

**UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN**  
**FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA**  
**INGENIERÍA MECATRÓNICA**

---

# **Diseño de Instrumentación Virtual para Teleoperación de Robot.**

---

Estancia de Investigación Semestral

Presenta: Angel David Sasso Villa

Asesor: M.I. Carlos Alberto Vaquera Hernández.

*Febrero-Junio 2021*

## **Diseño de Instrumentación Virtual para Teleoperación de Robot.**

## **Resumen**

En el presente trabajo se diseñará el prototipo de un instrumento virtual con el uso de un microcontrolador que por medio de software tendrá la capacidad de lectura, escritura, recepción y transmisión de señales. Se diseñará una interfaz de escritorio que muestre el estado de conexión de los componentes del robot, las lecturas de las señales medidas y permita asignar valores a los servomotores del robot para realizar una teleoperación del mismo.

## Contenido

I.	Introducción.....	6
II.	Objetivos del proyecto .....	7
	Objetivo General.....	7
	Objetivo Particular .....	7
III.	Marco Teórico .....	8
	Instrumento.....	8
	Instrumento Virtual.....	9
	Hardware.....	11
	Software .....	11
	Diseño de Instrumentos Virtuales .....	12
	Componentes clave del Instrumento Virtual desarrollado. ....	12
	ESP32 de espressif.....	12
	Raspberry Pi 3 Model B+ .....	13
	MQTT: El protocolo de comunicación estándar para el internet de las cosas (IoT) .....	13
	Node-RED.....	14
IV.	Metodología.....	16
	Análisis y Descripción del Sistema .....	16
	Características limitantes .....	18
	Arquitectura General .....	19
	Arquitectura del Software .....	21
	Software Embebido.....	21
	Software de Seguridad .....	22
	Software de Control .....	22
	Arquitectura del Hardware.....	23
	Conexiones .....	23
	Servicio de escritorio remoto .....	24
V.	Análisis de resultados.....	25

14. Conclusiones .....	37
15. Referencias Bibliográficas.....	38

## Índice de Figuras

Figura 1 Diferentes tipos de instrumentos.....	9
Figura 2. Características de un Instrumento Tradicional frente a uno Virtual. ....	10
Figura 3 estructura típica de un programa de instrumentación virtual.....	11
Figura 4. ESP32-C3-DevKitC-02.....	13
Figura 5. Raspberry Pi 3 Model B+ .....	13
Figura 6. Arquitectura MQTT .....	14
Figura 7. Ambiente de desarrollo de Node-RED.....	15
Figura 8. Fotografía lateral del robot a teleoperar. ....	16
Figura 9. Fotografía del robot a teleoperar, servomotores visibles.....	16
Figura 10. Diagrama adquisición de datos y escritura de señales .....	17
Figura 11. Componentes de instrumentación virtual, funcionalidad y elemento seleccionado	18
Figura 12. Diagrama de Estructura General .....	20
Figura 13. Diagrama Arquitectura con Detalles.....	21
Figura 14. Estructura de software para ESP32 .....	21
Figura 15. Archivos en la raspberrypi.....	22
Figura 16. Captura de pantalla del programa de control. Flujo principal .....	23
Figura 17. Estructura de payload .....	23
Figura 18. Diagrama de bloques de conexiones y funciones.....	23
Figura 19. Estructura de conexión desde servicio de escritorio remoto.....	24
Figura 20. Iniciando servicio de escritorio remoto. ....	25
Figura 21. Inicio de sesión en la cuenta del servicio de escritorio remoto. ....	25
Figura 22. Comando ejecutado desde la terminal para iniciar el proceso de Node-RED.....	27
Figura 23. Segunda ventana de terminal ejecutando el comando para iniciar el proceso que se realiza con Python.....	28
Figura 24. Segunda ventana mostrando como resultado un código "0", significa que esta conectado de manera exitosa. ....	28
Figura 25. Primer ventana mostrando el mensaje de conectado del proceso de Node-RED. ...	29
Figura 26. Ventanas de terminal minimizadas y dirección de la interfaz de usuario en el navegador.....	29

Figura 27. Ventana principal de la interfaz gráfica. ....	30
Figura 28. Introduciendo credenciales. ....	30
Figura 29. Pestaña en la cual se encuentra el instrumento virtual. ....	31
Figura 30. Ventana de operación del robot. ....	31
Figura 31. Moviendo el servomotor 1 (base). Led Indicador verde significa que no hay error..	32
Figura 32. Asignando valores a los primeros 3 servomotores. ....	32
Figura 33. Ventana terminal del proceso de python, mostrando un nuevo evento, se permitió el acceso a el usuario "david". ....	32
Figura 34. Ventana terminal mostrando información acerca de la última conexión. ....	33
Figura 35. Nombre de usuario alterado (no registrado) ....	33
Figura 36. Error causado por credenciales incorrectas. ....	33
Figura 37. Estado de Error: Error ....	34
Figura 38. Solicitud de acceso denegada al usuario: davidRRR ....	34
Figura 39. Se vuelve a colocar un usuario correcto, el acceso se permite. ....	34
Figura 40. Señal adquirida graficada contra el tiempo donde se observa el consumo bajo bloqueo (izquierda), reposo (centro) y nominal (derecha). ....	35
Figura 41. Pico de corriente por movimiento del servomotor. ....	35
Figura 42. Corriente consumida al mover el servomotor. ....	36
Figura 43. Corriente consumida al realizar esfuerzo manual al servomotor. ....	36

## **I. Introducción**

La instrumentación virtual es la integración entre una interfaz de usuario, adquisición de datos y elementos de software, para cumplir con la función de un instrumento tradicional, siendo el usuario quien define la funcionalidad mediante el software.

Un instrumento virtual es el software que el usuario diseña a su medida, agregando o quitando funcionalidad a su sistema en base a los requerimientos y necesidades del usuario final.

En este trabajo se utilizaron los conceptos de la instrumentación virtual para crear un instrumento que permita la teleoperación del sistema de servos del robot.

Con el uso de un microcontrolador ESP32 como tarjeta de adquisición de datos, el framework Node-RED para la elaboración del programa de control y creación de la interfaz de usuario, una Raspberry Pi 3 y el protocolo MQTT como bus de comunicación entre los distintos componentes; se diseñará la arquitectura del sistema que permitirá cumplir con el objetivo propuesto que es, la teleoperación de un robot.

Específicamente se hará uso de un brazo robótico el cual será el sistema teleoperado a partir de la manipulación de los servomotores que permiten el movimiento al brazo robótico.

## **II. Objetivos del proyecto**

### **Objetivo General**

Diseñar un instrumento virtual que adquiriera la señal de corriente de los servomotores del robot descrito para conocer su estado, muestre las variables de estado pertinentes y permita dirigirlo bajo demanda desde una interfaz de escritorio.

### **Objetivo Particular**

- Realizar un análisis y descripción del sistema a teleoperar.
- Diseñar un programa que sea capaz de realizar lectura, escritura, recepción y transmisión de señales digitales bajo demanda para la tarjeta de adquisición.
- Diseñar una interfaz de escritorio que muestre las variables de estados pertinentes a la teleoperación, tal como la señal de corriente de los servomotores y permita transmitir señales para actuar sobre la teleoperación del sistema.



### III. Marco Teórico

Los instrumentos virtuales representan un progreso desde los sistemas de instrumentación basados únicamente en el hardware a sistemas centrados en el software que aprovechan la potencia de cálculo, productividad y capacidad de conexión de las populares computadoras de escritorio y estaciones de trabajo. [1]

La instrumentación virtual es un concepto introducido por la compañía Nacional Instruments. En el año 1983, Truchard y Kodosky, decidieron enfrentar el problema de crear un software que permitiera utilizar el ordenador personal como instrumento para realizar mediciones.[2]

En sus orígenes, estos equipos, compuestos de una tarjeta de adquisición de datos con acondicionamiento de señales y el software apropiado estaban orientados a laboratorios, donde sus prestaciones eran muy requeridas por la gran precisión y capacidad de adecuar sus capacidades y cálculos de forma acorde al proceso que se estaba analizando. Con el tiempo, se fueron obteniendo cada vez soluciones y prestaciones más robustas en PC. [2]

Con los instrumentos virtuales se pueden construir sistemas que se ajusten exactamente a las necesidades del usuario (definidas por el mismo) en lugar de estar limitados por los instrumentos tradicionales de funciones fijas (definidos por el fabricante).

El término virtual nace precisamente a partir del hecho de que cuando se utiliza el PC como “instrumento” es el mismo usuario quien, a través del software define su funcionalidad y “apariencia” y por ello decimos que “virtualizamos” el instrumento, ya que su funcionalidad puede ser definida por el usuario y no por el fabricante. [1]

#### Instrumento

Un instrumento, es un objeto fabricado, herramienta, aparato, etc., que sirve para hacer una operación, producir un efecto, etc. [3]

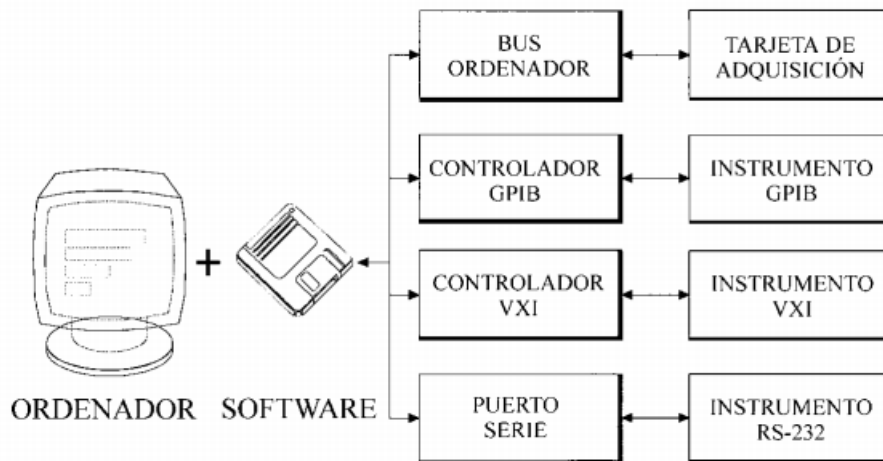


Figura 1. Diferentes tipos de instrumentos.

### Instrumento Virtual

Un instrumento virtual es la combinación de Hardware y elementos de Software, usados por un PC, para cumplir la función de un instrumento tradicional.[4]

Un instrumento virtual es software que el usuario emplea para desarrollar un sistema computarizado de pruebas y mediciones para controlar desde una computadora un dispositivo externo de medición, y para mostrar, probar o medir datos obtenidos por un dispositivo externo. [3]

De esta manera surge el concepto de instrumento virtual, como lo definieron en National Instruments: “un instrumento que no es real, se ejecuta en una computadora y tiene sus funciones definidas por software”. [2]

¿Por qué un computador digital?

Debido a que están basados en el PC, los instrumentos virtuales aprovechan inherentemente los beneficios de la última tecnología de las computadoras personales corrientes. Estos avances en tecnología y rendimiento incluyen poderosos procesadores y sistemas operativos, además de incorporar poderosas herramientas tales como Internet.

El uso de un computador digital en los sistemas de instrumentación y control ofrece:

- Mayor rendimiento del proceso y por lo tanto menores costos y mayor producción.
- Buena calidad y velocidad, necesarias para operar en tiempo real.
- Mayor seguridad (inmediata acción de corrección y activación de alarmas).
- Gran cantidad de información acerca del proceso.

Los instrumentos tradicionales tienen la característica de estar definidos por el fabricante. Tomemos de ejemplo un osciloscopio, este instrumento tiene aplicaciones específicas y limitadas por su hardware, esto porque el instrumento tradicional posee una o varias funciones específicas, lo que conduce a tener una capacidad limitada por el mismo instrumento (Hardware) en cuanto a funciones.

