



UNIVERSIDAD SANTO TOMÁS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA

T U N J A

TÍTULO PRÁCTICA

Instrumentación Electrónica e Implementación de Software sobre Máquina de Oxidación de Materiales

PROPONENTE

Ing. Est. Luis Felipe Narvaez Gomez
C.c: 1.049.652.438
Código estudiantil: 2312660

TUTOR

Ing. Sebastián Ibague Martinez

Tunja

Fecha de presentación (03, 12, 2022)

CONTENIDO

1. <u>FICHA TÉCNICA DE LA PRÁCTICA.....</u>	3
2. <u>PLANTEAMIENTO DEL PROBLEMA</u>	4
3. <u>OBJETIVOS.....</u>	4
4. <u>DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS</u>	5
4.1. ACTIVIDADES REALIZADAS EN EL PROYECTO.....	5
4.2. HERRAMIENTAS UTILIZADAS EN EL DESARROLLO DEL PROYECTO	58
5. <u>CONCLUSIONES.....</u>	59

1. FICHA TÉCNICA DE LA PRÁCTICA

Título	Instrumentación Electrónica e Implementación de Software sobre Máquina de Oxidación de Materiales
Nombre Estudiante	Luis Felipe Narvaez Gomez
Documento estudiante	1049652438
Correo electrónico	luis.narvaez@usantoto.edu.co
Entidad o sector de la empresa	Ing. Sebastián Ibagué Martínez
Lugar de ejecución de la práctica	Tunja, Boyacá
Duración	192 h - 0.263013 meses
Los abajo firmantes confirman que todos los datos incluidos en la presente propuesta son correctos y verídicos, que no incumplen ninguna ley o norma vigente. (Incluir nombres y firmas de estudiantes, director y tutor).	

2. PLANTEAMIENTO DEL PROBLEMA

La Facultad de Ingeniería Mecánica de la Universidad Santo Tomás Tunja posee el armazón compuesto de una máquina de oxidación de materiales que han mandado elaborar con anterioridad a terceros. Para su correcto funcionamiento se requiere de la adquisición de diferentes dispositivos y materiales que puedan dar uso a este armazón y en su conjunto hagan del instrumento una central que usar por parte de estudiantes y personal administrativo e la facultad para la resolución de proyectos e inicio de investigación de ciencias de materiales. A su vez para que todas estas piezas funcionen se requiere de la instrumentación electrónica y el despliegue de un software robusto que pueda interactuar con los diferentes modelos físicos de manera fiable con el fin de monitorear y controlar los procesos de oxidación de los elementos que se dispongan en la maquina a lo largo del tiempo.

Por estos motivos la Facultad de Ingeniería Mecánica en compañía con el Ingeniero contratado para el apartado de adquisición de materiales de Instrumentación general del equipo, Ing. Sebastián Ibagué Martínez solicitan a la Facultad de Ingeniería de Sistemas la intervención de uno de sus estudiantes para que en modalidad de Práctica empresarial inicie con la adecuación de modelos iniciales de software que permita la interacción de los modelos físicos esto reservado y limitado a un nivel propio de Práctica Empresarial, teniendo de esta manera una base sólida la cual poder escalar en el proceso de construcción y adecuación hasta de producto final de la Maquina de Oxidación de Materiales.

3. OBJETIVOS

1. Adecuar y configurar el sistema operativo y herramientas de software de uso en la Raspberry PI 400.
2. Detectar de posibles problemas físicos de conexión del hardware de la Raspberry PI 400 con lo codificado en los módulos de software.
3. Diseñar y construir PCB electrónicos que sean necesarias para la comunicación plena entre la Raspberry 400 y los demás sistemas físicos de la Maquina de oxidación de materiales.
4. Apoyar en el apartado de instrumentación electrónica y física de los dispositivos que necesita la Maquina de oxidación de materiales.
5. Diseñar uno o varios módulos de software que permitan la commutación eléctrica y electrónica de los dispositivos físicos de la Maquina de Oxidación de materiales haciendo provecho del ordenador Raspberry PI 400 y tarjetas electrónicas.
6. Diseñar uno o varios módulos de software que permitan la lectura de sensores analógicos y digital por parte de la Raspberry PI 400.
7. Diseñar uno o varios módulos externos de software sobre tarjetas electrónicas de apoyo en caso de ser necesarias para la adquisición de datos.

8. Realizar una conexión física y digital en caso de ser necesaria entre los diferentes módulos de hardware y software con la finalidad e que funcionen en sincronía al momento de interactuar con la máquina de oxidación de materiales.
9. Solucionar posibles malfuncionamientos de módulos de software previamente hechos y/o librerías que impidan el comportamiento adecuado de la toma de datos y conmutación a partir del puerto GPIO del a Raspberry PI 400.
10. Proponer los primeros módulos de software o uso de herramientas de terceros que permitan el uso de Dashboard para facilitar la accesibilidad y modo de interactuar del usuario con la máquina de oxidación de materiales.
11. Formular modelos de escalabilidad de los modelos de software y hardware propuestos en la practica empresarial para el pleno funcionamiento de la máquina de oxidación de materiales.

4. DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS

4.1. Actividades realizadas en el proyecto

En la presente practica empresarial se llevaron a cabalidad diferentes actividades que engloban el proceso de construcción e instrumentación electrónica sobre la máquina de oxidación de materiales, así como otras actividades que también engloban tareas relacionadas con el diseño, implementación y despliegue de software que permite la manipulación y programación secuenciada de algoritmos de trabajo sobre la misma máquina.

En esta practica empresarial La Facultad de Ingeniería Mecánica proporciono el armazón para la creación de una máquina que consiga el envejecimiento forzoso de la materia mediante luces infrarrojas, luces ultravioletas y la preparación por gas de un ambiente corrosivo.

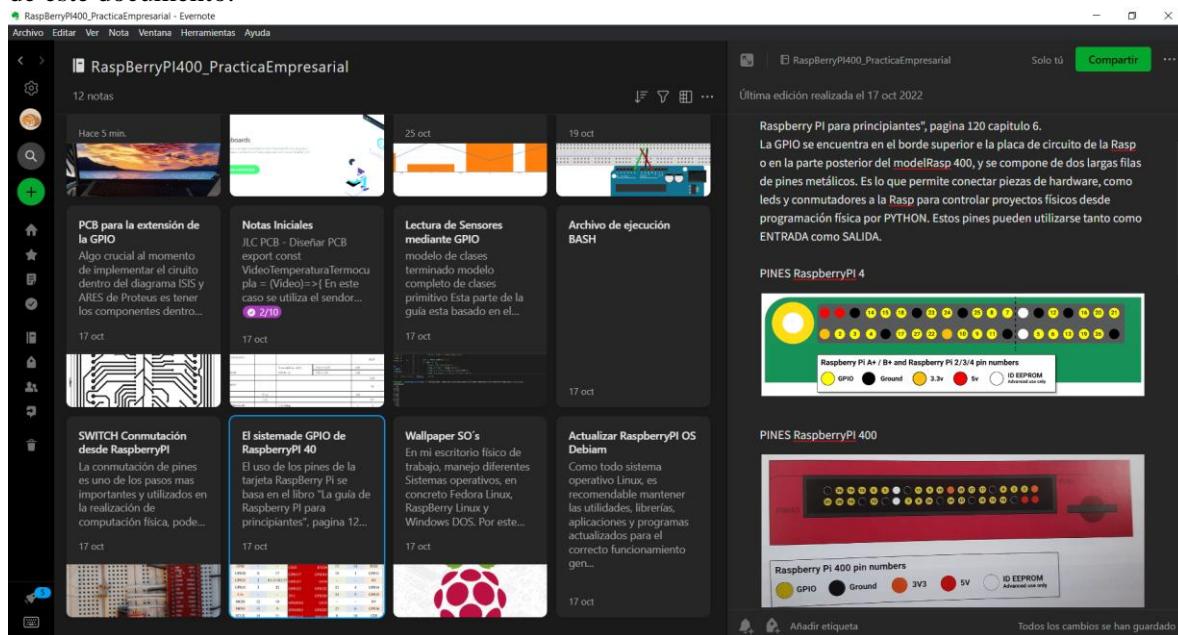
Con esta máquina se permitiría oxidar materiales de laboratorio para las practicas y trabajos de los estudiantes e Ingenieros de esta facultad y la Universidad.

Es aquí donde los Ingenieros ING. NELSON IVAN VILLAMIZAR CRUZ y ING. JUAN RODRIGO SALAMANCA SARMIENTO solicitaron el armado, instrumentación electrónica y desarrollo de software para lograr que esta maquina empiece a trabajar bajo el propósito de la oxidación de materiales.

En orden de pasante por practica empresarial de la Facultad de Ingeniería de Sistemas, el Ing. Luis Felipe Narvaez Gomez, es centrado para diseñar los diferentes módulos de software necesarios para la máquina de oxidación de materiales, con el propósito de que los mismos permitan a los diferentes usuarios interactuar y monitorear el proceso de corrosión. Los módulos de software a presentar son en versión de prueba y se le comunica al pasante sobre las limitaciones, requerimientos y algunas ideas extras de implementación que pueden trabajarse durante la ejecución de la práctica. De la misma manera, en caso de ser necesario, el pasante debido a que posee ya un titulo profesional en Ingeniería Electrónica, puede ser solicitado por el contratante para servir en el apoyo de diseño, soldado, creación y preparación de circuitos electrónicos, así como en el apartado de instrumentación electrónica sobre la maquina en tema.

Cabe mencionar que el pasante no tiene como tarea principal la elaboración de circuitos electrónicos si no la de los módulos de software, esto debido a la naturaleza de practica empresarial dada dentro de la facultad de Ingeniería de Sistemas.

Para llevar una bitácora de la realización de actividades se le recomienda al pasante tomar apuntes en algún medio digital que lo permita. Este software fue Evernote el cual permite tomar apuntes simultáneos en varios dispositivos con cuadros de código donde especificar el avance del trabajo realizado durante la practica empresarial y de estos archivos es que se extractara para la realización de este documento.



A continuación, en el presente documento se explicará las distintas actividades que el pasante de practica empresarial logro desarrollar dentro de la empresa:

4.1.1 Instalación y configuración de la ISO Raspberry PI Debian.

En esta actividad el pasante se le otorgo a su poder el modelo Raspberry PI 400, una memoria SD de 32GB, un cargador que hace de alimentación para la Raspberry PI 400 y un conector de video MicroHDMI/HDMI con la finalidad de que instale el sistema operativo Raspberry PI Debian en el dispositivo de computo y realice el software de practica en el (Este es un ambiente de desarrollo Linux de software y programación digital y programación física).



La instalación del Sistema operativo Raspberry PI Debian debe realizarse sobre una SD, en este caso una SanDisk de 10 generación de 32GB de almacenamiento (otros modelos de SD pueden presentar conflictos con la lectura de la Raspberry).

La Instalación de este sistema operativo se realizará siguiendo los pasos descritos en el libro "La Guías oficial de Raspberry para Principiantes - Como usar tu nuevo ordenador", en la página 214. Cabe mencionar que realmente la Raspberry PI 400 (Computadora de Raspberry donde se está ejecutando todo este trabajo) puede recibir cualquier Linux que se adapte a la especificaciones de Hardware, sin embargo se recomienda trabajar con la derivación de Debian para Raspberry ya que es un ambiente desarrollado específicamente para este dispositivo que sirve para el desarrollo y aprendizaje de sistemas informáticos y sistemas físicos puesto que ya posee la configuración adecuada para la lectura de los puerto de entrada, salida y alimentaciones de la computadora.

Parte de la documentación revisada no solo deriva del libro ya mencionado si también de registros explicados en las redes de difusión de la empresa Raspberry como las siguientes:

1. Twiter del autor de Raspberry: <https://twitter.com/ghalfacree>
2. Twiter oficial de Raspberry: https://twitter.com/Raspberry_Pi
3. Página Oficial de Raspberry: <https://www.raspberrypi.com/software/>

Lo primero es dirigirnos a la página principal de Raspberry para descargar la imagen o el generador de la imagen en la SD. En el enlace de la página oficial de Raspberry se encontrará el software de Raspberry Image, un programa para Windows SO el cual sirve para poder montar el SO de RaspberryPi Debian en cualquier unidad de almacenamiento. Seleccionar el botón "Descargar para Windows".

Instale el sistema operativo Raspberry Pi con Raspberry Pi Imager

Raspberry Pi Imager es la forma rápida y fácil de instalar el sistema operativo Raspberry Pi y otros sistemas operativos en una tarjeta microSD, lista para usar con su Raspberry Pi. [Mire nuestro video de 45 segundos](#) para aprender cómo instalar un sistema operativo usando Raspberry Pi Imager.

Descargue e instale Raspberry Pi Imager en una computadora con un lector de tarjetas SD. Coloque la tarjeta SD que usará con su Raspberry Pi en el lector y ejecute Raspberry Pi Imager.

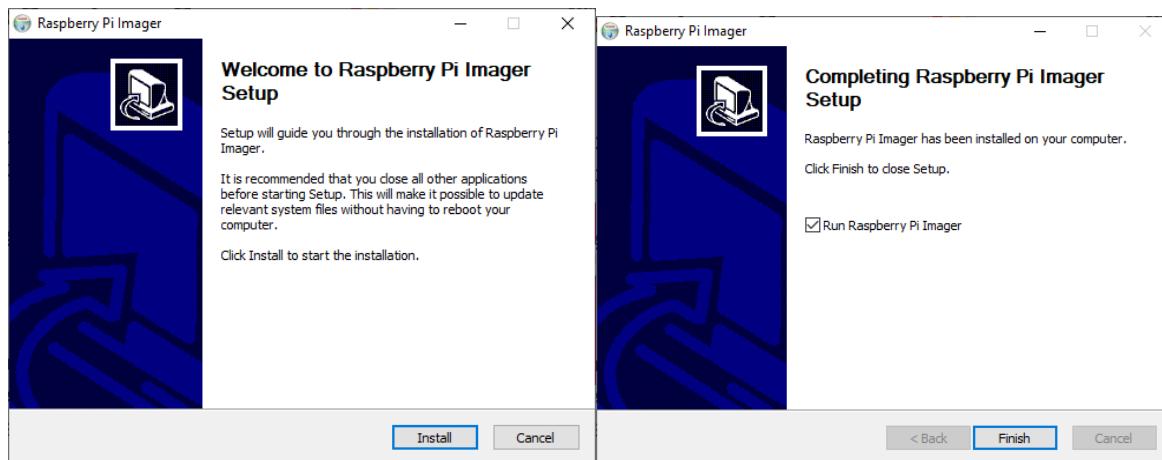
[Descargar para Windows](#)

[Descargar para macOS](#)

[Descargar para Ubuntu para x86](#)



El proceso de Instalación de Software es el siguiente:

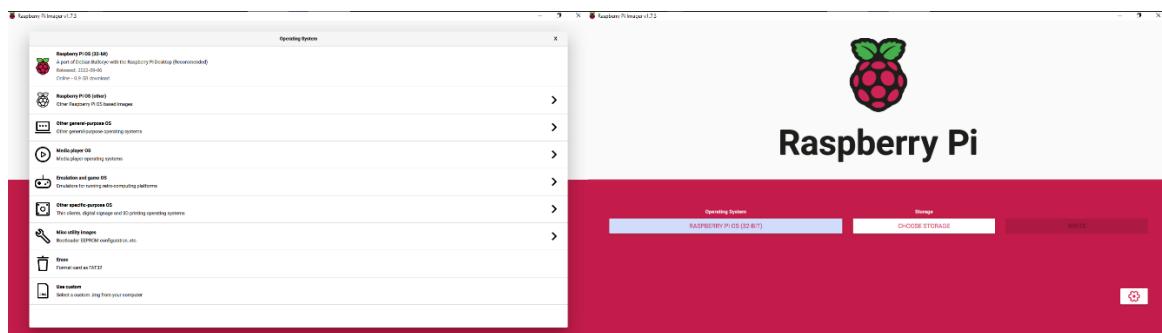


Una vez instalado el software se requiere Conectar la Tarjeta SD de 32GB SanDisk 10 Gen, parece que ya tenía instalado Raspberry en el Recovery. Se decide eliminar esta versión del SO para evitar problemas de lectura y configuración que puedan mal lograr la configuración del Hardware permanentemente.

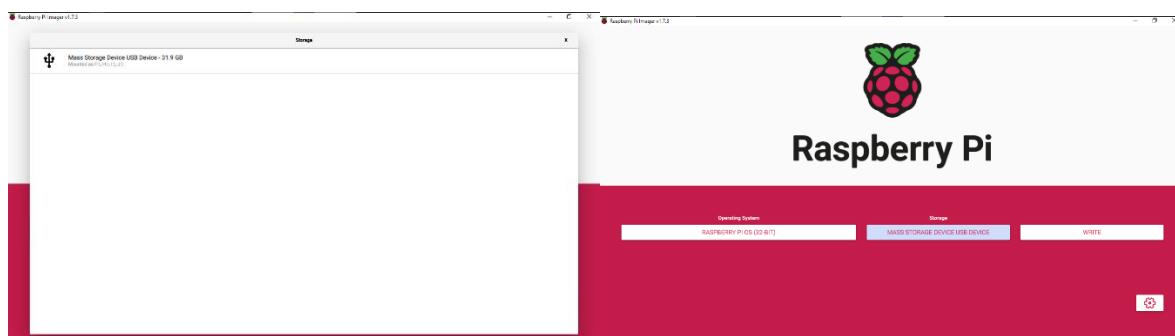
Para instalar la ISO en la memoria se mostrarán los pasos en las siguientes imágenes:



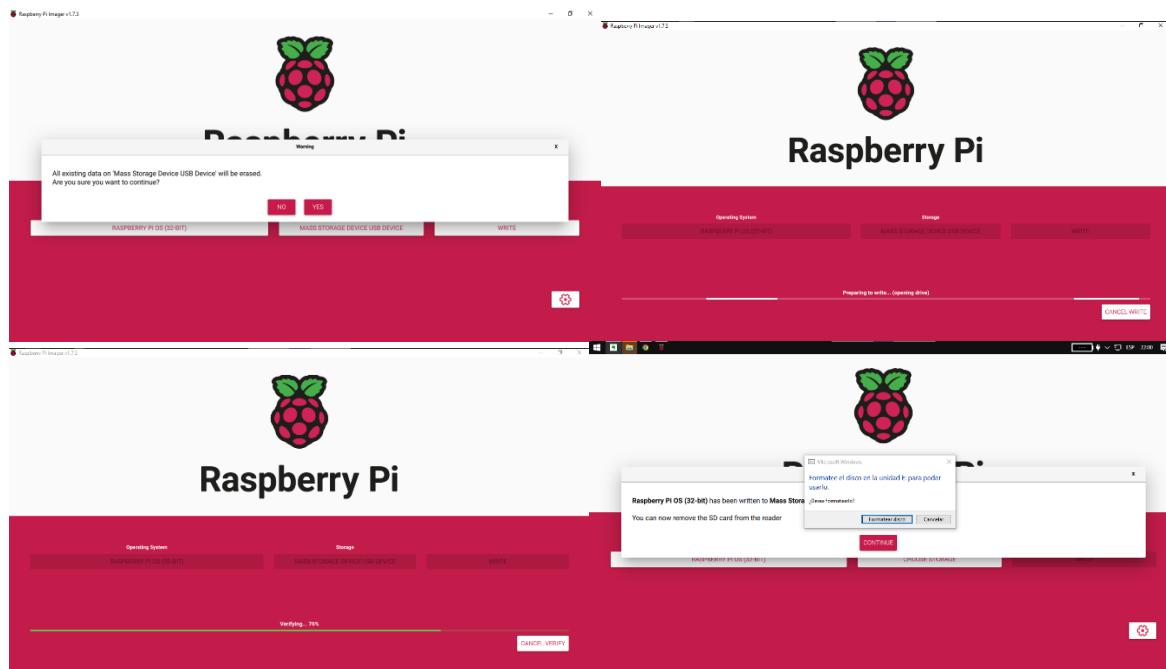
Seleccionamos la opción Choose OS para seleccionar que versión o herramienta del SO queremos instalar. Seleccionamos Raspberry PI 08(32-bit).



Se desconectan todas las USB externas para solo tener la SD y evitar problemas. Clic en Choose Storage. Aquí seleccionamos la memoria Sd insertada.



Una vez seleccionada la opción de la SO a instalar y donde se va a instalar; confirmar una conexión estable de la SD y dar clic en la opción WRITE.



Una vez instalado el SO en la SD la extraigo y la conecto físicamente a la Raspberry 400 por su ranura trasera. Cuando se encienda, la BIOS de la computadora leerá el SO en la SD y guiará a iniciar el sistema por ese lado, en caso de no tenerla se quedará en el Kernel de la BIOS.

Una vez se prende la Raspberry PI 400 accederemos al menú de instalación del SO Debian. La instalación la realice en Offline, No agrego WIFI porque lo trabajare con entrada de Ethernet en los procesos posteriores y por este momento de inicio sé que tengo la versión más actual de Debian Raspberry en la Tarjeta SD por lo que no hay que actualizar nada.

