenador, puesto que no requieren configuraciones adicionales y proporcionan a los ventiladores el voltaje y la corriente adecuada que aseguran una temperatura estable en este y del mismo modo evita un crecimiento exagerado en las dimensiones del microordenador [22].



Figura 3.1 Disipador de calor de Raspberry [22]

Proceso de Selección de Periféricos (Teclado y Mouse)

Los periféricos seleccionados poseen un puerto de conexión de tipo *USB 2.0* puesto que son compatibles con las ranuras de los computadores utilizados y que adicionalmente son los dispositivos a los que son conectados. Además, mantiene el

requisito establecido de utilizar equipos actuales con la finalidad de evitar problemas de compatibilidad.

Proceso de selección para el adaptador *HDMI a CSI*

Se compararon las dos opciones recomendadas por la organización de *KVM* para la entrada de video, *HDMI a USB* y *HDMI a CSI*, como muestra la Tabla 3.2.

Tabla 3.2 Comparación de Adaptadores [23], [24]

Adaptador	Controlador de Pi KVM	Soporte ante fallas	Estabilidad con Raspberry	Precio (\$)
HDMI a USB	Si	No	Inestable	19.90
HDMI a CSI	Si	Si	Estable	33.29

De donde se determinó que el adaptador adecuado es el que tiene ingreso de video *HDMI* con salida *CSI* puesto que el bus de datos del adaptador ingresa directamente a la ranura *CSI* de *Raspberry* como muestra la Figura 3.2. Dicho adaptador tiene soporte ante errores por parte de *KVM*, lo que soluciona cualquier problema de compatibilidad y estabilidad en la conexión. Finalmente ofrece una recepción de video más estable frente al adaptador *HDMI a USB* con una resolución de hasta 1080 (p) a 30 (fps) del video recibido por el microordenador [12].

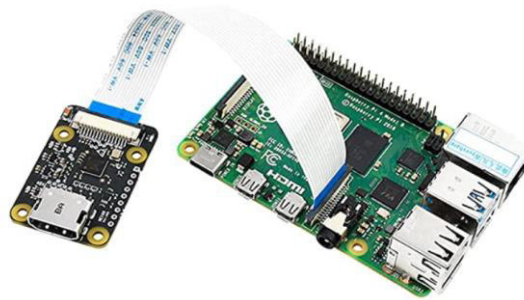


Figura 3.2 Conexión de adaptador *HDMI a CSI* [12]

Una vez escogidos los componentes físicos necesarios, el microordenador fue colocado, en conjunto con el disipador de calor, sus ventiladores y el adaptador *CSI* dentro de una carcasa protectora realizada de materiales reciclados con la finalidad de disminuir los desechos tecnológicos.

Computador de escritorio N°1 (Controlador)

El *CPU* cuenta con una salida de video *Video Graphics Array (VGA)* conectado a un monitor genérico. Sus periféricos (teclado y *mouse*) son conectados por medio de *USB 2.0* y está conectado mediante un cable *UTP* a la red local.

Computador de escritorio N°2 (Dispositivo controlado)

El *CPU* cuenta con una salida de video *HDMI*. Sus periféricos (teclado y *mouse*) para ser conectados contaron con un puerto *USB 2.0*, en el cual está conectado el cable *USB* tipo C del *Raspberry*. Del mismo modo a través de un cable *UTP* se encuentra conectado a un switch a la misma red local.

Software

Respecto al *software* utilizado, se indagó sobre controladores que permitan realizar los requerimientos propuestos. De donde se determinó que los controladores de *Pi KVM* y *Raspberry Pi Imager* son los adecuados para sustentar los requerimientos del prototipo, que fueron especificados en el punto 2 y del mismo modo, que puedan ser ejecutados sin inconvenientes por el *hardware* implementado. A continuación, se detalla el proceso de selección de cada uno de los controladores específicos mencionados.

Selección del controlador del prototipo

Una vez determinado que *Pi KVM* es la plataforma adecuada, se procedió a analizar las dos opciones de controlador, uno para *USB* (denominada “*HDMI-to-USB dongle v2*”) y otro para *CSI* (denominada *HDMI-to-CSI bridge v2*) presentados en la página oficial de *Pi KVM* [25]. En la Tabla 3.3 se muestra los parámetros de selección del controlador adecuado en base al microordenador seleccionado en el apartado de *hardware*.

Tabla 3.3 Parámetros de selección del controlador *KVM*

Controlador	<i>HDMI-to-CSI bridge v2</i>	<i>HDMI-to-USB dongle v2</i>
Soporte ante fallos por parte de <i>KVM</i>	Si	No
Estabilidad con <i>Raspberry Pi 4</i>	Estable	Inestable
Compatibilidad con <i>Raspberry Imager</i>	Si	Si
Compatibilidad con adaptador físico	Si	No

De esta comparación se concluye que el controlador adecuado es el *HDMI-to-CSI bridge v2* gracias a que es compatible con el dispositivo físico y presenta mayor estabilidad en su desempeño.

Selección del *software* de creación de imagen para *Raspberry*

Para crear la imagen para *Raspberry* del controlador proporcionado por *Pi KVM*, se escogió el programa *Raspberry Pi Imager*. El cual se encarga de convertir el controlador de *KVM* a un archivo de arranque que el microordenador pueda ejecutar desde la tarjeta *SD*.

La versión del programa utilizado es la versión 1.7.2 de *Raspberry Pi Imager* para el sistema operativo *Windows 10*, en donde se cargó la imagen del servidor *Pi KVM* con la finalidad de montar en la tarjeta *SD* el controlador “*HDMI-to-CSI bridge v2*” desde el cual se inicia el microordenador para mantenerse operativo.

3.3 Diseño del prototipo de KVM

Esquema general del prototipo

Dentro de este apartado se muestra el funcionamiento del proyecto, la forma en la que se conectan los elementos mencionados y el modo de operación del prototipo para controlar los dispositivos.

Conexión de equipos

La Figura 3.3 detalla la conexión entre los equipos seleccionados. Es así como se evidencia que el primer dispositivo que está conectado es el cable *UTP* para la conexión a la red *LAN*, a continuación, el adaptador *HDMI* a *CSI* está colocado en la ranura *CSI* para cámara del *Raspberry* y finalmente se energiza el microordenador por medio del cable *USB* tipo *C* del *Raspberry*. Este paso a su vez facilitará la emulación de teclado y *mouse* en el servidor de *Pi KVM*, pero se aclara que si el puerto al que está conectado el prototipo en el computador controlado no tiene una corriente de 3 (A) y 5 (V_{DC}) este no funcionará adecuadamente.

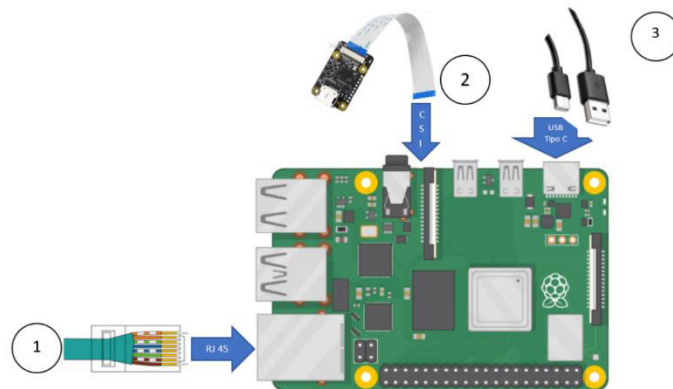


Figura 3.3 Conexión de elementos a *Raspberry Pi*

Diseño de carcasa protectora

De un *TV Box* en estado obsoleto se extrajo la carcasa protectora cuyas dimensiones son 11.7 (cm) de ancho por 11.7 (cm) de largo, 3 (cm) de alto y se reutilizó la ranura de salida del puerto *HDMI* de la carcasa. En el mismo costado, se realizó una apertura de 1.6 (cm) de ancho y de 1.45 (cm) de alto para facilitar la conexión del cable de *UTP* y el resto de los orificios de la carcasa fueron sellados para evitar el ingreso de impurezas. Así mismo se realizó una perforación en el costado derecho respecto a la ranura del *HDMI* para dejar libre el cable *USB* tipo *C*.

En la parte inferior de la carcasa (Ver Figura 3.5) se reutilizó las aperturas que daban la ventilación en el *TV Box* para asegurar la libre circulación del aire hacia los ventiladores del disipador de calor del *Raspberry*. La carcasa diseñada evita la libre manipulación de los componentes que puedan afectar su desempeño y durabilidad. La disposición de los elementos y agujeros necesarios se detalla en la Figura 3.4 y la Figura 3.6.

