

# CONEXIÓN DE UNA PANTALLA LCD 16x2 A UNA RASPBERRY PI 3 - MODELO B PARA LA PRESENTACIÓN DE MENSAJES

## 1 PLANTEAMIENTO DEL PROBLEMA

Se desconoce el funcionamiento, control y conexión de una Raspberry Pi a una pantalla LCD 16x2 para la presentación de mensajes, para lo cual se formularon las siguientes preguntas:

- ¿Cuál es el sistema operativo que utiliza la Raspberry Pi?
- ¿Cuál es el lenguaje de programación utilizado para controlar la pantalla LCD 16x2 para presentación de mensajes?
- ¿Cómo funciona la Pantalla LCD 16x2?

## 2 OBJETIVOS

### A. Objetivo general

Conectar una pantalla LCD 16x2 a una Raspberry Pi 3 - Modelo B para proyectar un mensaje en la LCD.

### B. Objetivos específicos

- Comprender la estructura de control básica de Python para controlar una pantalla LCD 16X2 mediante una Raspberry Pi.
- Conocer la función y estructura de cada pin que integra una LCD 16x2.
- Identificar los pines de configuración y las conexiones necesarias para establecer comunicación entre la pantalla LCD 16x2 y la Raspberry Pi.

## 3 ESTADO DEL ARTE

En 2018, Dr. IS Akila, Akshaya, Deepthi y Sivadharshini del Department of ECE, Coimbatore Institute of Technology, desarrollaron un lector de texto para discapacitados visuales utilizando Raspberry Pi, basado en un prototipo que ayuda al usuario a escuchar. Esto implica la extracción de texto de la imagen y convertir el texto a la salida de audio traducida al idioma inglés. Esto se hace usando una Raspberry Pi 3 modelo B y un módulo de cámara con el conceptos de Tesseract OCR [carácter óptico Motor de reconocimiento] y API de Google Speech [Application Program Interface] que es el textual entrada al motor del habla. El modelo esta programado utilizando el Lenguaje Python(Akshaya,2018,p.1) [1].

Mario Pattichis, Silvia Celedon-Pattichis y Carlos Lopez del Department of Electrical and Computer Engineering y Department of Language de la University of New Mexico en 2017, desarrollaron un proyecto de enseñanza de imágenes y procesamiento de video utilizando las matemáticas de la escuela media y la Raspberry Pi. El proyecto de ingeniería AOLME utiliza un currículum integrado que se basa en el uso de bibliotecas de Python que se ejecutan en la Raspberry Pi(Pattichis,2017,p.1) [2].

En general se demostró que el aprendizaje se facilita cuando se emplea una representación de imágenes y video digitales mientras se trabaja en un entorno de programación real trabajando en Linux y no requirió el uso de interfaces gráficas de usuario para aprender a programar(Pattichis,2017,p.1) [2].

En 2015, Manisha Sharma, Nidhi Agarwal, Dr. SRN Reddy realizaron el diseño y desarrollo de una placa hija para comunicación USB-UART entre Raspberry Pi y PC. Desarrollo de aplicaciones en tiempo real utilizando plataformas como Raspberry Pi que a menudo requieren Interfaz de diferentes sensores y comunicación. Pueden interconectarse fácilmente usando un puerto RS232, pero hoy en día se sustituye por USB. Este trabajo propone el diseño y desarrollo de una placa para comunicación USB a UART a la Raspberry Pi que proporciona una comunicación con diferentes dispositivos mediante el uso de USB a UART, con la ayuda de estos convertidores, se logra la transmisión de datos bidireccional FT232RL los IC CP2102 se utilizan en las juntas secundarias propuestas (Sharma,2015,p.1)[3].

En 2016, Ing. Felix Palacios desarrolló un Sistema de identificación de huella dactilares para el acceso a lugares restringidos y control de asistencia con protocolo de comunicación IEEE 802.11. El presente trabajo muestra la implementación de un Sensor Biométrico GT511C3, un Arduino Nano, una Raspberry Pi 2.0, una pantalla táctil LCD 3.5" y un módulo USB Wifi, el entorno de programación para el desarrollo del aplicativo Web, se hizo mediante los módulos Django y Adafruit sobre la base de Python. El sistema tiene la capacidad de registrar huellas de personas, las cuales se les asigna un identificador entero de entre 0 a 199 y almacenarlo en una base de datos instalada como es MySQL, además tiene la capacidad de almacenar nombres asociados al identificador generado, en la base de datos con el cual se puede habilitar el acceso hacia algún lugar específico(Palacios,2016,p.1) [4].

Holger Romero, Alonso Pacheco y Hernan Ramos estudiantes de la Universidad Católica de Santa María, Perú en 2017 implementaron un sistema de monitoreo para supervisar el consumo de energía eléctrica doméstica; esta propuesta requirió la utilización de la placa Arduino Mega, shield Ethernet, el sensor de corriente y la pantalla LCD. Para el monitoreo del consumo eléctrico local se utilizó la pantalla LCD y para un monitoreo remoto se desarrolló una página web alojada en el Shield Ethernet que permite analizar en tiempo real los eventos y alertas mediante cualquier navegador web. El proyecto brindó la suficiente información a los usuarios para controlar su consumo de energía y así tomar la mejor decisión para generar un ahorro continuo y fortalecer una conciencia amigable con el buen uso de la energía eléctrica(Romero,2017,p.1) [5].

Para el trabajo de investigación presente, se utilizó el entorno de programación del módulo Adafruit dirigido a una Raspberry Pi 3 modelo B y una pantalla LCD 16x2 (Palacios Felix, 2016), el entorno de programación para el aplicativo se hizo sobre la base de Python (IS Akila, Akshaya, Deepthi, Sivadharshini, 2018). El sistema tiene la capacidad de proyectar un mensaje en la pantalla LCD y borrarlo después de una previa instrucción (Manisha Sharma, Nidhi Agarwal, SRN Reddy, 2015).

## 4 MARCO TEÓRICO

### 4.1 Raspberry Pi 3 – modelo B

Raspberry Pi es un ordenador de placa reducida, ordenador de placa única u ordenador de placa simple (SBC) de bajo costo desarrollado en el Reino Unido.

Aunque no se indica expresamente si es hardware libre o con derechos de marca, es compatible con varios Sistemas operativos de libre acceso, siendo su sistema operativo oficial una versión adaptada de Debian, denominada Raspbian, aunque permite usar otros sistemas operativos, promueve principalmente el aprendizaje del lenguaje de programación Python – IDLE, otros lenguajes también soportados son: Tiny BASIC,11 C, Perl4 y Ruby.12

La Raspberry Pi 3 está construida alrededor del nuevo procesador BCM2837 ARMv8 de 64bits con 1,2GHz de velocidad, mucho más rápido y con mayor capacidad de procesamiento que sus antecesores, integra el chip BCM43143 que la dota con conectividad Wifi b/g/n y Bluetooth 4.1 LE de bajo consumo y cuenta con administración de energía mejorada que permite trabajar con más dispositivos USB externos, permite usar fuentes de hasta 2.5A para proveer más energía a los puertos USB.