Mas adelante si se descargaran paquetes software y librerías necesarias para el proyecto si se recomienda mantener actualizados los paquetes de configuración de aplicaciones con los siguientes comandos:

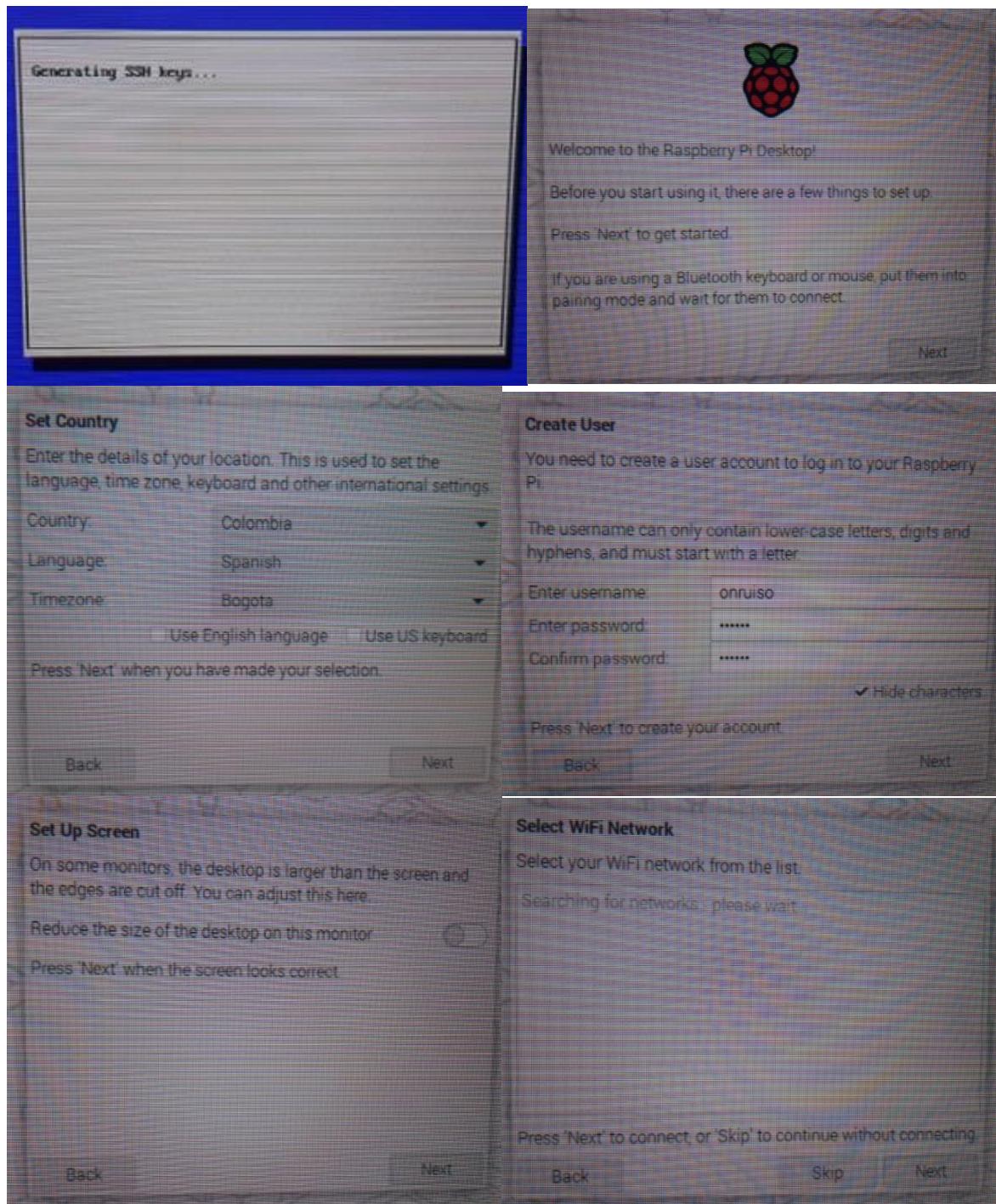
Primer comando:

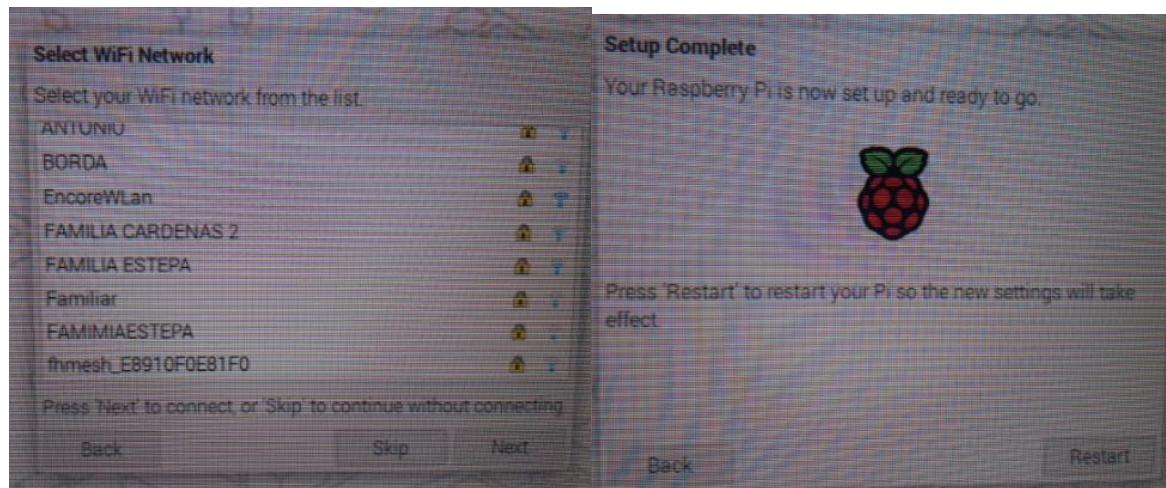
```
sudo apt-get update -y
```

Segundo comando:

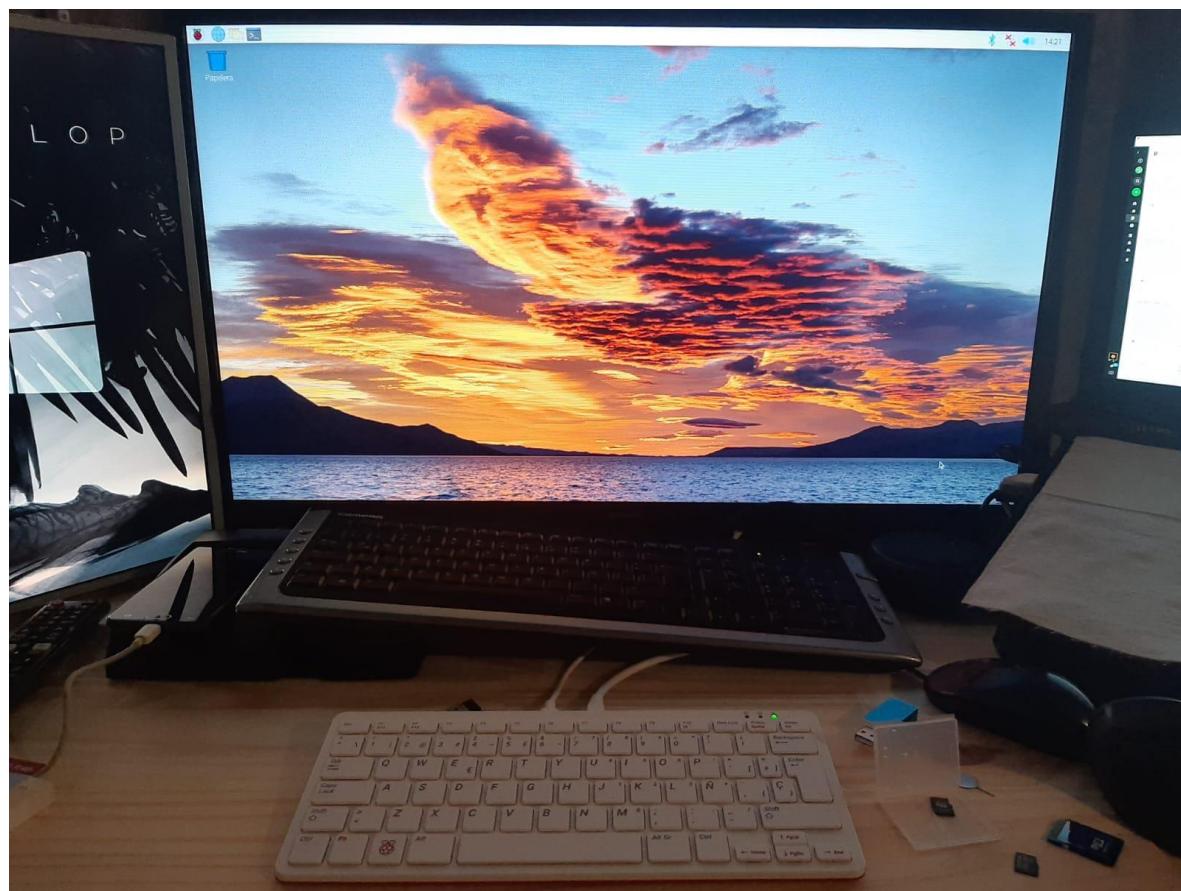
```
sudo apt-get upgrade -y
```

Este proceso es similar entre las dos Raspberries PI 400.





Al finalizar la instalación del sistema operativo, reiniciaremos la Raspberry. Al reiniciarse tendremos nuestro nuevo ambiente de Debian.



4.1.2 Actualizar Raspberry PI OS Debian.

Como todo sistema operativo Linux, es recomendable mantener las utilidades, librerías, aplicaciones y programas actualizados para el correcto funcionamiento general del sistema mismo y los trabajos que corren en él.

Sin embargo, hay ciertos proyectos que funcionan en versiones específicas de programas por lo que actualizar en general puede llegar a ser contra producente, por lo que se recomienda observar estos casos especiales.

El primer comando a utilizar es:

```
sudo apt-get update -y
```

El segundo comando a utilizar es:

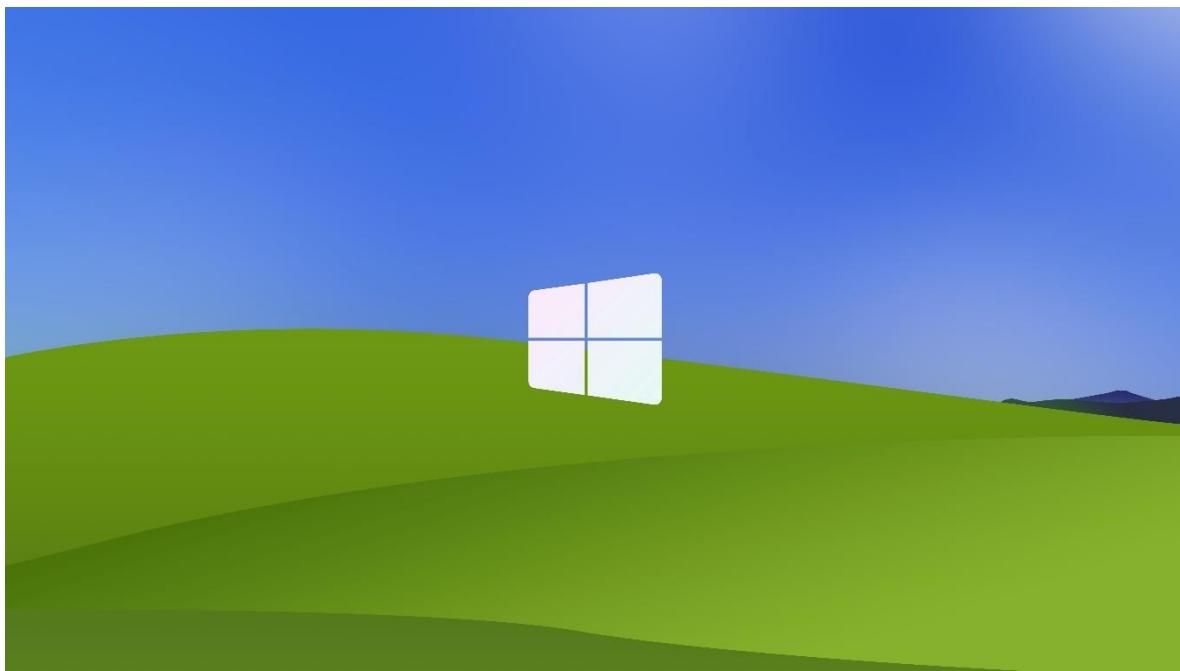
```
sudo apt-get upgrade -y
```

4.1.3 Preparación de Entorno de Trabajo entre Windows y Linux u otros dispositivos requeridos.

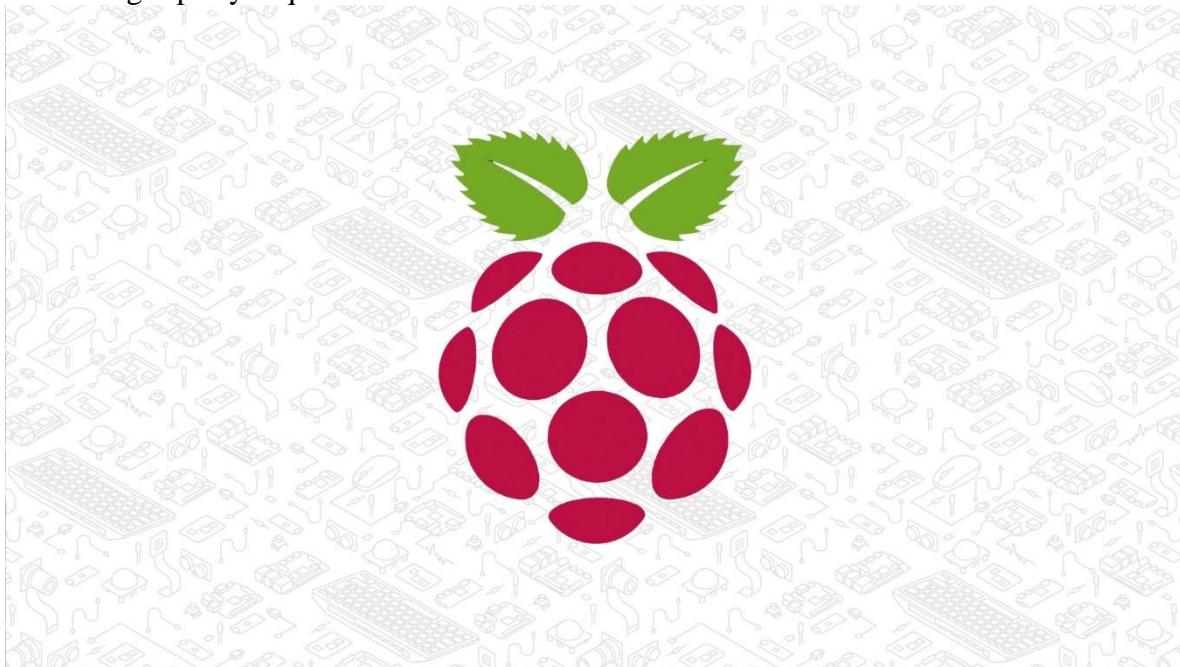
En mi escritorio físico de trabajo, manejo diferentes Sistemas operativos, en concreto Fedora Linux, Raspberry Linux y Windows DOS. Por este motivo y con la facilidad de manejo paralelo entre estos SOs, así como evitar confusiones que pueden mermar la productividad al momento de programar, testear y/o consultar información, yo cambio los Fondos de escritorio de cada máquina virtual con la finalidad de detectar visualmente en qué ambiente de trabajo estoy laborando en el momento de una tarea específica.

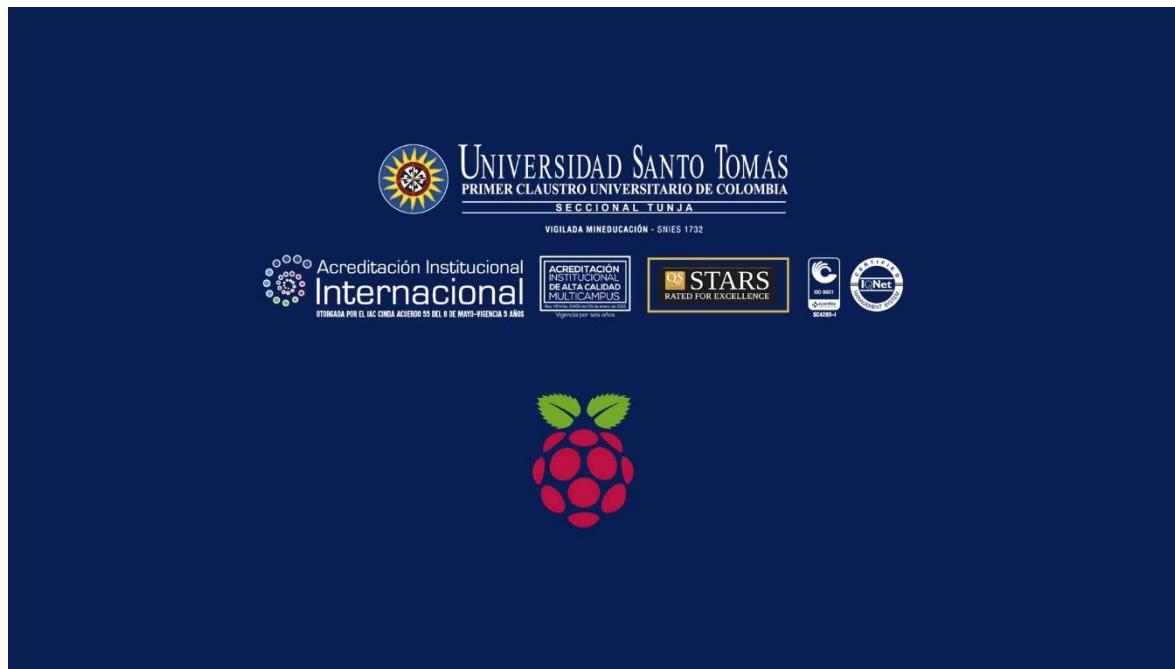
Por lo ya explicado, los wallpapers utilizados son:

1. Windows 10 Home Single Language



2. Raspberry PI Debian Linux. Se elaboro uno en FHD (1920*1080) y una en 4K de 16:9 (3840*2160). En la versión FHD se opto también por un modelo con los logotipos y esquemas de colores de la Universidad Santo Tomas.



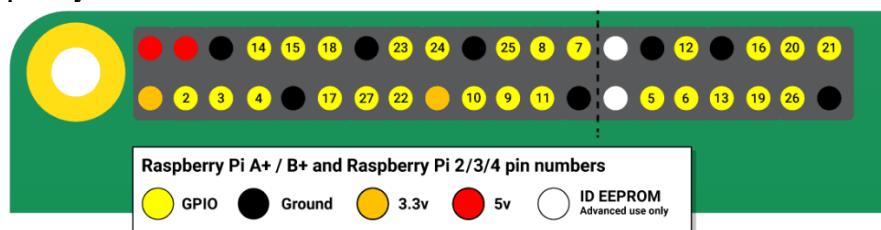


4.1.4 Sistema de lectura y acción GPIO de la RaspberryPI400: Comprensión y adecuación.

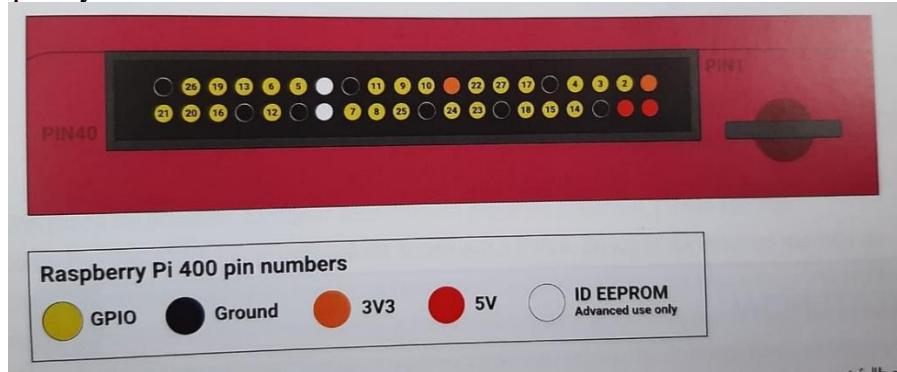
El uso de los pines de la tarjeta RaspBerry Pi se basa en el libro "La guía de Raspberry PI para principiantes", pagina 120 capitulo 6.

La GPIO se encuentra en el borde superior en la placa de circuito de la RaspBerry PI4, PI3 y PI3vB o en la parte posterior del modelo RaspBerry 400. Esta se compone de dos largas filas de pines metálicos en los que se permite conectar piezas de hardware comprendidas entre sensores y actuadores, como lo pueden ser leds y circuitos electrónicos de conmutación compleja o bien sensores analógicos o digitales. Estos pines son posibles de usar en la Raspberry para controlar y monitorear variables físicas del ambiente desde programación PYTHON u otros lenguajes que el desarrollador crea conveniente. De esta forma se puede decir que el apartado de GPIO es una canal de pines que pueden utilizarse tanto como ENTRADA y SALIDA.

PINES RaspberryPI 4.



PINES RaspberryPI 400



La conexión de la GPIO para su instrumentación electrónica en la máquina de oxidación de materiales no puede usarse directamente en la board, esto por problemas de conexión y individualización de los mismos pines. Para esto fue necesario implementar una extensión con board electrónica implementada que permitiera la extracción de los canales de conexión fuera del dispositivo central de cómputo Raspberry PI 400.

Name	wiringPi Pin	BCM GPIO	BCM GPIO	wiringPi Pin	Name
GPIO Extension Board					
3.3V	-	-	3V3	5V0	-
SDA	8	R1:0/R2:2	SDA1	5V0	-
SCL	9	R1:1/R2:3	SCL1	GND	-
GPIO7	7	4	GPIO4	TXD0	14
GND	RXD0	-	GND	RXD0	15
GPIO17	GPIO18	-	GPIO17	GPIO18	18
GPIO27	GPIO27	-	GPIO27	GND	-
GPIO22	GPIO22	R1:21/R2:27	GPIO22	GPIO23	23
3V3	GPIO24	-	3V3	GPIO24	24
SPIMOSI	GND	-	SPIMOSI	GND	-
SPIMISO	GPIO25	-	SPIMISO	GPIO25	25
SPISCLK	SPICE0	-	SPISCLK	SPICE0	8
GND	SPICE1	-	GND	SPICE1	7
ID_SD	ID_SD	-	ID_SD	ID_SD	1
0V	GPIO12	-	GND	SPICE1	11
ID_SDA	GND	-	ID_SDA	ID_SDA	31
GPIO05	GPIO05	-	GPIO05	GND	-
GPIO06	GPIO06	-	GPIO06	GPIO12	12
GPIO13	GPIO13	-	GPIO13	GND	-
GPIO19	GPIO18	-	GPIO19	GPIO16	16
GPIO26	GPIO20	-	GPIO23	GPIO12	26
GND	GPIO21	-	GPIO23	GND	-
		-	GPIO24	GPIO19	27
		-	GPIO25	GPIO26	28
		-	GND	GPIO20	0V
		-	GND	GPIO21	29
		-	GND	GPIO29	GPIO28

La GPIO esta conformada por distintos tipos de pines, cada uno para diferentes utilidades dentro del comportamiento físico y de comunicación que se quiera tener con la Raspberry PI 400, la función de estos pines se puede ver en la siguiente tabla:

3v3	3,3 voltios DC de potencia	Una fuente de energía de 3.3v DC encendida permanentemente, con el mismo voltaje que el de Raspberry PI utiliza internamente.
5v	5 voltios DC de potencia	Una fuente de energía de 5v DC encendida permanentemente, con el mismo voltaje que Raspberry PI toma del conector de alimentación micro USB.
Ground (GND)	0 voltios de potencia. Tierra del circuito.	Una conexión a tierra, usada para completar un circuito conectado a la fuente de alimentación.
GPIO XX	Pin "XX" entrada/salida de propósito general.	Los pines GPIO disponibles para tus programas, identificados por un número del 2 al 27.
ID EEPROM	Pines reservados para fines especiales.	Los pines se reservan para usarlos con la placa HAT y otros accesorios.

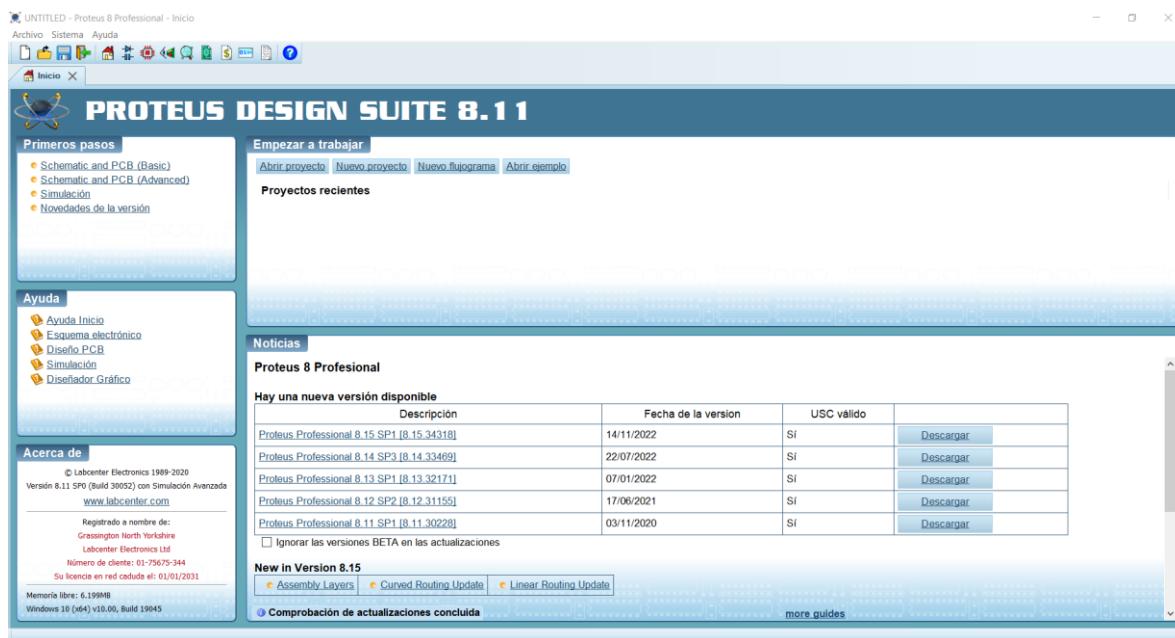
4.1.5 Diseño y Fabricación de Placa electrónica para la Board GPIO.