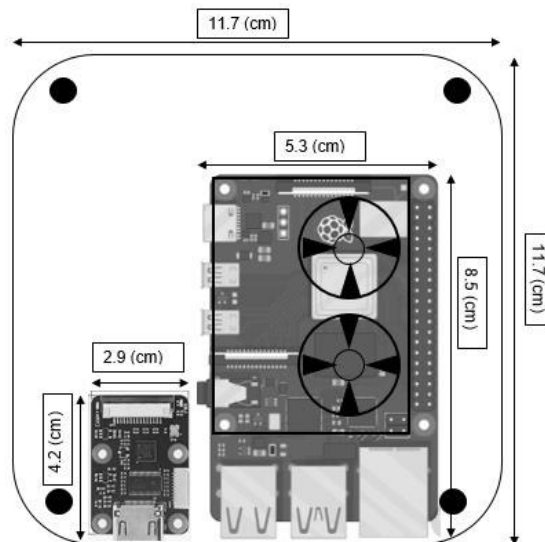


Figura 3.4 Disposición de equipos en la carcasa protectora

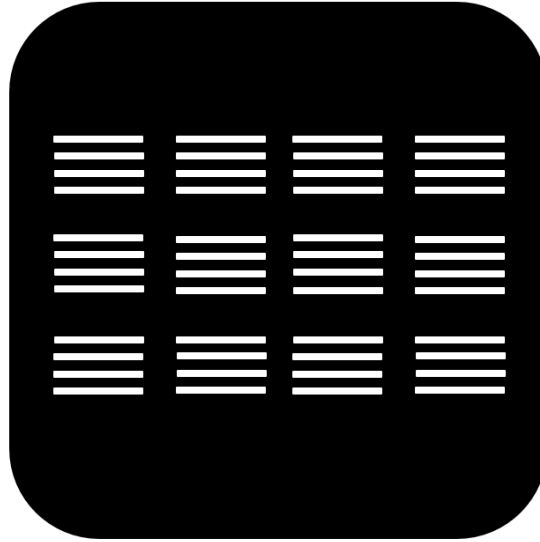


Figura 3.5 Vista Inferior / Orificios de ventilación

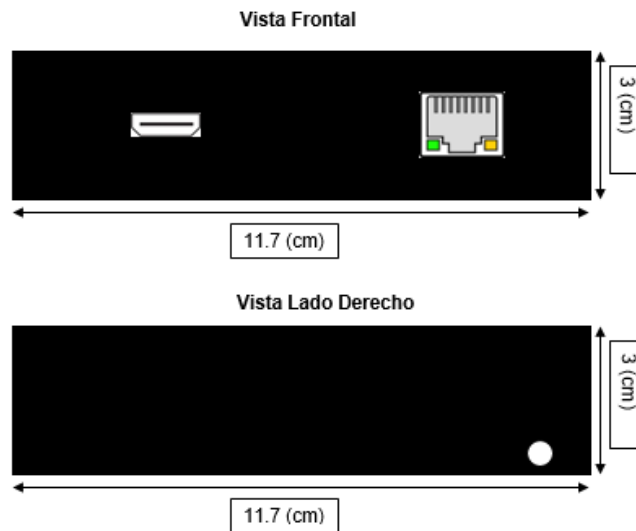


Figura 3.6 Vistas de la carcasa cerrada

Conexión de ordenadores a la placa *Raspberry*

Una vez completo el proceso de conexión al *Raspberry* se muestra la conexión del ordenador que es controlado con el microordenador. La salida de video *HDMI* del *CPU* está conectada al adaptador *HDMI* a *CSI* del *Raspberry*. El primer extremo del cable *USB* tipo *C* será conectado a un puerto *USB* 2.0 del *CPU* y el otro extremo será conectado al puerto *USB* tipo *C* del microordenador. Cómo se mencionó con anterioridad el microordenador está dentro de la red local para poder trabajar desde la *web* como muestra la Figura **3.7**.

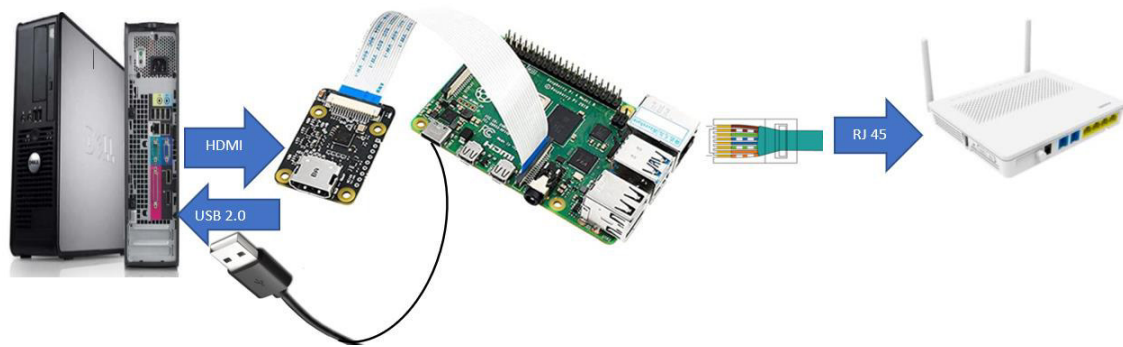


Figura 3.7 Conexión de computador controlado a Raspberry

Montaje de *Pi KVM* en la tarjeta *SD*

De la página oficial de Pi KVM se descargó la imagen que fue montada en el programa *Raspberry Pi Imager* el cual guardó esta imagen en la tarjeta *SD* como compatible para *Raspberry* como muestra la Figura 3.8 [26].

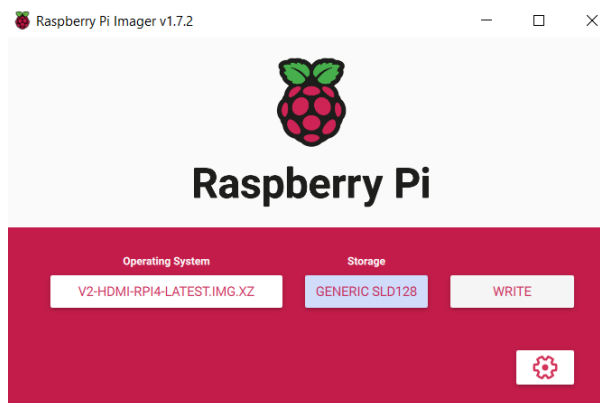


Figura 3.8 Montaje de Imagen de *KVM* en la tarjeta *microSD*

Asignación de IP al Raspberry

En el modem en donde se encuentra el servidor de *Dynamic Host Configuration Protocol* (*DHCP*) que administra la red local se reservó dinámicamente la *IP* 192.168.100.95 a la dirección *MAC* de la tarjeta de red del *Raspberry* para evitar que exista conflictos con las *IP* ya existentes de la red o cambios en las mismas, lo que puede causar problemas en el acceso al prototipo.

Instalación de *Putty*

Después de colocar la tarjeta *SD* dentro del microordenador se procedió a conectarlo en el equipo que va a controlar y este se encenderá automáticamente. Por otra parte, dentro del equipo controlador se instaló la aplicación *Putty*, mediante la cual se

estableció la conexión *SSH* al *Raspberry* con la dirección IP de este como muestra la Figura 3.9.

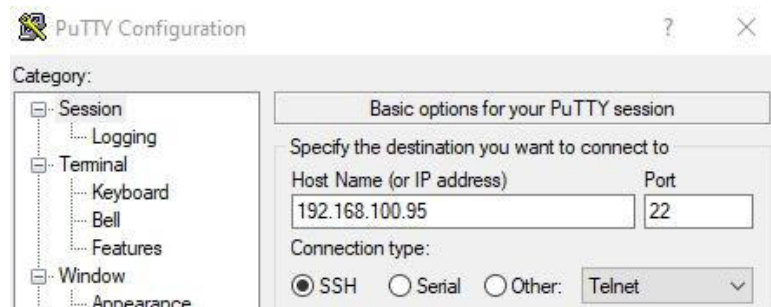


Figura 3.9 Instalación y ejecución de putty

Modificación de *script Override*

Dentro del programa *Putty* se inició sesión con el microordenador mediante el uso de la credencial por defecto *root* para usuario y contraseña como muestra la Figura 3.10.

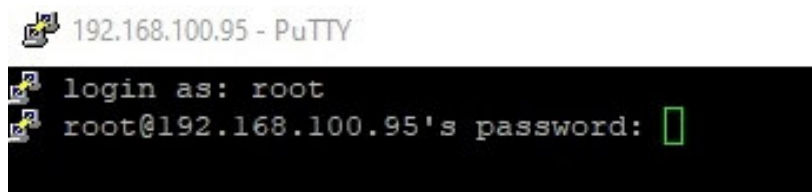


Figura 3.10 Inicio de sesión en Raspberry con Putty

Una vez iniciada sesión en el programa Putty, se procedió a modificar el *script* “*Override.yaml*” por medio del comando nano (Ver Figura 3.11) y dentro del archivo mencionado se añadieron las líneas de código mostradas en la Figura 3.12 con la finalidad de asegurar la compatibilidad con cualquier navegador en donde se utilice el *KVM* [27] [28].

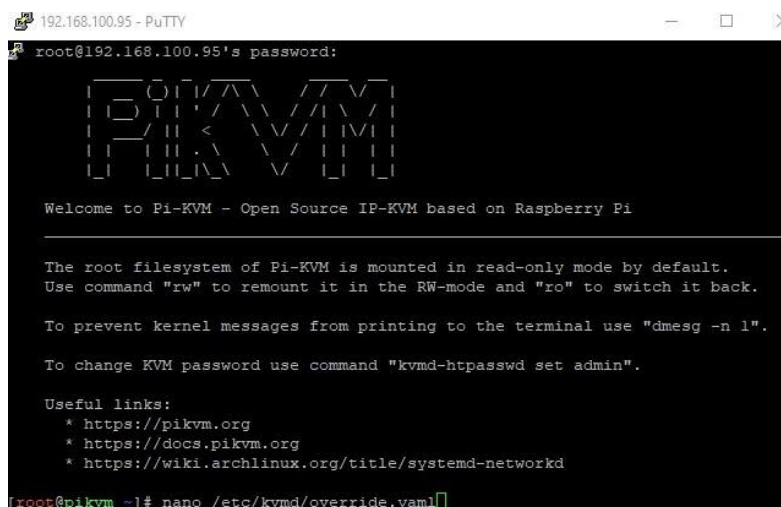


Figura 3.11 Ingreso al archivo *override.yaml*

```

GNU nano 6.2 /etc/kvmd/override.yaml
28 # drivers: ...
29 # scheme: ...
30 #
31 #####
32 # Example #
33 #####
34 #vnc:
35 # # See https://github.com/pikvm/pikvm/blob/master/pages/vnc.md
36 # keymap: /usr/share/kvmd/keymaps/ru # Set russian keymap
37 # auth:
38 # vncauth:
39 #     enabled: true # Enable auth via /etc/kvmd/vncpasswd
40
41 kvmd:
42   hid:
43     mouse:
44       absolute: false
45       horizontal wheel: false

```

Figura 3.12 Modificación de *script override.yaml*

Ejecución de *Pi KVM*

Una vez conectado el *Raspberry* a la red local y modificado el *script* fuente, desde el equipo que será el controlador se deberá ingresar a la dirección URL <https://pikvm/login/> o a la dirección IP del *Raspberry* puesto que la IP asignada al microordenador será un servidor virtual local (*localhost*) que permitirá administrar los dispositivos de manera remota. Una vez ingresado al enlace de *KVM* para inicio de sesión se ingresará las credenciales de acceso *admin* en usuario y *admin* en contraseña (Ver Figura 3.13) y finalmente seleccionar *KVM* como muestra la Figura 3.14.