## Pines GPIO Raspberry Pi 3 - Modelo

La distribución de los pines GPIO del modelo Raspberry Pi 3 es la siguiente:

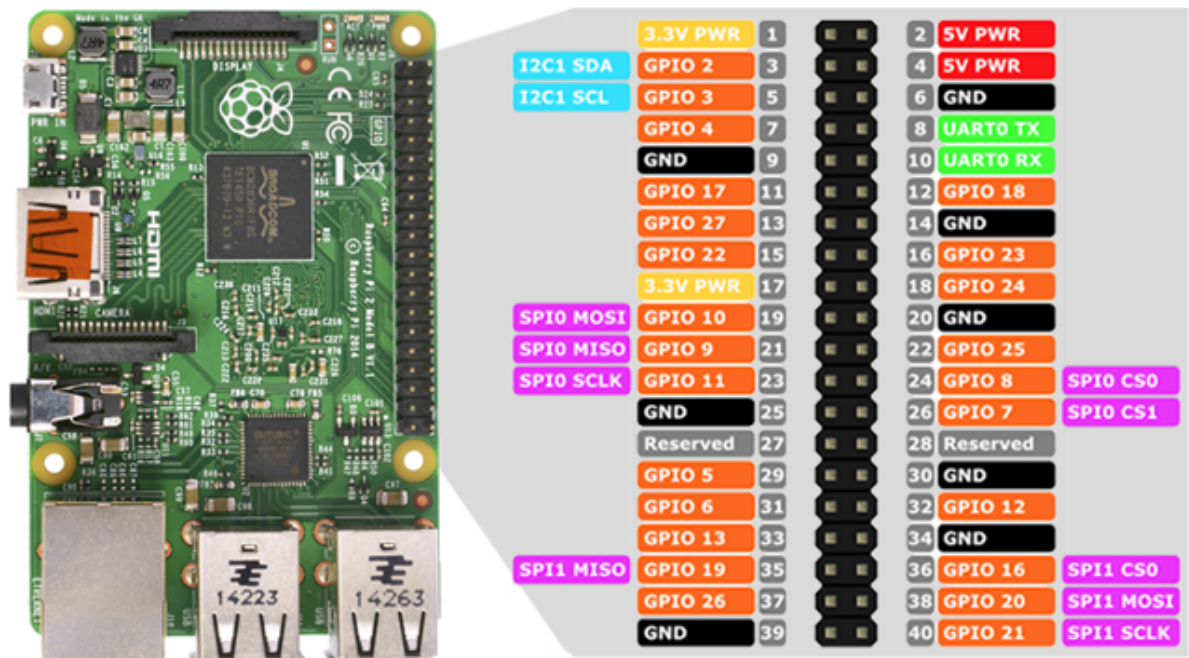


Fig 1. Pines GPIO para Raspberry Pi 2 Modelo B

- Amarillo (2): Alimentación a 3.3V.
- Rojo (2): Alimentación a 5V.
- Naranja (26): Entradas / salidas de propósito general. Pueden configurarse como entradas o salidas. Ten presente que el nivel alto es de 3.3V y no son tolerantes a tensiones de 5V.
- Gris (2): Reservados
- Negro (8): Conexión a GND o masa.
- Azul (2): Comunicación mediante el protocolo I2C para comunicarse con periféricos que siguen este protocolo
- Verde (2): Destinados a conexión para UART para puerto serie convencional
- Morado (5): Comunicación mediante el protocolo SPI para comunicarse con periféricos que siguen este protocolo.

Todos los pines son de tipo "unbuffered", es decir, no disponen de buffers de protección y puedes dañar la placa con un mal uso. Existen 2 formas de numerar los pines de la Raspberry Pi, en modo GPIO o en modo BCM. En el modo GPIO, los pines se numeran de forma física por el lugar que ocupan en la placa (representados por el color gris) viene siendo igual para todas las versiones (comenzamos a contar desde arriba a la izquierda y finalizamos abajo a la derecha).

## 4.2 Python – IDLE

IDLE (Integrated Development and Learning Environment) es un entorno de desarrollo integrado (IDE) para Python. El instalador de Python para Windows contiene el módulo IDLE de manera predeterminada. Debe instalarse utilizando los respectivos gestores de paquetes.

IDLE se puede usar para ejecutar una sola declaración como Python Shell y también para crear, modificar y ejecutar scripts de Python. IDLE proporciona un editor de texto con todas las funciones para crear scripts de Python que incluye funciones como resaltado de sintaxis, autocompletado y sangría inteligente. También tiene un depurador con características de pasos y puntos de interrupción.

## Módulo Adafruit CircuitPython CharLCD

Este módulo le permite escribir fácilmente el código de Python que controla una LCD de caracteres (ya sea con luz de fondo individual o con luz de fondo RGB). Se puede usar con cualquier placa de microcontrolador CircuitPython o con una computadora que tenga GPIO y Python gracias a la librería Adafruit *blinka*.

**”import adafruit\_character\_lcd.character\_lcd as characterlcd”** Si está utilizando una pantalla

LCD con retroiluminación RGB, recuerde que cada uno de los canales de color rojo, verde y azul debe conectarse a un pin de salida compatible con PWM en su computadora (pines 10, 12, 18, 21 en la Raspberry Pi) .

## 4.3 Pantalla LCD 16x2

LCD 16x2 se refiere a un pequeño dispositivo con pantalla de cristal líquido que cuenta con 2 filas, de 16 caracteres cada una, que se utiliza para mostrar información, por lo general alfanumérica. Las capacidades de estos dispositivos son altas, pues se puede mostrar todo tipo de información sin importar qué tipo de símbolos o caracteres sea, el número de píxeles que tiene cada símbolo o carácter varía dependiendo del modelo del dispositivo y cada artefacto.

Está controlada por un microcontrolador que está programado para dirigir el funcionamiento y la imagen mostrada en la pantalla, este tipo de artefacto tiene mínimo consumo de energía o corriente eléctrica, la programación es sumamente sencilla y es por lo general cargada por el fabricante.

## Pines de la Pantalla LCD 16X2

La distribución de los pines en la Pantalla LCD 16x2 es la siguiente:

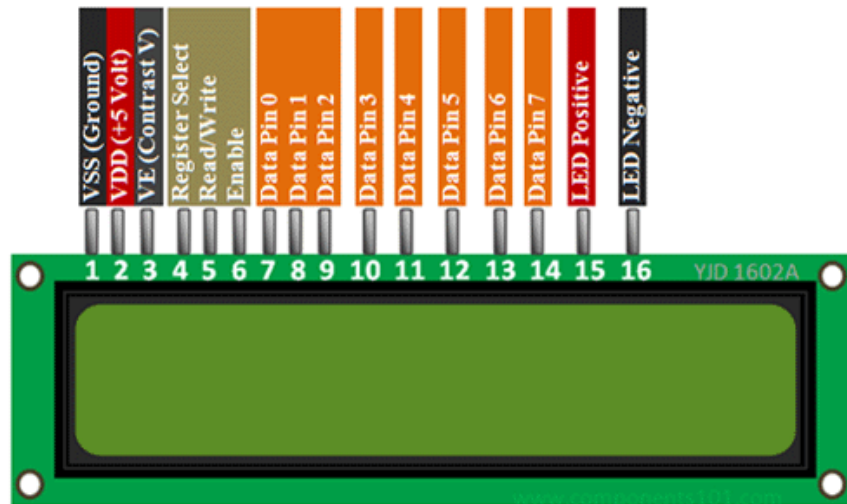


Fig 2. Pines LCD 16x2

- **Pines de alimentación**

- **Vss:** Gnd (Tierra)
- **Vdd:** Alimentación +5 voltios
- **Vee:** Corresponde al pin de contraste, lo regularemos con un potenciómetro de 10K conectado a RS.