Debido a que la board GPIO que hace de extensión para la comunicación y conmutación de la Raspberry PI 400 esta diseñada para ser usada sobre protoboard, se necesita adecuar su uso sobre una placa electrónica mas robusta para la máquina de oxidación de materiales, este diseño y elaboración de la PCB electrónica queda a cargo del pasante Ing. Felipe Narvaez.

Para el diseño y fabricación de la PCB electrónica que necesitamos, se dio uso al software de diseño y simulación de componentes y laboratorios electrónicos Proteus Desing Suite Labcenter Electronics en su versión Professional 8.11 en idioma español.

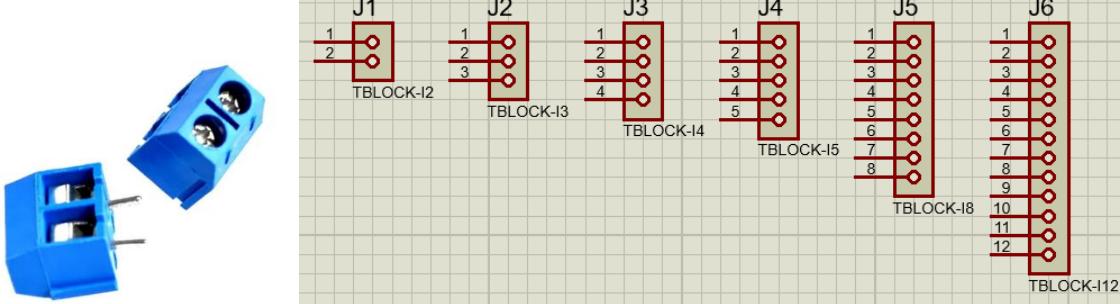
INFORME FINAL PRÁCTICA EMPRESARIAL

VERSIÓN: 1.0



Algo crucial al momento de implementar el circuito dentro del diagrama ISIS y ARES de Proteus es tener los componentes dentro de este software de simulación y armado de Boards electrónicas. Podemos utilizar componentes ya creados dentro de proteus para cablear nuestro circuito. Decidimos utilizar borneras típicas de conexión para asegurar un ajuste firme de cada cable a esta Board, esto aprovechando que las borneras poseen tornillos y un espacio suficiente para asegurar y conectar los cables que necesitemos.

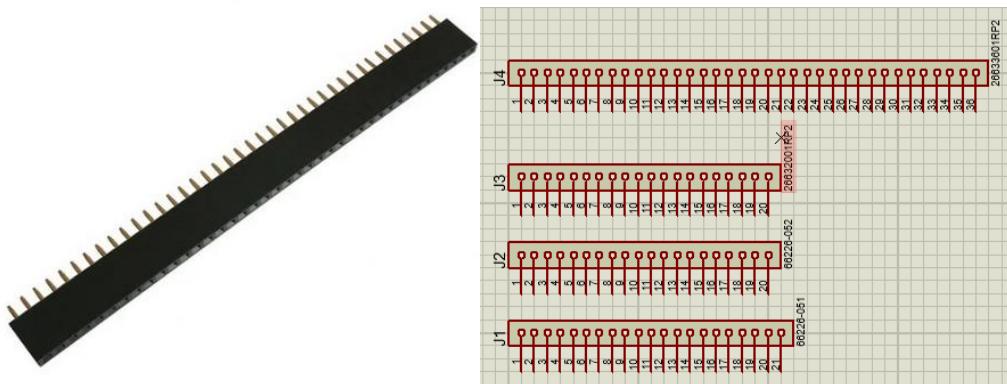
El componente de borneras luce como la imagen de la izquierda y se puede encontrar en proteus de diferentes números de pines, entre estos vamos a utilizar:



Otro componente que utilizaremos bastante son las resistencias eléctricas, estas son de 1/8 de Vatio pues no soportaran corrientes altas en el orden de los Amperios (A) y tampoco necesitamos que lo hagan pues las salidas y entradas a la GPIO no tienen estos valores eléctricos, solo corrientes bajas en el orden de los mili-Amperios (mA) máximo 1A. El componente se encuentra con el nombre de RES y luce así:

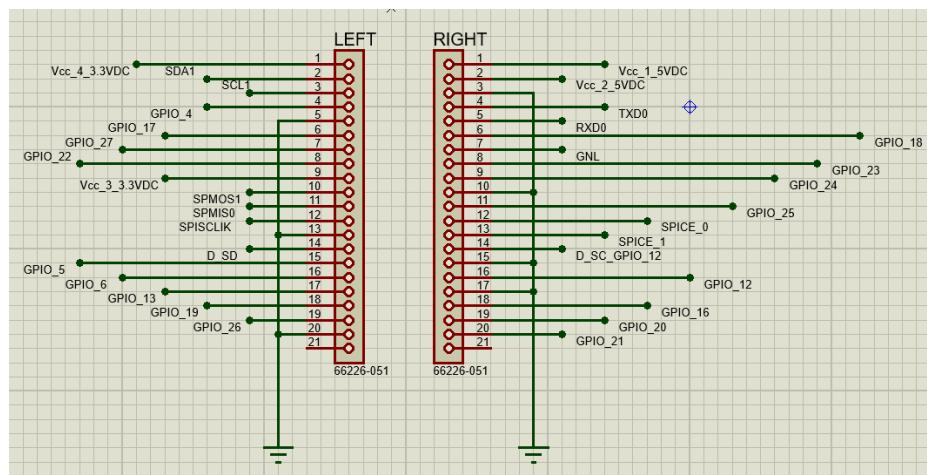


Ahora bien, para la conexión de la Extensión GPIO Board necesitamos una serie de regletas hembras que se acomoden a la separación entre pines propia de este componente, el problema, no existe algo ya consolidado por lo que podemos experimentar con los siguientes:

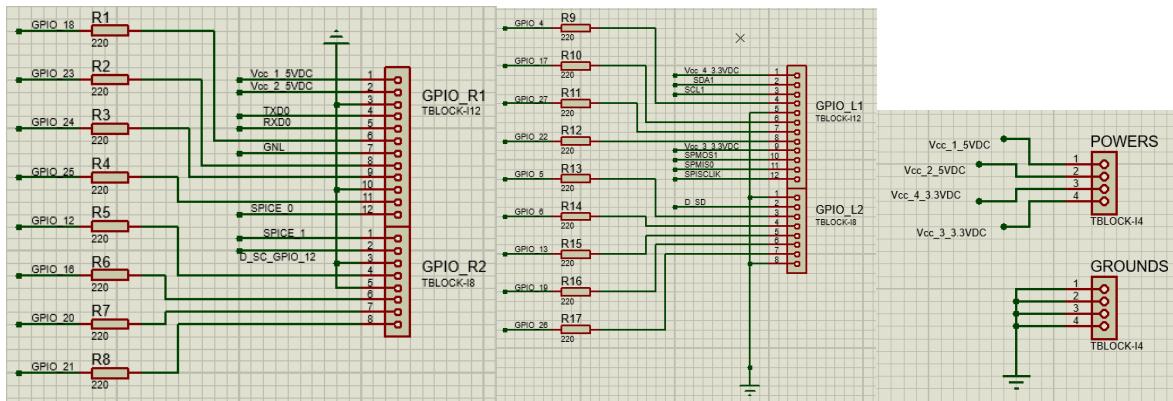


Utilizando los componentes vistos hasta el momento podemos diseñar todo nuestro circuito de la siguiente manera:

1. Regletas conexión directa con la Extensión GPIO Board.



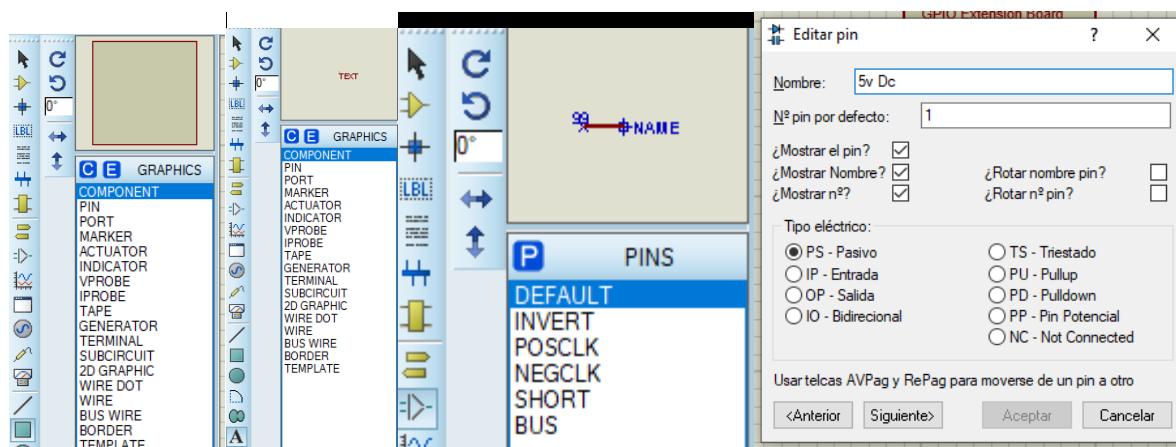
1. Borneras de conexión directa para el lado derecho de la extensión. Borneras de conexión directa para el lado izquierdo de la extensión. Borneras de alimentación.



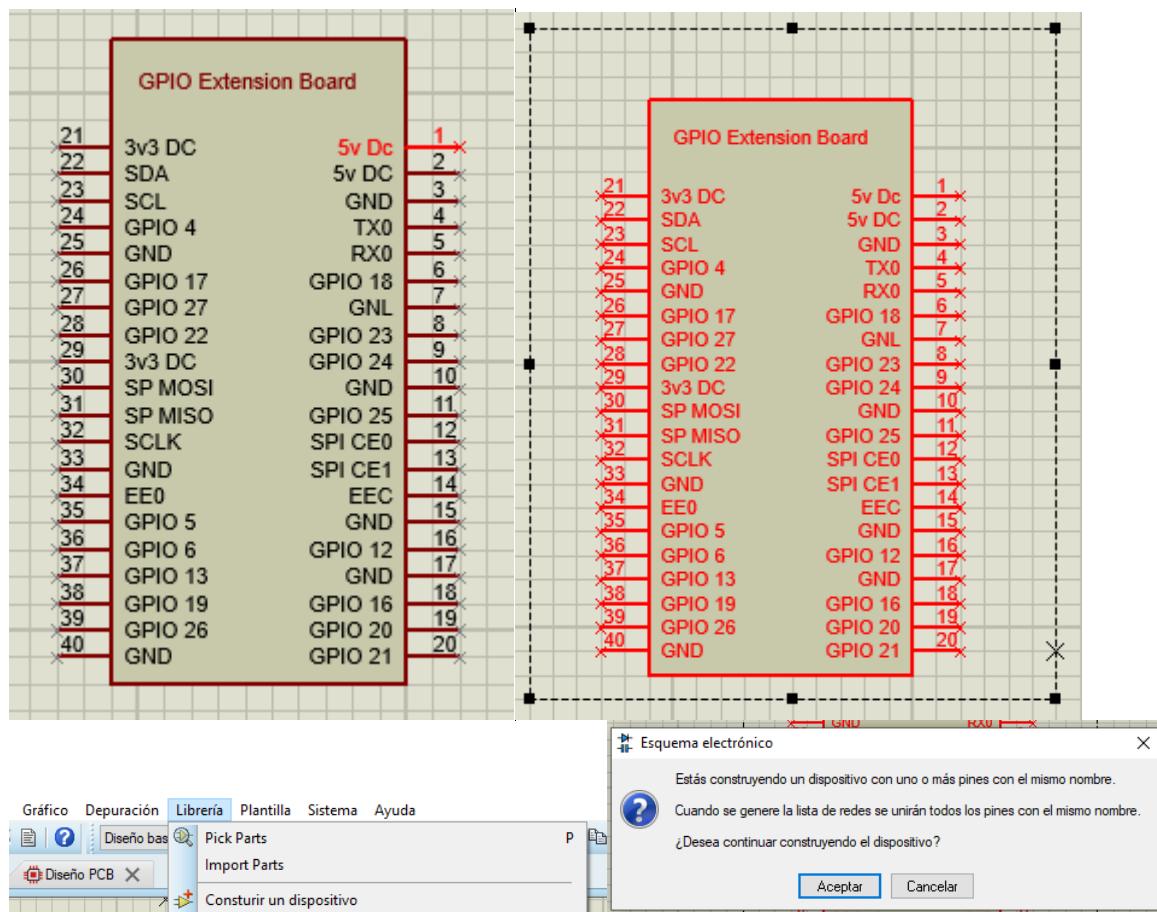
A este punto y revisando algunos problemas en el lado ARES del proyecto, nos damos cuenta que por más adaptaciones que se realicen a las regletas estas no cumplen con los parámetros que necesitamos para la conexión de la extensión, por lo que se resuelve con la idea de crear nuestro propio componente, tanto en ISIS como en ARES. Un componente que no encontramos y precisamos hacer es la propia extensión GPIO. Para esto creamos el componente con ayuda de las herramientas de la barra lateral del menú del software.



En la imagen se ve de forma horizontal pero realmente se encuentra de manera vertical. Para hacer el componente utilizamos las figuras de rectángulo o cuadrado de la barra lateral. Para hacer el título del componente seleccionamos la herramienta con el símbolo de "A". Para añadir los pines seleccionamos el símbolo de amplificador, allí seleccionamos default y con el creamos los pines. En este último paso debemos nombrar los pines y darles una numeración, esto se consigue dando doble clic sobre el pin que especificaremos como el siguiente:

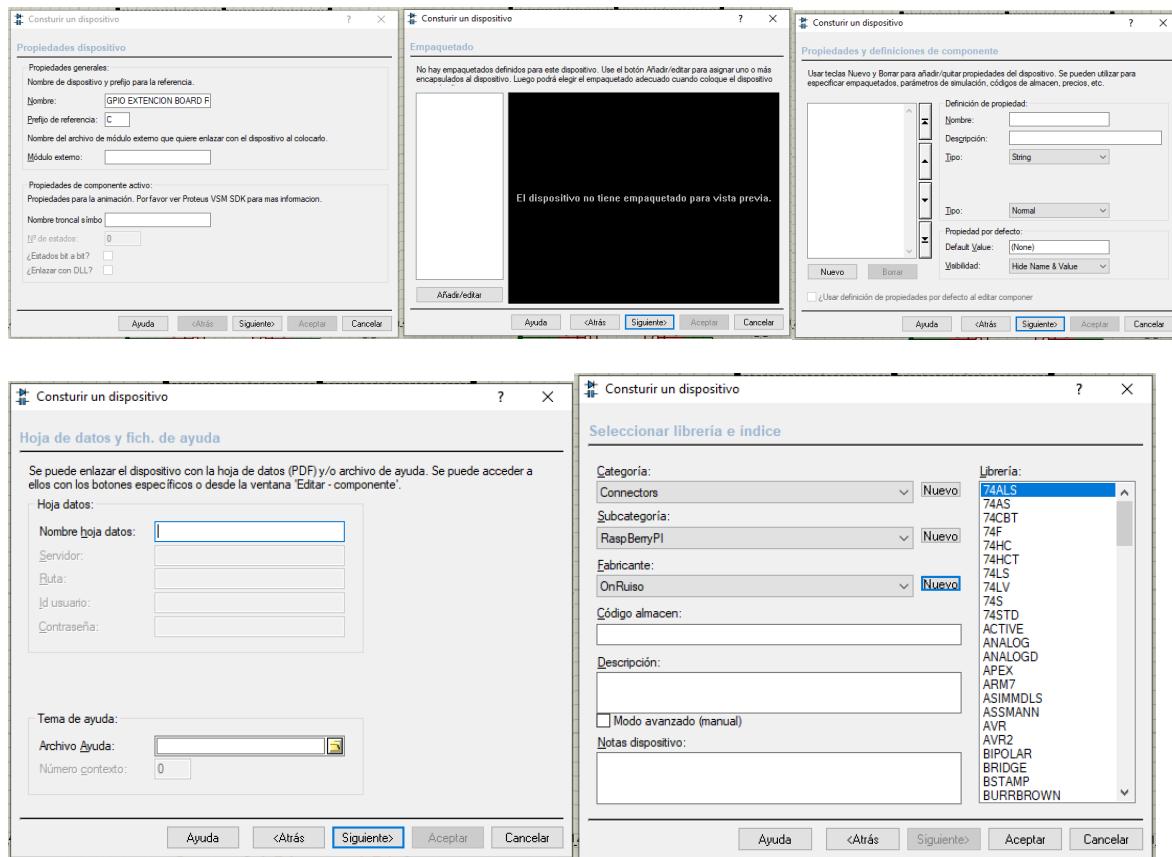


Una vez terminado tenemos todo el nuevo componente lucirá como el que se muestra a continuación. Ahora lo seleccionamos por completo, teniendo solo este elemento seleccionado vamos a la barra de menús de proteus y seleccionamos "Librería" y a continuación "construir componente".

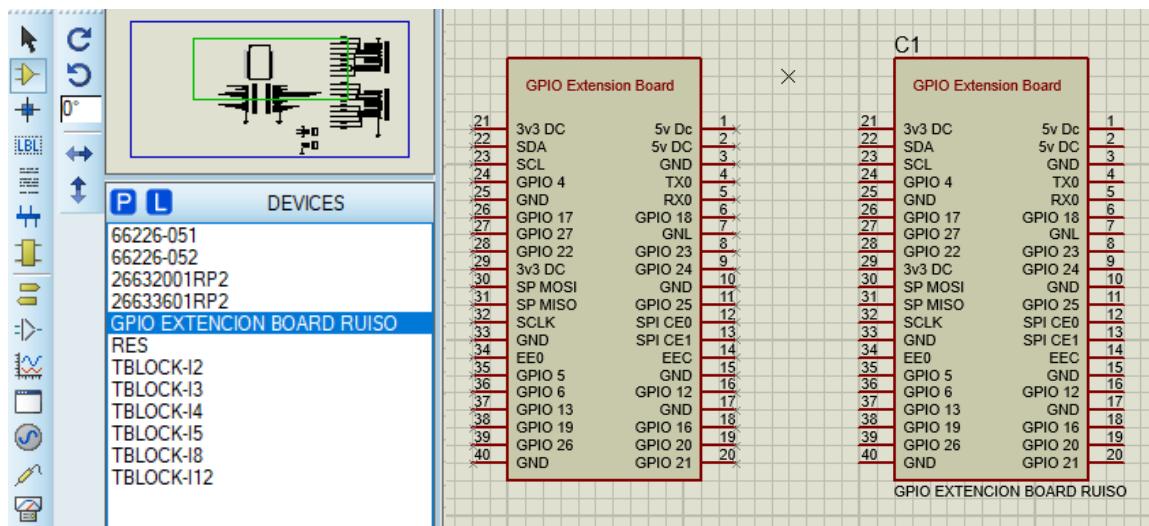


INFORME FINAL PRÁCTICA EMPRESARIAL

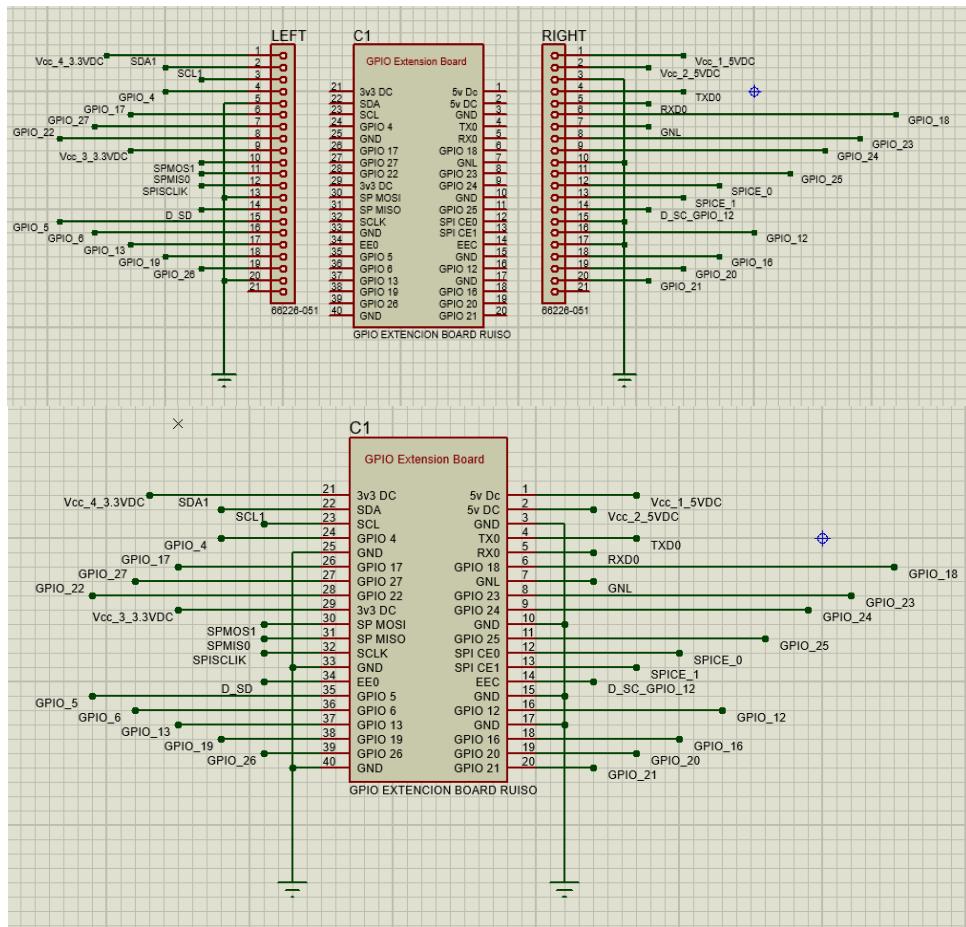
VERSIÓN: 1.0



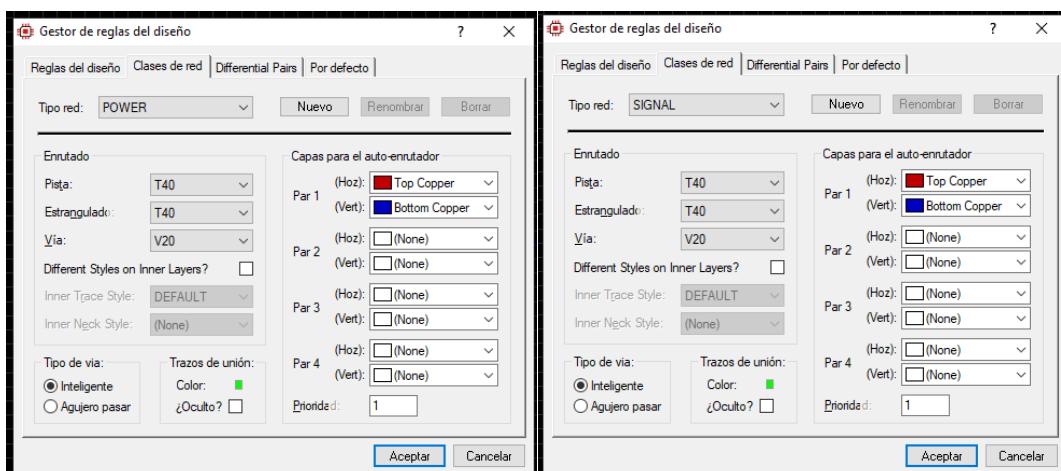
Y listo ya tenemos nuestro componente (sin conexión a ARES) en nuestro ISIS, de hecho, podemos verlo ya importado en nuestra barra de herramientas lateral. Algo a tener en cuenta es que el bosquejo que tenemos en la hoja, sigue siendo un dibujo editable, pero solo el componente que se encuentra en la barra de tareas será el utilizable. A la izquierda el bosquejo que hicimos y a la derecha el componente creado.