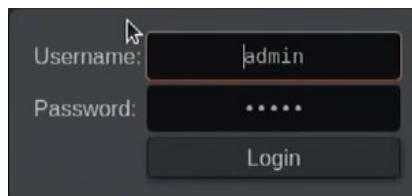


Figura 3.13 Inicio de sesión en *Pi KVM*

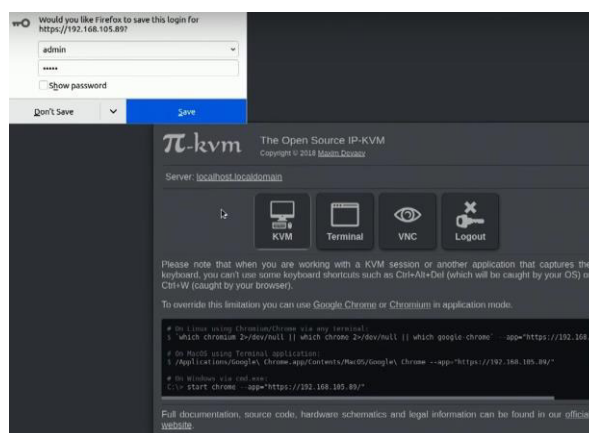


Figura 3.14 Selección de servidor *KVM*

3.4 Implementar el prototipo de *KVM*

Adaptación de la carcasa de TV Box

Se procedió a tapar los orificios de la carcasa con masilla para evitar que existan filtraciones no deseadas de impurezas y se abrió una nueva apertura para permitir el acceso al RJ 45 al *Raspberry* y al puerto *HDMI* en la cara frontal de la carcasa, y al lado derecho de ésta se realizó una nueva apertura para la salida del cable *USB* tipo C. Posterior a ello, se pintó la carcasa de color negro para que pueda estar operativa como se aprecia en la Figura 3.15.

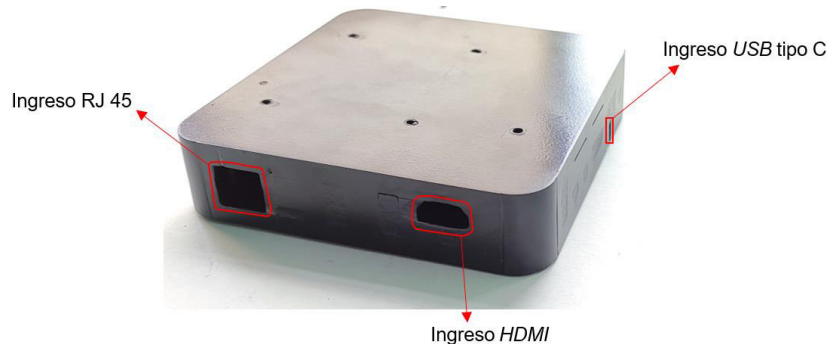


Figura 3.15 Carcasa protectora operativa

Montaje de *Raspberry Pi* sobre la carcasa

La Figura 3.16 muestra la disposición del *Raspberry Pi*, el *CSI*, y el cable *USB* tipo C en la carcasa diseñada de modo que el disipador de calor se encuentra sobre el microordenador con la finalidad de refrigerarlo y el *CSI* se encuentra en un costado para evitar que se maltrate el bus de datos y que pueda existir daños en el hardware del prototipo. Así mismo, el cable *USB* tipo C se encuentra conectado al microordenador y cada elemento fue colocado sobre alzas para que cada uno de estos no tenga contacto directo con la carcasa como demuestra la Figura 3.17. Una vez que todo fue colocado en orden se procedió a colocar la tapa de la carcasa como indica la Figura 3.18.

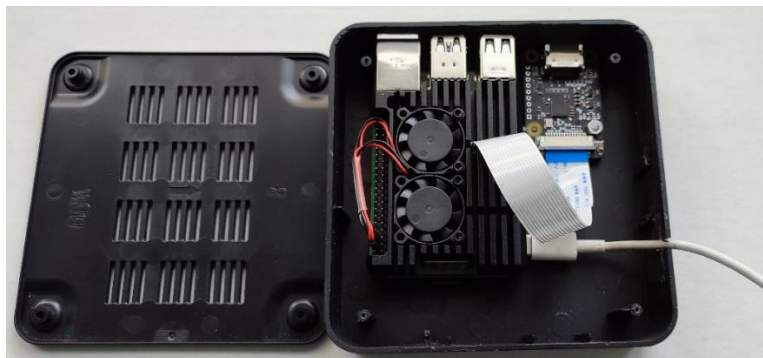


Figura 3.16 Disposición de elementos en carcasa protectora



Figura 3.17 Montaje de equipos sobre alzas

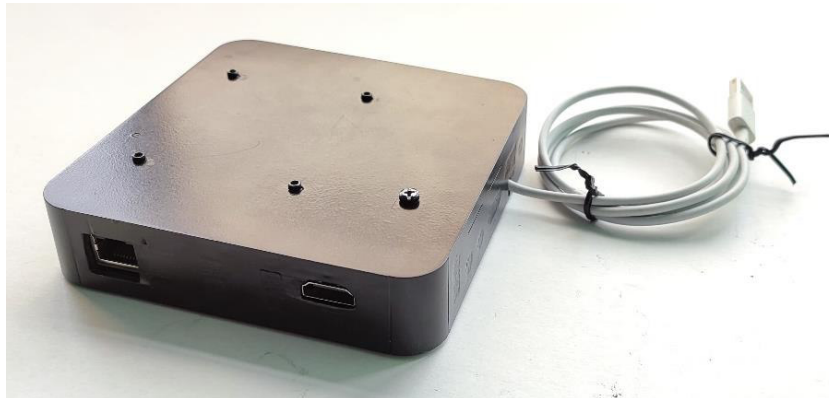


Figura 3.18 Prototipo terminado

Conexión de *Raspberry* a la red local

Una vez ensamblado el prototipo, este es conectado al *switch* al que se conecta también el computador desde el cual se controla el *KVM*. Como muestra la Figura **3.19**, la topología implementada indica una red interna y los puertos de conexión de los dispositivos, de donde se aprecia que el microordenador se conecta en el primer puerto del *switch* de 8 puertos y el computador que actúa como controlador se conectó al puerto número 2 del mismo *switch*, cada uno realizó esta conexión con cable *UTP* por medio de sus puertos RJ 45 respectivamente.

En el caso de la laptop, se la conectó al adaptador *CSI* por medio de un cable *HDMI* y el cable *USB* tipo C fue conectado tanto al puerto *USB* de la laptop como al *Raspberry*.

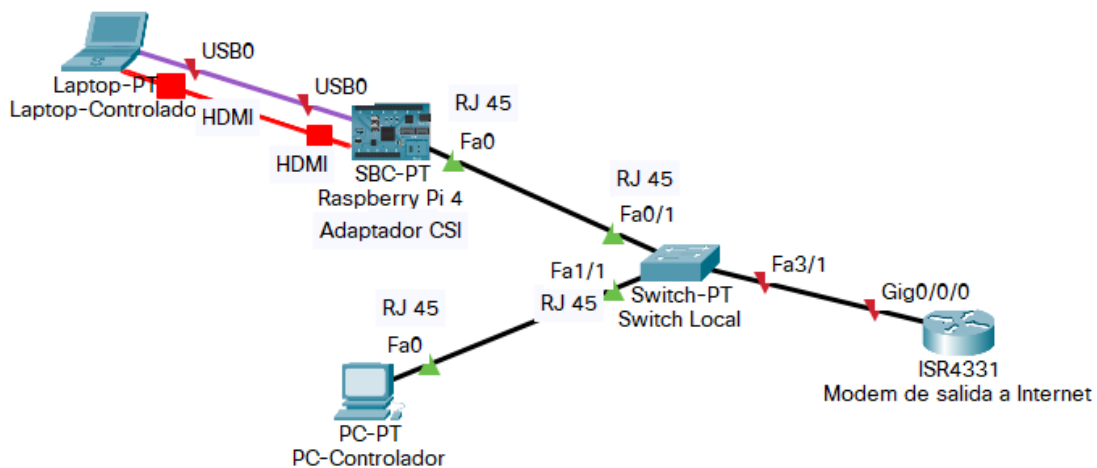


Figura 3.19 Conexión de equipos a la red local

3.5 Realizar pruebas de funcionamiento del prototipo

Acceso al servidor *PI KVM*

Desde el navegador *Brave* en el computador que realizará el control de manera remota, se ingresó al servidor mediante la dirección *IP* del prototipo y se inició sesión con las credenciales por defecto de *KVM* (*admin* en usuario y contraseña) únicamente para comprobar su funcionamiento (Ver Figura 3.20) posterior a ello se seleccionó la opción *KVM* como muestra la Figura 3.21.