En el instrumento virtual, el software es la clave del sistema, a diferencia del instrumento tradicional, donde la clave es el hardware. [1]

<u><b>Instrumento Tradicional.</b></u>	<u><b>Instrumento Virtual.</b></u>
.-Definido por el proveedor.	.-Definido por el usuario.
.-Posee una función específica, lo que conduce a tener una baja capacidad de interacción.	.-Sistemas orientados a la aplicación, con capacidad de interactuar con redes, periféricos y otras aplicaciones.
.-Se basa en el hardware.	.-Se basa en el software.
.-El costo de adquisición es alto.	.-Bajo costo. Reprogramable.
.-Tecnología base estable.  (ciclo de vida: 5 a 10 años)	.-Tecnología base en constante desarrollo. (ciclo de vida: 1 a 2 años)
.-Mínima economía de escala.	.-Máxima economía de escala.
.-Costo de desarrollo y mantenimiento elevados.	.-El uso de software minimiza los costos de desarrollo y mantenimiento.

*Figura 2. Características de un Instrumento Tradicional frente a uno Virtual.*

Algunas de las ventajas de los instrumentos virtuales es la creación de interfaces de usuario que mejor satisfagan el objetivo de la aplicación, pueden definir cómo y cuándo la aplicación adquiere datos desde el dispositivo, como los procesa, manipula, almacena y como se presentan los resultados al usuario.

Generalmente, todos los instrumentos virtuales propuestos son de bajo costo, comparado a los instrumentos tradicionales. Los sistemas más propuestos usualmente están basados en herramientas de LabVIEW o MATLAB, pero éstas son costosas y complejas. [5]

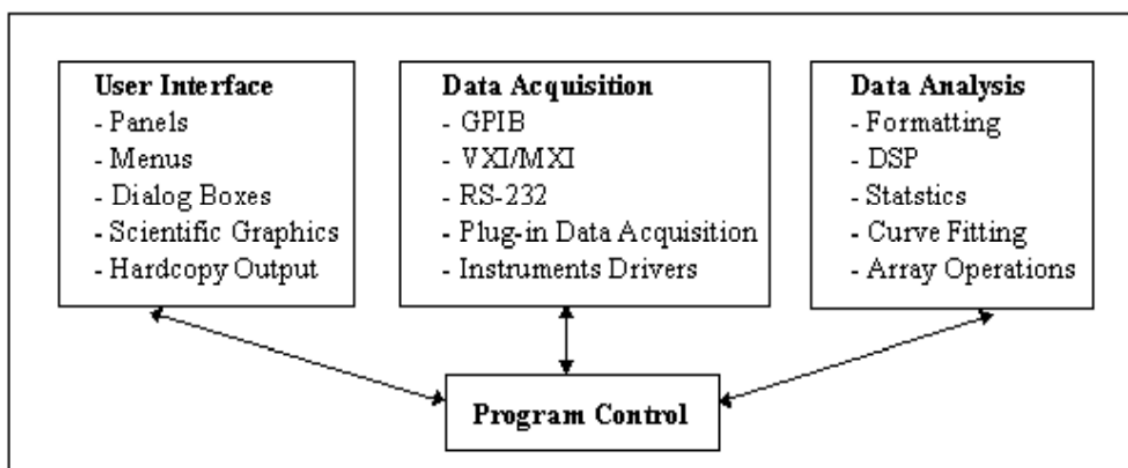
*Componentes que conforman un instrumento virtual.*

## Hardware

El hardware esta conformado por todos los elementos físicos que forman parte del sistema, en este caso del instrumento virtual, tales como la computadora personal, la tarjeta de adquisición, los transductores y actuadores.

## Software

El software esta conformado por todos los elementos de programación que forman parte del sistema, y son los que le dan la funcionalidad, flexibilidad, reusabilidad y personalización al sistema. Algunos ejemplos son el programa de control, el programa de comandos además de el código o los scripts que se utilizan para dar seguridad al sistema.



*Figura 3. estructura típica de un programa de instrumentación virtual*

### 1. Interfaz Gráfica

El editor de la interfaz de usuario permite construir con esfuerzo mínimo, objetos interactivos para el programa de control, de esta manera, la interfaz se adecuará a nuestras necesidades contando con controles e indicadores que permitan llevar a cabo la aplicación en curso.

### 2. Programa de Control

El programa de control nos permite controlar la aplicación: la adquisición de datos, el análisis de estos y la manipulación de la interfaz gráfica, para esta tarea los softwares

de instrumentación virtual proveen lenguajes de alto nivel, como pueden ser los lenguajes gráficos y el lenguaje C.

### 3. *Adquisición de Datos*

El software de instrumentación virtual permite, mediante una serie de órdenes, realizar operaciones como la lectura o la escritura en cualquier instrumento, la tarjeta de adquisición, etc.

### 4. *Análisis de Datos*

Después de adquirir los datos, en muchas ocasiones es necesario analizarlos, realizar operaciones estadísticas de los datos, su adecuación a un tipo de formato, etc.

La instrumentación virtual implica la adquisición de señales, el procesamiento, análisis, almacenamiento, distribución y despliegue de los datos e información relacionados con la medición de una o varias señales, la interfaz hombre-máquina, la visualización, monitorización y supervisión remota del proceso, la comunicación con otros equipos, etc. [2]

## Diseño de Instrumentos Virtuales

Para construir un instrumento virtual, sólo se requiere de un ordenador, una tarjeta de adquisición de datos con acondicionamiento de señales y el software apropiado. Debe realizar, como mínimo, las tres funciones básicas de un instrumento convencional: adquisición, análisis y presentación de datos. [2]

## Componentes clave del Instrumento Virtual desarrollado.

### ESP32 de espressif

El ESP32 es un sistema dual-core con dos CPUs de arquitectura Harvard Xtensa LX6, es un microcontrolador en el que está integrado un chip híbrido para conectividad Wi-Fi y Bluetooth de tal manera que tiene capacidad para una amplia variedad de aplicaciones. Puede ser utilizado como interfaz con otros sistemas para proveer funcionalidad por Wi-Fi y Bluetooth.

Brinda capacidad de realizar lectura de señales digitales y analógicas, así como de realizar operaciones y escribir señales digitales. Para este trabajo algunas de las características que se aprovecha de este componente es la escritura de señales de tipo PWM en los pines del microcontrolador, la lectura de señales por medio del ADC que contiene el microcontrolador, y los dos núcleos de procesamiento para tener un mejor rendimiento en la operación del sistema.



*Figura 4. ESP32-C3-DevKitC-02*

Mas información: <https://www.espressif.com/en/products/socs/esp32>

### Raspberry Pi 3 Model B+

La Raspberry Pi es una computadora de bajo costo y de tamaño compacto, usualmente se compara a el tamaño de una tarjeta de crédito, puede ser conectada a un monitor de computadora o a una televisión, usarse con un mouse y teclado estándar. Corre el sistema operativo Linux capaz de permitirle a las personas explorar la computación, usualmente se programa con Python, un lenguaje de programación popular y es capaz de hacer la mayoría de las tareas típicas de un computador de escritorio. Para este trabajo lo que la hace buen elemento es el poder instalar un “broker” (servidor) en ella, de tal modo que sirva como centro de intercambio de datos para la comunicación de los elementos del sistema.



*Figura 5. Raspberry Pi 3 Model B+*

Mas información en: <https://raspberrypi.cl/raspberry/>

MQTT: El protocolo de comunicación estándar para el internet de las cosas (IoT)

El protocolo MQTT fue diseñado como transporte de mensajes vía una publicación/suscriptor de peso ligero, lo cual lo hace ideal para conectar dispositivos de manera remota con un mínimo ancho de banda de red. Hoy en día MQTT es utilizado en una amplia variedad de industrias, desde la automotriz, de manufactura, telecomunicaciones, combustible, gas, etc.

Para este trabajo se planteó el uso de este protocolo por su capacidad de comunicar dispositivos de manera inalámbrica, por su velocidad de transmisión y por su intrínseco funcionamiento suscriptor/publicador.

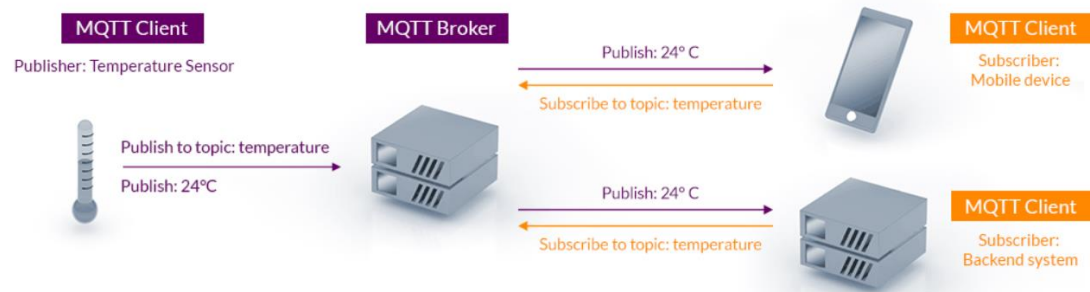


Figura 6. Arquitectura MQTT

Mas información en: <https://mqtt.org/>

## Node-RED

Node-RED es una herramienta creada por OpenJS Foundation para “cablear” dispositivos, APIs y servicios en línea en una nueva y más interesante manera. Es usado para aplicaciones de eventos bajo demanda y provee un ambiente basado en un navegador web que hace sencillo conectar con cable una amplia variedad de nodos, que pueden ser funciones, elementos gráficos, APIs, etc. y que pueden ser publicados con un solo click.

Node-RED contiene módulos, que permiten crear una interfaz de usuario de manera rápida y sencilla, además con JavaScript como lenguaje de programación, aplicar funciones y lógica a el proceso que se este llevando a cabo. Esta herramienta sopprta el uso de MQTT para comunicarse con otros dispositivos, por esto y por su capacidad de crear interfaces de usuario amigables es que se seleccionó para este trabajo.

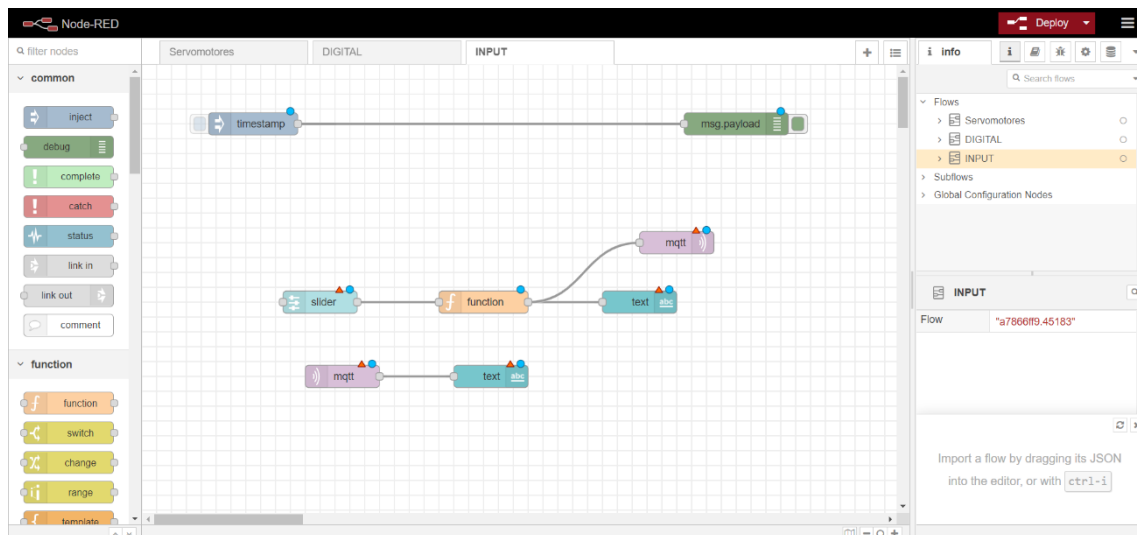


Figura 7. Ambiente de desarrollo de Node-RED

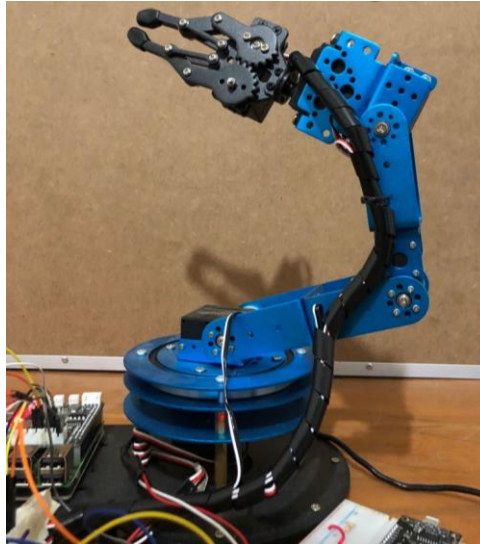
Mas información: <https://nodered.org/>



## IV. Metodología

### Análisis y Descripción del Sistema

El sistema a teleoperar es un brazo robótico de 5 grados de libertad con un sistema de servomotores como actuadores. Para este trabajo sólo 4 servomotores serán de interés. La operación de este sistema radica en manipular sus componentes, es decir, los servomotores, por lo tanto, es necesario saber el funcionamiento de estos componentes y sus características.