Al nuevo componente podemos conectar los caminos que necesitamos.

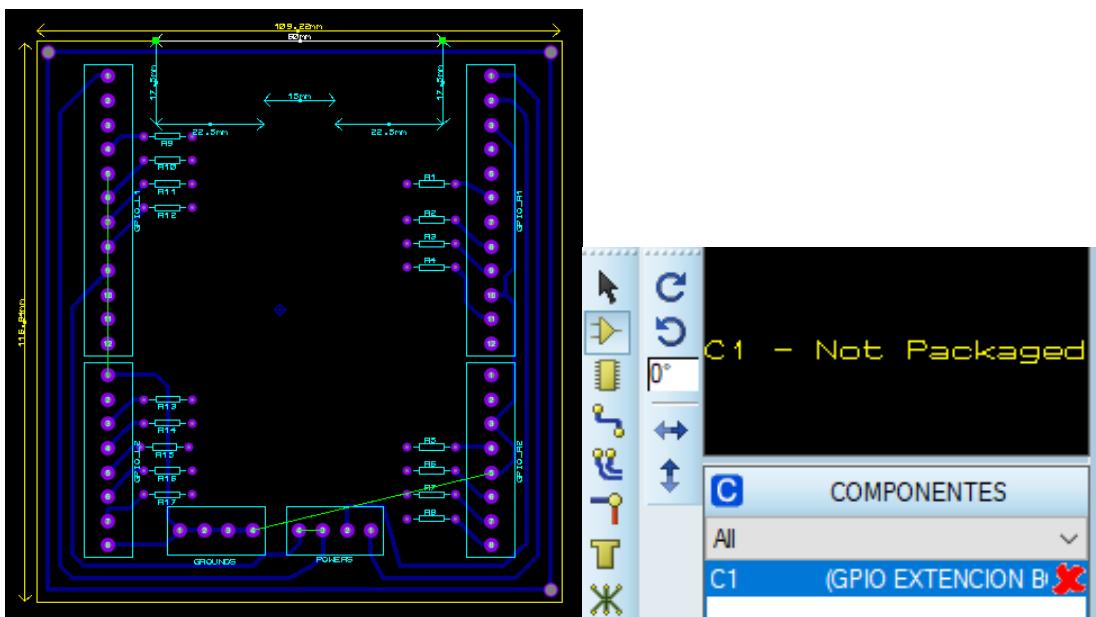


Ahora si, momento de pasar al ARES, aquí recomiendo manejar los siguientes parámetros para el grosor de nuestros caminos.

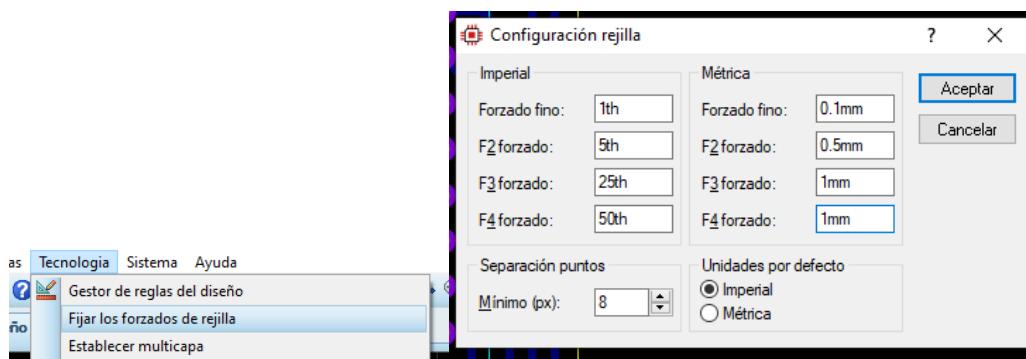


También es recomendable el uso de crear un espacio para la Board Física que vamos a utilizar, así no nos llevamos sorpresas con los tamaños de los circuitos que nos salgan. de la misma manera el uso de la regla de medición es algo importantísimo y por sobre todo tratar de NO utilizar el autorouting a menos que lo vea muy necesario, pues muchas veces este mecanismo nos generara pistas de cobre con interrupciones o mal colocadas.

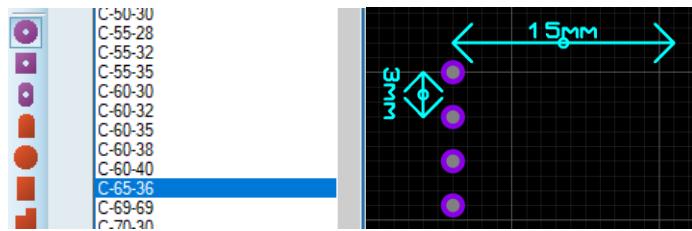
Este circuito este hecho a una capa por lo que solo se utilizara la capa Bottom Copper, la cual es fácilmente distinguible al trabajar con caminos de cobre azules. Cableando todos los componentes que tenemos al momento (sin contar la GPIO extensión) tenemos lo siguiente:



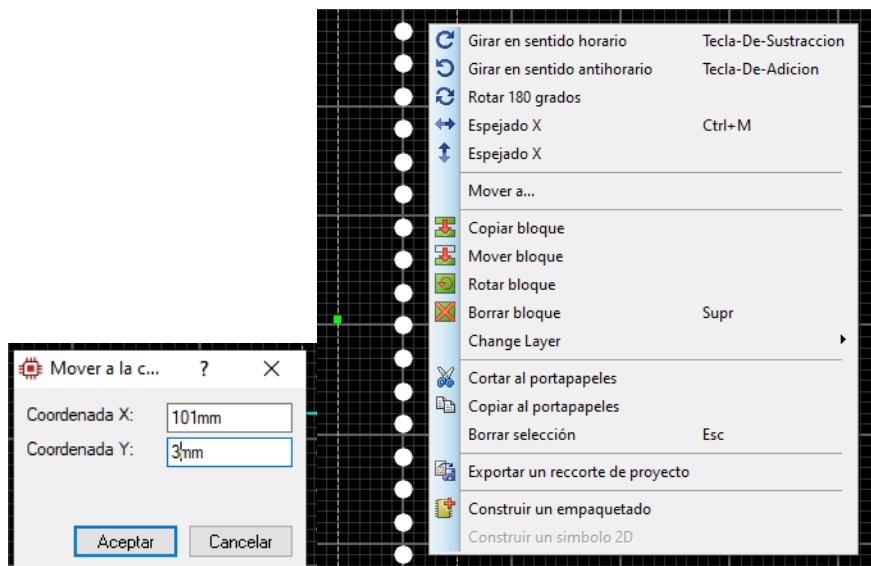
Para el componente faltante sabemos que el ancho de los pines es de 15mm (espacio entre lateral y lateral) mientras que el espaciado es de 3mm (espacio de pin a pin inmediatamente siguiente), además que son 20 pines por lado. Ahora creamos este componente. Voy a setear a conveniencia una malla de 1mm, esto quiere decir que cada cuadrado vale 1mm, de esta manera puedo crear de forma más fidedigna el componente y evitar los problemas que teníamos con los otros pre establecidos en proteus.



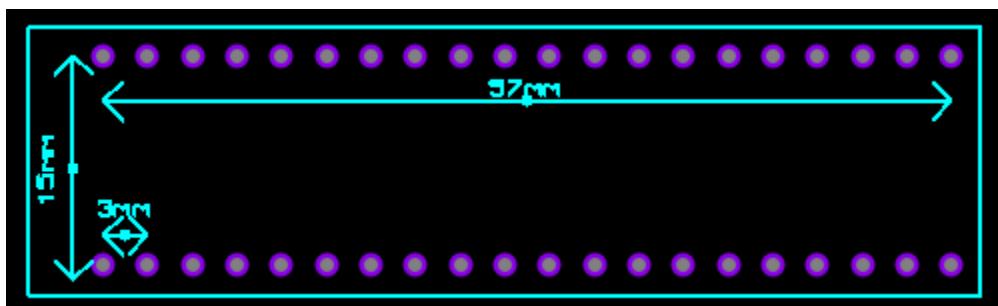
Voy a utilizar un pad o punto de soldadura similar al tamaño de las resistencias.



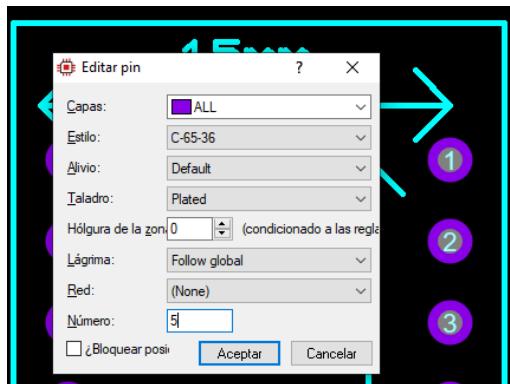
Para asegurarnos que los pines tienen la separación que necesitamos podemos ver sus propiedades de "move to" aunque en principio no los queramos mover. En caso de que no estén en 3mm podemos moverlos ahora si a que cumplan la separación prevista de pines.



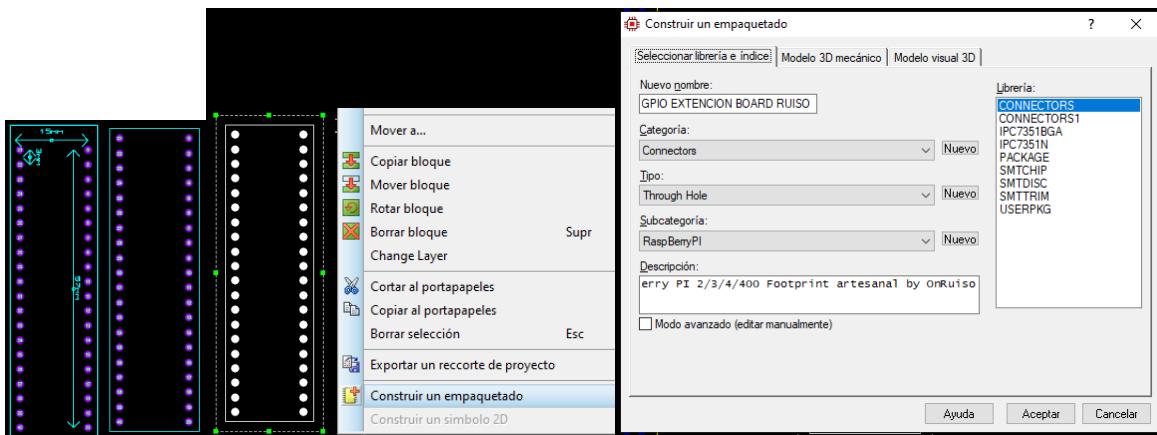
De esta manera vamos creando el componente y tendremos algo similar a lo siguiente:



Ya tenemos nuestro esquema del componente, ahora debemos asignar los pines, para esto damos doble clic sobre el pin que nos interesa y damos el número del pin que necesitamos.



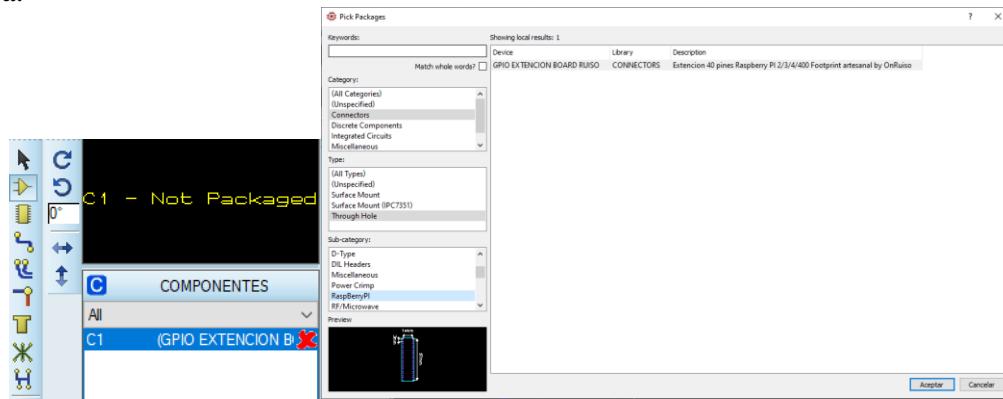
Solo queda confirmar que tengamos la enumeración correcta en nuestros pines. La importancia de la enumeración de los pines es la de coincidir con los pines de nuestro esquemático en ISIS. Ahora sí, vamos a crear este componente ARES de la siguiente forma:



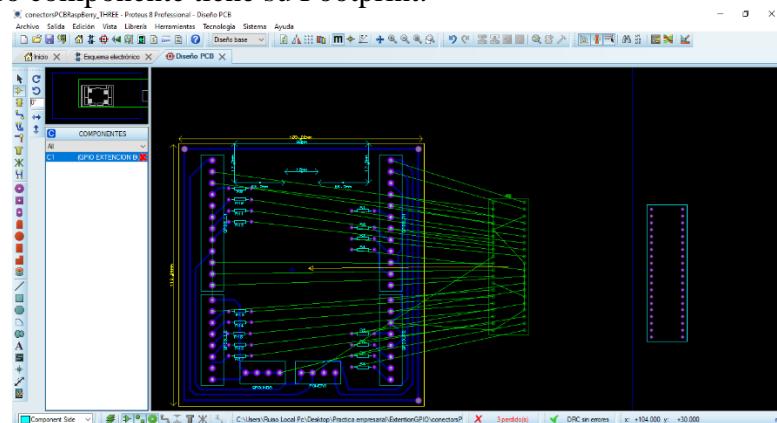
De la misma manera nos quedaremos con nuestro bosquejo en la zona del ARES, por seguridad se recomienda sacarla fuera del borde azul de operación del ARES.



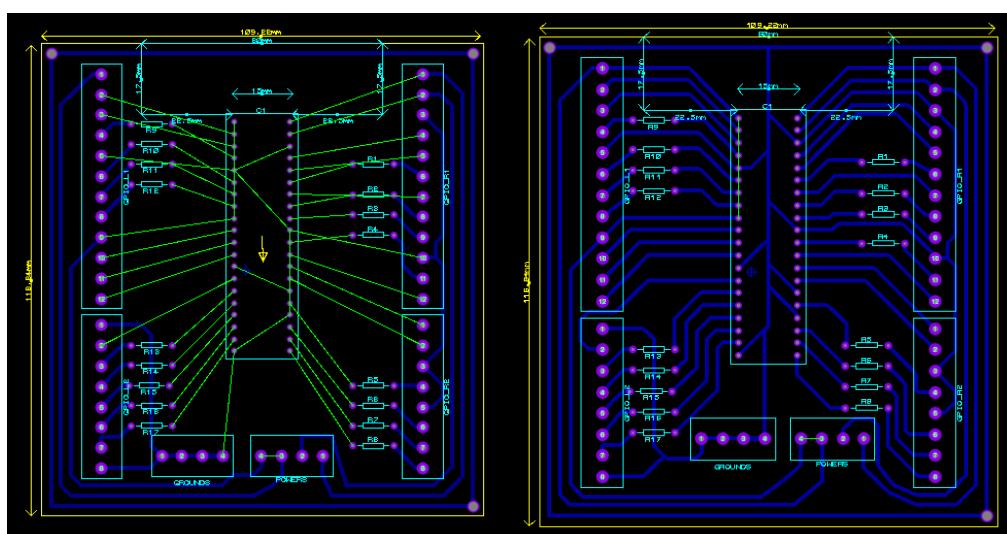
Ahora voy a mi componente y en el selecciono mi Footprint recién creado de la siguiente manera.

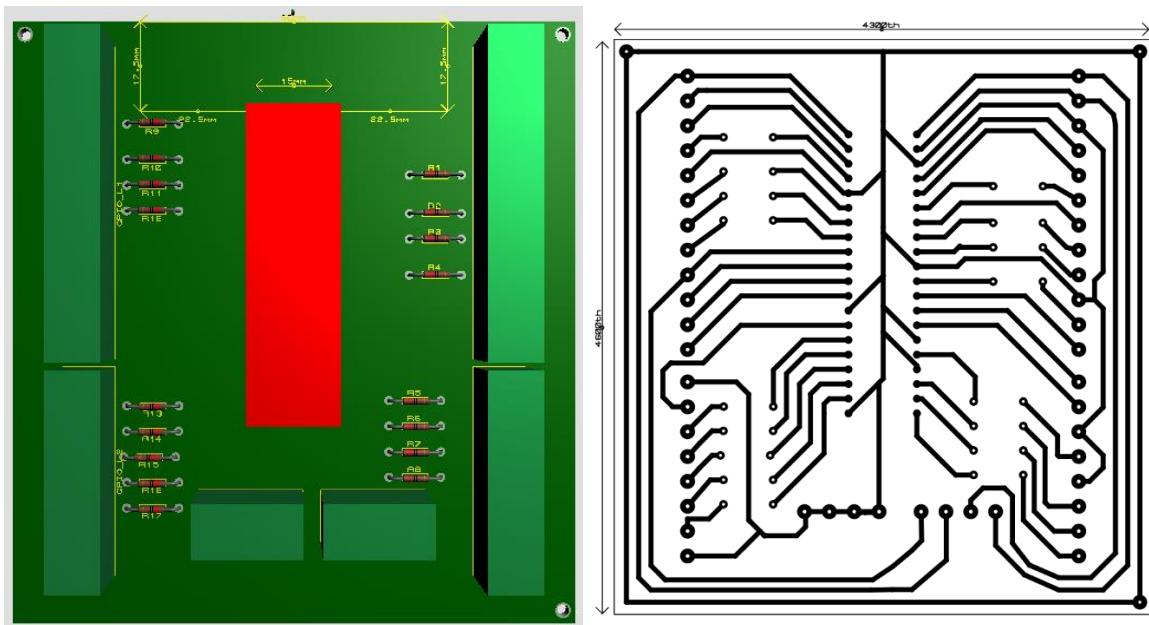


Ahora si nuestro componente tiene su Footprint.



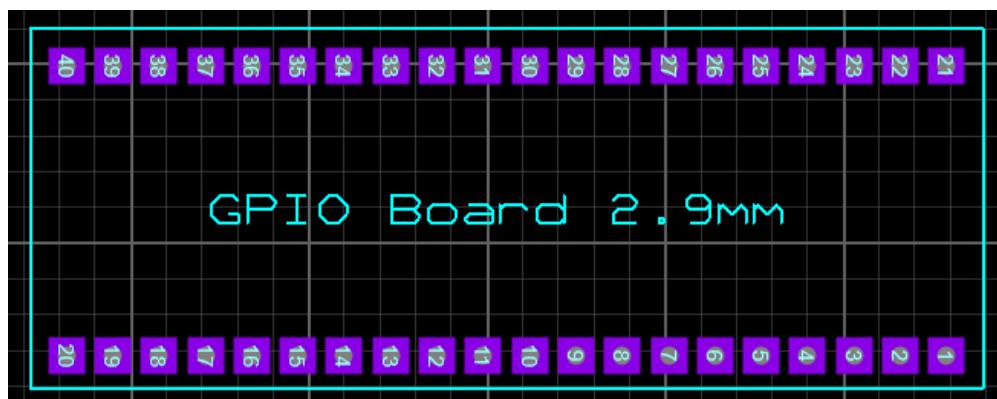
Ya solo nos queda cablear el nuevo componente a nuestro circuito electrónico.





En caso de que vuestra Extensión GPIO sea más pequeña, probablemente es que la separación entre pines sea distinta, por ejemplo, una separación de 2.9mm o una separación de 2mm. Para estos casos hay que crear un nuevo modelo Footprint que cumpla con las especificaciones que necesita. Luego se le asigna al componente este nuevo footprint.

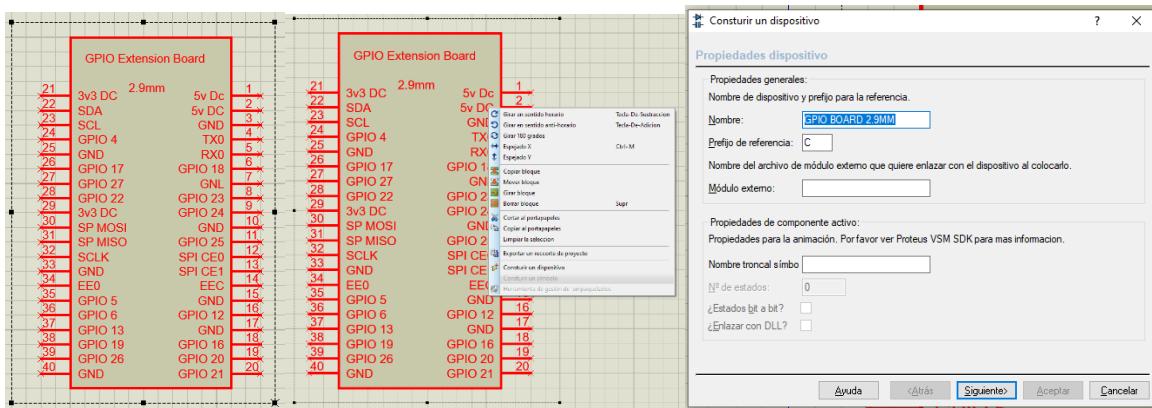
Por ejemplo, Una Board de 2.9mm luce de la siguiente manera:



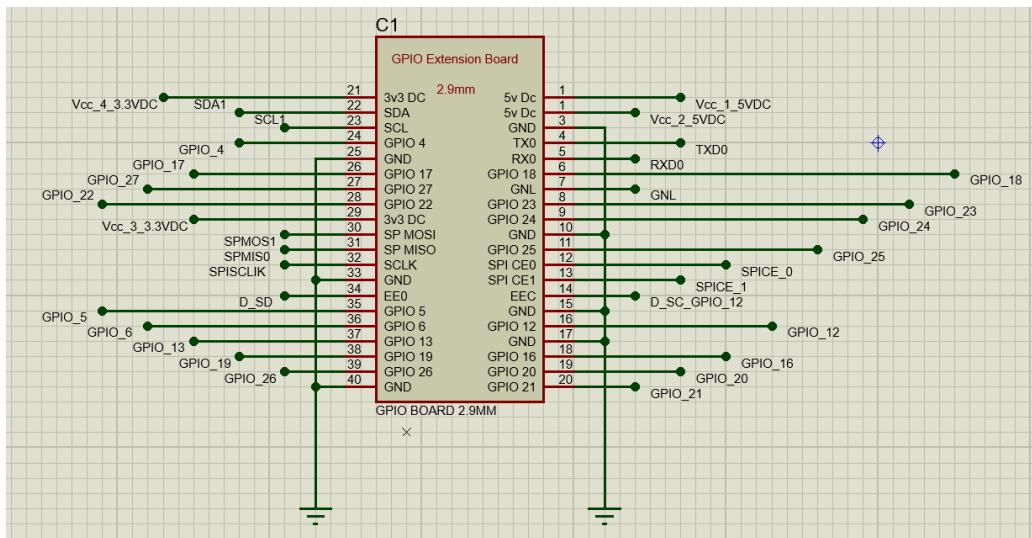
Para reasignar su footprint, vamos al componente desde IRIS, dentro de la barra lateral del sub-menu de componentes (Amplificador Amarillo), dando clic derecho sobre el componente creado con anterioridad y seleccionar "gestionar empaquetado". Si la versión de Proteus no permite añadir un nuevo empaquetado desde esta forma, lo prudente es crear un nuevo componente desde el apartado de ISIS de la siguiente forma:

INFORME FINAL PRÁCTICA EMPRESARIAL

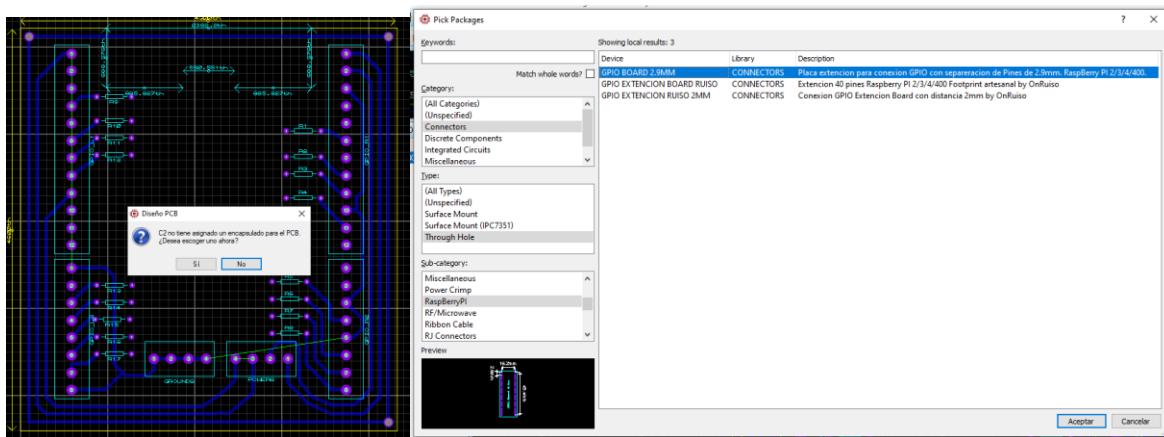
VERSIÓN: 1.0



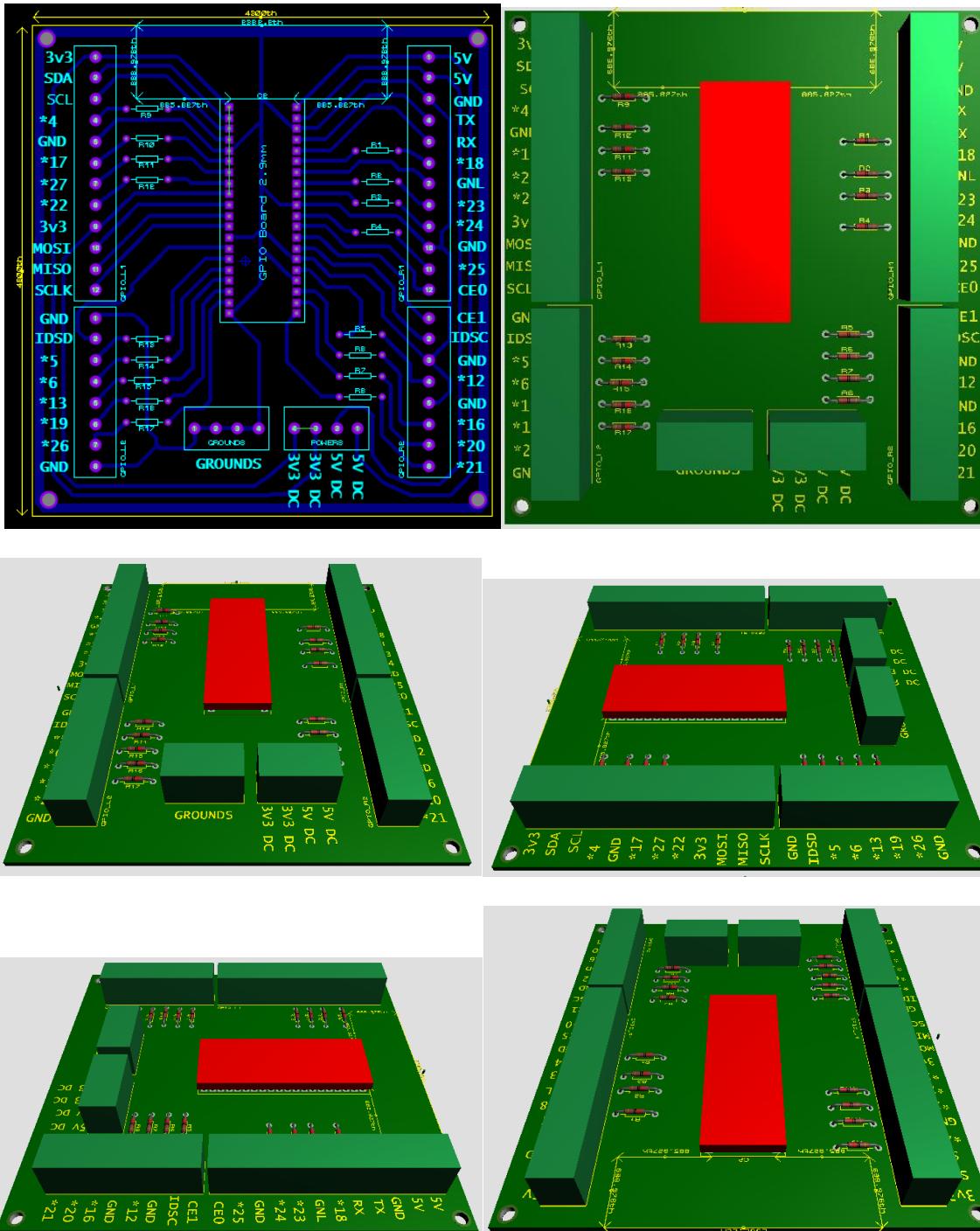
Al crearlo lo reemplazamos por su homólogo anterior.



Dentro del ARES asignamos el nuevo Footprint.

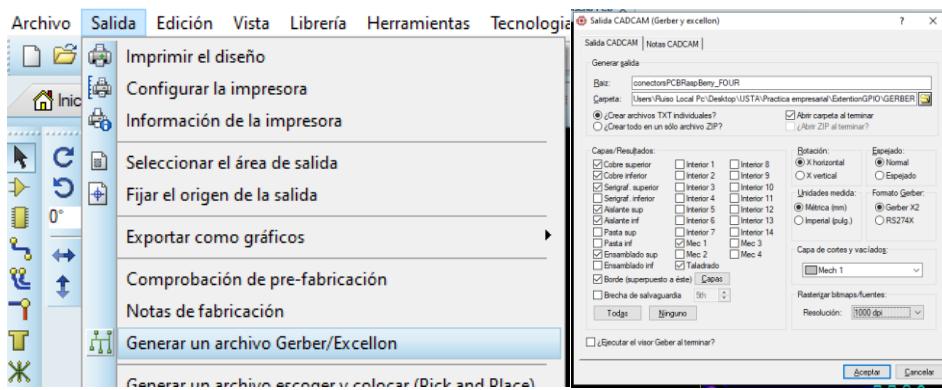


Ahora podemos empezar a nombrar los pines de la siguiente manera.

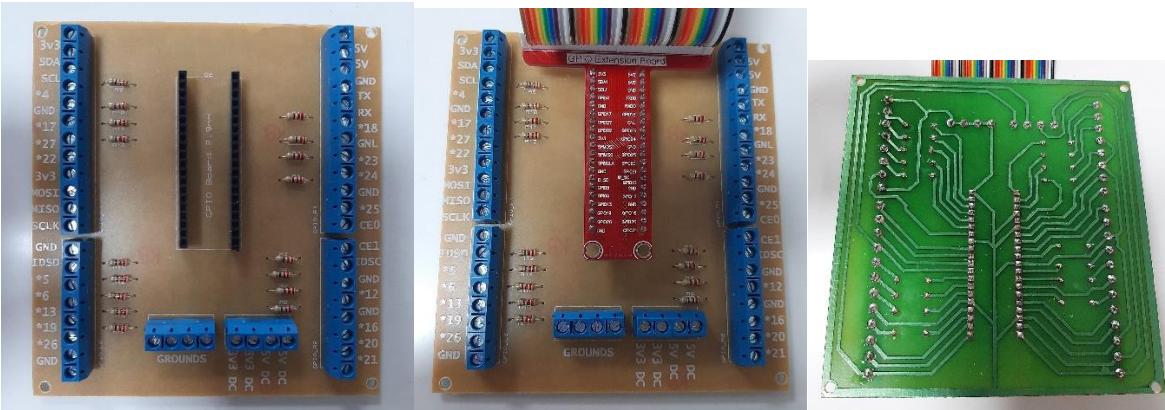


Para poder enviar este circuito a impresión de PCB, podemos bien exportar un archivo de Imagen SVG o un archivo PDF con la finalidad de obtener los caminos de cobre y la tipografía que diseñamos, sin embargo, el método más profesional de realización de la PCB

electrónico es mandar estos archivos a una empresa de realización industrial de tarjetas electrónicas, para esto se debe exportar un archivo Gerber/Excellon.



Una vez tenemos los GERBER podemos ordenar la fabricación de la PCB. Con ella lista podemos soldar los componentes e implantar la board GPIO para ser usada dentro de la máquina de oxidación de materiales. El resultado físico final es el siguiente:



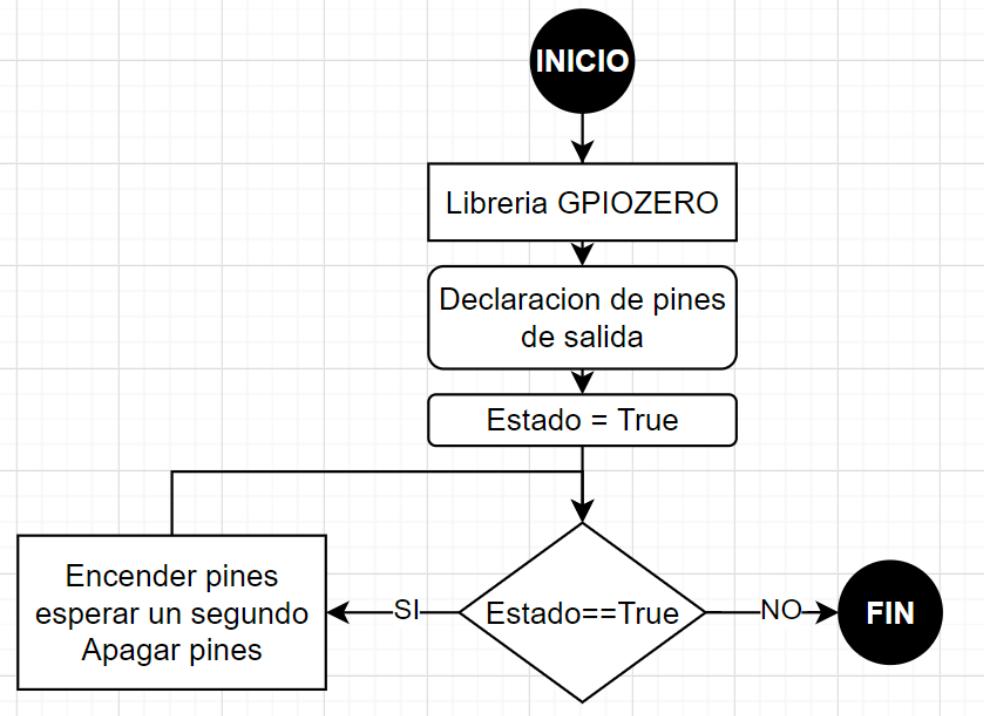
4.1.6 Software Switch: Comutación a partir de la RaspberryPI 400 y el canal de GPIO.

La conmutación de pines es uno de los pasos más importantes y utilizados en la realización de computación física, poder encender y apagar los pines de nuestra RaspberryPI puede ser utilizado en un sin fin de productos y proyectos y es por este motivo que se realiza este procedimiento, primeramente.

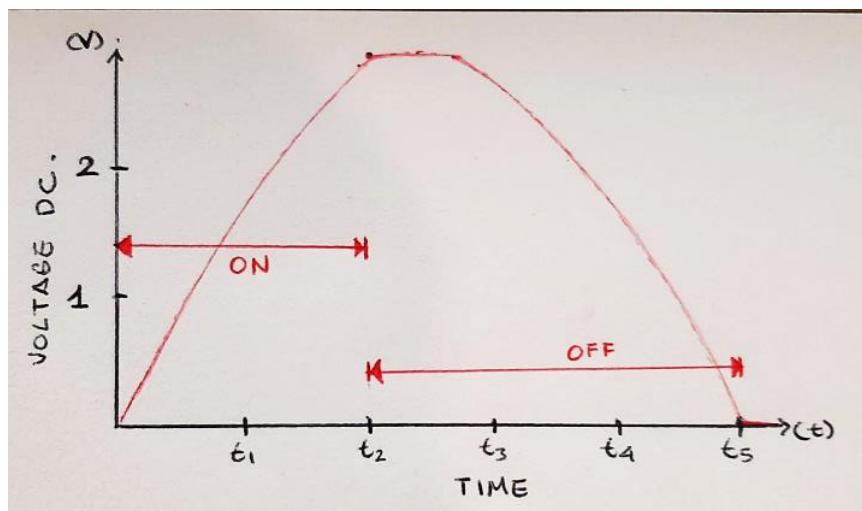
Los diferentes avances en las versiones y módulos de conmutación a partir de Software se desarrollaron en base a dos diferentes librerías de control de la placa GPIO integrada en la Raspberry PI 400, los resultados según el orden de bitácora del pasante son los siguientes:

Encender y apagar un pin de la Raspberry de forma cíclica.

La primera forma de código esta basada en el siguiente diagrama de flujo:

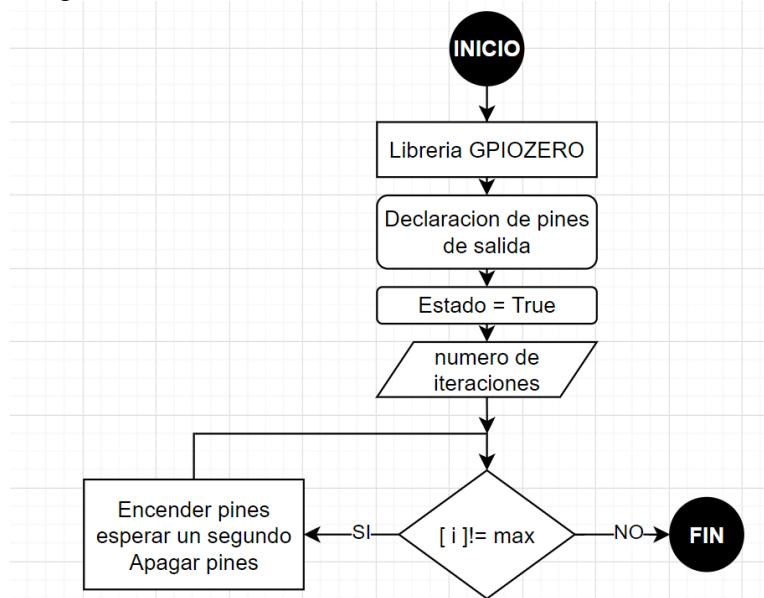


Por análisis observacional en la lectura dada por el multímetro, el encendido del PIN de salida es casi instantáneo, con un posible retraso en las decenas de (ms) o menos (1ms - 99ms) sin embargo el apagado del PIN demora mucho más, en el rango de las centenas en (ms) (100ms - 999ms) probablemente medio segundo, un poco más o un poco menos. Un comportamiento que tener en cuenta en la realización de proyectos.

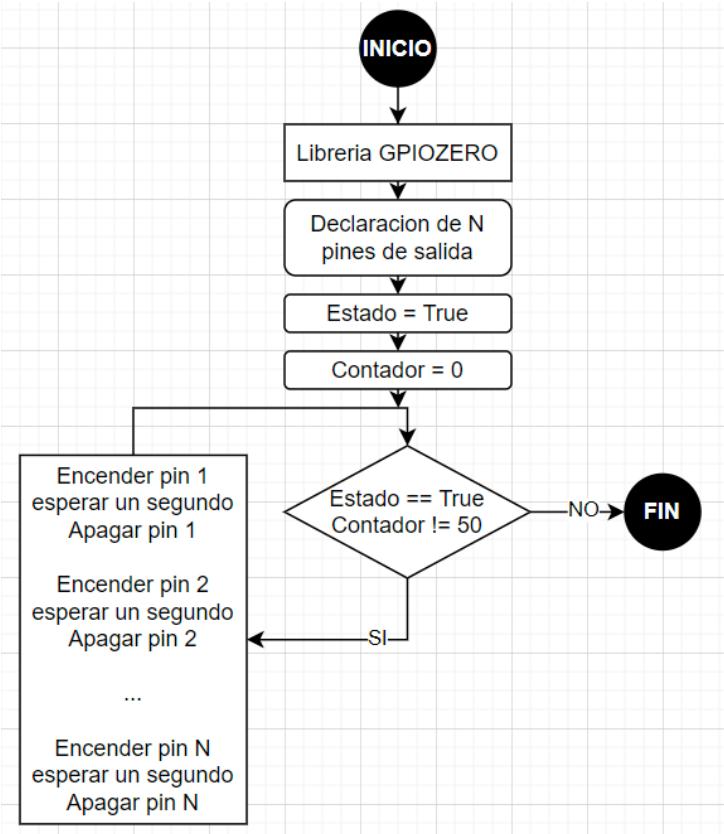


Encender y apagar un pin de la RaspberryPI con un ciclo límite

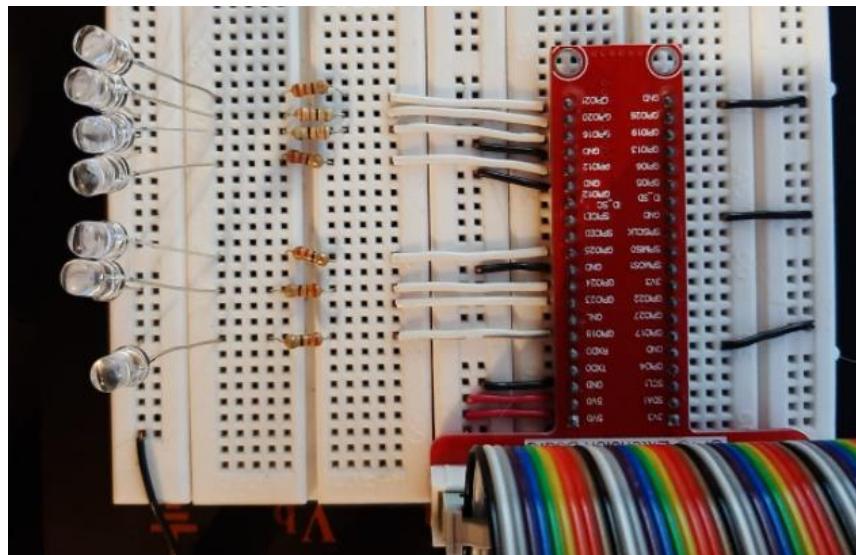
El código sería lo siguiente:



Podemos alterar un poco el código inicial para testear cada uno de los pines GPIO de la siguiente manera:

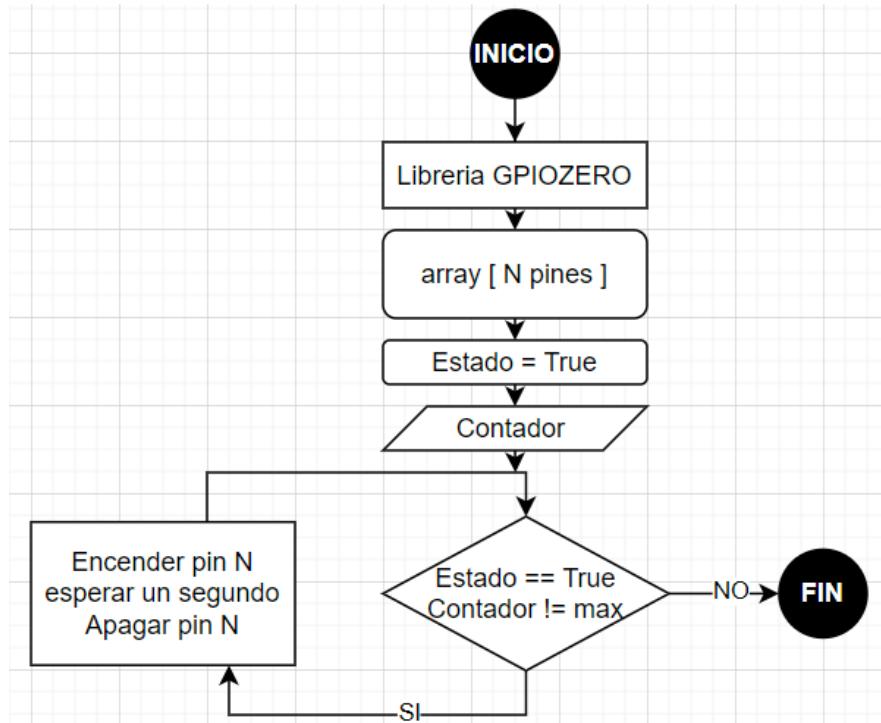


El lateral que estamos encendiendo es el siguiente:



Optimizar - Encender y apagar varios pines de forma cíclica límite.

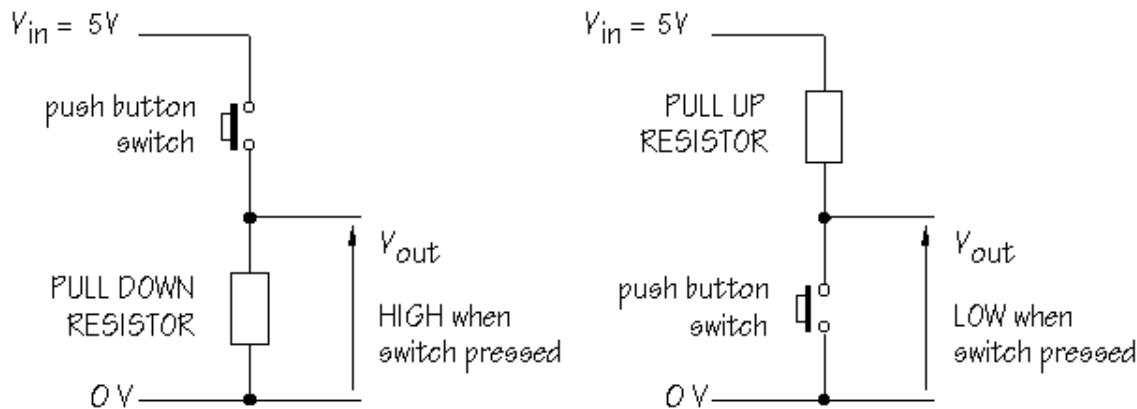
Ahora bien podemos mejorar el anterior código cíclico y preguntando por el número de ciclos a realizar por el usuario de la siguiente manera:



Aquí debemos tener muy en cuenta un comportamiento de los pines de la raspberry y es que el primer led a iluminarse siempre demorara más que los demás en cada ciclo, o bien el led 18 tiene ese comportamiento, es cuestión de observación específica en este caso.

Encender o apagar un pin mediante la detección de un botón

Primero tenemos que recordar que cuando trabajamos con micro controladores PIC o Arduino, la detección de un botón puede darse de dos maneras distintas, esto es detectando un 0 o un 1, físicamente esto se traduce en detectar una bajada de voltaje en un pin digital de entrada o una subida de voltaje respectivamente. Para lograr lo anterior se dispone de una conexión de un pulsador en conexión a tierra o a Vcc según convenga de la siguiente manera.