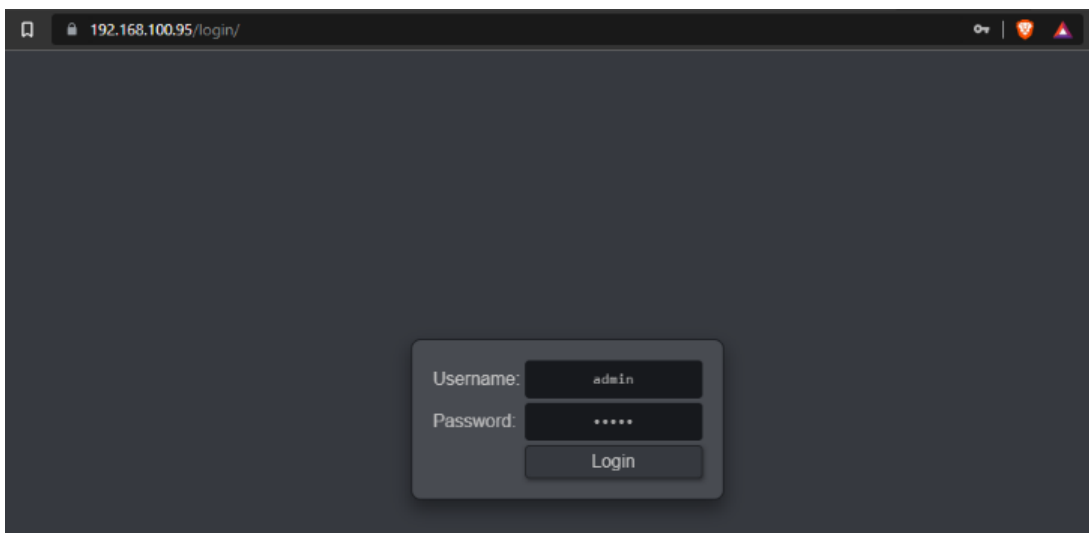


Figura 3.20 Inicio de sesión en *Pi KVM*



Figura 3.21 Ingreso al servidor

Una vez dentro, el servidor comenzará a cargar su interfaz de control como se aprecia en la Figura 3.22.



Figura 3.22 Interfaz de control de *Pi KVM*

Comprobación de operabilidad

Una vez que el sistema se encendió correctamente se procedió a revisar que todos los componentes estén operativos. Para lo cual se revisó en la esquina superior derecha que todos los componentes estén de color verde, indicador que aclara que se puede hacer uso de estos como muestra la Figura 3.23. En caso de que exista algún error en la conexión, los indicadores estarán de color naranja (Ver Figura 3.24) y de no tener ninguna respuesta de estos o si se encuentran inactivos, estarán de color gris como se aprecia en la Figura 3.25.

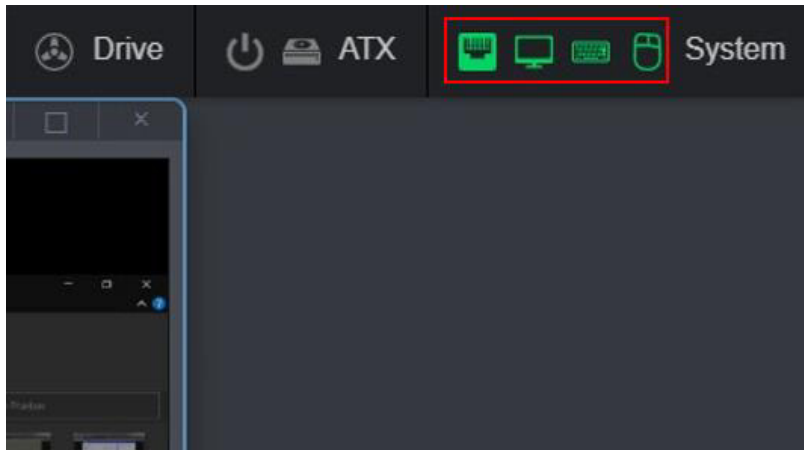


Figura 3.23 Prototipo operativo

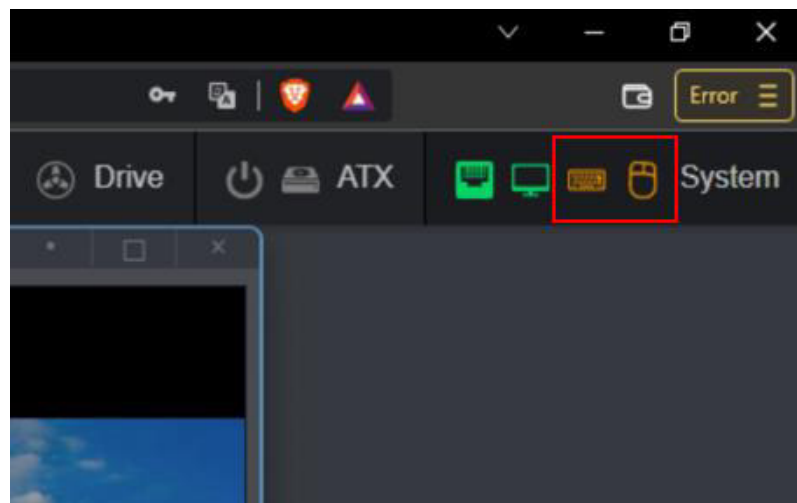


Figura 3.24 Error de conexión

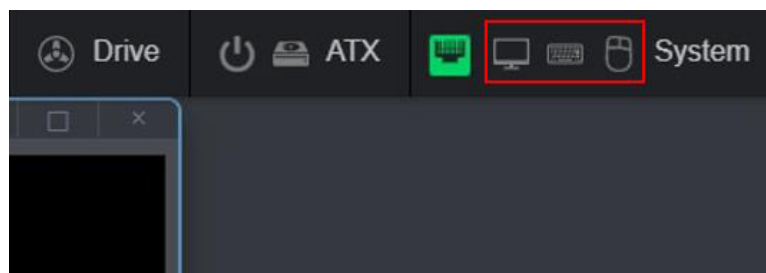


Figura 3.25 Periféricos inactivos

En caso de que todo se encuentre operativo pero la pantalla muestre el mensaje “*No Signal*” será problema de compatibilidad del adaptador de video como muestra la Figura 3.26, esto es el caso del adaptador *HDMI a USB*.

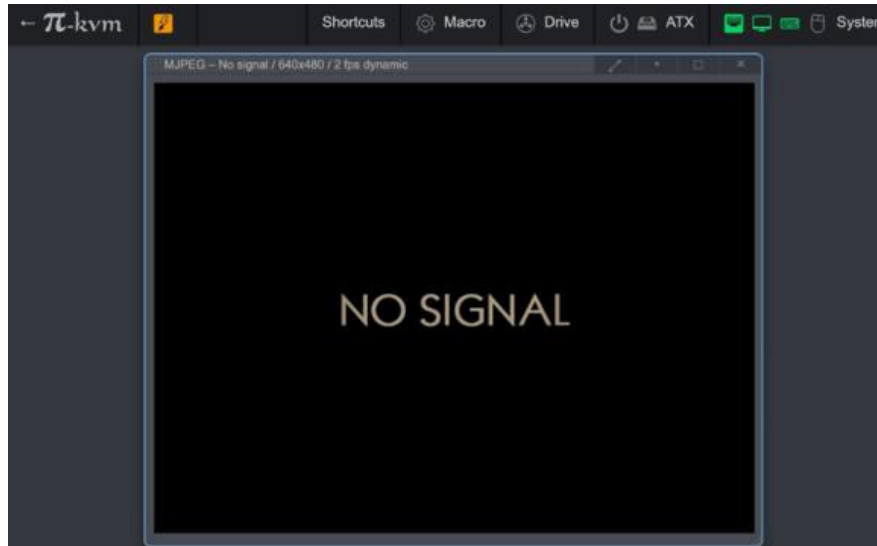


Figura 3.26 Incompatibilidad de adaptador de video

Ingreso al *BIOS* del computador controlado

Al estar lista la interfaz de control se procedió a reiniciar el computador controlado para acceder al *BIOS* de este (Ver Figura 3.27). Antes de que se encienda el computador por medio de la tecla suprimir se accedió a la interfaz mencionada como muestra la Figura 3.28 y una vez dentro se efectuó una revisión de las configuraciones.