- **Pines de control**

- **RS:** Corresponde al pin de selección de registro de control de datos (0) o registro de datos(1). Es decir el pin RS funciona paralelamente a los pines del bus de datos. Cuando RS es 0 el dato presente en el bus pertenece a un registro de control/instrucción. y cuando RS es 1 el dato presente en el bus de datos pertenece a un registro de datos o un carácter.
- **RW:** Corresponde al pin de Escritura(0) o de Lectura(1). Nos permite escribir un dato en la pantalla o leer un dato desde la pantalla.
- **E:** Corresponde al pin Enable o de habilitación. Si E(0) esto quiere decir que el LCD no esta activado para recibir datos, pero si E(1) se encuentra activo y podemos escribir o leer desde el LCD.

- **Pines de Bus de datos**

- El Bus de datos bidireccional comprende desde los pines D0 a D7. Para realizar la comunicación con el LCD podemos hacerlo utilizando los 8 bits del bus de datos(D0 a D7) o empleando los 4 bits mas significativos del bus de datos(D4 a D7).

## 5 DIAGRAMAS

## 5.1 DIAGRAMA DE FLUJO

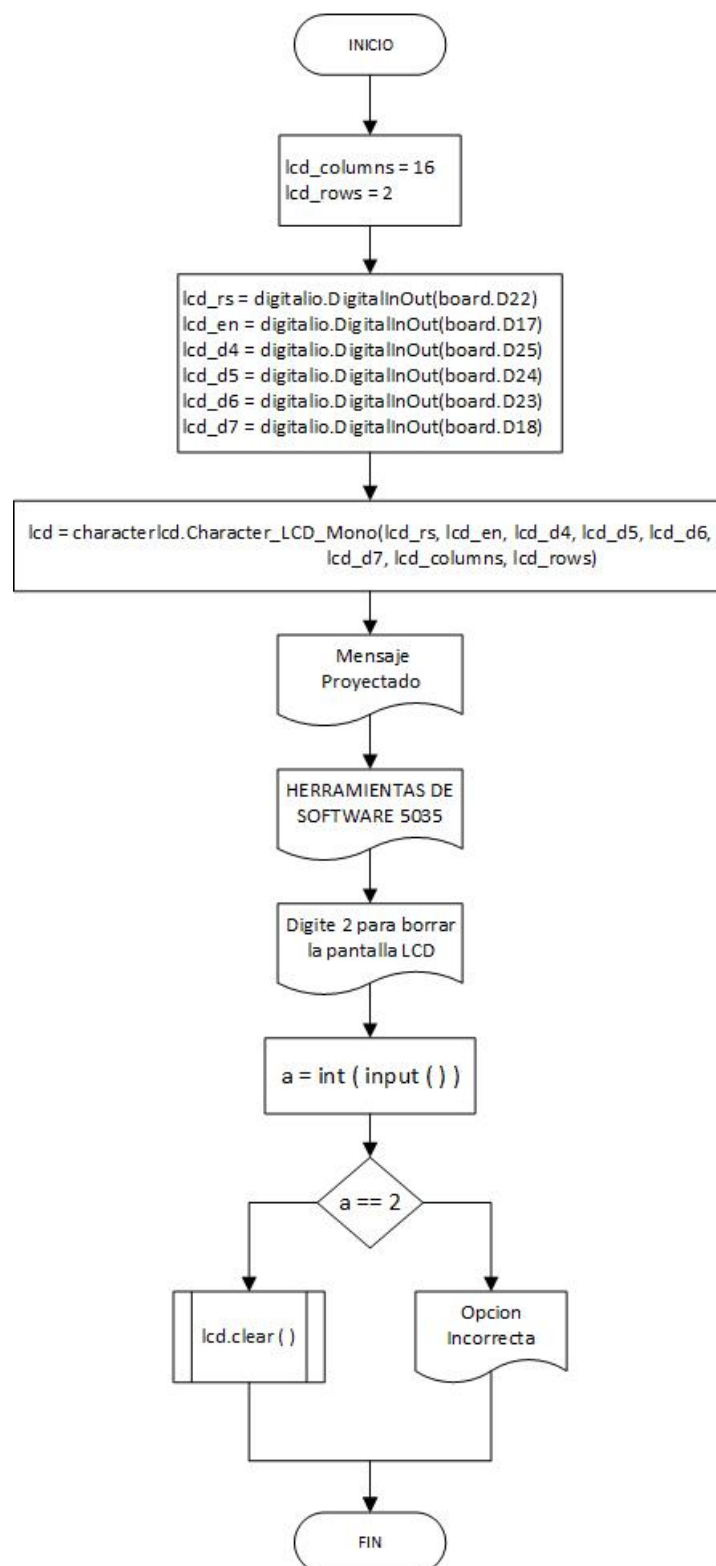


Fig 3. Diagrama de Flujo

## 5.2 DIAGRAMA ELECTRÓNICO

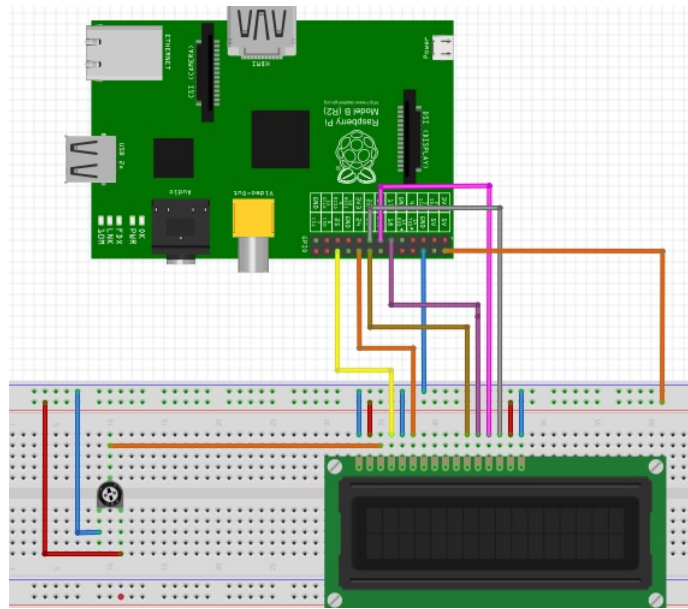


Fig 4. Diagrama Electrónico

## 6 LISTA DE COMPONENTES

- Raspberry Pi
- raspbian
- Python 3IDLE
- 1 Cable HDMI a VGA
- Potenciómetro 10K
- Cables conectores
- Pantalla LCD 16x2
- Proto board
- Micro SD de 8 GB

## 7 MAPA DE VARIABLES

VARIABLE	DEFINICIÓN OPERACIONAL
Tarjeta Micro SD 8 GB	En este dispositivo de almacenamiento guardamos el sistema operativo Raspbian, para la ejecución de la Raspberry Pi.
Raspberry Pi	Miniordenador que sirve para realizar la interfaz con la pantalla LCD.
Pantalla LCD 16 x 2	Dispositivo empleado para la visualización de contenidos o información gráfica.
Pines Raspberry Pi	Pin de tierra, conectado a los pines VSS, RW, K de la LCD
GND (Pin 6)	
+ 5V (Pin 2)	Es el pin de alimentación que proporciona 5V de energía, va conectado al pin VDO y A de la LCD
GPIO 25 (Pin 22)	Sirven para establecer la comunicación entre la Raspberry Pi y la pantalla LCD, van conectados a los pines D4 , D5, D6, D7 respectivamente.
GPIO 24 (Pin 18)	
GPIO 23 (Pin 16)	
GPIO 18 (Pin 12)	
GPIO 22 (Pin 15)	
GPIO 17 (Pin 11)	Sirve para establecer la selección de modo; se conecta al pin RS de la LCD
	Permite habilitar o deshabilitar la comunicación con la LCD, se conecta al pin E