*Figura 8. Fotografía lateral del robot a teleoperar.*



*Figura 9. Fotografía del robot a teleoperar, servomotores visibles.*

Como se comentó anteriormente los servomotores realizan su movimiento dependiendo de una señal de entrada de tipo PWM, donde se varía el ciclo de trabajo, es decir, la relación del tiempo de un pulso positivo con el tiempo de un pulso negativo en un determinado periodo de tiempo. Las características principales de estos servomotores son las siguientes.

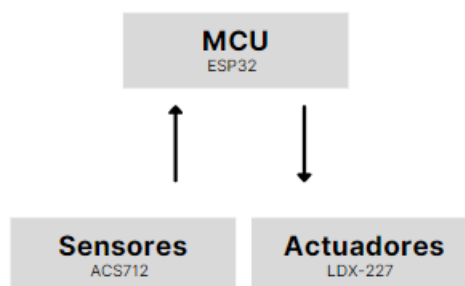
- Frecuencia: 50 Hz
- Periodo de pulso: 20ms
- Ancho de pulso: 500-2500 us

Con estos datos podemos enviarle al servomotor una señal PWM con un ancho de pulso variable entre 500 y 2500 microsegundos con un periodo de 20 milisegundos.

Por otra parte, para cerciorarnos del movimiento del servomotor podemos observar la corriente que se encuentra recibiendo, teniendo estos componentes una corriente mínima sin carga de 100mA. El servomotor para realizar su movimiento necesita consumir corriente, para transformar esa energía eléctrica en energía mecánica, de manera que, la corriente suministrada al servomotor aumenta si el servomotor se encuentra en movimiento. Además, si el servomotor se encuentra con un obstáculo externo, que le impida llegar a su posición indicada, el mismo requerirá de más corriente para superar la fuerza externa y llegar a su posición.

Resumiendo lo anteriormente comentado, necesitamos de una fuente de señal PWM y la lectura de la corriente que se encuentra circulando en los servomotores.

Esto se logra con la implementación de un microcontrolador ESP32 actuando como una tarjeta de adquisición de datos para realizar lecturas de un sensor de corriente mientras que al mismo tiempo permita la escritura de señales PWM hacia los componentes.



*Figura 10. Diagrama adquisición de datos y escritura de señales*

El software que se embeba en este componente será el que le dé la funcionalidad a este, y por lo tanto se debe de diseñar de forma escalable, modular, reusable, extensible y sobre todo

simple, llevando un orden en el flujo de programa y tomando en consideración la comunicación con otros dispositivos.

### Características limitantes

La característica esencial del trabajo es que el sistema tenga la capacidad de ser teleoperado, es decir, que una persona mediante un medio tecnológico y de forma remota, actúe sobre los componentes del sistema, siendo esos componentes los servomotores del robot y el medio tecnológico un protocolo de comunicación inalámbrico como lo es MQTT.

Dentro de esta situación de operar de manera remota, viene consigo el uso de un protocolo de comunicación inalámbrico, y con esto que los demás elementos o dispositivos tengan la capacidad de comunicarse a través de ese mismo protocolo.

La herramienta que se seleccionó para la creación de la interfaz gráfica y el programa de control es Node-RED ya que tiene soporte para el uso de este protocolo, y además permite una creación rápida de interfaces gráficas que cuya funcionalidad se encuentra directamente ligada a un programa de control, que es el que interactúa con los demás elementos del sistema.

Otra característica limitante es el uso de hardware, si bien es necesario tener componentes físicos, los conceptos de la instrumentación virtual hacen énfasis en el software, de manera que se limitará el uso de hardware al mínimo posible, siendo el hardware preestablecido de una computadora personal la que junto con el software adecuado maneje la carga del sistema.

COMPONENTE	FUNCIÓN	ELEMENTO
INTERFAZ DE USUARIO	VISUALIZACION E INTERACCIÓN	NODE-RED
PROGRAMA DE CONTROL	SOFTWARE DE APLICACIÓN	NODE-RED
PROTOCOLO DE COMUNICACIÓN	COMUNICACIÓN ENTRE COMPONENTES	MQTT RASPBERRY PI
ANALISIS Y PROCESAMIENTO DE DATOS	FORMATEO Y OPERACIONES	ESP32 NODE-RED
ADQUISICION DE DATOS	SOFTWARE EMBEBIDO	ESP32

Figura 11. Componentes de instrumentación virtual, funcionalidad y elemento seleccionado

## Arquitectura General

Para el diseño de la arquitectura se clasificó en etapas, enlistando las características y el seguimiento de cada una de ellas.

- Adquisición de datos
  - Señal. (Voltaje)
  - Tarjeta DAQ. (ESP32)
  - BUS de Comunicación.
  - Programación. (Software embebido)
- Procesamiento de datos
  - Operaciones o funciones matemáticas.
- Transmisión de datos de manera inalámbrica
  - Protocolo MQTT

Nota: Los datos pueden ser transmitidos de manera continua o bajo demanda.

- Seguridad
  - Control de acceso al sistema.
- Transmisión de datos
  - Protocolo MQTT

Nota: Datos continuos o sobre demanda, gran cantidad de datos o un dato.

- Procesamiento de datos
  - Operaciones o funciones matemáticas.
- Visualización de los datos
  - Interfaz gráfica en navegador. (Node-RED)

La instrumentación virtual es el conjunto de estos componentes, los cuales ya han sido descritos, la integración de estos y su funcionamiento se visualizan en la siguiente imagen.

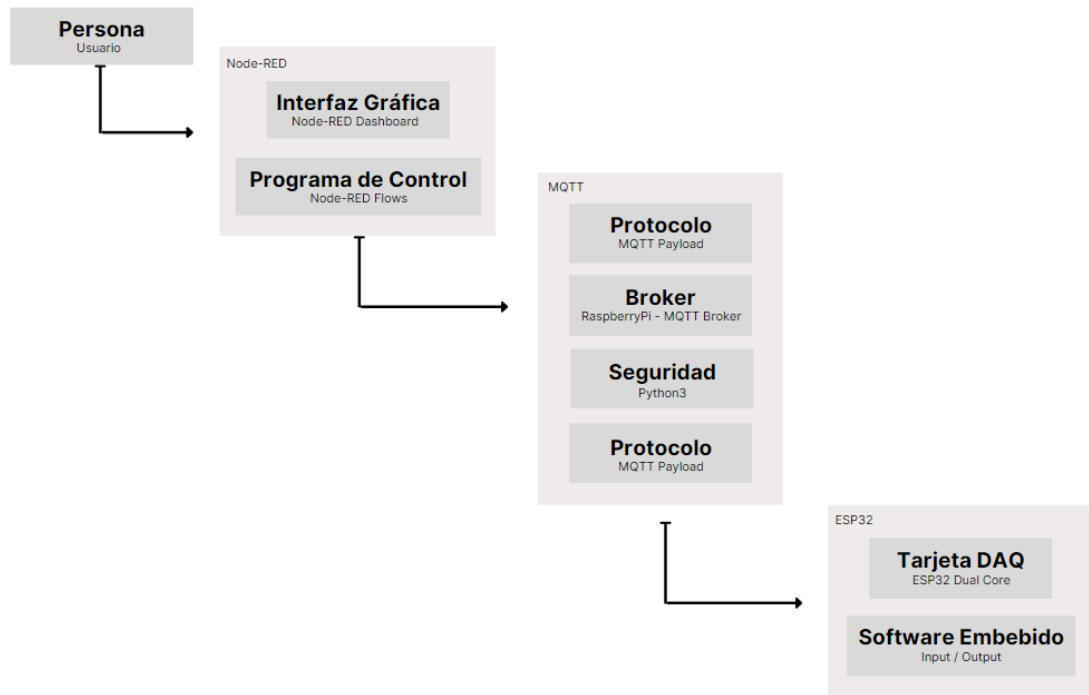


Figura 12. Diagrama de Estructura General

Para diseñar la arquitectura general del sistema se partió de la funcionalidad, el usuario debe poder manipular los actuadores del sistema desde la interfaz gráfica, y en esta misma observar las señales adquiridas de los sensores. Esta información debe ser obtenida, procesada, transmitida, procesada nuevamente y visualizada en la interfaz gráfica, esto para cumplir con las funciones de adquisición, análisis y presentación de datos y además, agregar la función de operación desde la interfaz de usuario hasta los actuadores. Utilizando los componentes esenciales que conforman la instrumentación virtual se denominaron los elementos que se utilizarán para cada componente.