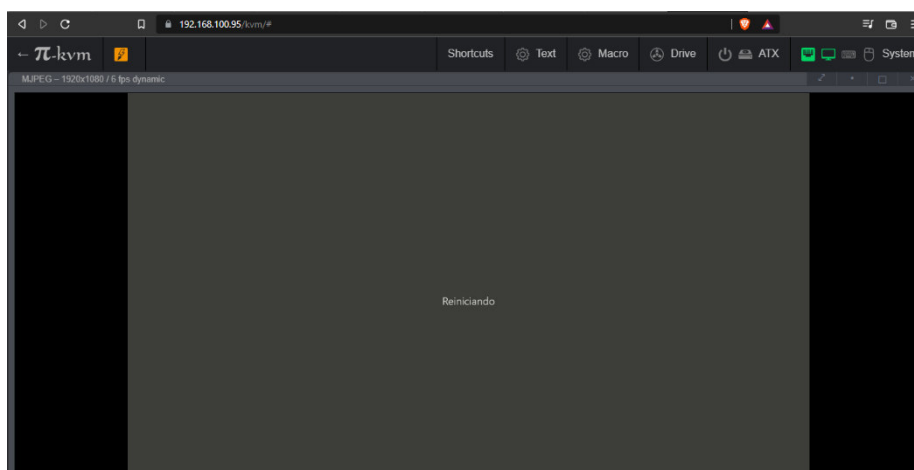


Figura 3.27 Reinicio del dispositivo controlado

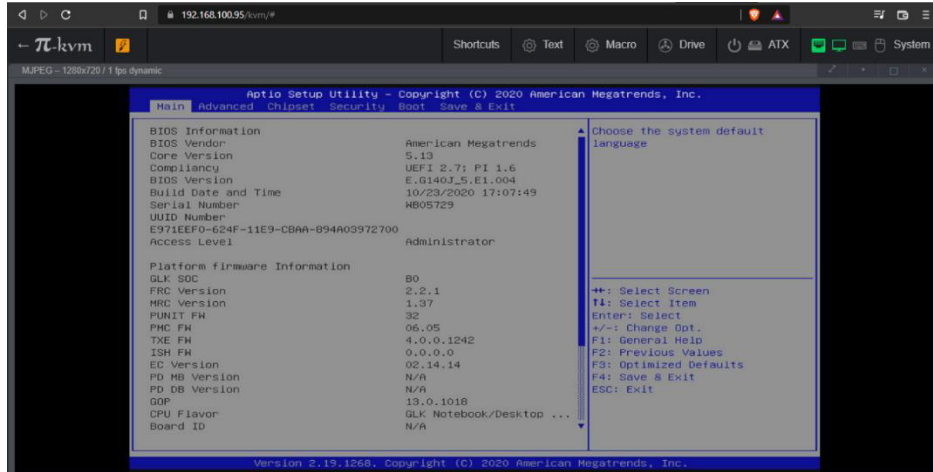


Figura 3.28 Revisión del BIOS del computador controlado

Al terminar se procede a reiniciar el equipo (Ver Figura 3.29) y una vez que se encienda estará listo para su uso como muestra la Figura 3.30.

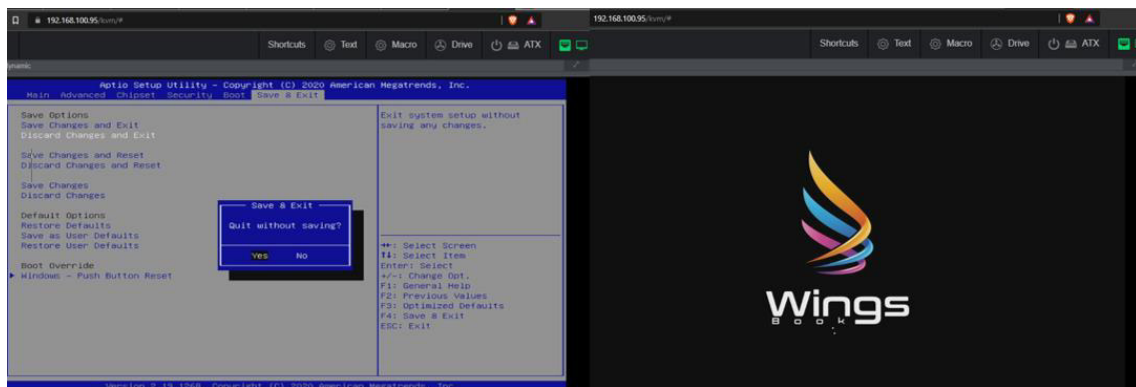


Figura 3.29 Reinicio y salida del BIOS



Figura 3.30 Computador operativo

Creación de carpeta y reubicación de esta

Dentro del disco local C se creó una carpeta mediante el controlador *KVM* en el computador que maneja el prototipo. La carpeta en cuestión llevó el nombre “Prueba 1” como muestra la Figura 3.31 y a continuación se la reubicó dentro de la carpeta “Tesis” en el mismo disco local C como se aprecia en la Figura 3.32.

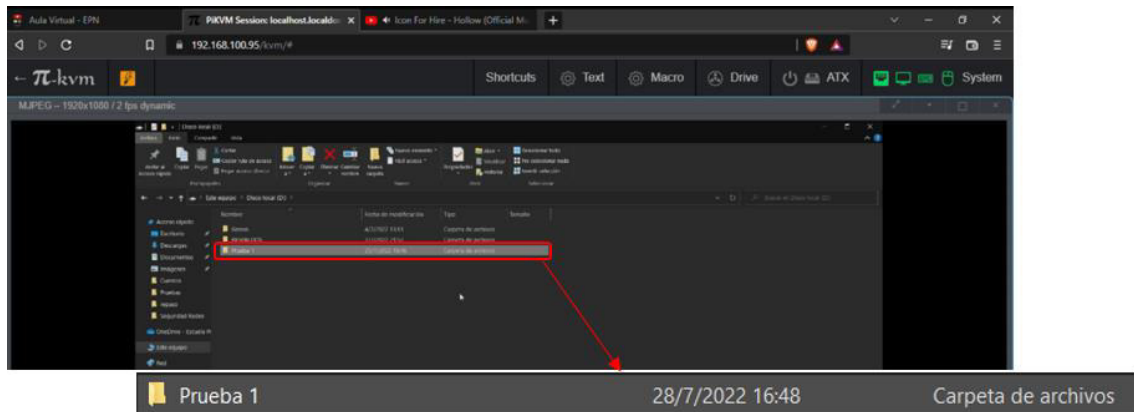


Figura 3.31 Creación de carpeta

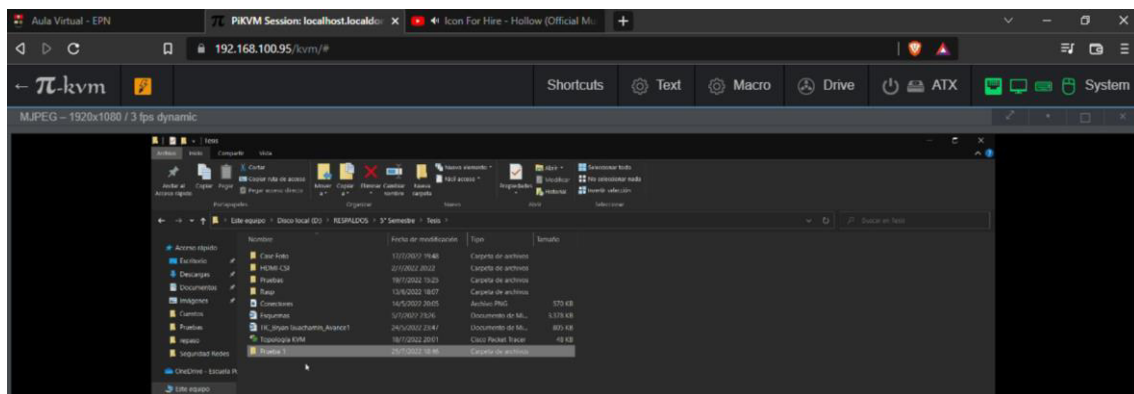


Figura 3.32 Reubicación de carpeta

Revisión de IP del equipo desde los símbolos del sistema

En el computador controlado, se buscó la aplicación Símbolo del Sistema como se aprecia en la Figura 3.33 y una vez dentro, se ejecutó el comando *ipconfig* y se pudo revisar la *IP* del equipo como muestra la Figura 3.34.

