Principales tarjetas de desarrollo Raspberry Pi, Arduino y Micro Bit

K. Peña, E. Simbaña, MJ. Vizuete

Resumen

PCB son las siglas de Placa de Circuito Impreso, por sus siglas en inglés (Printed Circuit Board).La placa electrónica es un soporte físico en donde se instalan componentes electrónicos y eléctricos que se interconectan entre ellos. Estos componentes pueden ser, chips, condensadores, diodos, resistencias, conectores, etc.

El uso de las PCB fue un paso de gigante en la evolución de los dispositivos electrónicos, ya que proporcionó un método innovador para la conexión de elementos sin necesidad de utilizar cables eléctricos.

I. INTRODUCCIÓN

El objeto de investigación de este artículo es comprender el funcionamiento,componentes ,pines E/S ,lenguaje de programación en el que se desarrolla y sus principales aplicaciones, lo cual se logrará a través de la abstracción de información, para así poder conocer las funcionalidades que tiene.
A partir de la investigación se busca explicar más a fondo los componentes de estas placas y algunas reglas que lo caracterizan, de tal manera que la información sea comprensible y detallada. También se busca desarrollar ejemplos básicos donde se evidencie la aplicación de las principales placas.

II.MARCO TEÓRICO

A.¿Qué es Raspberry Pi?

Es un ordenador del tamaño de una tarjeta de crédito de bajo coste y tamaño reducido, creada con el objetivo de estimular la enseñanza de la informática en las escuelas, aunque no empezó su comercialización hasta el año 2012.
[1]Consta de una placa base sobre la que se monta un procesador, un chip gráfico y memoria RAM. Esta placa que soporta varios componentes necesarios en un ordenador común y es capaz de comportarse como tal. Su concepto es el de un ordenador desnudo de todos los accesorios que se pueden eliminar sin que afecte al funcionamiento básico (Schmidt, M. 2014).

B.¿Que es Micro Bit?

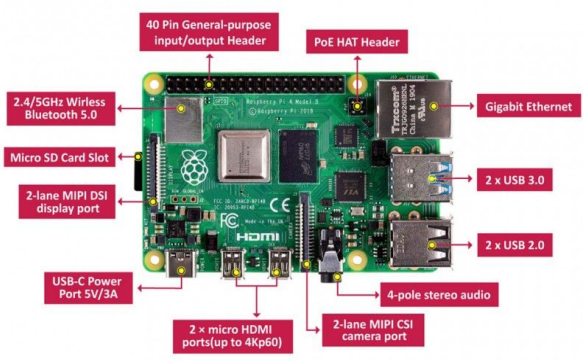
La micro:BIT es una pequeña tarjeta programable, con un costo asequible a cualquier bolsillo (unos 19€, ojo el cable va aparte). Aun cuando su tamaño es muy reducido, incorpora gran cantidad de sensores y actuadores lo que unido a que usa un software Open Source, hacen de la micro:BIT una plataforma ideal para introducirse en el mundo de la programación de robots.[5]
La naturaleza fácilmente accesible del dispositivo. significa que puede usarse para enseñar conceptos de programación y computación a niños de cualquier edad, su robustez y su pequeño tamaño significan que es extremadamente portátil y adecuado para usar proyectos(Eckel, J. R. 1967).

C.¿Qué es Arduino?

Arduino Uno es una placa electrónica basada en el microcontrolador ATmega328. Cuenta con 14 entradas/salidas digitales, de las cuales 6 se pueden utilizar como salidas PWM (Modulación por ancho de pulsos) y otras 6 son entradas analógicas. (Artero,Ó.T,2013). Además, incluye un resonador cerámico de 16 MHz, un conector USB, un conector de alimentación, una cabecera ICSP y un botón de reseteo. La placa incluye todo lo necesario para que el microcontrolador haga su trabajo, basta conectarla a un ordenador con un cable USB o a la corriente eléctrica a través de un transformador.[8]