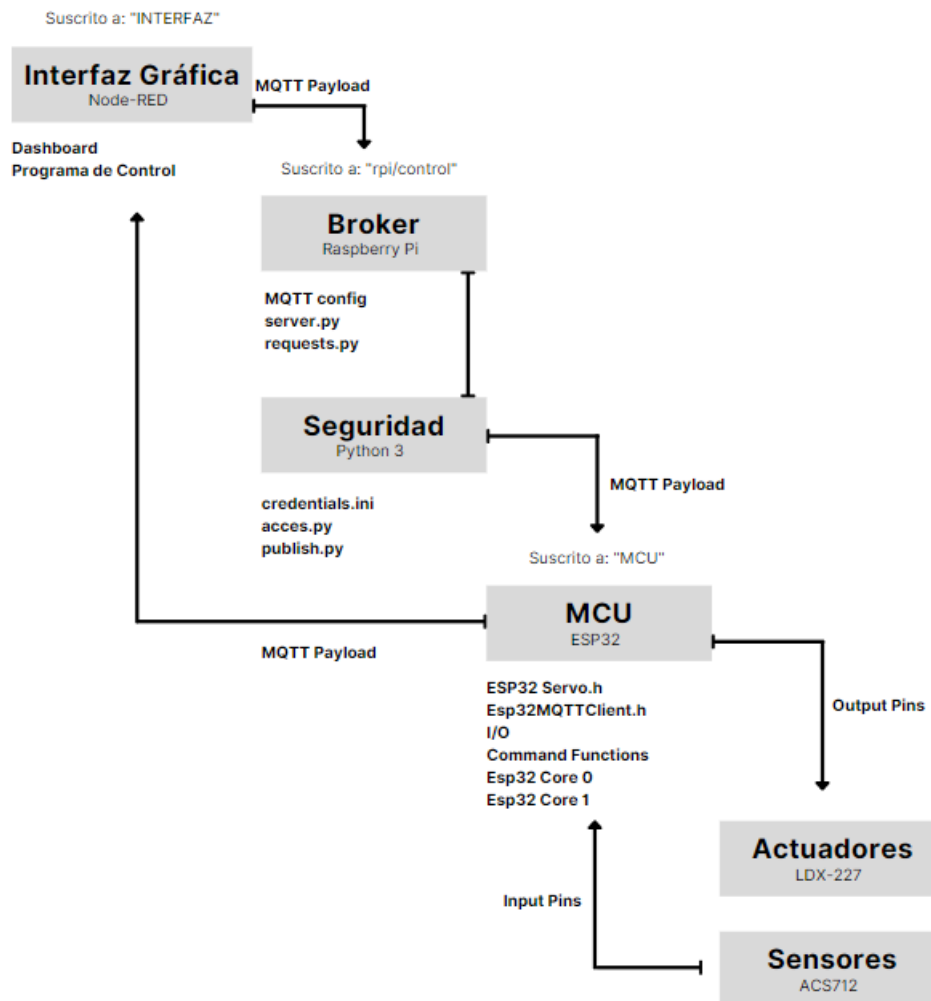


Figura 13. Diagrama Arquitectura con Detalles

## Arquitectura del Software

### Software Embebido

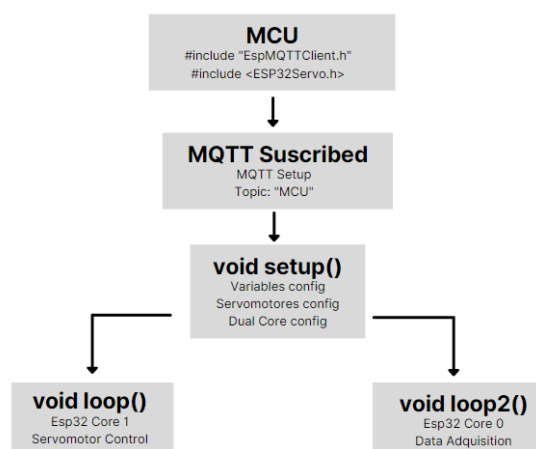


Figura 14. Estructura de software para ESP32

## Software de Seguridad

El software que tiene la función de actuar como una capa de seguridad, para controlar el acceso a los controles del sistema y solo el usuario autorizado pueda realizar la operación deseada. Esto permite el control del acceso a los controles del sistema, de modo que es necesario que este registrado y sea un usuario autenticado. Esto se logra mediante la implementación de varios archivos que se ejecutan en la Raspberry Pi.

El archivo “credentials.ini” es un archivo clave/valor el cual tiene registrados los nombres de usuario con los que se puede acceder a los controles en Node-RED, así como sus contraseñas. A cualquier usuario/contraseña que no este registrado en este archivo, se le negará el acceso.

Por medio de scripts de Python, se puede correr el programa que sirve de servidor para el sistema, controlando el flujo de los mensajes que se hayan enviado al ESP32 desde la interfaz de usuario en Node-RED. Además, se controla el acceso al mismo.

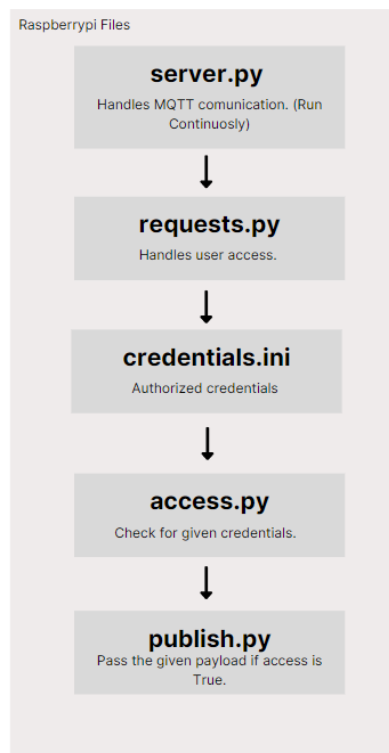


Figura 15. Archivos en la raspberry

## Software de Control

La interfaz de usuario es lo que el usuario ve y con lo que puede interactuar, el software o programa de control es aquel que no se ve, es como esta construida la lógica que deberán seguir los elementos en la interfaz de usuario. En este proyecto, el software de control esta constituido por varias pestañas conocidas como “flows” o “flujos” en node-red.

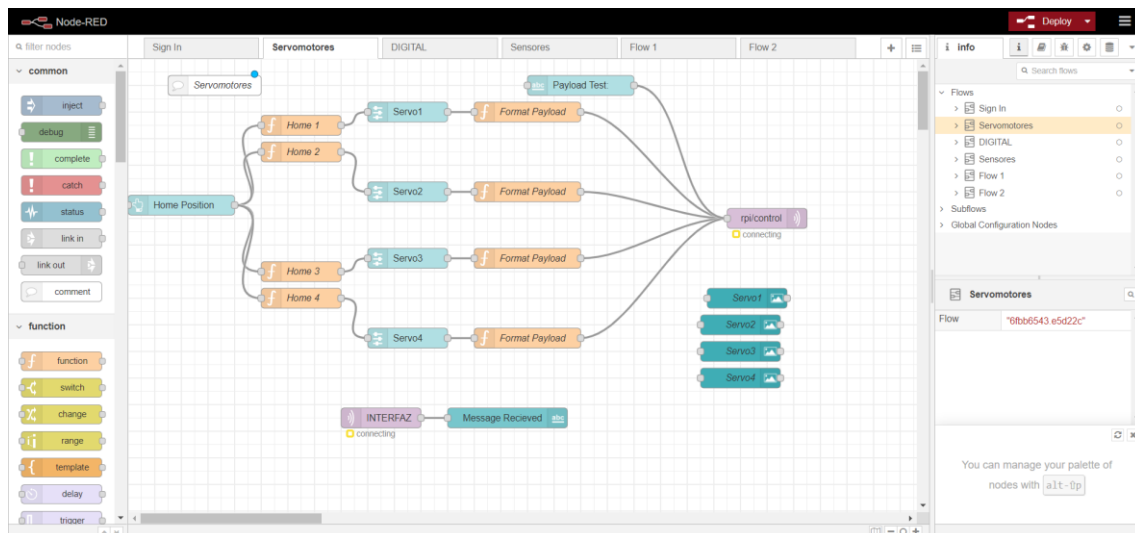


Figura 16. Captura de pantalla del programa de control. Flujo principal

Un elemento muy importante del funcionamiento es el “Payload”, esto es el mensaje completo que es enviado desde la interfaz de usuario hacia los demás componentes. Se envía aquí toda la información necesaria para realizar una acción, cada parámetro debe ir en su debido orden según marca la descripción.

	<i>request_payload</i>					
Description	user	password	command	pin	value	
Example	david	sasso	PWM	14	180	
	credentials		payload			

Figura 17. Estructura de payload

## Arquitectura del Hardware

### Conexiones

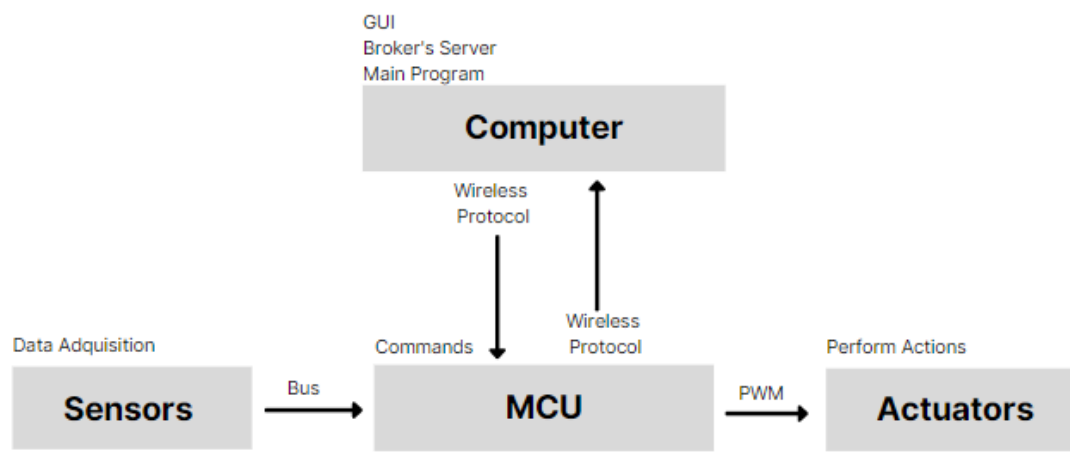


Figura 18. Diagrama de bloques de conexiones y funciones.



## Servicio de escritorio remoto

Para el acceso al sistema, que se encuentra en una red local (A) con acceso a internet, desde otra red local (B) que también cuenta con acceso a internet se utilizó un servicio de escritorio remoto, que permite la conexión segura a través de internet entre dos dispositivos, registrados en la cuenta del servicio.

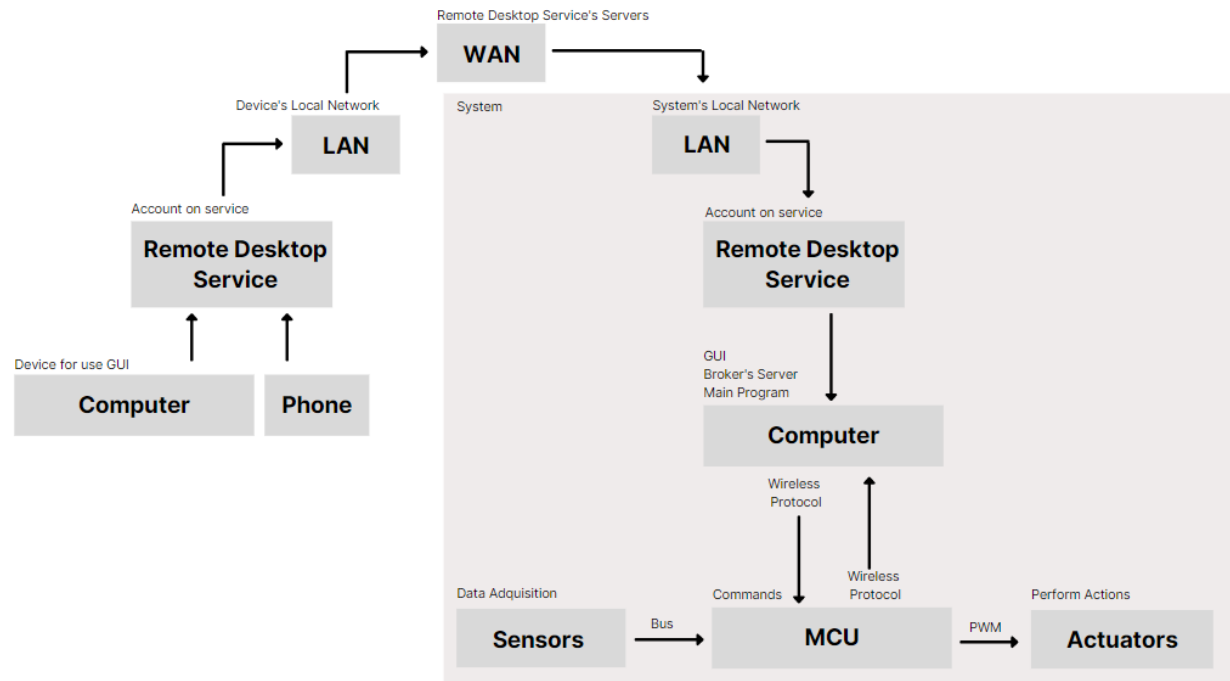


Figura 19. Estructura de conexión desde servicio de escritorio remoto

## V. Análisis de resultados

A continuación, se muestra de manera textual y visual el funcionamiento y se discuten sus resultados.