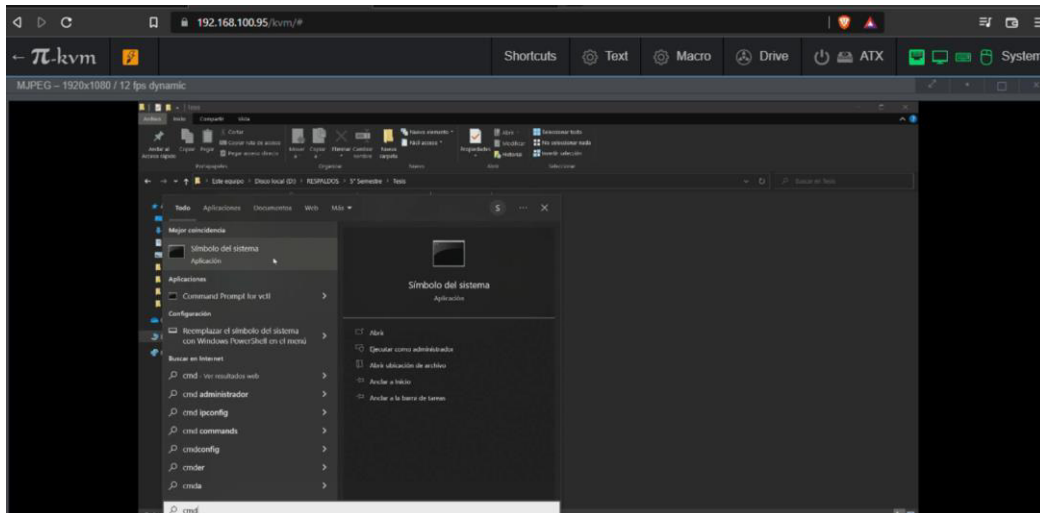


Figura 3.33 Ingreso a símbolo del sistema

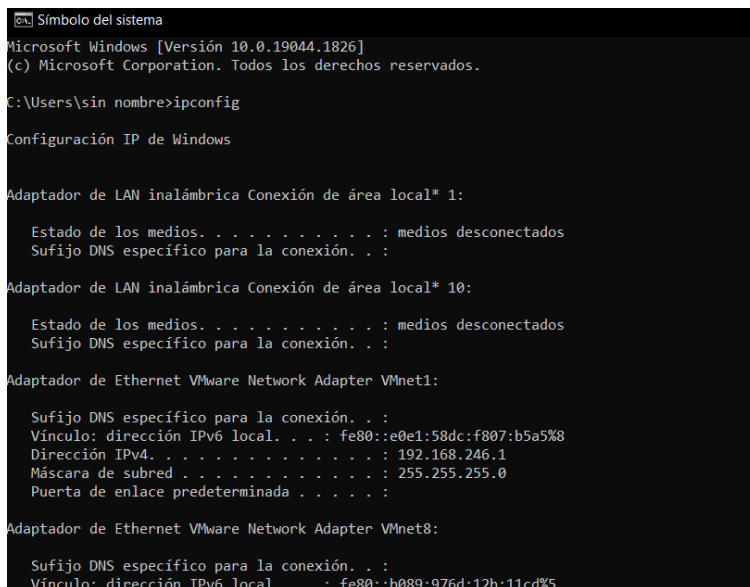


Figura 3.34 Ejecución de comando *ipconfig*

Recorte de pantalla

Para demostrar el control de teclado y *mouse* del prototipo se ingresó en el apartado “*system*” dentro de la interfaz de *KVM* y se seleccionó la opción “*show keyboard*” para obtener un teclado virtual que ejecute comandos dentro del computador controlado como muestra la Figura 3.35. Posterior a ello se ejecutó con las teclas el comando “*Windows + Shift + S*” para poder realizar el recorte como se aprecia en la Figura 3.36. Finalmente, se revisó en la sección de notificaciones la captura realizada como se ve en la Figura 3.37.

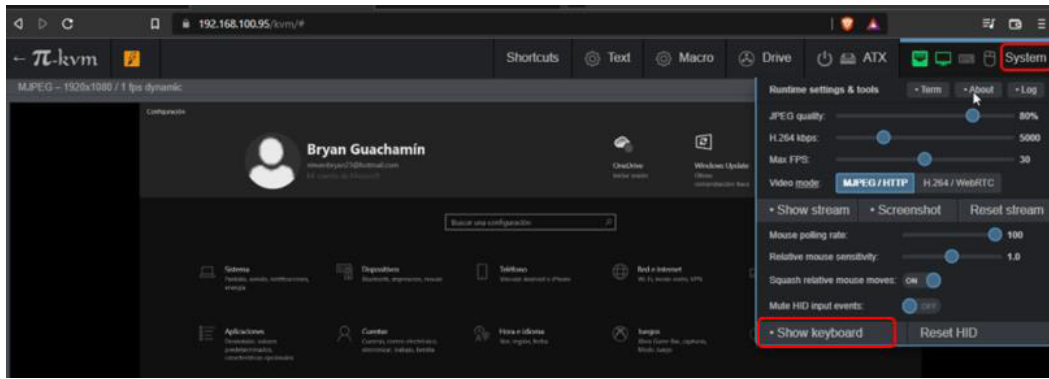


Figura 3.35 Ingreso al teclado de KVM

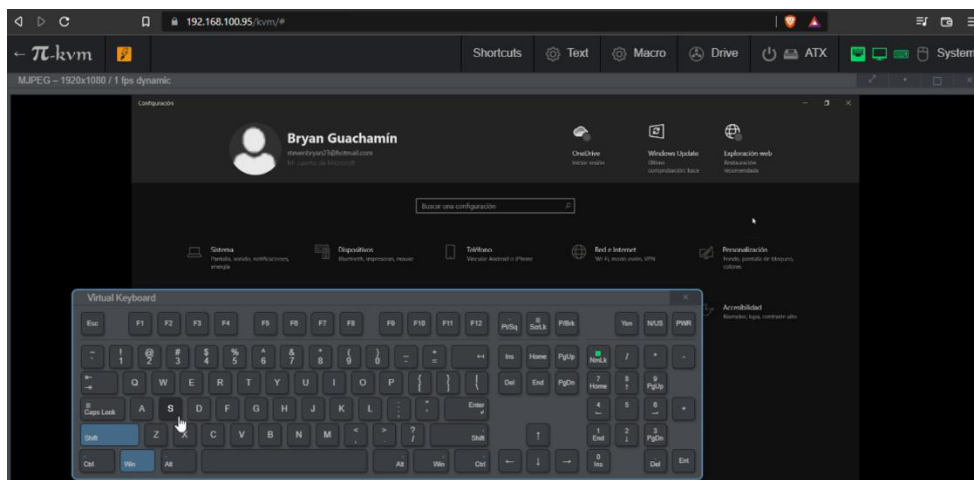


Figura 3.36 Ejecución del comando *Windows + Shift + S*

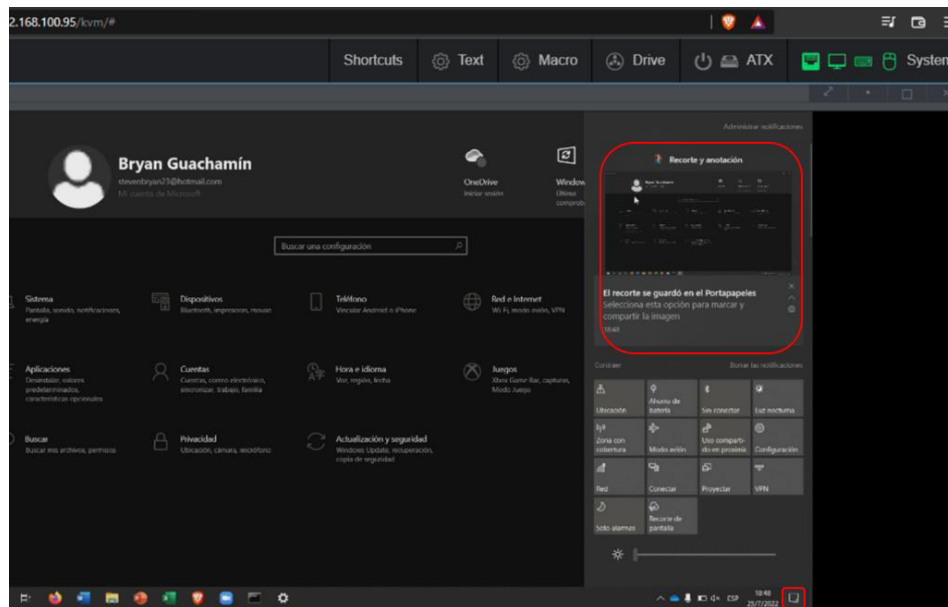


Figura 3.37 Comprobación de lo realizado

Reproducción de video

Dentro del navegador *Brave* en la máquina controlada se accedió al reproductor *YouTube* como se ve en la Figura 3.38. A continuación, se reprodujo un video para evidenciar la capacidad del prototipo de visualizar todo lo que se encuentre en la máquina que está siendo controlada como muestra la Figura 3.39.

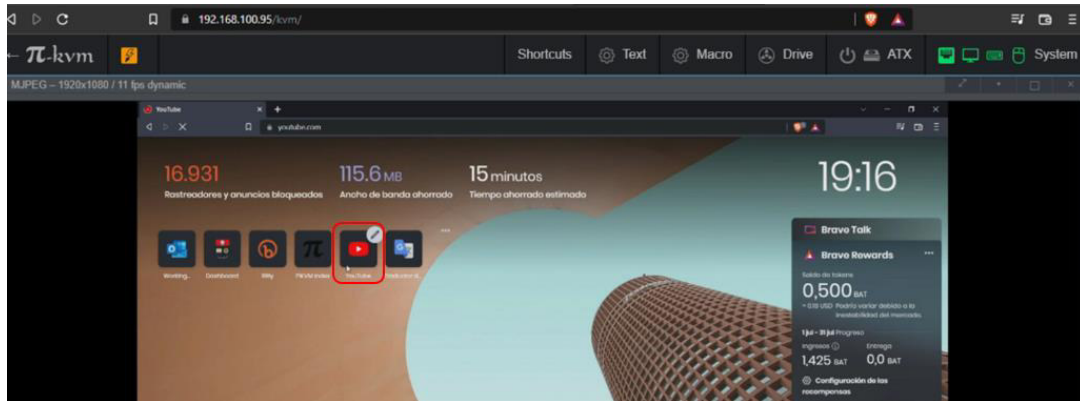


Figura 3.38 Ingreso a *YouTube*

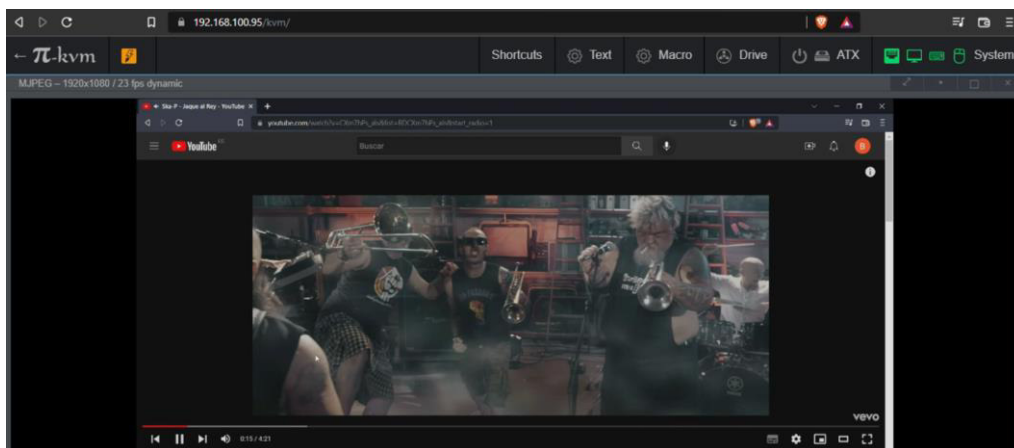


Figura 3.39 Reproducción de video

Funcionamiento y mantenimiento del prototipo

El video explicativo en donde se demuestra el funcionamiento del prototipo se encuentra en el Código QR del video de las pruebas de funcionamiento realizadas y del mismo modo en video de mantenimiento del prototipo en el Código QR del video de mantenimiento del prototipo.