## 8 EXPLICACIÓN DEL CÓDIGO FUENTE



```
*untitled*
File Edit Format Run Options Window Help

#-----LIBRERIAS-----

import board      # Esta libreria sirve para asignar constantes fijas a los pines
                  # del tablero, esto hace que usar el módulo de la placa sea
                  # más seguro y confiable

import digitalio   # Este módulo sirve utilizar la pantalla LCD con una
                  # retroalimentación de un solo color, en el caso que se
                  # desee una retroalimentación RGB (varios colores) se debe
                  # definir los pines de salida según su ubicación en el board
                  # y el color

import adafruit_character_lcd.character_lcd as characterlcd
|
    # Este módulo permite escribir fácilmente el código de Python
    # que controla una LCD de caracteres (ya sea con luz de fondo
    # individual o con luz de fondo RGB)

#----- TAMAÑO DE LA PANTALLA LCD -----

lcd_columns = 16   # Estas líneas de código definen el tamaño de caracteres de
lcd_rows = 2       # nuestra LCD, en este caso la pantalla LCD es de 16 columnas
                  # y 2 filas, si se tiene una LCD de caracteres de diferente
                  # tamaño se debe modificar estos valores

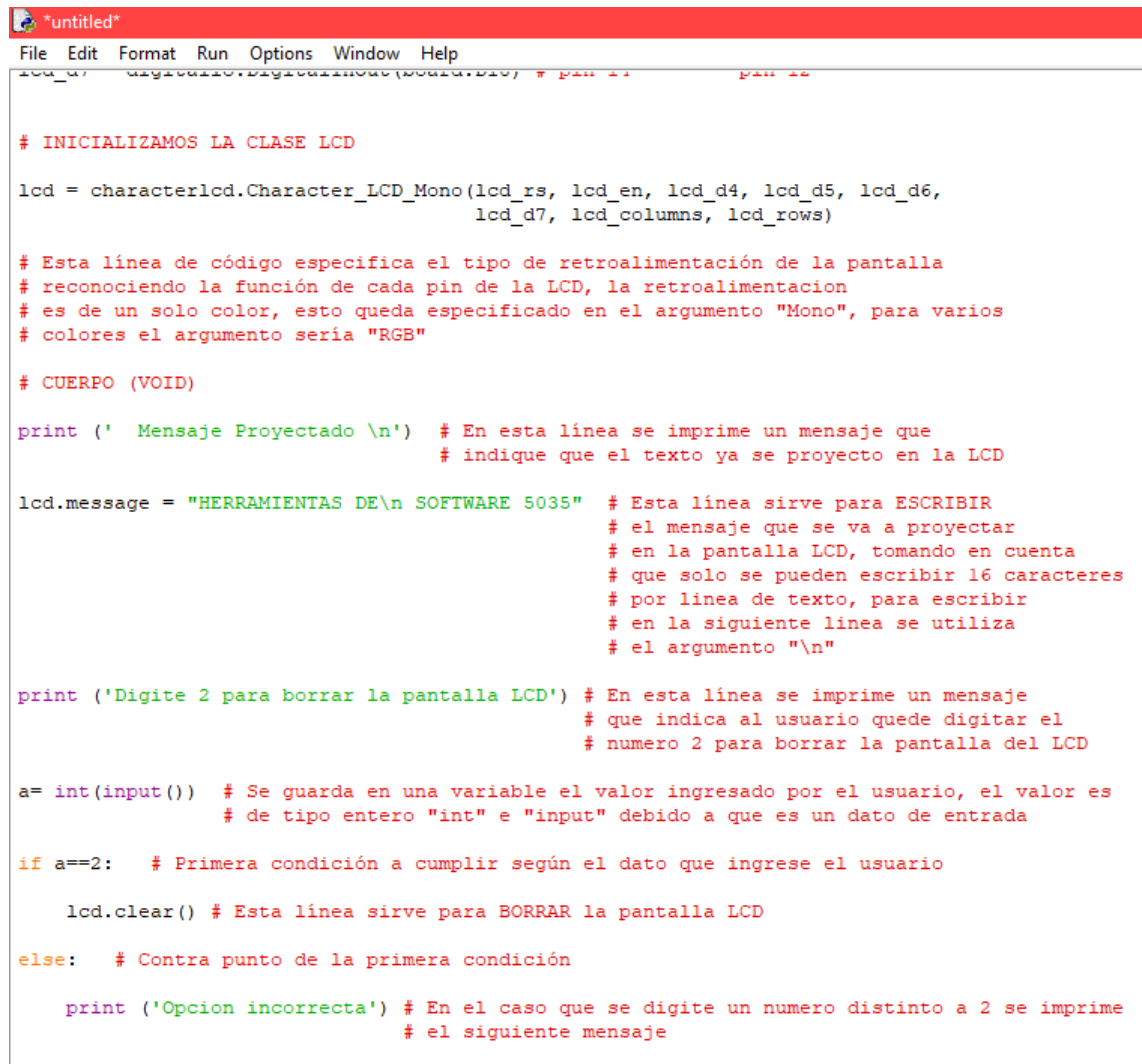
#---- ASIGNACIÓN DE LOS PINES DE CONFIGURACIÓN DE LA RASPBERRY PI Y LCD -----

#Según la ubicación en la placa :          -LCD-      -RASPBERRY PI 3-MODELO B-
lcd_rs = digitalio.DigitalInOut(board.D22) # pin 4          pin 15
lcd_en = digitalio.DigitalInOut(board.D17) # pin 6          pin 11
lcd_d4 = digitalio.DigitalInOut(board.D25) # pin 11         pin 22
lcd_d5 = digitalio.DigitalInOut(board.D24) # pin 12         pin 18
lcd_d6 = digitalio.DigitalInOut(board.D23) # pin 13         pin 16
lcd_d7 = digitalio.DigitalInOut(board.D18) # pin 14         pin 12

# INICIALIZAMOS LA CLASE LCD

lcd = characterlcd.Character_LCD_Mono(lcd_rs, lcd_en, lcd_d4, lcd_d5, lcd_d6,
                                     lcd_d7, lcd_columns, lcd_rows)
```

Fig 5. Código Fuente



```
*untitled*
File Edit Format Run Options Window Help

# INICIALIZAMOS LA CLASE LCD

lcd = characterlcd.Character_LCD_Mono(lcd_rs, lcd_en, lcd_d4, lcd_d5, lcd_d6,
                                       lcd_d7, lcd_columns, lcd_rows)

# Esta línea de código especifica el tipo de retroalimentación de la pantalla
# reconociendo la función de cada pin de la LCD, la retroalimentación
# es de un solo color, esto queda especificado en el argumento "Mono", para varios
# colores el argumento sería "RGB"

# CUERPO (VOID)

print (' Mensaje Proyectado \n') # En esta línea se imprime un mensaje que
                                # indique que el texto ya se proyectó en la LCD

lcd.message = "HERRAMIENTAS DE\n SOFTWARE 5035" # Esta línea sirve para ESCRIBIR
                                                # el mensaje que se va a proyectar
                                                # en la pantalla LCD, tomando en cuenta
                                                # que solo se pueden escribir 16 caracteres
                                                # por línea de texto, para escribir
                                                # en la siguiente línea se utiliza
                                                # el argumento "\n"

print ('Digite 2 para borrar la pantalla LCD') # En esta línea se imprime un mensaje
                                                # que indica al usuario quede digitar el
                                                # número 2 para borrar la pantalla del LCD

a= int(input()) # Se guarda en una variable el valor ingresado por el usuario, el valor es
               # de tipo entero "int" e "input" debido a que es un dato de entrada

if a==2: # Primera condición a cumplir según el dato que ingrese el usuario
    lcd.clear() # Esta línea sirve para BORRAR la pantalla LCD
else: # Contra punto de la primera condición
    print ('Opcion incorrecta') # En el caso que se digite un número distinto a 2 se imprime
                                # el siguiente mensaje
```