Para comenzar con la operación es necesario comenzar la ejecución de ciertos procesos en la computadora, ya que no están configurados para su ejecución automática al momento de iniciarla.

1. Debemos iniciar el servicio de escritorio remoto, en la computadora desde la cual se desea realizar la teleoperación del Sistema de Instrumentación Virtual que en lo sucesivo se denominará SIV.

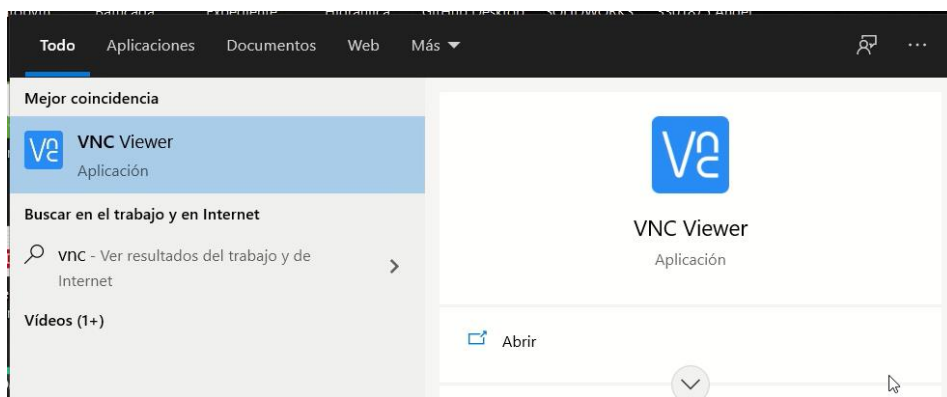


Figura 20. Iniciando servicio de escritorio remoto.

2. Se necesita iniciar sesión con la cuenta que se encuentre activa en el SIV.

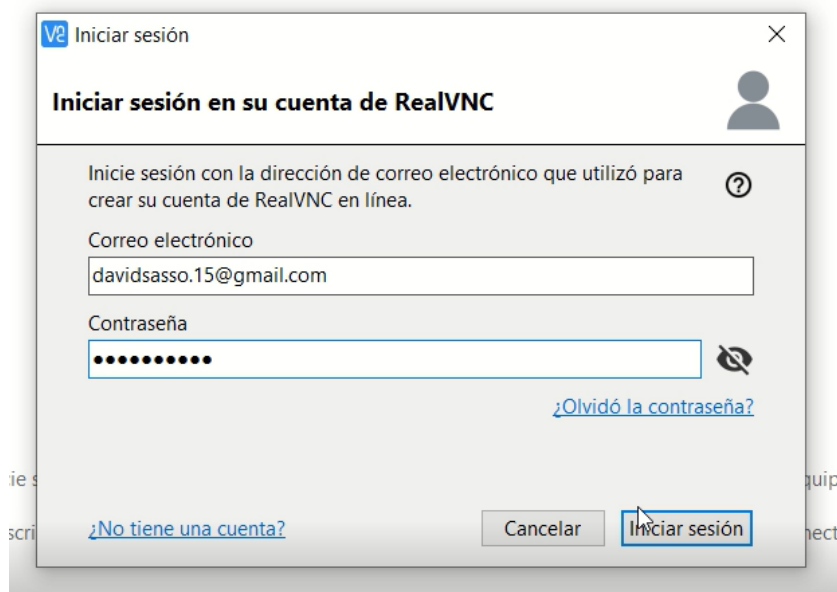


Figura 21. Inicio de sesión en la cuenta del servicio de escritorio remoto.

- Una vez iniciada la sesión y siempre y cuando la computadora del SIV se haya registrado en la cuenta, deberá aparecer como registrada. Se procede a acceder a ella.

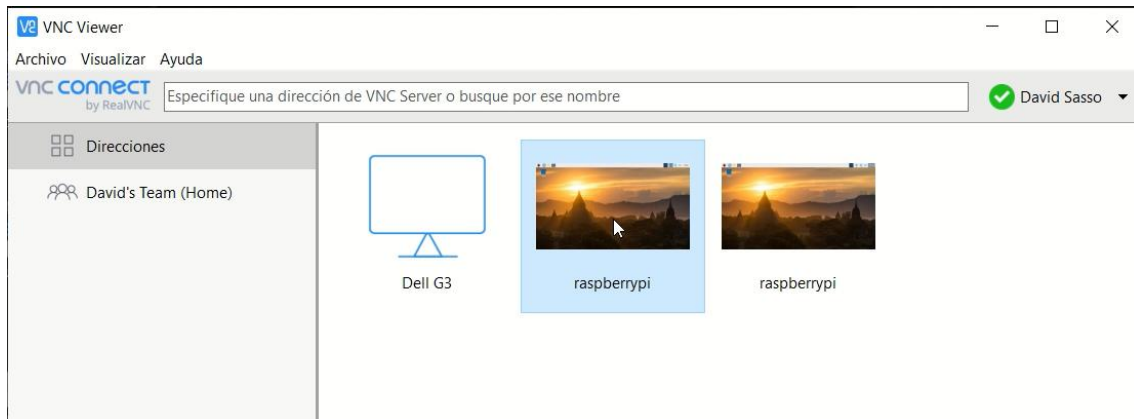


Figura 22. Selección de computadora para servicio remoto.

- Se requerirán las credenciales de algún usuario de esa computadora.

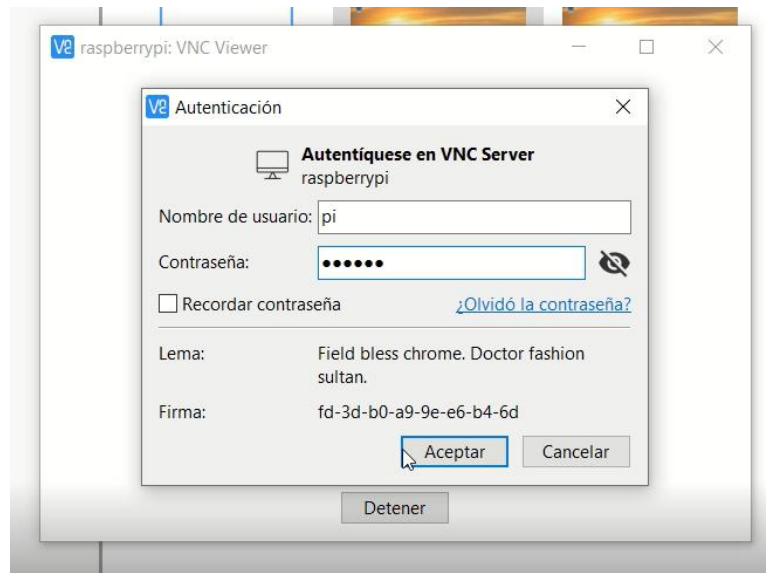


Figura 23. Ingreso de credenciales locales de la computadora remota.

- Si las credenciales son correctas y el equipo se encuentra activo, nos dará acceso a la computadora.

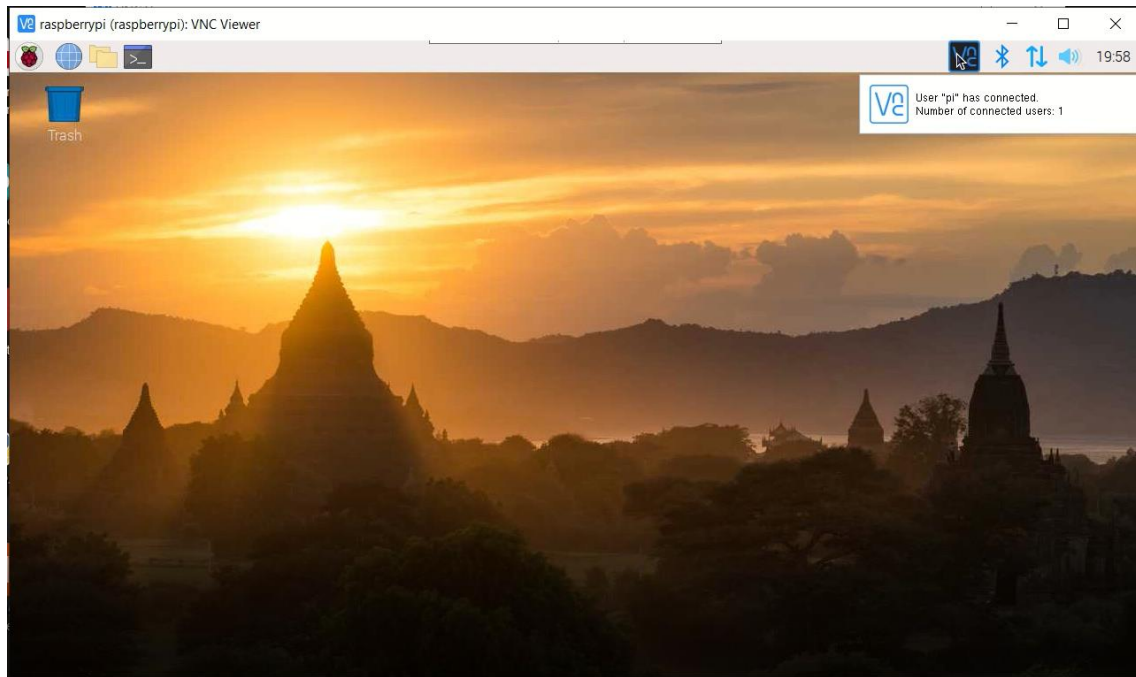


Figura 24. Escritorio de la computadora a través del servicio.

6. Se procede a ejecutar los procesos que deben estar activos para el funcionamiento del SIV. Estos son: <server.py> y <node-red>. Esto será llevado a cabo desde la terminal de la computadora.