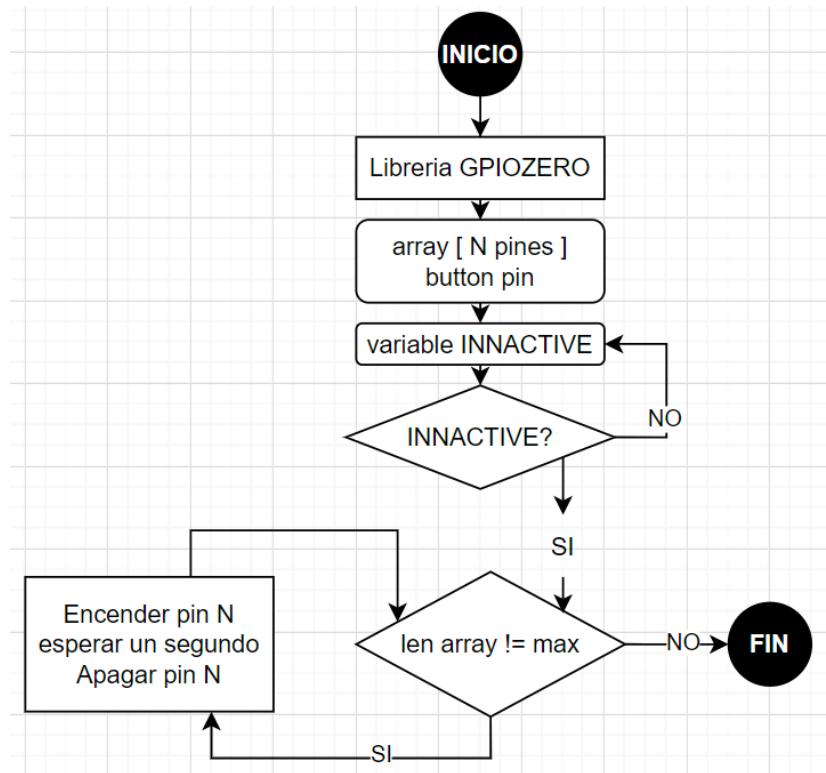
Como se puede ver en la anterior imagen, la conexión PULL DOWN mantiene una conexión a tierra o 0V DC, al pulsar el pulsador el flujo de corriente de V_{in} dirige la corriente a V_{out} siendo este último conectado directamente al pin del microcontrolador. Ahora en caso contrario el PULL UP mantiene de forma permanente un voltaje en 5v DC de V_{in} en el V_{out} , al momento de pulsar el pulsador la corriente se dirija al camino de menos resistencia, es decir tierra, leyendo el pin del controlador una disminución del voltaje.

El siguiente código puede ser utilizado para cualquiera de las dos configuraciones, el aspecto a cambiar será el siguiente:

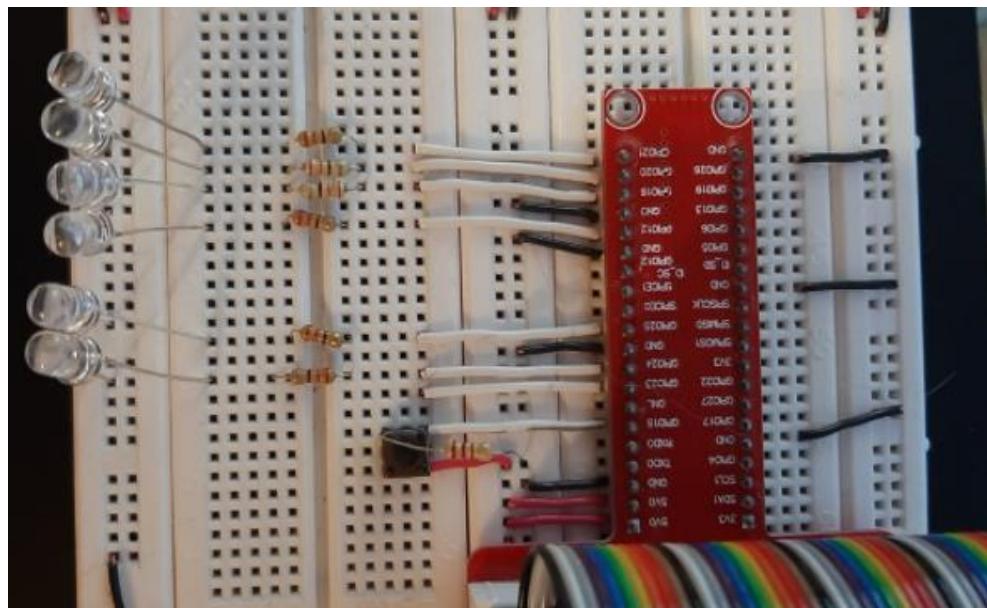
```
while(button.wait_for_()):
    for i in range(len(button)):
        led = LED(button[i])
        led.wait_for_inactive()
        led.on()
```

Recordar que la función WAIT_FOR_ACTIVE o WAIT_FOR_INACTIVE es propia de la librería de GPIOZERO, con versiones mas antiguas de esta librería es posible trabajar con datos básicos de 1

y 0 para la lógica de programación. La configuración utilizada para este circuito fue de tipo PULL DOWN.



El circuito montado luce de la siguiente manera:



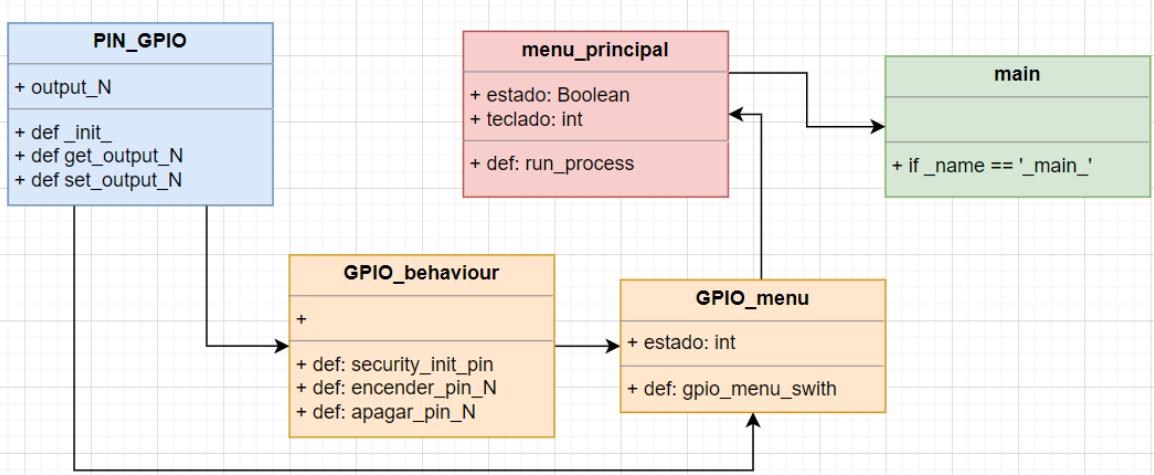
Encender o apagar un pin desde la división de modelos y controladores

Una primera idea de implementación del módulo de conmutación de los pines de la GPIO con el resto de módulos del proyecto fue la de realizar una versión mejorada del código dividida en el clásico paradigma de programación orientado a objetos POO en la forma de Modelos, Vistas y Controladores. Este primer esquema semi completo se trabajó bajo la librería de GPIOZERO.



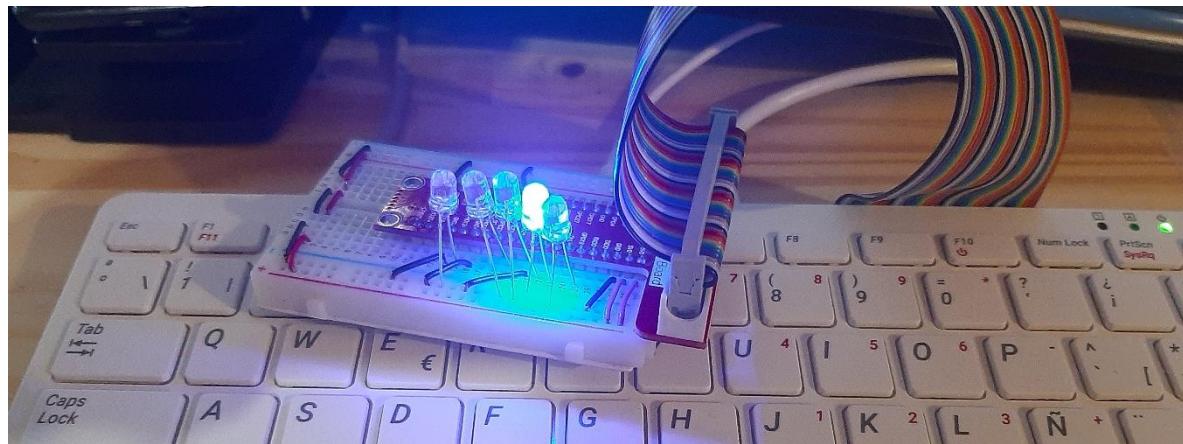
En la realización de este modelo se tomaron las siguientes premisas en su construcción:

1. El Main es el código principal que se encarga de cargar a forma global todos los parámetros de funcionamiento de cada uno de los métodos del programa general. Este llama a ejecutar el menú principal de acceso del usuario.
2. El menú principal es la vista del proyecto, en el esta codificado las opciones que tiene el usuario dentro del sistema de conmutación. Este llama la función el controlador principal de menú.
3. El controlador GPIO_menu contiene la serie de métodos que se encargan de evaluar el requerimiento del usuario del sistema y determina según la entrada por teclado o por un botón si se debe apagar o encender un pin. Este llama a los controladores de GPIOBehaviour.
4. En el controlador secundario están los métodos que dan un estado de encendido o apagado a los pines previamente configurados de la GPIO. Este utiliza la definición de pines del modelo.
5. El modelo PIN_GPIO basado en la librería GPIOZERO define las salidas de los pines para ser utilizadas por el sistema.

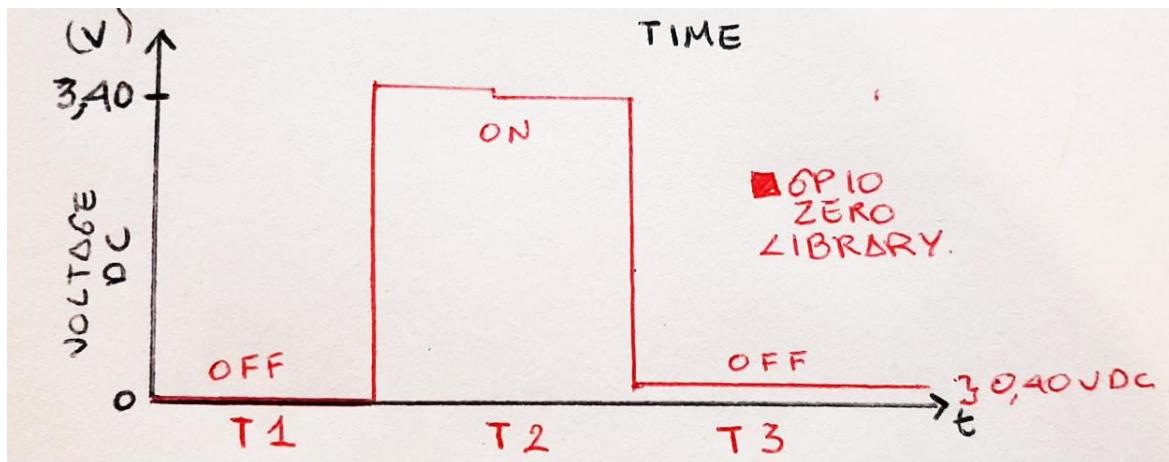


Este modelo de trabajo del código, aunque prometedor en la organización de los módulos de trabajo del sistema, hace responder mal las entradas y salidas de la GPIO, incrementando los retardos de encendido y apagado del sistema a la vez que demorando y anulando lecturas de datos de sensores

tanto análogos como digitales. Por este motivo se decide cambiar de forma de modulación del programa y de librerías a utilizar.



Con el uso de la librería GPIOZERO se observa a través de multímetro de que el estado inicial d ellos pines de la GPIO una vez llamados dentro del código no es de 0VDC, mas bien es de aproximadamente 0.20V DC a 0.40V DC, lo cual puede incurrir en falsos positivos a la electrónica sensible que puede entender este mínimo voltaje como un 1 y empezar a que la máquina elabore una rutina sin el usuario requerirlo. Para una máquina de oxidación de materiales es algo sumamente peligroso.



En caso de encender un pin con el uso de la librería GPIOZERO se recomienda asegurar el pin de salida con la implementación de una resistencia en serie al pin de 220 ohmios a 330 ohm, con la finalidad que el mínimo voltaje existente tras la constante comutación vuelva a 0VDC y se entienda como un 0 lógico en las tarjetas electrónicas.



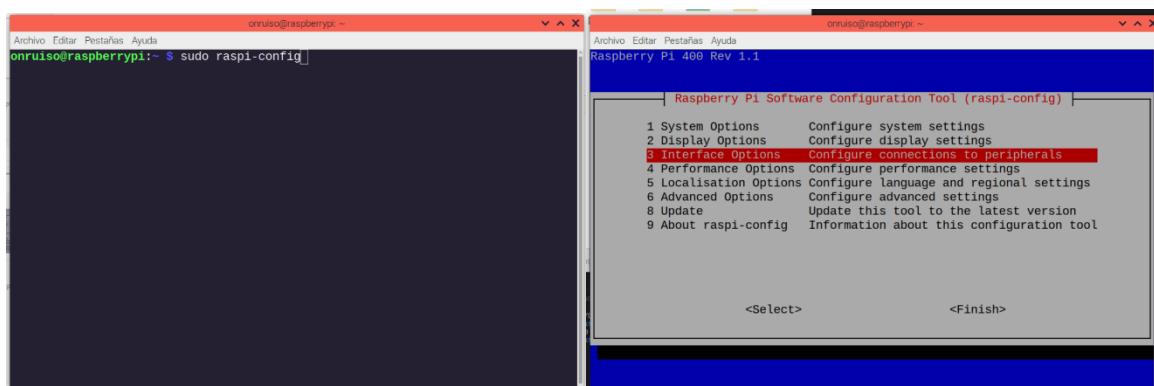
4.1.7 Software Sensores: Lectura de sensores Digitales mediante GPIO.

Para poder leer un SENSOR DIGITAL por medio de la entrada GPIO que tiene la Raspberry PI 400 en la parte anterior, debemos primero entrar a la "BIOS" de esta computadora y activar el protocolo de Raspberry PI 400 para la lectura por los canales de la Board.

Basta con acceder a la terminal e ingresar el siguiente comando:

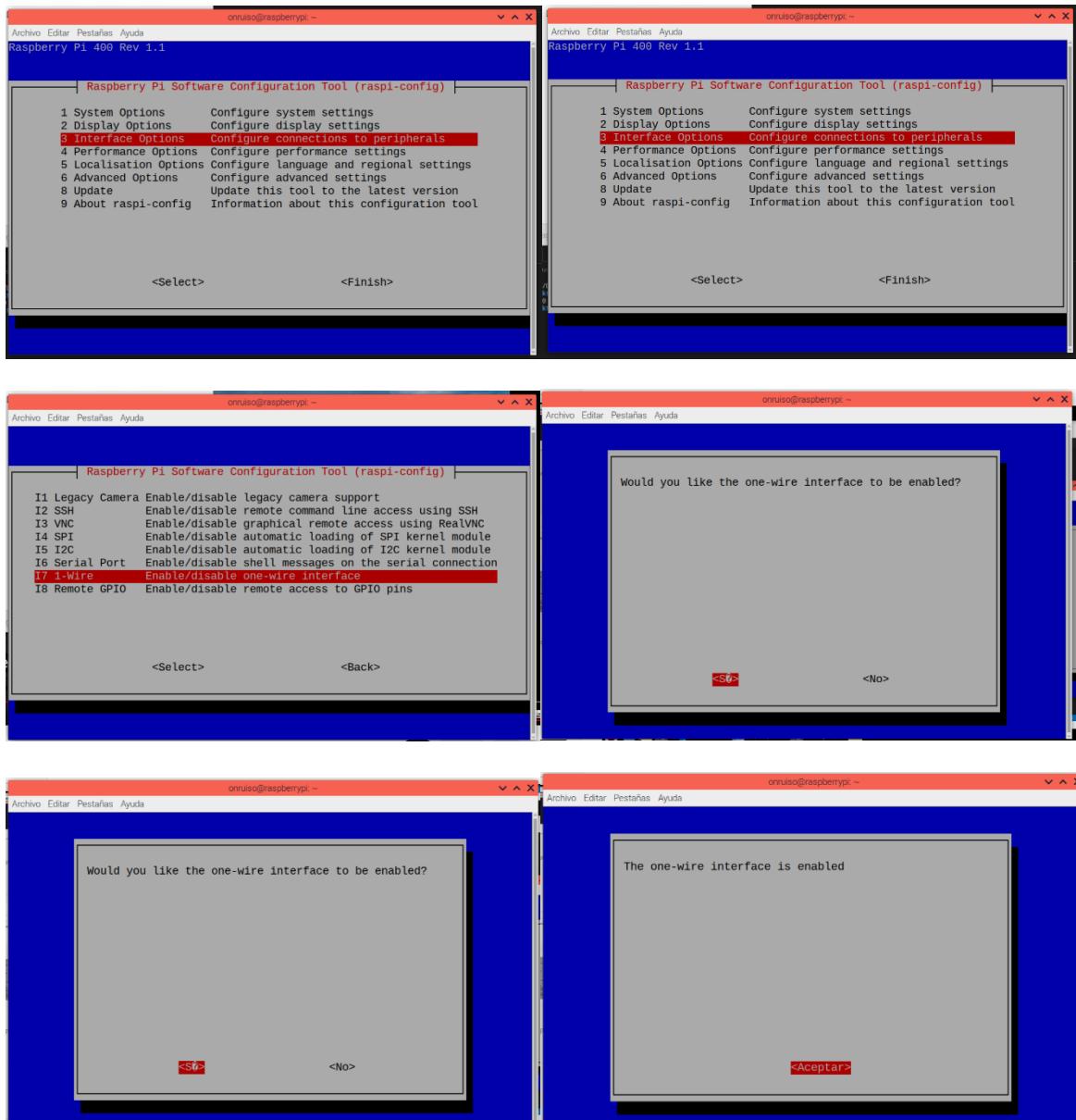
```
sudo raspi-config
```

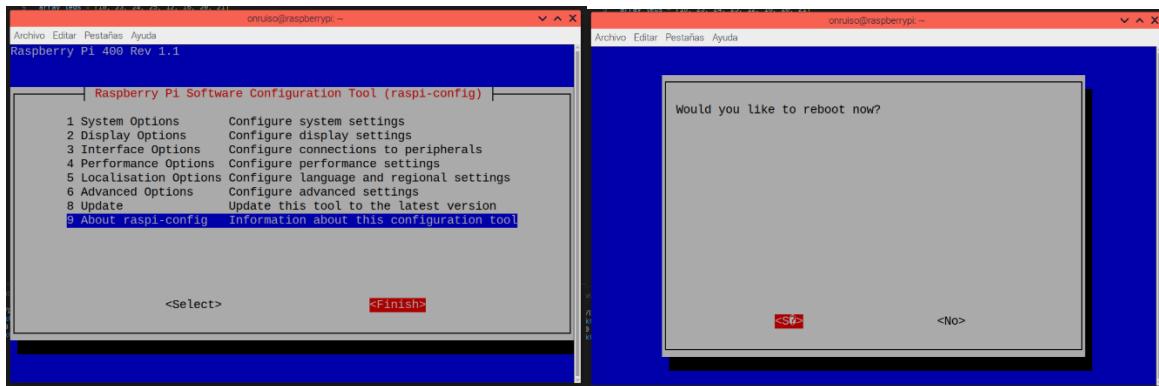
Al escribir esto en la terminal se nos presentara en la misma una interfaz grafica de apariencia parecida al aseso de la BIOS donde podremos trabajar con todos los elementos internos de la Raspberry, el proceso de activación del canal es el siguiente:



INFORME FINAL PRÁCTICA EMPRESARIAL

VERSIÓN: 1.0





Solo queda Reiniciar la computadora. Una vez reiniciado el sistema los puertos estarán preparados para leer los datos que entrar por parte de los sensores, sin embargo, queda saber en que parte de la maquina está entrando esta información. Vamos a navegar hasta la siguiente dirección, donde se estarán almacenando los datos que se están recopilando por el sensor.

```
cd /sys/bus/w1/devices
```

El sensor se conecta al GPIO4 en nuestra conexión Física. Los datos son resibidos por la RspBerry PI 400 como información de un componente integrado a la placa por lo que su estado y precisamente las lecturas que necesitamos yacen dentro de la memoria ROM de este dispositivo, concretamente en la dirección de archivo a la que accedimos con anterioridad.



Si bien por medio de esta canal se puede conectar cualquier tipo de sensor digital, la primera prueba del sensor se hará con la Termocupla DS18B20, este es un sensor de temperatura con cobertura de aluminio que permite tomar mediciones de calor en ambientes húmedos gracias a su diseño el cual posee una "case" que le ayuda a proteger los circuitos internos de cortos eléctricos y oxidación.



Las especificaciones de funcionamiento de este sensor son:

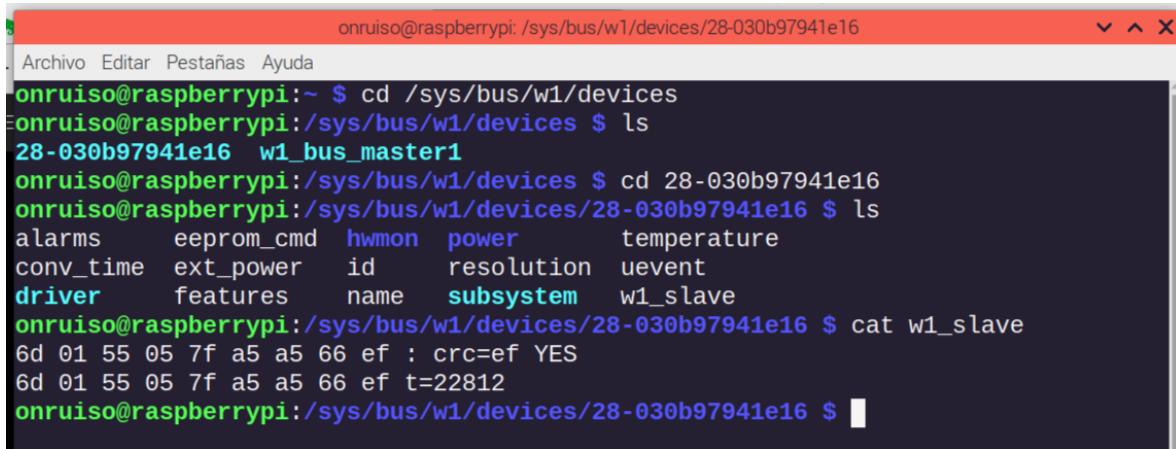
ORDERING INFORMATION

PART	TEMP RANGE	PIN-PACKAGE	TOP MARK
DS18B20	-55°C to +125°C	3 TO-92	18B20
DS18B20+	-55°C to +125°C	3 TO-92	18B20
DS18B20/T&R	-55°C to +125°C	3 TO-92 (2000 Piece)	18B20
DS18B20+T&R	-55°C to +125°C	3 TO-92 (2000 Piece)	18B20
DS18B20-SL/T&R	-55°C to +125°C	3 TO-92 (2000 Piece)*	18B20
DS18B20-SL+T&R	-55°C to +125°C	3 TO-92 (2000 Piece)*	18B20
DS18B20U	-55°C to +125°C	8 µSOP	18B20
DS18B20U+	-55°C to +125°C	8 µSOP	18B20
DS18B20U/T&R	-55°C to +125°C	8 µSOP (3000 Piece)	18B20
DS18B20U+T&R	-55°C to +125°C	8 µSOP (3000 Piece)	18B20
DS18B20Z	-55°C to +125°C	8 SO	DS18B20
DS18B20Z+	-55°C to +125°C	8 SO	DS18B20
DS18B20Z/T&R	-55°C to +125°C	8 SO (2500 Piece)	DS18B20
DS18B20Z+T&R	-55°C to +125°C	8 SO (2500 Piece)	DS18B20

El sensor se encuentra en el folder que es nombrado como 28-030... ingresamos a el con ayuda del comando "cd". Este archivo es el nombre o etiqueta que se asigna a los valores de entrada puestos en la ROM de la computadora.