Fig 6. Código Fuente

## LIBRERIAS

- *import board*

Esta librería sirve para asignar constantes fijas a los pines del tablero, esto hace que usar el módulo de la placa sea más seguro y confiable

- *import digitalio*

Este módulo sirve utilizar la pantalla LCD con una retroalimentación de un solo color, en el caso que se desee una retroalimentación RGB (varios colores) se debe definir los pines de salida según su ubicación en el board y el color

- *import adafruit\_character\_lcd.character\_lcd as characterlcd*

Este módulo permite escribir fácilmente el código de Python que controla una LCD de caracteres (ya sea con luz de fondo individual o con luz de fondo RGB)

## TAMAÑO DE LA PANTALLA LCD

- `lcd_columns = 16`
- `lcd_rows = 2`

Estas líneas de código definen el tamaño de caracteres de la LCD, en este caso la pantalla LCD es de 16 columnas y 2 filas, si se tiene una LCD de caracteres de diferente tamaño se debe modificar estos valores

## ASIGNACIÓN DE LOS PINES DE CONFIGURACIÓN DE LA RASPBERRY PI Y LCD

Según la ubicación en la placa:	LCD	RASPBERRY PI 3-MODELO B
<code>lcd_rs = digitalio.DigitalInOut(board.D22)</code>	pin4	pin15
<code>lcd_en = digitalio.DigitalInOut(board.D17)</code>	pin6	pin11
<code>lcd_d4 = digitalio.DigitalInOut(board.D25)</code>	pin 11	pin22
<code>lcd_d5 = digitalio.DigitalInOut(board.D24)</code>	pin12	pin18
<code>lcd_d6 = digitalio.DigitalInOut(board.D23)</code>	pin 13	pin16
<code>lcd_d7 = digitalio.DigitalInOut(board.D18)</code>	pin 14	pin12

## INICIALIZAMOS LA CLASE LCD

- `lcd = characterlcd.Character_LCD_Mono(lcd_rs, lcd_en, lcd_d4, lcd_d5, lcd_d6, lcd_d7, lcd_columns, lcd_rows)`

Esta línea de código especifica el tipo de retroalimentación de la pantalla reconociendo la función de cada pin de la LCD, la retroalimentación es de un solo color, esto queda especificado en el argumento "Mono", para varios colores el argumento sería "RGB"

## CUERPO (VOID)

- `print (' Mensaje Proyectado ')`

En esta línea se imprime un mensaje que indique que el texto ya se proyectó en el LCD

- `lcd.message = "HERRAMIENTAS DE SOFTWARE 5035"`

Esta línea sirve para ESCRIBIR el mensaje que se va a proyectar en la pantalla LCD, tomando en cuenta que solo se pueden escribir 16 caracteres por línea de texto

- `print ('Dígitate 2 para borrar la pantalla LCD')`

En esta línea se imprime un mensaje que indica al usuario que digite el número 2 para borrar la pantalla del LCD

- `a = int(input())`

Se guarda en una variable el valor ingresado por el usuario, el valor es de tipo entero "int" e "input" debido a que es un dato de entrada

- *if a==2:*

Primera condición a cumplir según el dato que ingrese el usuario

- *lcd.clear()*

Esta línea sirve para BORRAR la pantalla LCD

- *else:*

Contra punto de la primera condición

- *print ('Opcion incorrecta')*

En el caso que se digite un numero distinto a 2 se imprime el siguiente mensaje

## 8.1 EJECUCIÓN DEL TRABAJO DE INVESTIGACIÓN

1. Para ejecutar el código en Python en la Raspberry pi 3 – modelo B, primero se debe instalar el sistema operativo en el micrordenador, la descarga del software se la realiza directamente de la página oficial de Raspberry: <https://www.raspberrypi.org/>



Fig 7. Página principal de Raspberry pi

2. Descargar la interfaz NOOBS, por defecto tiene precargado el sistema operativo RASBIAN, en esta interfaz se va a implementar el código fuente de Python para realizar el control de leds, NOOBS se descarga como un archivo comprimido.



Fig 8. Página de descarga NOOBS

3. Para guardar el sistema operativo se utiliza una tarjeta Micro SD, que debe estar formateada, el programa para formatear la tarjeta es de libre acceso en el siguiente link: <https://www.sdcard.org/downloads/formatter...>

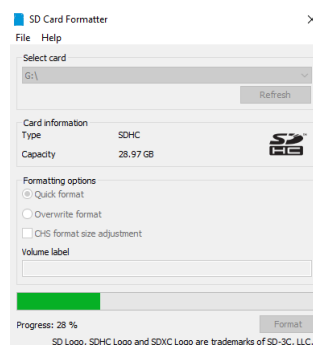


Fig 9. Programa para formatear unidades de almacenamiento

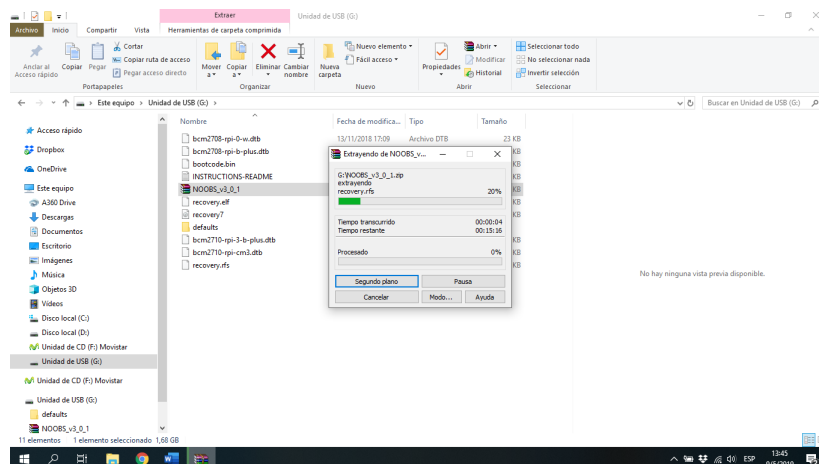


Fig 10. Interfaz de NOOBS en la tarjeta Micro SD

4. Conectamos el dispositivo a una fuente de poder de 5v-3A y un cable HDMI hacia un periférico que admita este tipo de conexión, al momento que Raspberry ejecuta la interfaz, automáticamente se empieza a

expandir el sistema operativo en la Micro SD, seleccionamos instalar Raspbian, al cabo de unos minutos el dispositivo está listo para ser utilizado, Raspbian por defecto tiene instalado Python 3 IDLE.