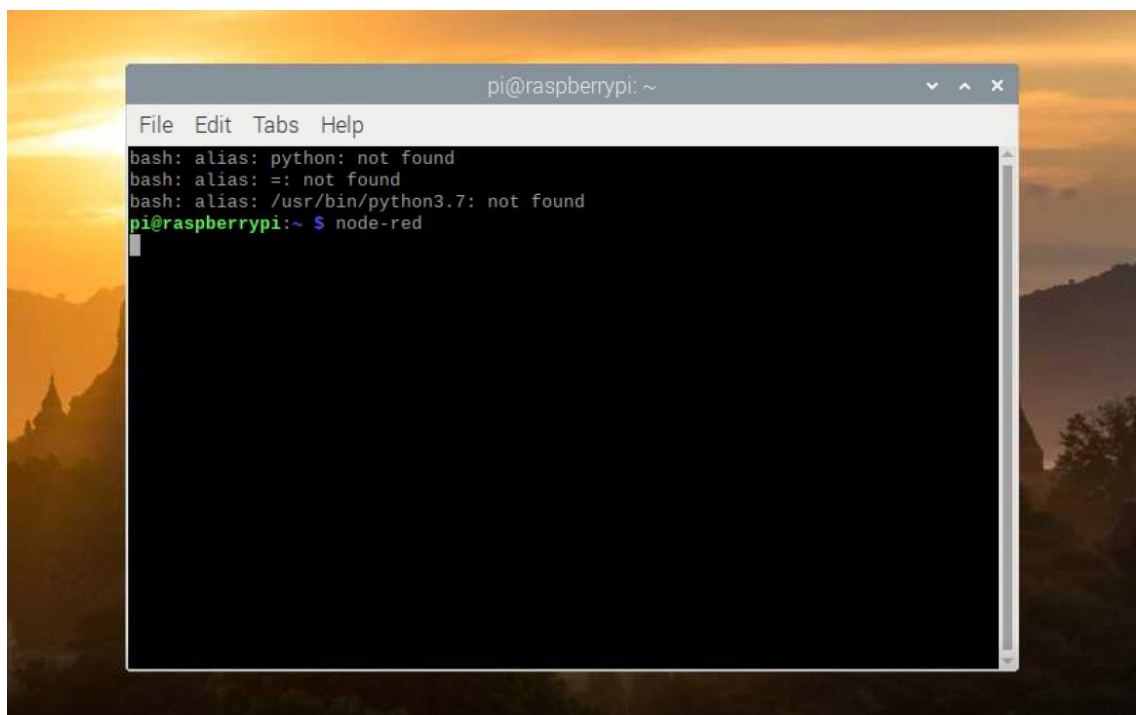
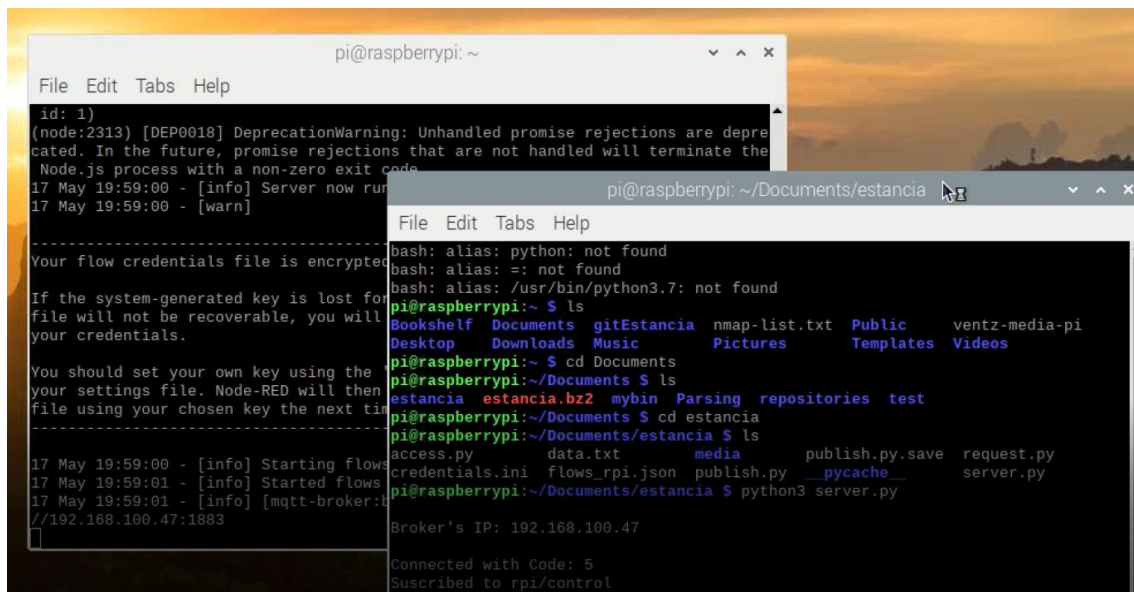


Figura 25. Comando ejecutado desde la terminal para iniciar el proceso de Node-RED.

7. Se ejecuta el comando para iniciar el proceso de Node-RED. Y en otra ventana de terminal se ejecuta el comando para iniciar el proceso llevado a cabo con Python.



```
id: 1)
(node:2313) [DEP0018] DeprecationWarning: Unhandled promise rejections are depre
cated. In the future, promise rejections that are not handled will terminate the
Node.js process with a non-zero exit code
17 May 19:59:00 - [info] Server now run
17 May 19:59:00 - [warn]

-----
Your flow credentials file is encrypted
If the system-generated key is lost fo
file will not be recoverable, you will
your credentials.

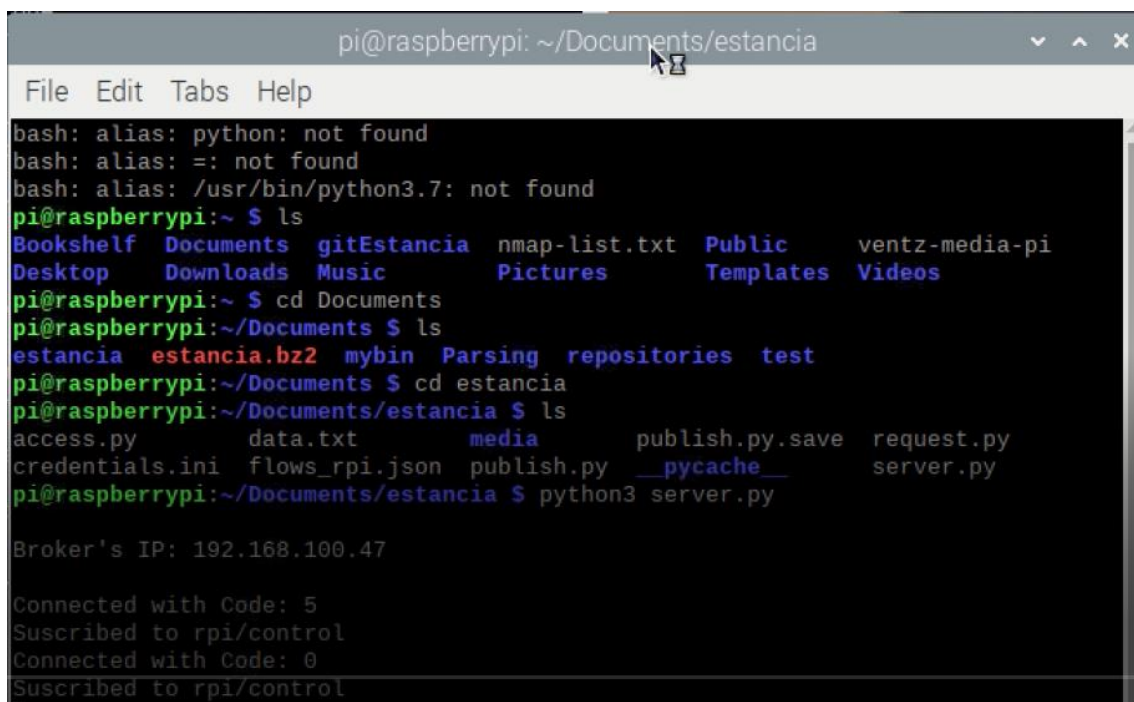
You should set your own key using the
your settings file. Node-RED will then
file using your chosen key the next ti
-----
17 May 19:59:00 - [info] Starting flows
17 May 19:59:01 - [info] Started flows
17 May 19:59:01 - [info] [mqtt-broker:
//192.168.100.47:1883

bash: alias: python: not found
bash: alias: =: not found
bash: alias: /usr/bin/python3.7: not found
pi@raspberrypi:~ $ ls
Bookshelf  Documents  gitEstancia  nmap-list.txt  Public  ventz-media-pi
Desktop    Downloads  Music        Pictures        Templates  Videos
pi@raspberrypi:~ $ cd Documents
pi@raspberrypi:~/Documents $ ls
estancia  estancia.bz2  mybin  Parsing  repositories  test
pi@raspberrypi:~/Documents $ cd estancia
pi@raspberrypi:~/Documents/estancia $ ls
access.py      data.txt      media      publish.py.save  request.py
credentials.ini flows_rpi.json publish.py  __pycache__      server.py
pi@raspberrypi:~/Documents/estancia $ python3 server.py

Broker's IP: 192.168.100.47

Connected with Code: 5
Suscribed to rpi/control
```

Figura 26. Segunda ventana de terminal ejecutando el comando para iniciar el proceso que se realiza con Python.



```
pi@raspberrypi: ~/Documents/estancia

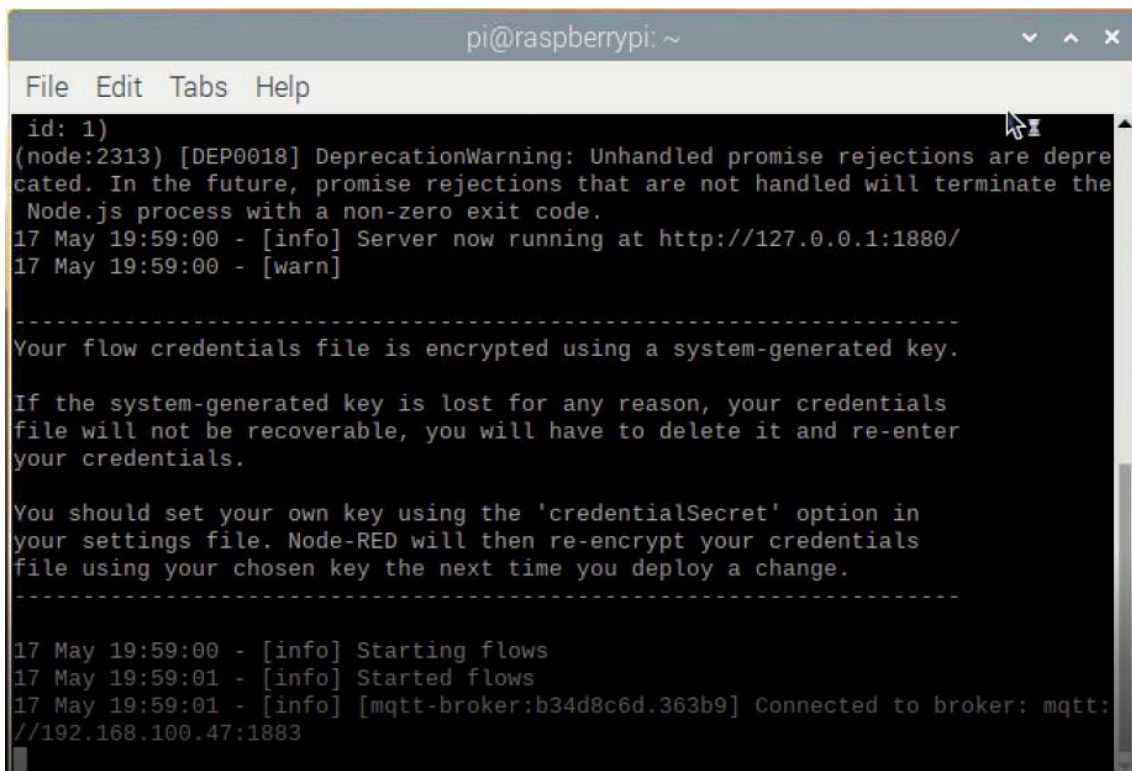
File Edit Tabs Help

bash: alias: python: not found
bash: alias: =: not found
bash: alias: /usr/bin/python3.7: not found
pi@raspberrypi:~ $ ls
Bookshelf  Documents  gitEstancia  nmap-list.txt  Public  ventz-media-pi
Desktop    Downloads  Music        Pictures        Templates  Videos
pi@raspberrypi:~ $ cd Documents
pi@raspberrypi:~/Documents $ ls
estancia  estancia.bz2  mybin  Parsing  repositories  test
pi@raspberrypi:~/Documents $ cd estancia
pi@raspberrypi:~/Documents/estancia $ ls
access.py      data.txt      media      publish.py.save  request.py
credentials.ini flows_rpi.json publish.py  __pycache__      server.py
pi@raspberrypi:~/Documents/estancia $ python3 server.py

Broker's IP: 192.168.100.47

Connected with Code: 5
Suscribed to rpi/control
Connected with Code: 0
Suscribed to rpi/control
```

Figura 27. Segunda ventana mostrando como resultado un código "0", significa que está conectado de manera exitosa.



```
pi@raspberrypi: ~  
File Edit Tabs Help  
id: 1)  
(node:2313) [DEP0018] DeprecationWarning: Unhandled promise rejections are deprecated. In the future, promise rejections that are not handled will terminate the Node.js process with a non-zero exit code.  
17 May 19:59:00 - [info] Server now running at http://127.0.0.1:1880/  
17 May 19:59:00 - [warn]  
  
-----  
Your flow credentials file is encrypted using a system-generated key.  
  
If the system-generated key is lost for any reason, your credentials file will not be recoverable, you will have to delete it and re-enter your credentials.  
  
You should set your own key using the 'credentialSecret' option in your settings file. Node-RED will then re-encrypt your credentials file using your chosen key the next time you deploy a change.  
-----  
  
17 May 19:59:00 - [info] Starting flows  
17 May 19:59:01 - [info] Started flows  
17 May 19:59:01 - [info] [mqtt-broker:b34d8c6d.363b9] Connected to broker: mqtt://192.168.100.47:1883
```

Figura 28. Primera ventana mostrando el mensaje de conectado del proceso de Node-RED.