Costo del proyecto

A continuación, se detalla en la Tabla 3.4 el listado de los componentes necesarios para la implementación de prototipo, así como sus respectivas cantidades (dadas en unidades u horas según corresponda) en conjunto con los costos totales y unitarios de

estas. Hay que considerar que cada uno de los valores son referenciales de plataformas digitales (Mercado Libre y Amazon) y tiendas físicas de la ciudad de Quito.

Tabla 3.4 Costo total del proyecto

Material	Cantidad (u)	Costo unitario (\$)	Costo total (\$)
<i>Raspberry Pi 4</i>	1	\$110,00	\$110,00
<i>Adaptador CSI</i>	1	\$111,00	\$111,00
<i>Cable HDMI</i>	1	\$4,50	\$4,50
<i>Cable UTP RJ 45 de 1.5 (m)</i>	2	\$1,25	\$2,50
<i>Computador</i>	2	\$400,00	\$800,00
<i>Switch 8 Puertos</i>	1	\$15,00	\$15,00
<i>Disipador de calor</i>	1	\$26,90	\$26,90
<i>Cable USB tipo C</i>	1	\$2,50	\$2,50
<i>Carcasa de TV Box</i>	1	\$5,00	\$5,00
<i>Mano de obra (h)</i>	28	\$10,00	\$280,00
Total			\$1.357,40

4 CONCLUSIONES

- Una vez terminado el análisis realizado para la selección de *hardware*, se determinó que el *Raspberry Pi 4B* gracias a que posee 1 (GB) de *RAM* y una ranura *CSI* compatible con el adaptador *HDMI* a *CSI* aporta estabilidad en el procesamiento de datos, así como reducción en las latencias de la transmisión de datos del control de los dispositivos, además en el mercado nacional su precio y disponibilidad es accesible, posicionándolo así como el microordenador más adecuado para el desarrollo del prototipo.
- Como el microordenador está sujeto a largos periodos de tiempo de uso fue necesario mantener baja la temperatura de este, por lo que se concluyó que un disipador de calor con ventiladores de velocidad constante fue necesario para refrigerar el dispositivo y de esta manera evitar daños en el prototipo.
- El adaptador *HDMI* a *USB* al ser conectado al puerto *USB* del *Raspberry* presenta un problema de compatibilidad evidenciado en la interfaz *web* al mostrar el mensaje “*No Signal*” a pesar de ser el modelo adecuado y ser reconocido correctamente por el microordenador, este adaptador no pudo transmitir video, mientras que el adaptador *HDMI* a *CSI* ofrece una transmisión

de video clara y sin interferencias, de donde se concluye que el adaptador *HDMI* a *CSI* es el más adecuado para el prototipo gracias a su compatibilidad con el microordenador y la calidad del video generada.

- El prototipo se conecta directamente al computador que se va a controlar mediante un cable *USB* tipo C que permite la transmisión de datos para emular en el dispositivo controlado el teclado y *mouse* físicos que permitirán el acceso al *BIOS* de este y también este permite el paso de energía para que el prototipo se encienda automáticamente, es así como se concluye que el prototipo solamente puede administrar y controlar un computador a la vez, lo cual al momento es una limitante del dispositivo.
- Se concluye que el prototipo no puede reproducir audio en el computador que actúa como controlador debido a que el adaptador *HDMI* a *CSI* es conectado en la ranura *CSI* del microordenador, la cual únicamente permite el ingreso de video a pesar de que la conexión *HDMI* es capaz de transmitir audio y video al mismo tiempo, por lo que en caso de necesitar el audio del computador se deberá utilizar un adaptador que cuente con una salida de audio capaz de conectarse a los puertos *GPIO* del microordenador.
- El software de *Pi – KVM* fue seleccionado debido a que es autónomo y no necesita ser levantado sobre un sistema operativo como *Linux* o *Windows*, sino que con el simple hecho de montarlo como imagen para *Raspberry* en una tarjeta *SD* estuvo listo para funcionar. Finalmente, la versión del controlador para *CSI* se seleccionó en base al *hardware* determinado por la compatibilidad entre los equipos como se explicó con anterioridad.
- Al ensamblar el prototipo, se pudo evidenciar que el bus de datos del adaptador *HDMI* a *CSI* no estaba estático en una posición y que el microordenador al aire libre corría el riesgo de ser manipulado inadecuadamente, el disipador de calor también estaba en peligro de que impurezas ingresen en los ventiladores, razón por la cual se diseñó la carcasa protectora con la finalidad de proteger al prototipo por completo de la libre manipulación, impurezas y movimientos bruscos, en conclusión, se aisló al prototipo de su entorno dejando únicamente acceso a las ranuras necesarias como son las del RJ 45, del cable *USB* y del ingreso *HDMI*.
- Si el disipador de calor no tiene ventilación no puede expulsar el aire caliente y refrigerar el microordenador, por este motivo la carcasa protectora cuenta con

orificios en la tapa protectora con la finalidad de permitir un flujo de aire adecuado para mantener la temperatura de operación adecuada en el *Raspberry*.

- La plataforma de *Pi – KVM* ofrece una interfaz *web* para hacer uso del microordenador que está conectado al computador que está siendo controlado, por lo que se concluye que el prototipo puede ser utilizado a través de cualquier navegador siempre y cuando el computador desde el cual se ingrese al *KVM* pertenezca a la misma red *LAN* que el microordenador y este último esté conectado y listo para su uso.
- Gracias a que se puede acceder a esta página *web* para administrar los dispositivos que estén conectados al prototipo desde cualquier navegador, no se requiere la presencia física de una persona para realizar cambios a nivel del *BIOS* o del sistema operativo del computador que será controlado.
- El acceso al *BIOS* de un computador es un factor diferenciador del prototipo planteado respecto a programas controladores remotos similares que pierden la administración de un equipo al ser reiniciados, por lo que se concluyó que el presente proyecto solucionó este inconveniente mediante la emulación de los periféricos mencionada con anterioridad, sin embargo, se presenta una limitante puesto que si se accede desde un dispositivo móvil (*smartphone* o *Tablet*) no será posible realizar modificaciones debido a que estos dispositivos no cuentan con un ingreso de teclado o mouse independiente sino que son manejados a través de la pantalla táctil.
- Una vez realizada la implementación se concluyó que el servidor debe contar con su propia *IP* para que pueda operar correctamente, es decir, el microordenador no puede estar sujeto a una asignación de *IP* dinámica, por lo que la *IP* asignada al *Raspberry* fue reservada dinámicamente en el servidor *DHCP* de la red para evitar que esta dirección vuelva a ser utilizada en otro dispositivo de la red y ocasione conflicto en los mismos.
- Para algunos navegadores el prototipo presentaba retardos o confusiones entre el manejo del teclado y *mouse* físico de la computadora que actúa como controlador y el entorno virtual de la página *web* de *KVM*, esto fue el motivo por el que se realizaron cambios en el script *Override* con la finalidad de brindar mayor compatibilidad y una mejor captura de los emuladores de teclado y *mouse* del servidor de *KVM* por lo que se concluye que el cambio realizado mejoró el rendimiento del prototipo en cualquier navegador.

5 RECOMENDACIONES

- Se recomienda como trabajo a futuro el desarrollo de un proyecto que implemente un KVM que permita controlar a otros dispositivos desde un dispositivo móvil puesto que sería una mejora significativa al permitir usar el KVM desde distintos dispositivos.
- Para implementaciones de periodos ininterrumpidos de tiempo (24 horas al día los 7 días a la semana) es recomendable colocar ventiladores de velocidad variable para que en caso de aumento de temperatura se refrigere el microordenador de mejor manera.
- En el caso de que una persona necesite observar los cambios que se están realizando en el dispositivo controlado sin realizar ninguna acción, se recomienda acceder desde un dispositivo móvil (*smartphone* o *tablet*) como observador.
- Como futuro proyecto es recomendable realizar una investigación sobre el posible uso de un adaptador que cuente con salida de audio compatible con los puertos *GPIO* del *Raspberry* para tener un manejo remoto más inmersivo desde el computador que está actuando como controlador.
- Se recomienda como futuro proyecto restringir el acceso al *KVM* desde un único dispositivo controlador a la vez puesto que al momento se puede ingresar desde varios equipos controladores simultáneamente, lo que puede causar conflicto en la manipulación del equipo controlado.
- Es recomendable que para futuras versiones del controlador *KVM* se desarrolle una política de seguridad que permita usar las credenciales de acceso por defecto únicamente para el primer inicio de sesión y una vez que se esté dentro se solicite el cambio inmediato de las credenciales de acceso, esto con la finalidad de evitar realizar el cambio manualmente e incluir mayor seguridad en el inicio de sesión del prototipo.
- En caso de que se necesite energizar al microordenador de manera externa debido a que no se abastece con la energía proporcionada por el puerto *USB* del dispositivo controlado, se puede utilizar un *USB* tipo *C splitter* para tener por separado el cable de energía con el voltaje de 5 (V_{DC}) y la corriente de 3 (A) para su correcto desempeño y el cable de datos para emulación de teclado y *mouse*.