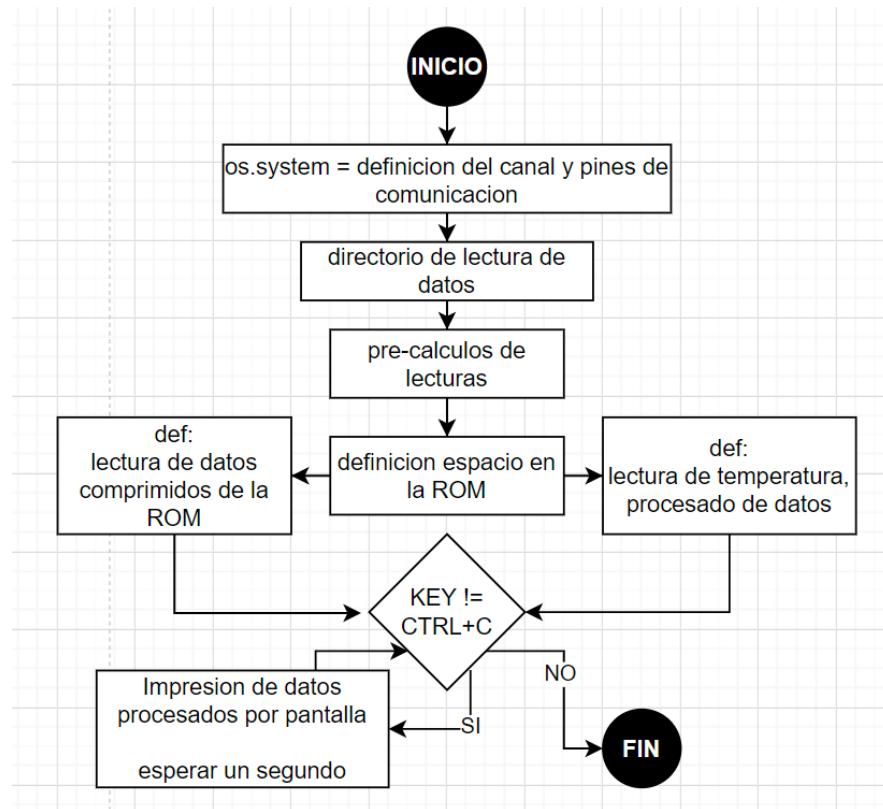
```
onruiso@raspberrypi:~ $ cd /sys/bus/w1/devices
onruiso@raspberrypi:/sys/bus/w1/devices $ ls
28-030b97941e16 w1_bus_master1
onruiso@raspberrypi:/sys/bus/w1/devices $ cd 28-030b97941e16
onruiso@raspberrypi:/sys/bus/w1/devices/28-030b97941e16 $ ls
alarms eeprom_cmd hwmon power temperature
conv_time ext_power id resolution uevent
driver features name subsystem w1_slave
onruiso@raspberrypi:/sys/bus/w1/devices/28-030b97941e16 $
```

De aquí el archivo que nos interesa es el llamado "w1_slave" pues este es el que contiene los datos de temperatura. Ingresamos a este archivo con ayuda del comando "cat".



```
onruiso@raspberrypi:~ $ cd /sys/bus/w1/devices
onruiso@raspberrypi:/sys/bus/w1/devices $ ls
28-030b97941e16  w1_bus_master1
onruiso@raspberrypi:/sys/bus/w1/devices $ cd 28-030b97941e16
onruiso@raspberrypi:/sys/bus/w1/devices/28-030b97941e16 $ ls
alarms  eeprom_cmd  hwmon  power  temperature
conv_time  ext_power  id  resolution  uevent
driver  features  name  subsystem  w1_slave
onruiso@raspberrypi:/sys/bus/w1/devices/28-030b97941e16 $ cat w1_slave
6d 01 55 05 7f a5 a5 66 ef : crc=ef YES
6d 01 55 05 7f a5 a5 66 ef t=22812
onruiso@raspberrypi:/sys/bus/w1/devices/28-030b97941e16 $
```

Como podemos ver en el archivo tenemos 2 líneas, la primera nos marca que los datos que están siendo recibidos son válidos, mientras que la segunda nos marca que después de "t=" el valor de la temperatura. Ahora si vamos a leer los datos de temperatura tal cual en el script de python.



Al ejecutar este código sacaremos en pantalla lo siguiente:

Teoricamente la GPIO puede recibir cualquier tipo de sensor digital, en caso de esta practica empresarial, podría utilizarse también la termocupla MAX 6675 para monitorear aspectos externos a la cámara de oxidación de materiales.

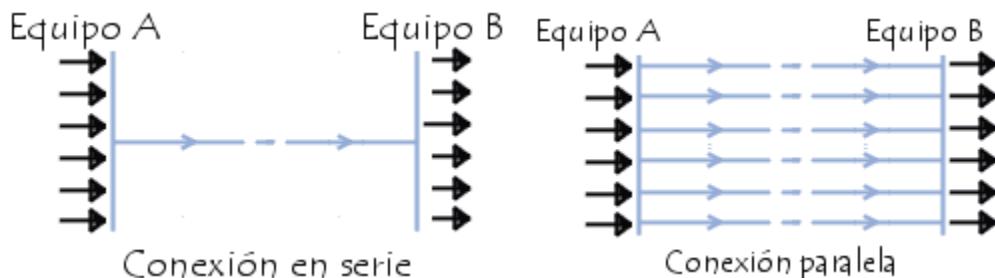


La ficha técnica para su uso es la siguiente:

ORDERING INFORMATION

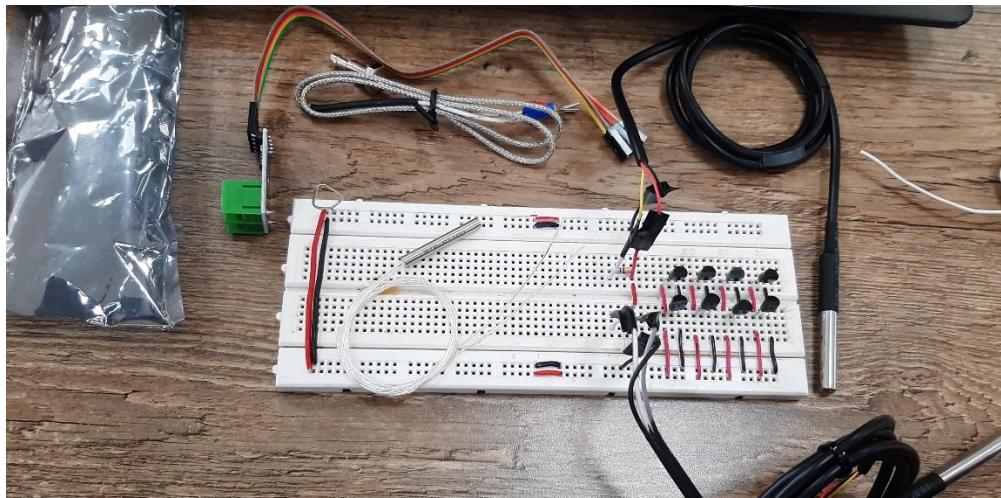
PART	TEMP RANGE	PIN-PACKAGE	TOP MARK
DS18B20	-55°C to +125°C	3 TO-92	18B20
DS18B20+	-55°C to +125°C	3 TO-92	18B20
DS18B20/T&R	-55°C to +125°C	3 TO-92 (2000 Piece)	18B20
DS18B20+T&R	-55°C to +125°C	3 TO-92 (2000 Piece)	18B20
DS18B20-SL/T&R	-55°C to +125°C	3 TO-92 (2000 Piece)*	18B20
DS18B20-SL+T&R	-55°C to +125°C	3 TO-92 (2000 Piece)*	18B20
DS18B20U	-55°C to +125°C	8 µSOP	18B20
DS18B20U+	-55°C to +125°C	8 µSOP	18B20
DS18B20U/T&R	-55°C to +125°C	8 µSOP (3000 Piece)	18B20
DS18B20U+T&R	-55°C to +125°C	8 µSOP (3000 Piece)	18B20
DS18B20Z	-55°C to +125°C	8 SO	DS18B20
DS18B20Z+	-55°C to +125°C	8 SO	DS18B20
DS18B20Z/T&R	-55°C to +125°C	8 SO (2500 Piece)	DS18B20
DS18B20Z+T&R	-55°C to +125°C	8 SO (2500 Piece)	DS18B20

Aquí se debe tener en cuenta que, tras diferentes pruebas y cambios en el código, la lectura simultánea de diferentes sensores digitales no ha podido ser posible, esto es debido a que no es entendible si la lectura es paralela como en otros Microcontroladores de menor gama como Arduino o al contrario es en Bus serial de datos como en el ESP8266. Siendo así se recomienda a este punto de la práctica empresarial trabajar con un Microcontrolador externo conectado a la RaspBerry PI para que haga de tarjeta de adquisición de datos y poder trabajar con ellos desde el ordenador.

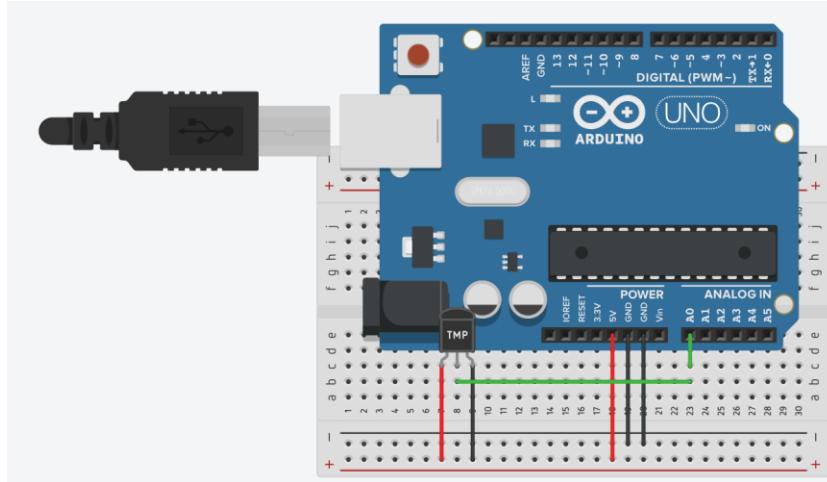


4.1.8 Software Sensores: Lectura de sensores con Microcontrolador Externo.

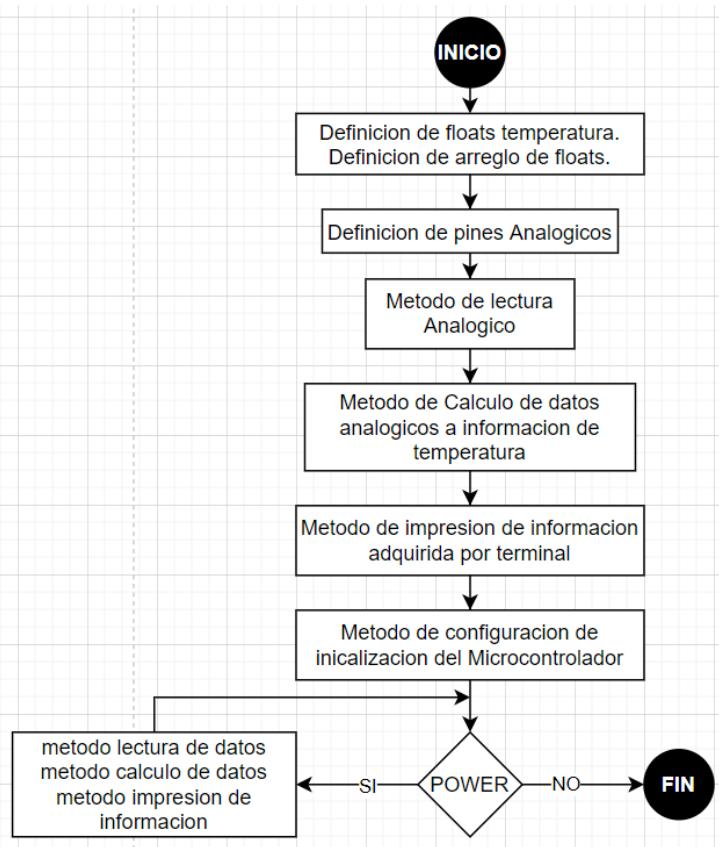
El registro de datos de los sensores digitales y análogos puede delegarse de la Raspberry a una placa de microcontrolador sencilla como lo es un Arduino. Gracias a su conexión serial con una computadora es posible conectar la placa para que sirva de extensión de adquirirán de los datos que de otra manera engorrosa de trabajar o por lo menos lo es a nivel de un proyecto de práctica empresarial.



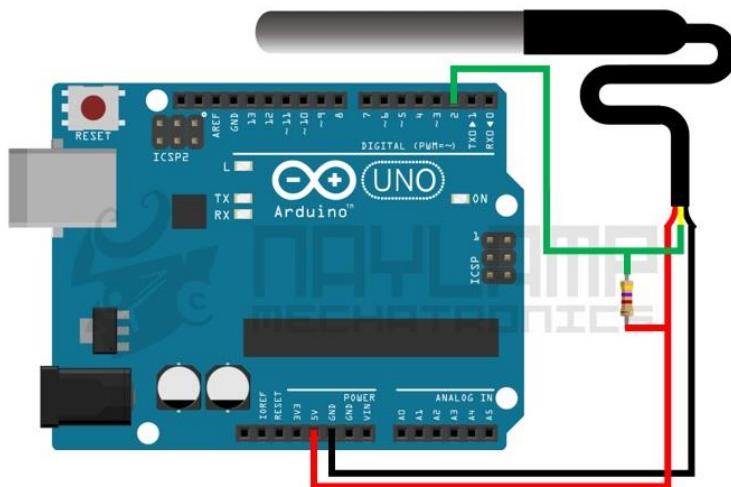
Como primer trabajo de pruebas se plantea la adquisición de 8 sensores de temperatura análogos LM35 esto con la finalidad de probar la conexión en paralelo de cada uno de los sensores y verificar que la integridad de los datos se mantenga. Un esquema similar al aplicado en la vida real es el siguiente.



El código utilizado para la adquisición de estos valores son:

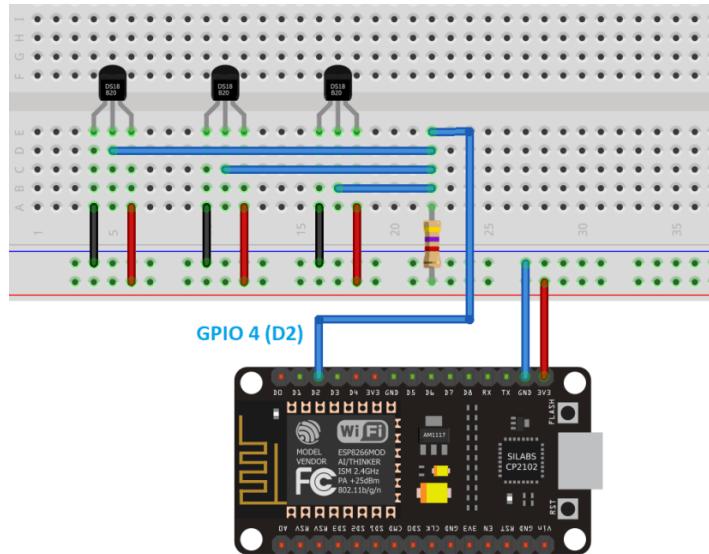


Con la estructura lógica que posee el anterior código se puede formular una conexión simultánea de sensores digitales como se ve en el siguiente esquema.



Sin embargo en este momento de la Práctica empresarial también se empieza a observar la idea de quizás no trabajar con un microcontrolador Arduino como tarjeta electrónica adquisidora, si no a su vez o en reemplazo trabajar con una tarjeta electrónica ESP8266, esto es debido a que, aunque no es objetivo propiamente de la práctica, el usuario final requiere de un dashboard para visualizar los datos que se recogen con los sensores, esto pues para facilitar el monitoreo del trabajo de la

máquina, así como la facilidad de accesibilidad que tendrá el usuario con el sistema. Podría verse algo de esta manera:



Ahora bien, ¿Por qué una ESP8266? Esto es debido a que esta placa electrónica posee un módulo de conexión Wifi integrado que le permite enviar los datos directamente a una dashboard de uso libre en internet. Sin embargo, esto mismo tiene sus problemas de implementación y su adecuación esta por fuera de la práctica empresarial.

En este momento del desarrollo de la práctica empresarial queda ensamblar el sistema físico y electrónico con el software para dar veracidad de los datos adquiridos desde el Arduino hacia la Raspberry PI 400.

4.1.9 Software Dashboard: Propuestas a futuro de la realización de una interfaz gráfica para el apartado del monitoreo.

Como ya lo habíamos mencionado con anterioridad, para facilitar la interacción que tiene el usuario en con el apartado de monitoreo del sistema es necesario que se pueda facilitar de una Dashboard, la misma puede ser programada por el pasante de práctica empresarial o bien puede ser adecuada haciendo uso de software libre en línea.

Lo importante en esta sección de actividades es tener en cuenta que el desarrollo propio de la dashboard no es uno de los objetivos a fines dentro de la práctica empresarial es más un estudio que puede iniciarse en el mismo lapso de tiempo, esto es debido a que la realización por completo de una interfaz gráfica final con la que se pueda quedar este trabajo, debido a su extensión y complejidad es un proyecto solo por si mismo y el estudiante de Ingeniería de Sistemas por recomendación y en parte orden de la Facultad está limitado en el nivel de aportaciones que puede hacer a una empresa, el trabajo dado al pasante debe ser acomodado al nivel de horas mínimas de trabajo solicitado en la modalidad de práctica.

Sabiendo esto, se inicia un estudio de posibles dashboards que pueden ser desplegadas para cumplir el rol de monitoreo visual de los sensores que la maquina utiliza. Una de las primeras opciones es la de realizar una interfaz grafica local programada desde cero, sabiendo que buena parte del código se esta realizando sobre el lenguaje de programación Python, se busca un framework que pueda ayudar con esta tarea dando por encontrado el de DASH.

Uno de los bloques de alta importancia en este proyecto es poder graficar en tiempo real los diferentes datos recogidos por los sensores y actuadores de la maquina en cuestión. sin embargo, nos topamos con un gran problema, y es que la traficación de los datos recolectados no siempre es la mejor presentable para un usuario final con librerías clásicas como lo son matplotlib y pandas, además estas tienen el inconveniente de ser algo complicadas al momento de modificar para que funcionen a tiempo real.

Algunos problemas que se tienen comúnmente al utilizar otras librerías son:

1. Interactividad: prácticamente hay nula interacción con los gráficos que se generan, solo se muestran los datos recopilados.
2. Incrustación: si podemos incrustar gráficos de otras aplicaciones a matplotlib, pero suele ser un proceso tardado y complicado.
3. Estética: los gráficos exportados por estas librerías, como matplotlib, son básicos para la visualización de datos específicos en el ámbito investigativo y educativo, pero no carecen de estética para funcionar como base para la presentación de un proyecto final ante un cliente.

Por esto y otros motivos, en este proyecto trabajaremos como base en la realización de la dashboard con el FRAMEWORK de Python DASH, el cual es conocido por servir para la creación de aplicaciones web analíticas. En el caso que solo queramos visualización de datos rápidos, prototipos, etc; es siempre recomendable solo usar matplotlib por su buena integración con la librería de Pandas.

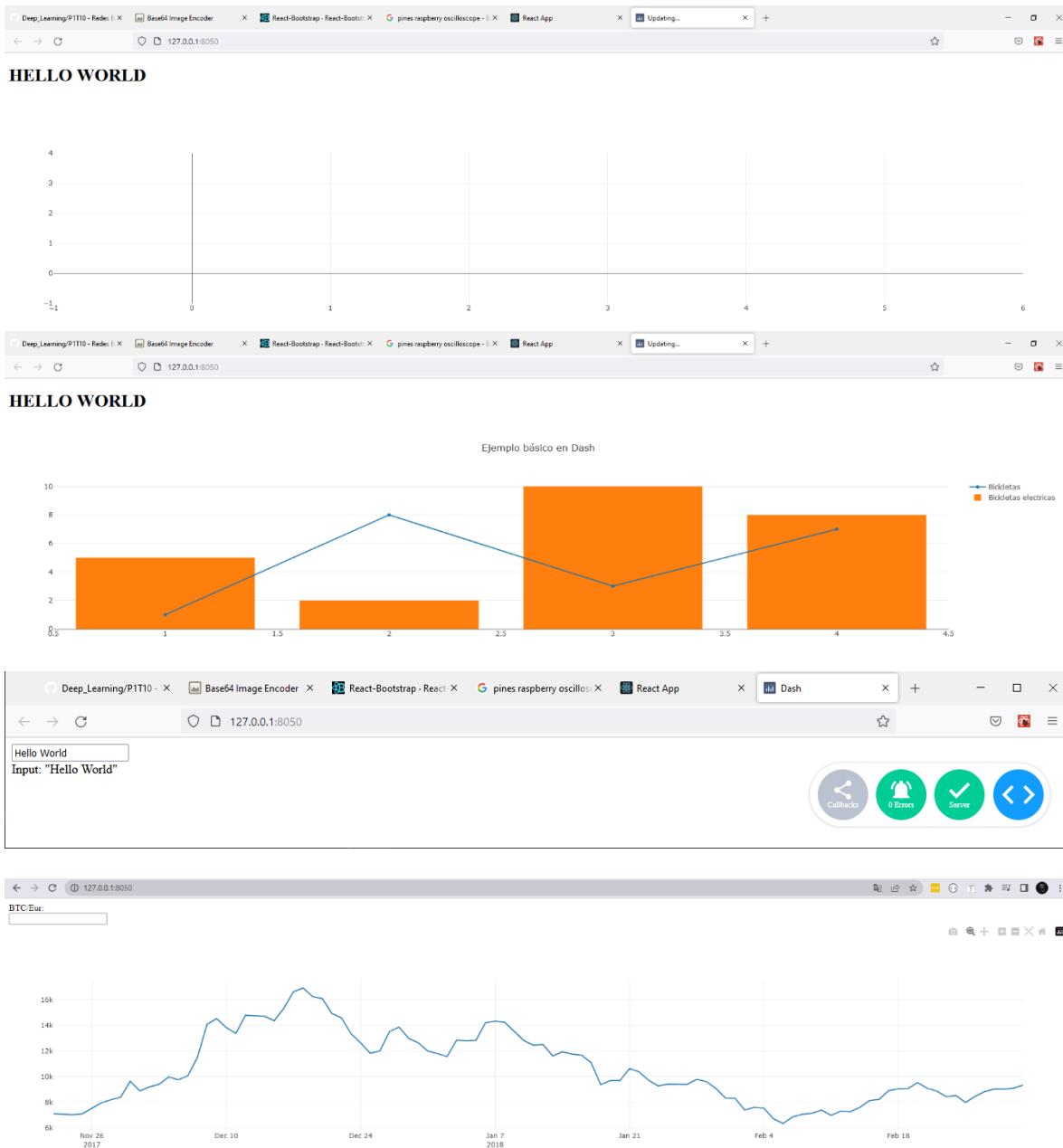
Algunas características que nos ofrece Dash son:

1. Ligereza: Las aplicaciones Dash requieren muy poco tiempo para empezar y son extremadamente ligeras en de Python puro.
2. Control Directo: Dash proporciona una interfaz sencilla para vincular controles de interfaz de usuario, como controles deslizantes, desplegables y gráficos, con el código de análisis de datos de Python.
3. Completa Personalización: Cada elemento estético de una aplicación Dash es personalizable. Las aplicaciones Dash se crean y se publican en la Web, por lo que está disponible toda la potencia de CSS.

Con Dash el pasante inicia un curso en línea para el desarrollo de interfaces web lanzadas desde Python. Algunas de las diferentes interfaces alcanzadas a hacer con este framework son las siguientes:

INFORME FINAL PRÁCTICA EMPRESARIAL

VERSIÓN: 1.0



Para este punto de la Práctica empresarial se había realizado una buena parte del curso de manejo de Dash con Python sin embargo el apartado estético, aunque existía y era personalizable por el usuario, era engorroso de tipar dentro del lenguaje de programación y no daba resultados profesionales a los que se esperaban tener en comparación con otras plataformas especialidades ya existentes. Por este motivo se plantea la idea de poder desarrollar a futuro fuera de la práctica, una dashboard basada en ReactTS en caso de continuar con la idea de una interfaz de uso local.