- Una vez iniciados estos dos procesos se pueden minimizar estas dos ventanas de terminal, abrir una pestaña del navegador e introducir la dirección en la cual está la interfaz gráfica. En este caso es: <http://localhost:1880/ui/>

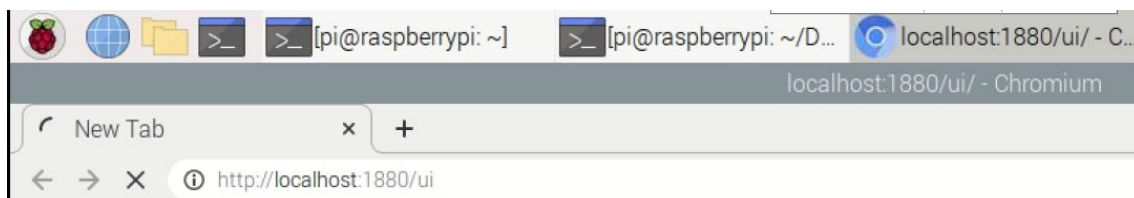


Figura 29. Ventanas de terminal minimizadas y dirección de la interfaz de usuario en el navegador.

- Hasta este punto ya podemos comenzar a utilizar la interfaz de usuario.

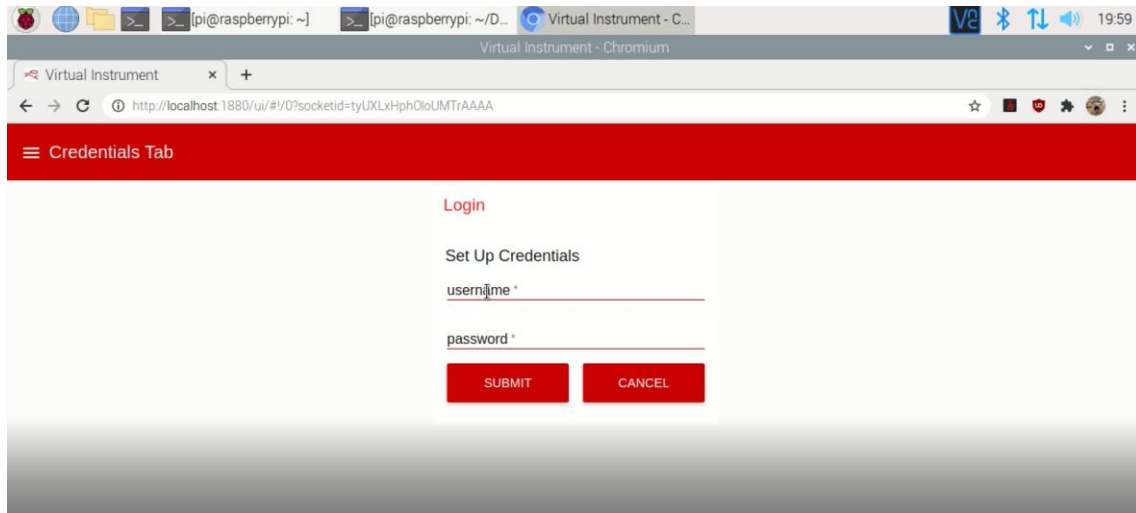


Figura 30. Ventana principal de la interfaz gráfica.

10. Introducimos las credenciales para poder utilizar el SIV.

Figura 31. Introduciendo credenciales.

11. Nos dirigimos a la pestaña del instrumento virtual.



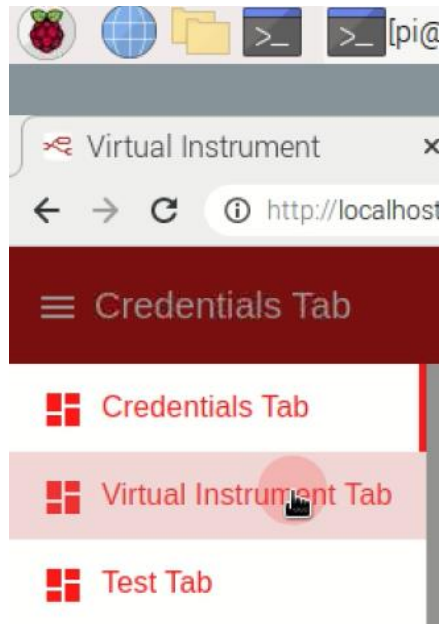


Figura 32. Pestaña en la cual se encuentra el instrumento virtual.

12. Podemos seleccionar un servomotor dependiendo el grado de libertad que queramos operar, como lo muestran las imágenes de referencia, y asignarle una posición.

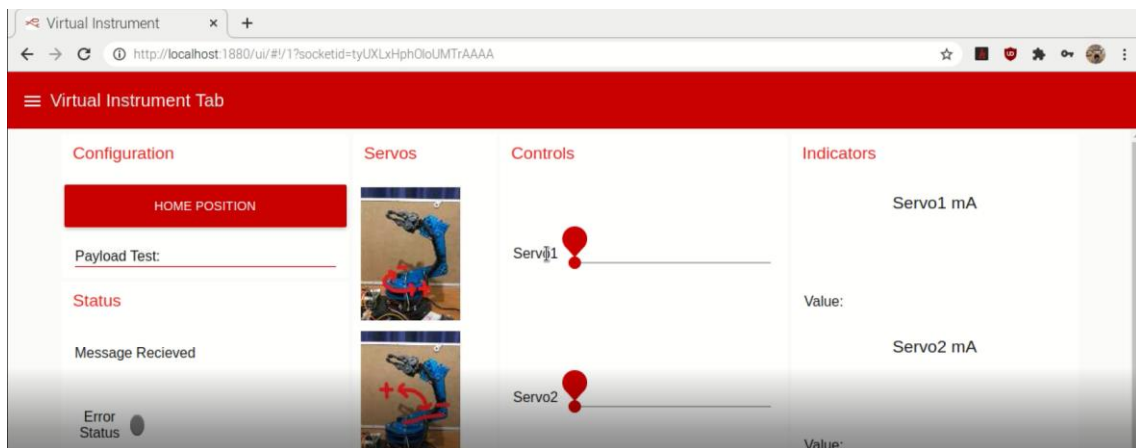


Figura 33. Ventana de operación del robot.



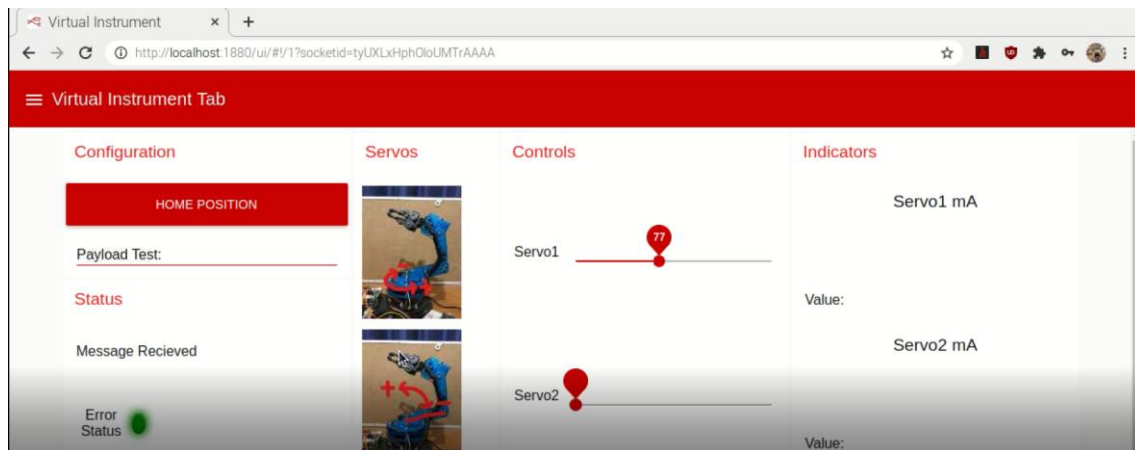


Figura 34. Moviendo el servomotor 1 (base). Led Indicador verde significa que no hay error.

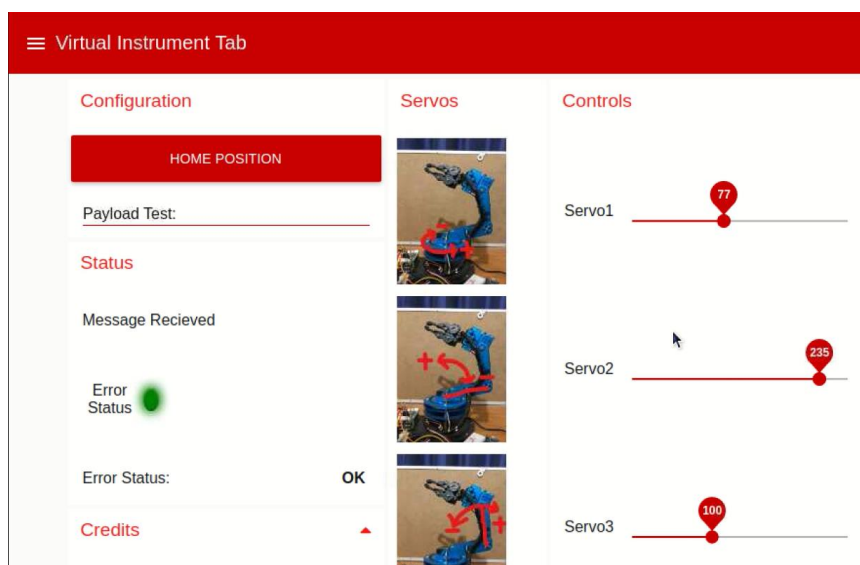


Figura 35. Asignando valores a los primeros 3 servomotores.

```

pi@raspberrypi: ~/Documents/estancia
File Edit Tabs Help
-----NEW REQUEST-----
Access Granted: david
Client mosqpub|2673-raspberrypi sending CONNECT
Client mosqpub|2673-raspberrypi received CONNACK (0)
Client mosqpub|2673-raspberrypi sending PUBLISH (d0, q0, r0, m1, 'ERROR', ... (2
bytes))
Client mosqpub|2673-raspberrypi sending DISCONNECT
Message is: PWM,26,100
Client mosqpub|2677-raspberrypi sending CONNECT
Client mosqpub|2677-raspberrypi received CONNACK (0)
Client mosqpub|2677-raspberrypi sending PUBLISH (d0, q0, r0, m1, 'MCU', ... (10 b
ytes))
Client mosqpub|2677-raspberrypi sending DISCONNECT
Information of Connection:
I
Topic: MCU
CMD: PWM

```

Figura 36. Ventana terminal del proceso de Python, mostrando un nuevo evento, se permitió el acceso a el usuario "david".

```
pi@raspberrypi: ~/Documents/estancia
File Edit Tabs Help
-----NEW REQUEST-----
Access Granted: david
Client mosqpub|2683-raspberrypi sending CONNECT
Client mosqpub|2683-raspberrypi received CONNACK (0)
Client mosqpub|2683-raspberrypi sending PUBLISH (d0, q0, r0, m1, 'ERROR', ... (2
bytes))
Client mosqpub|2683-raspberrypi sending DISCONNECT
Message is: PWM,14,170
Client mosqpub|2687-raspberrypi sending CONNECT
Client mosqpub|2687-raspberrypi received CONNACK (0)
Client mosqpub|2687-raspberrypi sending PUBLISH (d0, q0, r0, m1, 'MCU', ... (10 b
ytes))
Client mosqpub|2687-raspberrypi sending DISCONNECT

Information of Connection:
Topic: MCU
CMD: PWM
PIN: 14
Value: 170
```

Figura 37. Ventana terminal mostrando información acerca de la última conexión.