5. Mediante cables macho-hembra y macho-macho conectamos los pines de la Raspberry Pi hacia la pantalla LCD, según se muestra en la siguiente tabla:

RASPBERRY PI 3		LCD	
GND	PIN 6	VSS	PIN 1
		RW	PIN 5
		K(-)	PIN 16
5V PWR	PIN 2	VDD	PIN 2
		A(+)	PIN 15
GPIO 25	PIN 22	D4	PIN 11
GPIO 24	PIN 18	D5	PIN 12
GPIO 23	PIN 16	D6	PIN 13
GPIO 18	PIN 12	D7	PIN 14
GPIO 22	PIN 15	RS	PIN 4
GPIO 17	PIN 11	E	PIN 6
POTENCIOMETRO		LCD	
TERMINAL	SEÑAL	Vo/VE	PIN 3
GND	PROTO -		
5V	PROTO +		

fig 11. Pines de conexión Raspberry Pi- LCD 16x2

6. Para finalizar, encendemos la Raspberry al momento que corremos el código fuente se puede observar como se proyecta el mensaje en la pantalla LCD.

## 9 DESCRIPCIÓN DE PRERREQUISITOS Y CONFIGURACIÓN

Tener el sistema operativo Raspbian ya que ahí viene incluido Python el lenguaje de programación utilizado para presentar un mensaje en la pantalla LCD 16x2.

- Una Raspberry Pi 2, 3 o 3+
- Un PC o Mac con Etcher.io instalado para preparar la tarjeta microSD.
- Fichero imagen con la última versión de Raspbian Stretch Lite descargada desde la web de Raspberry Pi.
- Una tarjeta SD de 4GiB, si es más pequeña puede que no funcione correctamente. Es recomendable usar una más grande, ya que con los tamaños de archivos que se manejan actualmente 4GiB (menos lo que ocupe el sistema y las aplicaciones) se quedan en poco.

- Tener conexión a internet no es un requisito fundamental, pero si recomendado, ya que permitirá instalar parte de las actualizaciones durante el proceso de instalación.
- Es esencial el módulo Adafruit CircuitPython CharLCD ya que esta biblioteca es compatible con la LCD de caracteres estándar.

## 10 APORTACIONES

Aprovechando la interfaz de Python se realizó un programa más amigable con el usuario, donde para proyectar o borrar el mensaje, primero se imprime una pregunta y según la respuesta que ingrese por teclado se realiza la acción indicada.

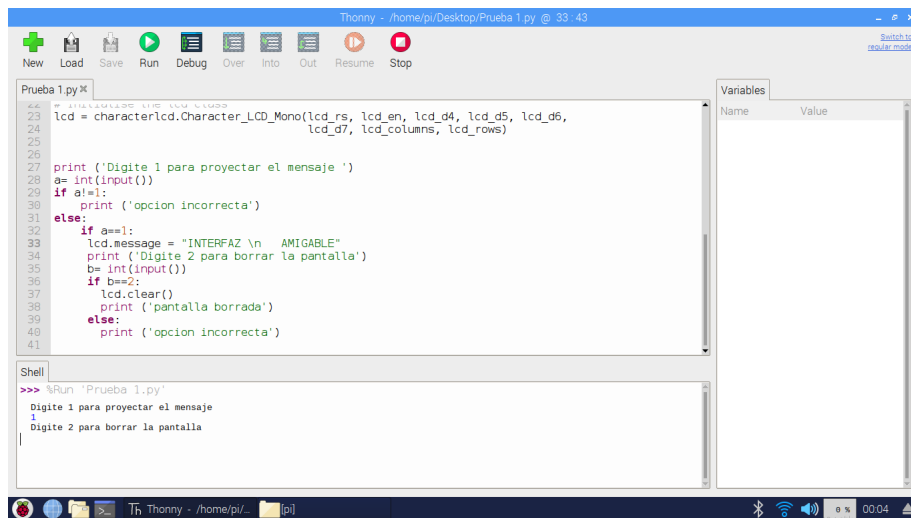


Fig 12. Interfaz de Python

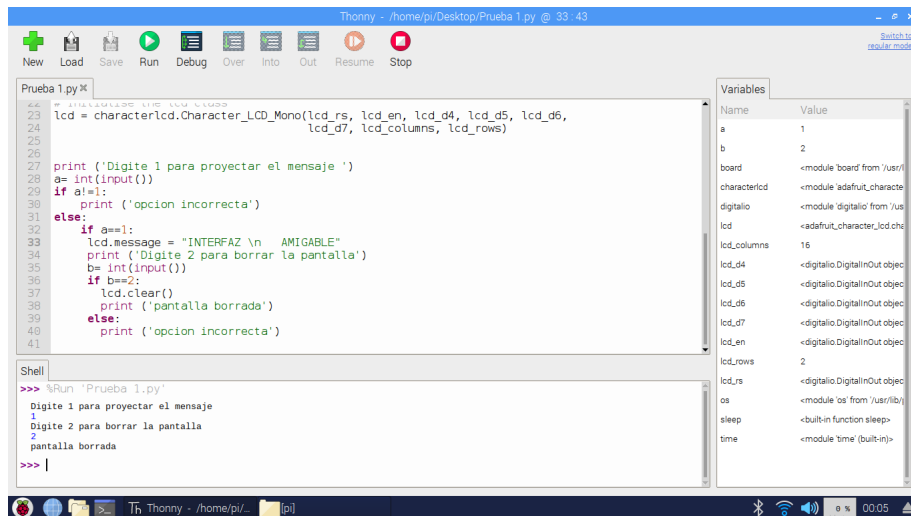


Fig 13. Interfaz de Python

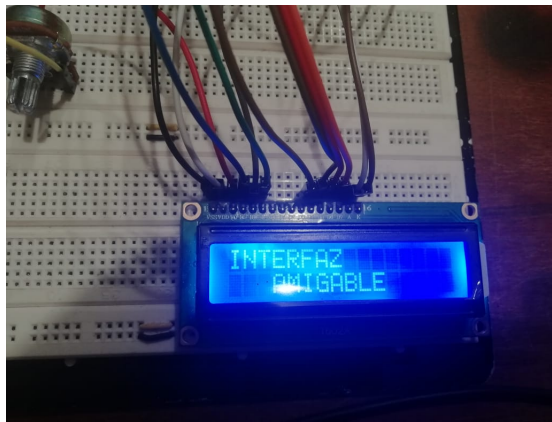


Fig 14. Mensaje proyectado en la pantalla LCD

## 11 CONCLUSIONES

En conclusión:

- La mejor forma de establecer la comunicación entre la Raspberry Pi y la pantalla LCD, es mediante la utilización de la interfaz de Python, simplemente se debe importar la librería Adafruit y definir los pines de salidas de datos, los dispositivos actuarán según las instrucciones que se indiquen en el código fuente.
- Los pines de la pantalla LCD cumplen con una función específica para la transmisión y respuesta de datos, esto facilita la interacción entre el usuario y la consola, resultado de dicha interacción es la proyección del mensaje en la LCD.
- Tanto la Raspberry Pi como la pantalla LCD son dispositivos fácilmente adaptables a un entorno de programación con Python siempre y cuando se limite de forma correcta sus variables y se identifique tanto sus pines de control de datos como los pines de configuración y de alimentación.

## 12 RECOMENDACIONES

- Se recomienda utilizar un potenciómetro de 10K para controlar el contraste de la pantalla LCD, debido a que si se conecta directamente a la Raspberry Pi se corre el riesgo de quemar los leds de la pantalla.
- Es recomendable utilizar mínimo 4 pines de los 8 pines de Datos de la pantalla LCD, para establecer una óptima comunicación entre los dispositivos.
- Al momento de escribir el mensaje que se va a proyectar en la pantalla, se debe tener en cuenta que la LCD solo soporta dos filas de 16 caracteres cada una.