También se empieza a indagar en el uso de plataformas de uso externo que sean libres, la documentación que poseen al respecto y las formas en que pudiéramos utilizarlas. En este proceso nos hayamos con las siguientes plataformas:

1. Thinger io.

The screenshot shows the Thinger.io Cloud Console interface. The left sidebar has a dark theme with various navigation options: Statistics, Devices, Dashboards (selected), Data Buckets, Endpoints, Access Tokens, Assets, File Storages, Products, Projects, Plugins, Administration, User Accounts, Cluster Hosts, Domains, Brands, and OAuth2 Clients. The main content area is titled "Dashboards". It contains a sub-section titled "Dashboards" with a brief description: "A dashboard is a data visualization tool that transforms, displays, and organizes a collection of data captured and transmitted by IoT devices." Below this is a green button labeled "+ Create a Dashboard". To the right of the text is a 3D rendering of a smartphone displaying the Thinger.io logo, with the word "Dashboards" written in blue next to it. At the bottom left of the main content area is a link "Why use dashboards?". The top of the browser window shows several tabs related to the platform.

2. Shinyapps.io

The screenshot shows the shinyapps.io admin dashboard. The left sidebar has a dark theme with sections: Tablero (selected), Aplicaciones, and Cuenta. The main content area has a light blue header with the shinyapps.io logo and a user account icon. Below the header, there's a message: "En el futuro, puede administrar sus tokens desde la página de tokens del menú de configuración." A section titled "PASO 3: IMPLEMENTAR" contains instructions: "Una vez rsconnect-python que se haya configurado el paquete, estará listo para implementar su primera aplicación ejecutando el siguiente comando en su terminal." with a code snippet: "rsconnect implemente ruta brillante/a/su/aplicación --nombre mecelescis --título nombre-de-su-aplicación". There are two sections at the bottom: "¿QUÉ HAY DE NUEVO?" and "APLICACIONES RECIENTES". The "APLICACIONES RECIENTES" table has three columns: Identificación, Nombre, and Estado. It shows one entry: "Sin aplicaciones recientes". The footer of the page includes copyright information: "© 2022 Posit Software, PBC | Todos los derechos reservados | Términos de Uso".

Para el uso de estas plataformas se destina un correo electrónico y un listado de credenciales nuevas y ajenas alas cuentas naturales del pasante de práctica, las mismas se darán como parte del aporte del estudiante al cliente para que así este ultimo pueda hacer uso de las cuentas y su posterior personalización.

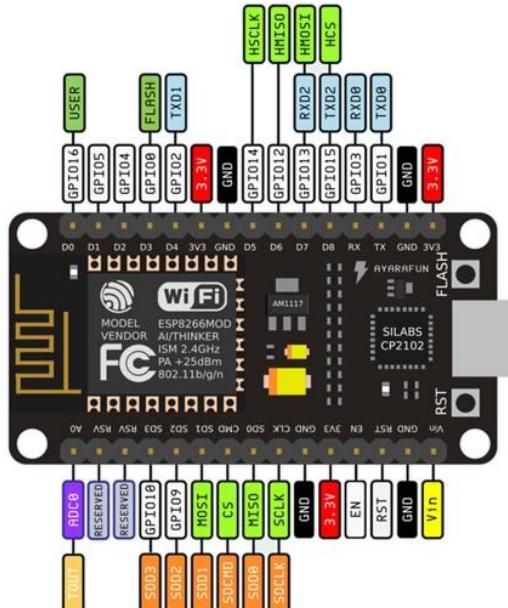
Cada una de las plataformas de Monitoreo ofrece poder elaborar un tablero completo de formas de graficación de datos comprendidos entre diagramas 2D de líneas, diagramas de barras, diagramas de tanques, botones LEDs, representación numérica en tiempo real, etc. Así como también ofrecen la interacción por Callback donde no solo se pueda recibir un dato de la maquina a la plataforma si no que la misma pueda devolver un dato en respuesta, ya sea automático o en acción con la interacción del usuario lo que abre un camino de escalabilidad al proyecto más fuerte.

El problema sin embargo esta en el primer camino, poder llevar los datos capturados por las tarjetas electrónicas hasta la dashboard. En este proceso se observa que existe muy poca documentación al respecto por parte de ambas plataformas, esto comprendido entre manuales, foros, documentos varios y videos en línea. Para este punto la mas desarrollada por el nivel de uso en línea encontrada es Thinger io.

Con Thinger io nos topamos con un fuerte desarrollo de proyectos en apartados del IOT Internet of Things, donde la comunidad y la propia empresa responsable de la plataforma a adecuado una serie de librerías y métodos de emparejamiento que permitan enlazar los datos capturados de forma local y mandarlos a la nube mediante el uso de redes especializadas como:

3. Red LORA
4. Red SigFox
5. Red WIFI
6. Red Bluetooth BLE
7. Red Ziegbe

Visto eso la opción explicada con anterioridad en este proyecto del uso de la tarjeta electrónica ESP8266 como capturadora de datos, toma mayor fuerza, pues la misma posee un módulo integrado e conexión WIFI y existe (aunque poca) documentación de proyecto IOT con este modulo y su enlace con la plataforma.



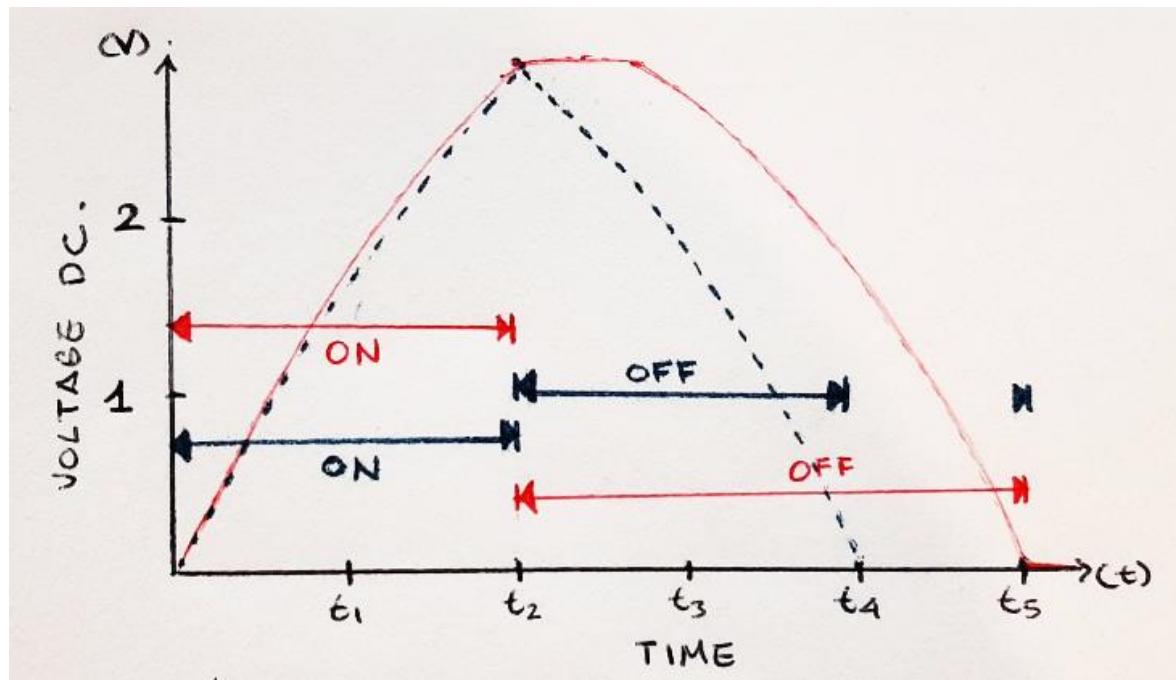
Sin embargo, su uso trastoca un problema y es que para su configuración se necesitan una serie de credenciales para su conexión con los Routers de la Universidad Santo Tomás. También se baraja la idea de trabajar con la misma plataforma, pero con la Raspberry, el problema es que para trabajar con las librerías creadas oficialmente para conectar con Thinger io, supone rehacer el proyecto a cabalidad, despreciando todo lo que se logró avanzar en Python y rehacerlo en los ficheros de conexión de esta librería.

En este estado de la práctica empresarial se resuelve trabajar mientras tanto con la tarjeta de adquisición de datos Arduino Nano y una posible implementación, pero por fuera de práctica empresarial con la tarjeta ESP8266.

4.1.10 Software: GPIOZERO a RPI.GPIO.

Luego de revisar documentación en distintos foros de internet, videos relacionados e informes oficiales de Raspberry, se llegó a la conclusión que la librería GPIOZERO nos provee de una serie de problemas cuando la implementamos, esto en la lectura de datos al tener falsos positivos o acciones nulas desde el programa al comportamiento físico de la tarjeta, como en la comutación la cual es demorada y poco fiable de ejecutar. La solución a esto es replantear el código hasta el momento realizado, pero con la librería RPI.GPIO la cual permite una lectura fidedigna de los datos de sensores analógicos y una comutación con casi cero tiempos de delay entre el encendido y apagado de los pines.

De la siguiente imagen la línea en rojo corresponde al comportamiento del voltaje con la librería GPIOZERO mientras que la línea azul corresponde al voltaje en pines con la librería RPI.GPIO.



Este nuevo comportamiento de los pines por el nuevo uso de otra librería más antigua a la recomendada por la página oficial de Raspberry en su documentación y la tercera edición de su

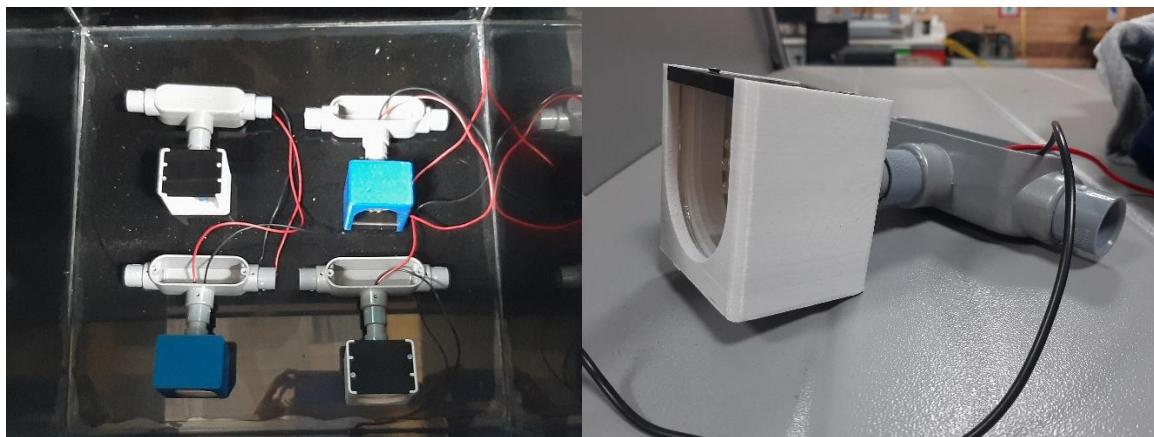
libro de introducción al uso del dispositivo, propone hacer una reedición de los módulos ya existentes de conmutación y lectura de datos.

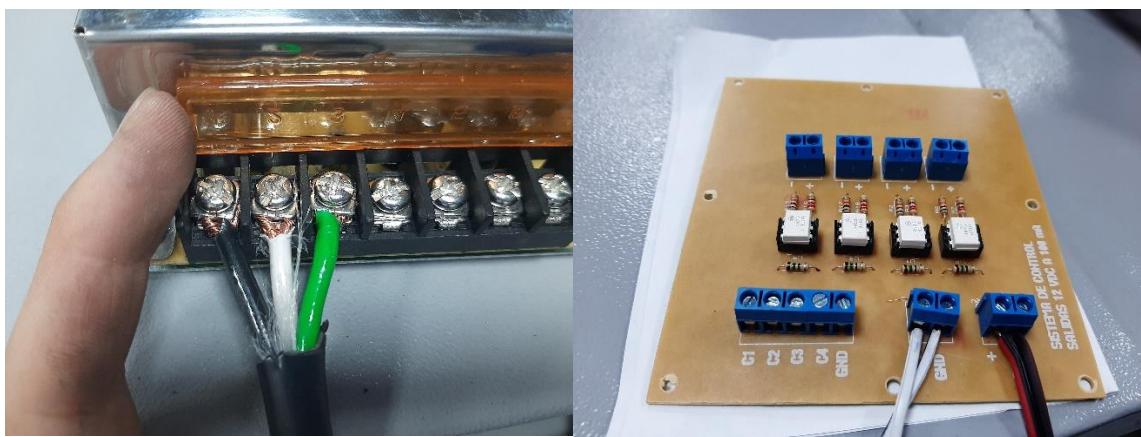
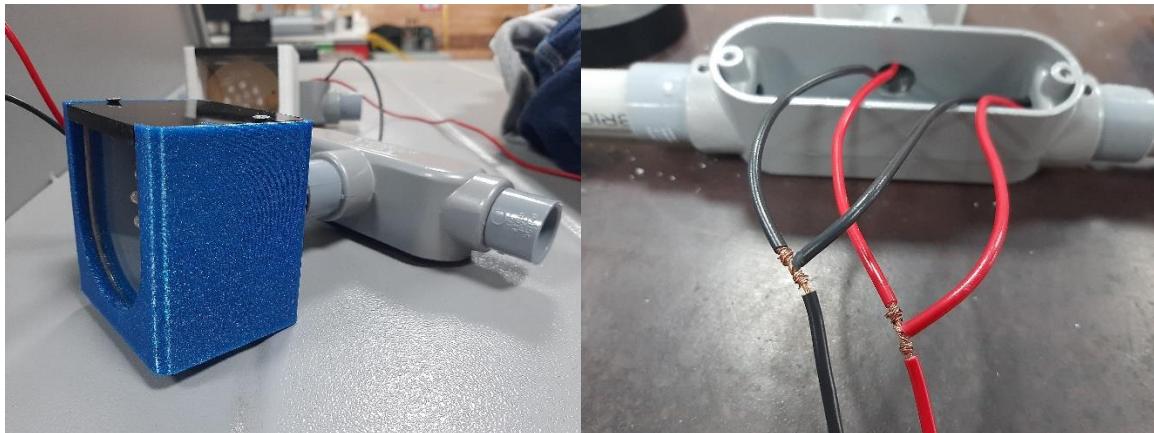
Los nuevos módulos ya no trabajan de la forma MODEL/VIEW/CONTROLLER o por lo menos no de forma estricta, mas ahora trabajan de forma individual y se centran cada uno en pocos archivos con funciones únicas.

Los nuevos códigos en estructura lógica siguen siendo iguales a sus predecesores sin embargo disminuye la cantidad de archivos individuales que se necesitan para su uso y la lógica interna con la que se trabaja fielmente con los 1 y 0 lógicos que se reciben y dan mediante los pines de la GPIO y es que en vez de esperar sucesos desde las librerías de funciones de GPIOZERO, con la librería RPIO.GPIO se puede preguntar directamente por el estado lógico de un pin en concreto, facilitando así la codificación e un programa.

4.1.11 Instrumentación Electrónica y preparación de módulos de Software.

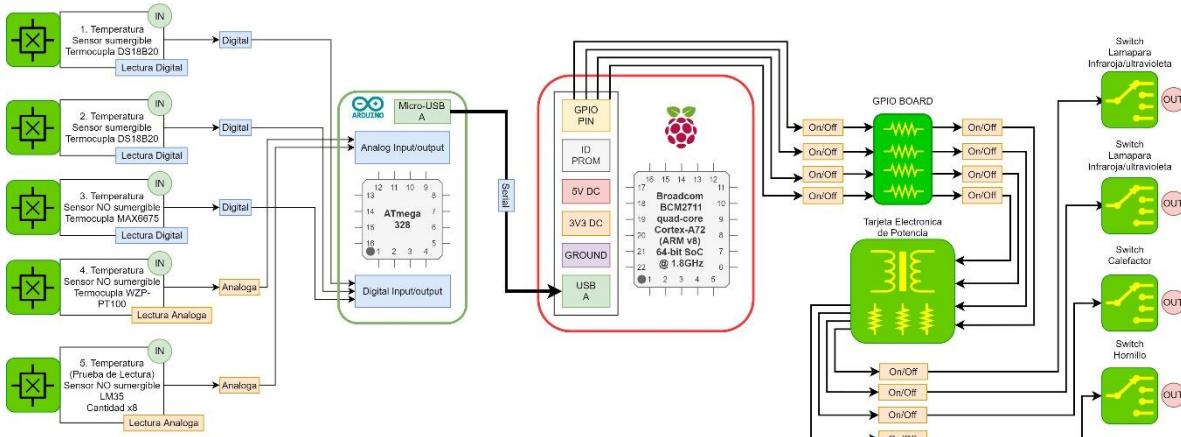
En este ultimo apartado de actividades, el pasante colabora con la adecuación, armado e instrumentación de los diferentes equipos mecánicos, eléctricos y placas electrónicas desarrolladas para la maquina de oxidación de materiales, a la vez que se va robando y adecuando los diferentes módulos de software con la conexión directa de la boards desarrolladas con el ordenador personal de la Raspberry PI 400. Algunas imágenes de este apartado que a la fecha sigue vigente son las siguientes:







La instrumentación electrónica y la adecuación de los módulos de software realizados en la práctica empresarial sobre los dispositivos instalados es una labor que aun se sigue desarrollando. La conexión del ordenador personal Raspberry PI 400 con el resto de la maquina se puede resumir en el siguiente grafico el cual es de tener en cuenta al momento de realizar todas aquellas conexiones digitales y físicas que sean necesarias.



Descripción de la actividad	Fecha inicio	Fecha entrega	Producto entregado
Instalación y configuración de la ISO Raspberry PI Debian.	01-06-2022	02-12-2022	Raspberry PI 400 con SO operativo montado actualizado y configurado para el desarrollo del proyecto.
Actualizar Raspberry PI OS Debian	01-06-2022	02-12-2022	Comandos correspondientes a Debian listos en un archivo BAT para que puedan ser ejecutados por el usuario final para actualizar diariamente el equipo.
Preparación de Entorno de Trabajo entre Windows y Linux u otros dispositivos requeridos.	01-06-2022	En proceso	Preparación del entorno de trabajo a nivel visual configuración de “ventanas” y habilitación de propiedades propias del modelo para el funcionamiento adecuado con el proyecto.
Sistema de lectura y acción GPIO de la RaspberryPI400: Comprensión y adecuación.	01-06-2022	02-12-2022	Adecuación e interpretación de los puertos de comunicación GPIO para programación Física, aspecto clave del proyecto.
Diseño y Fabricación de Placa electrónica para la Board GPIO.	01-06-2022	02-12-2022	PCB electrónica en la cual insertar la Board GPIO para un correcto y como uso de la GPIO de las Raspberry Pi 400 en la máquina de oxidación de materiales.
Software Switch: Comutación a partir de la RaspberryPI 400 y el canal de GPIO.	01-06-2022	02-12-2022	Módulos de software y adaptación de nuevas características de comunicación de la GPIO con el medio exterior para comutación de pines.
Software Sensores: Lectura de sensores Digitales mediante GPIO.	01-06-2022	02-12-2022	Adaptación del modulo 1-Wire propio de la Raspberry GPIO con los módulos de software desarrollados en la practica para la lectura de

			sensores digitales.
Software Sensores: Lectura de sensores con Microcontrolador Externo.	01-06-2022	02-12-2022	Módulos de software para la lectura de sensores analógicos y digitales por comunicación paralela en placas electrónicas externas a la Raspberry PI 400.
Software Dashboard: Propuestas a futuro de la realización de una interfaz gráfica para el apartado del monitoreo	01-06-2022	02-12-2022	Documentación y módulos de software de prueba para la recomendación del uso de plataformas de terceros y/o plataformas locales para el monitoreo de la sensorica de la máquina de oxidación de materiales.
Software: GPIOZERO a RPI.GPIO	01-06-2022	02-12-2022	Adecuación de todos los módulos iniciales de conmutación y lectura de datos de la Raspberry PI 400 al cambiar de librería y lógica aplicada en los diferentes procesos.
Instrumentación Electrónica y preparación de módulos de Software.	01-06-2022	En Proceso	Implementación y despliegue sobre la máquina de oxidación de materiales de los diferentes dispositivos, electrónicos, mecánicos y eléctricos. A su vez la adecuación final de los módulos de iniciales de conmutación y lectura de sensores para el correcto funcionamiento inicial de la máquina.

4.2. Herramientas utilizadas en el desarrollo del proyecto

Herramienta utilizada	Para qué la utilizó
Herramienta varia como taladros, cortafríos, multímetro digital, bisturí, alambre de cobre, cable d cobre, alicates, pinzas, termo encogible, estaño, crema para soldar, extensiones eléctricas, tubos de PVC, etc.	Instrumentación electrónica.
Proteus 8	Diseño y fabricación de PCB

Visual Studio Code	Programación de módulos de Software.
Raspberry Debian	Sistema operativo Linux en el que se desarrolló el proyecto
Rasberry PI 400, Cargador Raspberry, memoria Sandisk 10 gen, conector microHDMI/HDMI	Ordenador personal Raspberry donde se desarrolló el proyecto.
Lector de tarjetas SD a USB-A	Adaptador para instalar el Sistema operativo.
Github	Para mantener un versionamiento de software.
Evernote	Para realizar las bitácoras y procedimientos generales que se realizaron en la práctica y en base a esto realizar el presente documento.
Thinger io, Gmail, ShinyApps.	Plataformas utilizadas para generar las cuentas necesarias para la propuesta de Dashboard.
Elementos diversos eléctricos, electrónicos y mecánicos.	Dispositivos y materiales utilizados para la implementación de la maquina de oxidación de materiales.
Arduino Nano y ESP8266.	Tarjetas electrónicas externas para la adquisición de datos de sensores digitales y análogos.
Draw io	Para realizar gran parte de los diagramas presentados en este documento.

5. CONCLUSIONES

1. En la presente práctica empresarial se logró realizar la adecuación de los diferentes módulos de software necesarios para el correcto funcionamiento de la máquina de oxidación de materiales, así como la correcta configuración del ambiente y propiedades de funcionamiento del ordenador personal Raspberry PI 400. Este apartado es sumamente importante pues el software es el mediador directo entre las acciones que quiere ejecutar el usuario sobre la maquina y los diferentes dispositivos mecánicos, eléctricos y electrónicos que lo conforman, siendo así una práctica orientada a la programación Física.
2. A pesar que los diferentes módulos de software entregados requieren de una adaptación posterior a la práctica, estos significan un inicio en la escalación del proyecto y la realización de la máquina de oxidación de materiales hasta llevarla a un punto de pleno funcionamiento.
3. Se recomendó diferentes opciones de configuración y adaptación de nuevos módulos electrónicos y de software frente a las diferentes dificultades que se fueron observando en la realización de este proyecto como lo fue la engorrosa lectura de los sensores digitales

directamente en la Raspberry Pi, adecuando microcontroladores externos que suplan esta función.

4. Se logro un satisfactorio acompañamiento en el apartado de instalación e instrumentación de los diferentes dispositivos eléctricos, mecánicos y electrónicos que conforman la máquina de oxidación de materiales.
5. Se logro diseñar y presentar una PCB electrónica que ayudase con la cómoda conexión de dispositivos electrónicos a la GPIO de la Raspberry dentro de la máquina de oxidación de materiales.
6. Se propuso diferentes formas en las que se puede implementar el uso de Dashboard en la máquina de oxidación de materiales para un posterior desarrollo que cumpla con los requerimientos de accesibilidad y usabilidad de la máquina de oxidación de materiales y sus variables ambientales.