Figura 38. Nombre de usuario alterado (no registrado)

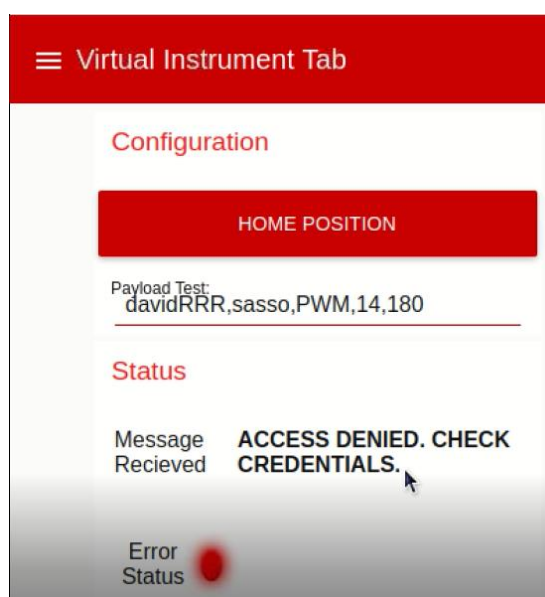


Figura 39. Error causado por credenciales incorrectas.

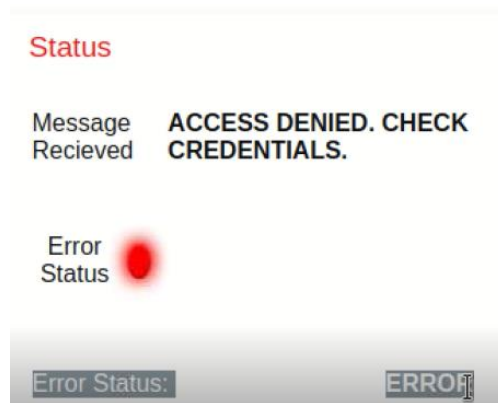


Figura 40. Estado de Error: Error

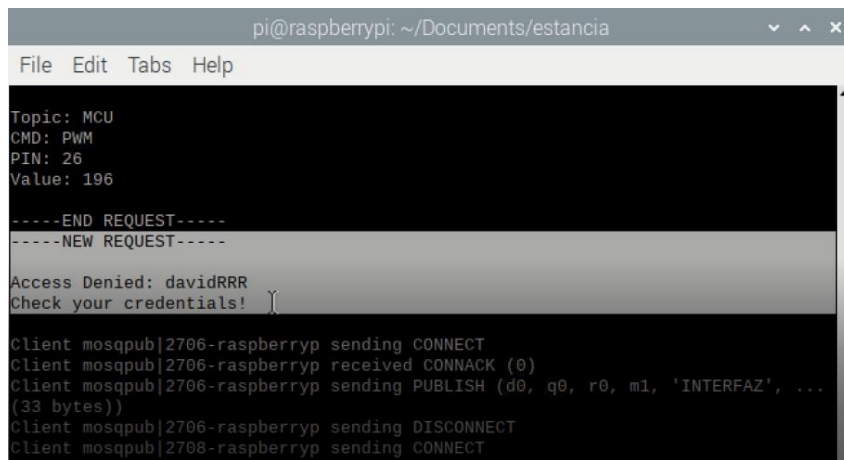


Figura 41. Solicitud de acceso denegada al usuario: davidRRR

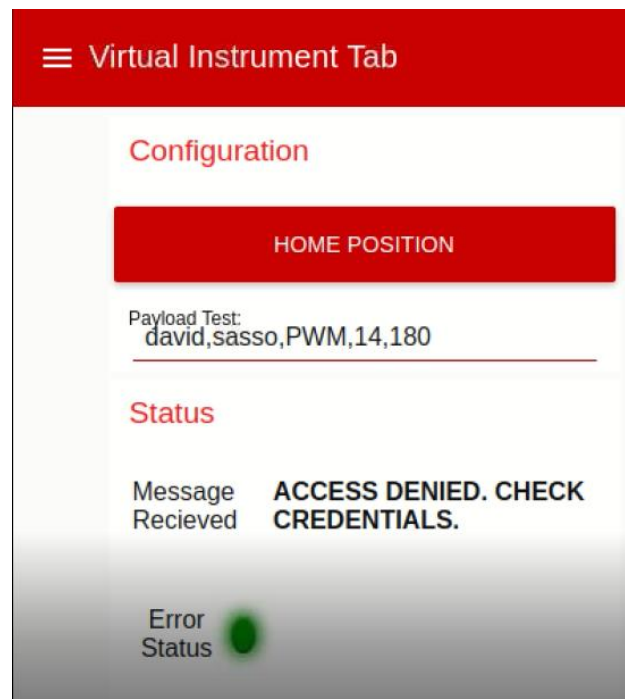
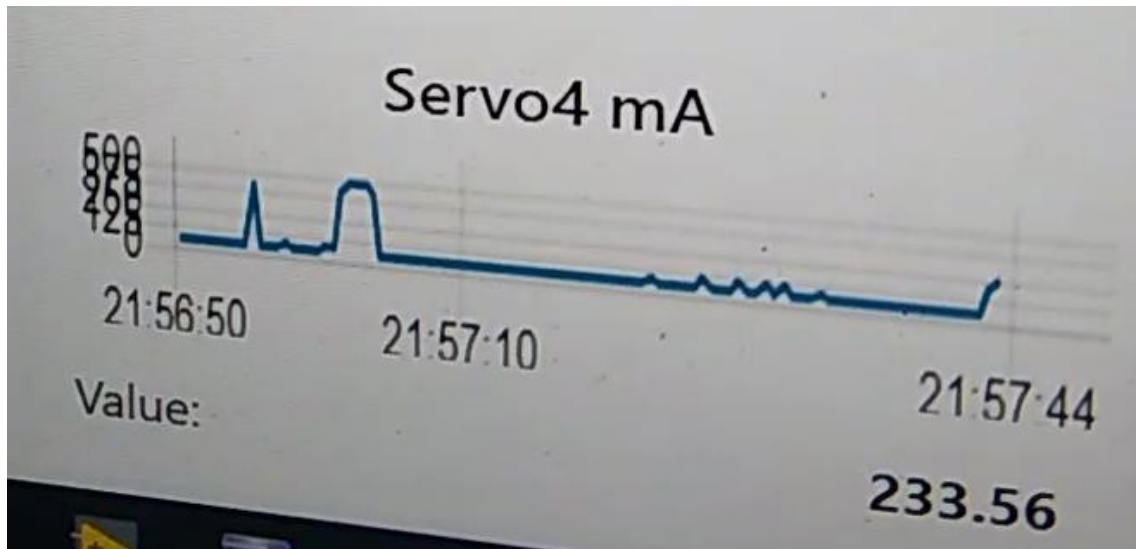


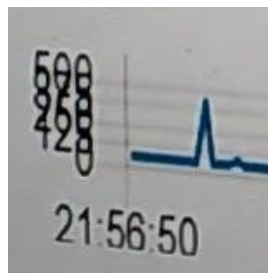
Figura 42. Se vuelve a colocar un usuario correcto, el acceso se permite.

13. Para la parte de los sensores, la señal adquirida es procesada y enviada a través del protocolo MQTT a la interfaz, la cual muestra su valor en una gráfica.



*Figura 43. Señal adquirida graficada contra el tiempo donde se observa el consumo bajo bloqueo (izquierda), reposo (centro) y nominal (derecha).*

Como se puede observar en la figura anterior, el servomotor identificado como el numero cuatro fue movido, teniendo al inicio de la gráfica un consumo aproximadamente de 32mA cuando se encuentra en reposo. En el primer pico se le asignó un ángulo destino desde la interfaz, no muy alejado del actual, es por eso, que la corriente sube hasta aproximadamente 400 mA y desciende.



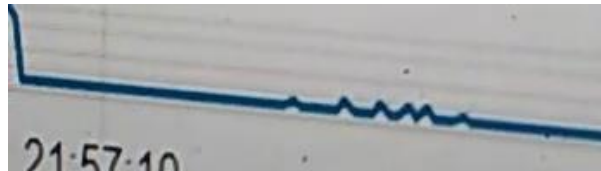
*Figura 44. Pico de corriente por movimiento del servomotor.*

En el segundo pico se le fue asignado un valor de ángulo destino considerablemente alejado del actual para graficar este consumo de corriente de aproximadamente 500 mA, es por eso, que se forma una curva y se aprecia que el consumo de corriente en la parte superior de la curva se mantiene por un tiempo más prolongado.



*Figura 45. Corriente consumida al mover el servomotor.*

Por último, las pequeñas variaciones que se aprecian en la parte derecha representan esfuerzos aplicados manualmente al servomotor, intentando alterar su posición.



*Figura 46. Corriente consumida al realizar esfuerzo manual al servomotor.*

## **VI. Conclusiones**

Con todo lo mencionado anteriormente en este documento se demuestra que fue posible diseñar el sistema, implementarlo y así cumplir con los objetivos establecidos. Se esperaba tener la posibilidad de operación remota de parte de un usuario y aún y dejando de lado el servicio de escritorio remoto, sigue siendo una operación remota al no estar físicamente conectada la computadora al microcontrolador. Hubo una reestructuración en la cuestión de seguridad, que fue la añadidura de credenciales de usuario. Gracias a que la estructura estaba realizada de manera modular, se pudo llevar a cabo de manera sencilla la implementación de este punto.

Personalmente comprobé que el analizar el problema en cuestión y realizar una lista de los requerimientos y características limitantes ayuda al diseño del sistema, y al tener un buen diseño con una buena arquitectura, la implementación es mucho más rápida de llevar a cabo.

Es importante poder identificar herramientas que nos ayuden a cumplir los objetivos propuestos, para realizar de manera más rápida las actividades. Para finalizar el desarrollo de este tipo de proyectos debe cumplir con el acrónimo SMore (scalable, modular, reusable, extensible, simple) para en un futuro si la situación lo requiere llevar a cabo cambios significativos de una manera efectiva y eficaz.

## VII. Referencias Bibliográficas

- [1] – Borrero Serrano, Maria España, (2011), Herramienta Software para el Control Remoto de una Fuente de Alimentación Mediante una Interfaz Gráfica., Sevilla España.
- [2] – Muñoz, Ivan Pulido, (2017), Control Remoto de Instrumentación para Radiocomunicación mediante Python., Sevilla España.
- [3] – Antoni Mànuel, Domingo Biel. Joaquim Olivé, Jordi Prat. Francesc J. Sánchez., (2002), Instrumentación Virtual Adquisición, Procesado y Análisis de Señales., 2ed., Catalunya Barcelona, UPC.
- [4]-- Jesús A. Calderón-Vielma,(2002), Instrumentación Virtual, Bogotá Colombia.
- [5] -- El Hammoumi et al. Renewables, (2018), Low-cost virtual instrumentation of PV panel characteristics using Excel and Arduino in comparison with traditional instrumentation., Springer.