## 13 CRONOGRAMA

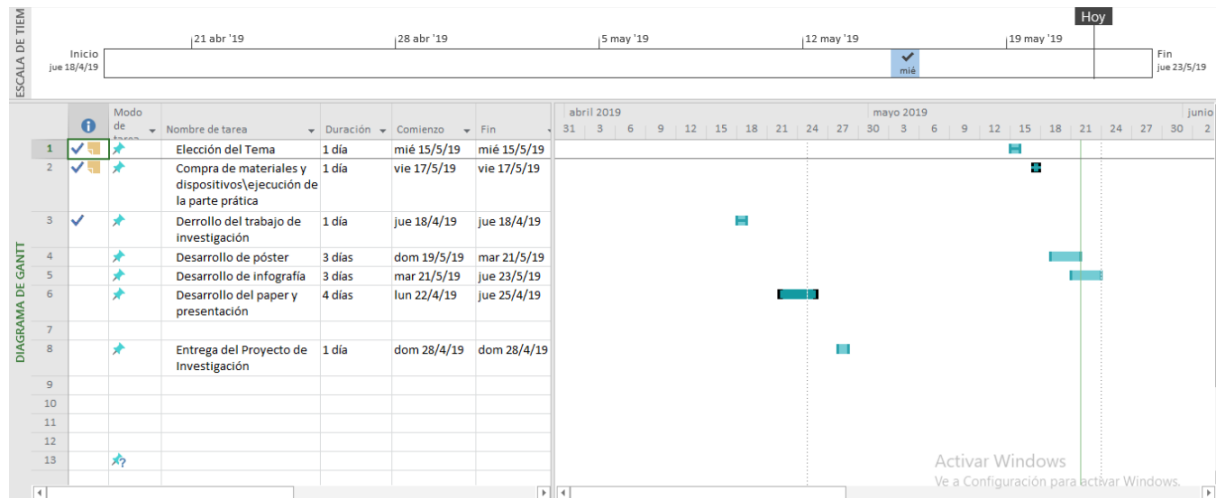


Fig 15. Cronograma de actividades

## References

- [1] Akila, IS, Akshaya, B., Deepthi, S., y Sivadharshini, P. (2018). Un lector de texto para discapacitados visuales utilizando Raspberry Pi. 2018 Segunda Conferencia Internacional sobre Metodologías de Computación y Comunicación (ICCMC). doi: 10.1109 / icmc.2018.8487513
- [2] Pattichis, MS, Celedon-Pattichis, S., y LopezLeiva, C. (2017). Enseñanza de imágenes y procesamiento de video utilizando las matemáticas de la escuela media y la Raspberry Pi. Conferencia Internacional IEEE 2017 sobre Acústica, Procesamiento de Voz y Señal (ICASSP). doi: 10.1109 / icassp.2017.7953378
- [3] Sharma, M., Agarwal, N. y Reddy, SRN (2015). Diseño y desarrollo de placa hija para comunicación USB-UART entre Raspberry Pi y PC. Conferencia internacional sobre computación, comunicación y automatización. doi: 10.1109 / ccaa.2015.7148532
- [4] Palacios, F. (2016). Sistema de identificación de huella dactilares para el acceso a lugares restringidos y control de asistencia con protocolo de comunicación IEEE 802.11. ISBN-13: 978-0-13-235613-8
- [5] Montes, H., Pacheco, A., Ramos, H. (2017). Monitoreo del Consumo de Energía Eléctrica Doméstica con Arduino. 15th LACCEI International Multi-Conference for Engineering, Education, and Technology: "Global Partnerships for Development and Engineering Education", 19-21 July 2017, Boca Raton Fl, United States.
- [6] Tutorials Teacher. (2018). Python-IDLE. Blog de Tecnologías de la Información. Recuperado el 7 de mayo del 2019 de: <https://www.tutorialsteacher.com/python/python-idle>
- [7] Shead, S. (2012). Raspberry Pi delivery delays leave buyers hungry (and angry). Hardware. Recuperado el 8 de mayo del 2019 de: <https://www.zdnet.com/article/raspberry-pi-delivery-delays-leave-buyers-hungry-and-angry/>
- [8] Raspberry Shop. (2019). Hardware y Accesorios. Raspberry Pi 3 modelo B. Recuperado el 8 de mayo del 2019 de: <https://www.raspberrypi-shop.es/raspberry-pi-3.php>

## 14 ANEXOS

### A. MANUAL DE USUARIO

1. Se deben conectar los cables macho-hembra y macho-macho entre la LCD y los pines de la Raspberry, verificamos que el circuito se encuentre correctamente armado tomando como guía la siguiente imagen:

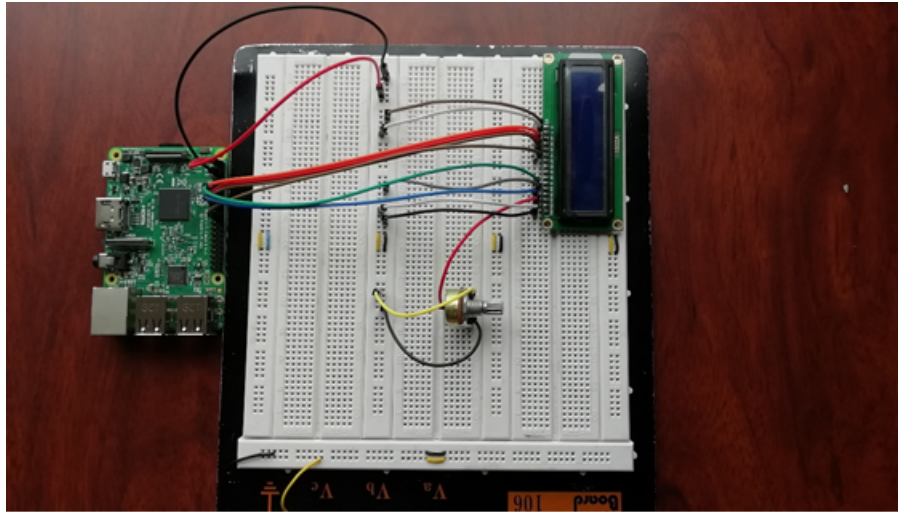


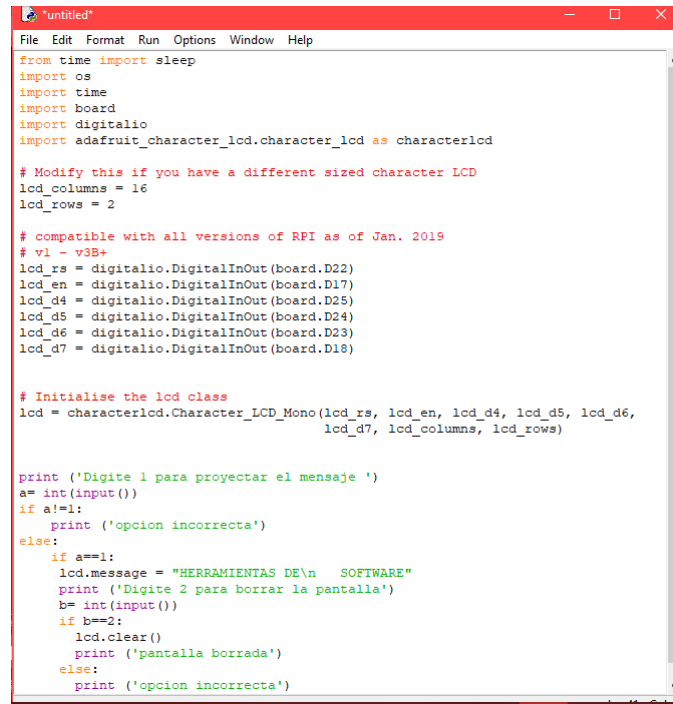
Fig 16. Implementación del Circuito

2. Para encender la Raspberry conectamos un cargador de 5 V y mínimo 1.5 A de alimentación., se conecta el cable HDMI a la Raspberry Pi y al monitor o televisor, de esta forma podremos visualizar la ventana principal del sistema operativo.