- Si no se posee la carcasa de un TV Box, es recomendable realizar una carcasa protectora manualmente en madera o imprimirla en 3D en plástico para evitar que el prototipo sea maltratado o manipulado de forma inadecuada considerando el lugar en donde se debe dejar orificios de conexión y para que el ventilador no se obstruya con la finalidad de mantener refrigerado el microordenador.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] C. Helena, «CAF Banco de Desarrollo de América latina,» 24 Mayo 2020. [En línea]. Available: <https://bit.ly/3PmOOeC>. [Último acceso: 15 Mayo 2022].
- [2] J. Pastor, «Xataka,» 25 Julio 2019. [En línea]. Available: <https://www.xataka.com/ordenadores/raspberry-pi-4-analisis-caracteristicas-precio-especificaciones>. [Último acceso: 30 Mayo 2022].
- [3] A. Delgado, «GEEKNETIC,» 21 Noviembre 2020. [En línea]. Available: <https://www.geeknetic.es/Raspberry-Pi/que-es-y-para-que-sirve>. [Último acceso: 23 Mayo 2022].
- [4] S. d. Diccionario, «significadodeldiccionario,» [En línea]. Available: <https://www.significado-diccionario.com/GPIO>. [Último acceso: 23 Mayo 2022].
- [5] Abbreviation Finder, «Abbreviation Finder,» [En línea]. Available: <https://www.abbreviationfinder.org/es/acronyms/csi.html>. [Último acceso: 23 Mayo 2022].
- [6] name, «Todo Siglas,» 23 Mayo 2020. [En línea]. Available: <https://www.todosiglas.com/siglas-de/delivered-source-instructions/>. [Último acceso: 23 Mayo 2022].
- [7] Tech Lib, «Techlib,» [En línea]. Available: <https://techlib.net/definicion/sd.html>. [Último acceso: 23 Mayo 2022].
- [8] Red Hat, «¿Qué son las KVM?,» Red Hat, 21 Marzo 2018. [En línea]. Available: <https://www.redhat.com/es/topics/virtualization/what-is-KVM>. [Último acceso: 26 Julio 2022].

- [9] T. Kurek, «KVM hypervisor: a beginners' guide,» Ubuntu Blog, 8 Septiembre 2021. [En línea]. Available: <https://ubuntu.com/blog/kvm-hypervisor>. [Último acceso: 26 Julio 2022].
- [10] N. Tech, «Youtube,» 20 Noviembre 2020. [En línea]. Available: <https://www.youtube.com/watch?v=pIP9Y1likRg>. [Último acceso: 15 Mayo 2022].
- [11] Pi KVM, «Pi-KVM,» [En línea]. Available: <https://pikvm.org/>. [Último acceso: 15 Mayo 2022].
- [12] BFab, «amazon,» 5 Enero 2022. [En línea]. Available: <https://www.amazon.com/-/es/Adaptador-CSI-compatible-Raspberry-soporte/dp/B095K343L2>. [Último acceso: 30 Mayo 2022].
- [13] P. Santamaría, «Xataka,» 4 Octubre 2018. [En línea]. Available: <https://bit.ly/3zHCHUe>. [Último acceso: 19 Junio 2022].
- [14] Raspberry Pi Foundation, «Raspberry Pi OS,» [En línea]. Available: <https://www.raspberrypi.com/software/>. [Último acceso: 23 Mayo 2022].
- [15] H. México, «PuTTY en programación, aprende qué es y cómo utilizarlo,» Hostgator blog, 13 Septiembre 2021. [En línea]. Available: <https://www.hostgator.mx/blog/putty-en-programacion-aprende-que-es/>. [Último acceso: 6 Julio 2022].
- [16] Red Hat, «¿Qué es YAML?,» Red Hat, 18 Junio 2021. [En línea]. Available: <https://www.redhat.com/es/topics/automation/what-is-yaml>. [Último acceso: 18 Julio 2022].
- [17] H. Marques Fernandes, «Archivo YAML – ¿Qué es y para qué sirve?,» marquesfernandes.com, 01 Abril 2020. [En línea]. Available: <https://marquesfernandes.com/es/tecnologia-es/file-yaml-what-and-and-for-that-serves/>. [Último acceso: 18 Julio 2022].
- [18] b. ghimire, «Introducción a YAML para principiantes,» GEEKFLARE, 11 Enero 2021. [En línea]. Available: <https://geekflare.com/es/yaml-introduction/>. [Último acceso: 18 Julio 2022].

- [19] mdevaev, «Github,» 24 Mayo 2022. [En línea]. Available: <https://github.com/pikvm/pikvm#diy-getting-started>. [Último acceso: 30 Mayo 2022].
- [20] NayLamp, «NayLamp Mechatronics,» 4 Agosto 2021. [En línea]. Available: <https://naylampmechatronics.com/raspberry-pi/255-raspberry-pi-3-b.html>. [Último acceso: 30 Mayo 2022].
- [21] F. Guillaume, «LD LC High-Tch Experience,» 12 Junio 2019. [En línea]. Available: <https://www.ldlc.com/es-es/ficha/PB00246555.html>. [Último acceso: 30 Mayo 2022].
- [22] Dynamo electronics , «DISIPADOR CALOR RASPBERRY PI 4 CON 2 VENTILADORES,» Dynamo Electronics, 5 Marzo 2022. [En línea]. Available: <https://dynamoelectronics.com/tienda/disipador-raspberry-pi-4-con-2-ventiladores/>. [Último acceso: 2 Julio 2022].
- [23] B&H, «ANDYCINE U2H41,» B&H Photo-Video-Audio, 2020. [En línea]. Available: https://www.bhphotovideo.com/c/product/1590561-REG/andycine_u2h41_hdmi_to_usb_2_0.html. [Último acceso: 16 Junio 2022].
- [24] Waveshare, «Computadoras de Placa Reducida,» Amazon, 3 Enero 2022. [En línea]. Available: <https://www.amazon.com/Adapter-Support-Raspberry-Standard-Compatible/dp/B093LVV3L1>. [Último acceso: 19 Junio 2022].
- [25] Pi-KVM, «pi-kvm.org,» Pi-KVM, 20 Noviembre 2020. [En línea]. Available: <https://pikvm.org/download/>. [Último acceso: 15 Julio 2022].
- [26] Pi KVM, «Here you can find ready-made PiKVM images for Raspberry Pi,» Pi KVM, 20 Marzo 2022. [En línea]. Available: <https://pikvm.org/download/>. [Último acceso: 2 Julio 2022].
- [27] Pi KVM, «Github,» 10 Mayo 2022. [En línea]. Available: <https://github.com/pikvm/pikvm#diy-getting-started>. [Último acceso: 15 Mayo 2022].
- [28] Pi-KVM, «Pi-KVM,» 2020. [En línea]. Available: <https://kaydron1000.github.io/pikvm/docs/pikvm-software/building-os>. [Último acceso: 15 Mayo 2022].

[29] S. Sistemas, solvetic.com, 16 Noviembre 2018. [En línea]. Available: <https://www.solvetic.com/tutoriales/article/6414-como-instalar-y-configurar-firewalld-en-centos-y-ubuntu/>. [Último acceso: 30 Noviembre 2021].

7 ANEXOS

La lista de los **Anexos** se muestra a continuación:

ANEXO I. Certificado de originalidad

ANEXO II. Enlaces a videos explicativos

ANEXO I: CERTIFICADO DE ORIGINALIDAD

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. XX de XXXX de 2022

De mi consideración:

Yo, NOMBRE, en calidad de Director del Trabajo de Integración Curricular titulado NOMBRE DEL TRABAJO DE INTEGRACIÓN CURRICULAR elaborado por el estudiante NOMBRE ESTUDIANTE de la carrera en TECNOLOGÍA SUPERIOR EN REDES Y TELECOMUNICACIONES, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del xxx%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el link del informe generado por la herramienta Turnitin.

LINK

Atentamente,

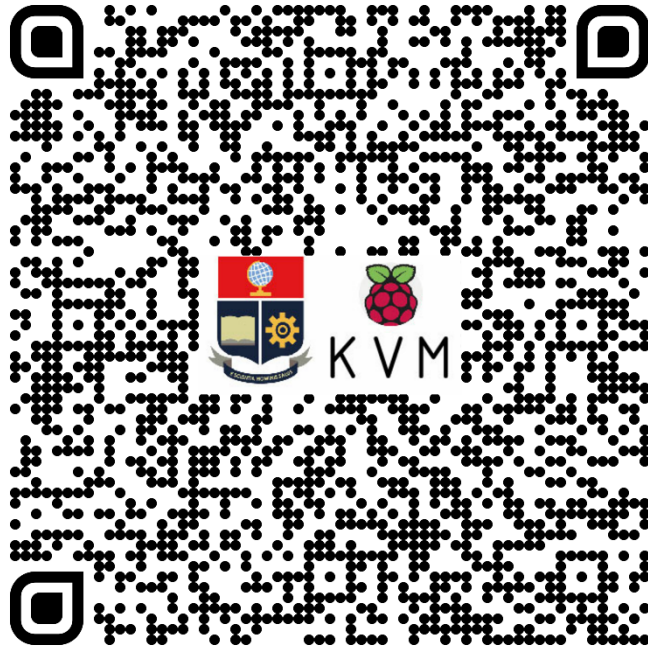
NOMBRE

Docente

Escuela de Formación de Tecnólogos

ANEXO II: Enlaces a videos explicativos

Código QR del video de las pruebas de funcionamiento realizadas



Código QR del video de mantenimiento del prototipo