Fig 17. Alimentación y conexión de la Raspberry Pi

3. Esperamos un momento que la interfaz inicie. Al momento que ingresamos a la pantalla principal del sistema operativo de la Raspberry (Raspbian), iniciamos Python y ejecutamos el código de programación.



```

from time import sleep
import os
import time
import board
import digitalio
import adafruit_character_lcd.character_lcd as characterlcd

# Modify this if you have a different sized character LCD
lcd_columns = 16
lcd_rows = 2

# compatible with all versions of RPI as of Jan. 2019
# v1 - v3B+
lcd_rs = digitalio.DigitalInOut(board.D22)
lcd_en = digitalio.DigitalInOut(board.D17)
lcd_d4 = digitalio.DigitalInOut(board.D25)
lcd_d5 = digitalio.DigitalInOut(board.D24)
lcd_d6 = digitalio.DigitalInOut(board.D23)
lcd_d7 = digitalio.DigitalInOut(board.D18)

# Initialise the lcd class
lcd = characterlcd.Character_LCD_Mono(lcd_rs, lcd_en, lcd_d4, lcd_d5, lcd_d6,
                                     lcd_d7, lcd_columns, lcd_rows)

print ('Digite 1 para proyectar el mensaje ')
a= int(input())
if a!=1:
    print ('opcion incorrecta')
else:
    if a==1:
        lcd.message = "HERRAMIENTAS DE\n SOFTWARE 5035"
        print ('Digite 2 para borrar la pantalla')
        b= int(input())
        if b==2:
            lcd.clear()
            print ('pantalla borrada')
        else:
            print ('opcion incorrecta')

```

Fig 18. Código Fuente

4. Se observa que en la pantalla LED se proyecta el mensaje “HERRAMIENTAS DE SOFTWARE 5035”, mientras que en la ventana principal de Python se imprime el mensaje “Digite 2 para borrar la pantalla”



Fig 19. Mensaje proyectado en la LCD

5. Para borrar el mensaje de la pantalla LCD, simplemente se debe digitar 2 y presionar enter, la pantalla LCD se borra y en la ventana principal de Python se imprime “mensaje borrado”, si se desea volver a proyectar el mensaje se debe ejecutar nuevamente el programa.

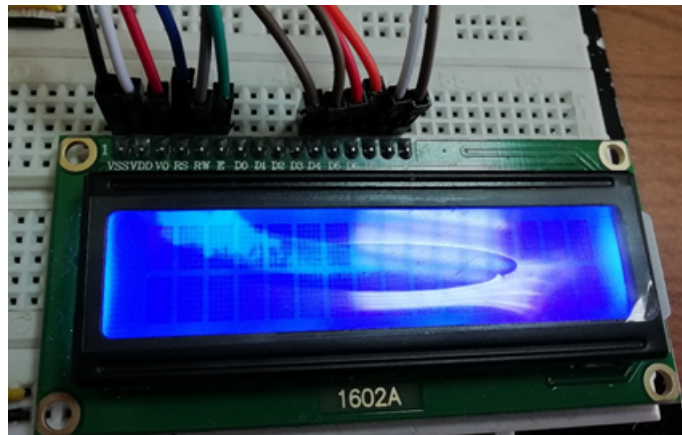


Fig 20. Mensaje Borrado

## B. FOTOGRAFÍAS



Fig 21. Elaboración del circuito



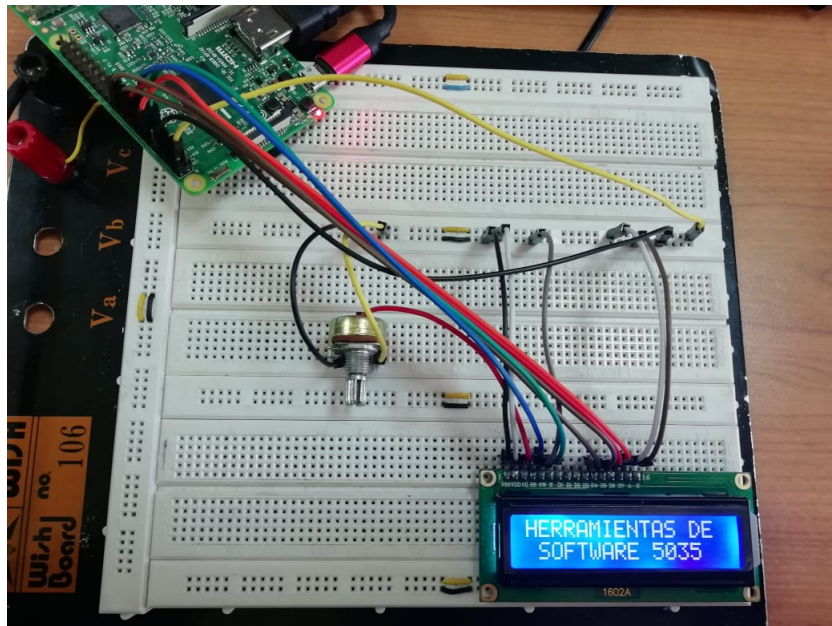


Fig 22. Conexión pantalla LCD 16x2-Raspberry Pi

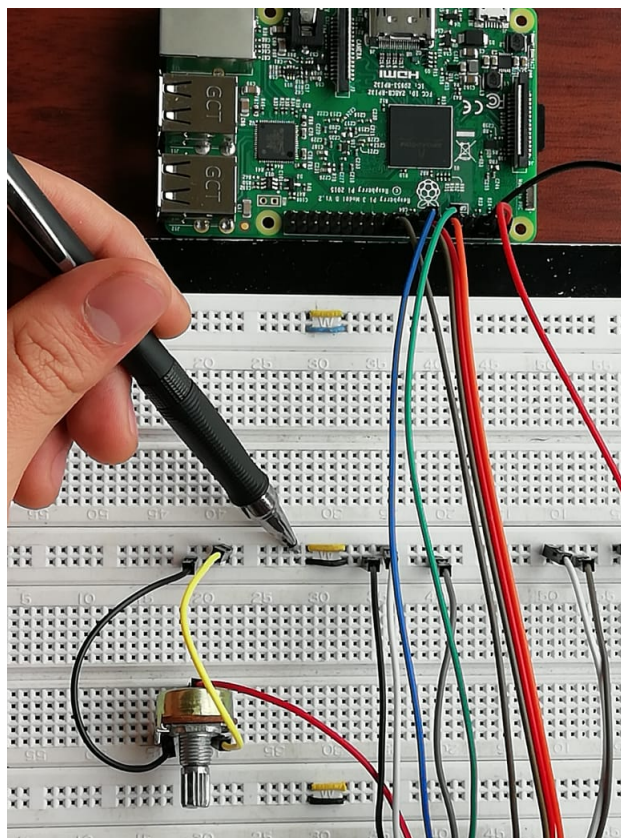


Fig 23. Implementación del Circuito