III. HARDWARE

D. HARDWARE RASPBERRY

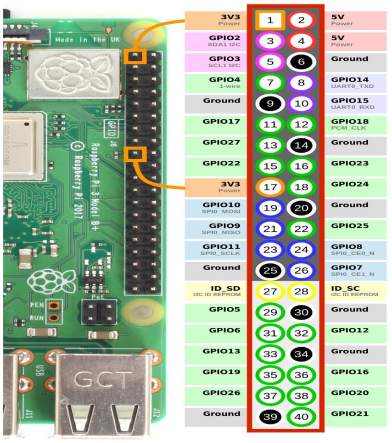


- **Sistema en un chip:** Broadcom BCM2711 con arquitectura de 64 bits.
- **CPU:** Cuenta con un procesador de cuatro núcleos a 1,5 GHz con brazo Cortex-A72.
- **GPU:** VideoCore VI (con soporte para OpenGL ES 3.x)
- **Memoria:** Memoria de 1, 2 o 4 GB de RAM LPDDR4 a una velocidad de 2.400MHz.
- **Conectividad:** Wi-Fi 802.11b/g/n/ac (2.4GHz y 5GHz), Bluetooth 5.0 con BLE (low energy), puerto Gigabit Ethernet (el cual es nativo y dedicado, por lo que conseguiremos velocidades de más de 900Mbps reales en la red local), 2 puertos USB 3.0 y 2 puertos USB 2.0.
- **Video y sonido:** cuenta con 2 puertos micro-HDMI que admite pantallas de 4K
  Puertos: GPIO 40 pines 2 x micro HDMI 2 x USB 2.0 2 x USB 3.0 CSI (cámara Raspberry Pi) DSI (pantalla tactil) Micro SD Conector de audio jack USB-C (alimentación).
- **Alimentación:** Vía USB tipo C con 5V y 3A; vía PoE con HAT podremos alimentarlo a través del estándar PoE con un cable de red, proporcionándole alimentación vía switch PoE.
- **Expansión:** Cabezal GPIO de 40 pines compatibles con otras Raspberry Pi.
- **Tarjetas micro SD:** Soporta micro SD para el sistema operativo y también para almacenamiento de datos.

La característica de las Raspberry que más destaca, es su apariencia desnuda. Para completarlo se necesitará una fuente de alimentación, un monitor o un televisor, cables para conectarse al monitor (normalmente un cable micro HDMI), un ratón y un teclado. Lo cual permite crear una máquina propia.[3]

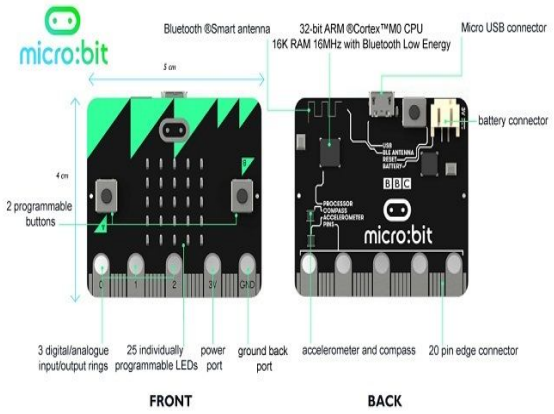
1)Los pines GPIO (General Purpose Input Output):

Estos pines son digitales, lo que significa que sólo pueden tener dos estados, apagado o encendido. Pueden tener una dirección para recibir o enviar corriente (entrada, salida respectivamente) y todo esto es totalmente controlable por lenguajes de programación
como Python, JavaScript, node-RED y otros. Los pines trabajan con una tensión de 3,3 V y un consumo máximo de corriente de 16 mA. Esto significa que podemos suministrar energía de forma segura desde un solo pin GPIO a través de una resistencia y uno o dos LEDs (Halfacree, G.2014).
Para controlar GPIO con Python, lo primero es importar una librería de código escrito previamente. El más común y difundido es el RPi.GPIO, utilizado para crear miles de proyectos desde los primeros días de la Raspberry Pi.[2]



E) HARDWARE MICRO BIT

La placa Micro:bit. Esta placa fue desarrollada bajo el amparo de la BBC con el objetivo de ser regalada a los niños y niñas de Gran Bretaña como punto de partida para su aprendizaje tecnológico.[6]



En tan solo 4×5 cm2 podemos encontrar:

Procesador

El procesador BBC micro: bit es un nRF51822 nórdico,que contiene un ARM Cortex-M0 de un solo núcleo que se ejecuta a 16Mhz, con 16 KB de aleatorio memoria de acceso (RAM) y 256 KB de memoria no volátil (NVM) para almacenamiento de programas(Young Jr, F. S.1967)..

25 LEDs.

Se pueden programar de forma independiente y permiten mostrar números, letras e imágenes. Si el texto o la cifra no caben en el display se desplazan de forma automática.

Sensor de Luz. Los LEDs también tiene la posibilidad de ser usados como sensor de luz ambiente.

Pulsadores. Existen 2 botones, etiquetados como A y B. Se puede detectar la pulsación independiente de cada uno de ellos.

Conectores. Situados en la parte inferior de la placa, dispone de 25 conexiones que permiten conectar otros sensores y actuadores.

Sensor de temperatura. Permite conocer a la micro:BIT la temperatura ambiente.

Acelerómetro. Activada cuando se mueve la placa, permite conocer aceleraciones y giros a los que se somete la placa.

Brújula digital. Permite conocer la desviación respecto el Norte Magnético.

Radio. Permite conectarse inalámbricamente con otras micro:BITs.

Bluetooth. Ideal para conectarse e intercambiar datos inalámbricamente con otros dispositivos que dispongan de este tipo de conexión.

USB. Usado para descargar los programas a la memoria de la tarjeta y para alimentar eléctricamente la micro:BIT.

Conector de batería. Permite suministrar electricidad mediante dos pilas AAA o una batería. La tarjeta carece de interruptor, por lo que cuando se conecta la fuente de alimentación se ejecuta de forma automática el código que haya en memoria.

F)HARDWARE ARDUINO

1) Botón de reset: Sirve para inicializar nuevamente el programa cargado en el microcontrolador de la placa.

2-3) Pines o puertos de entrada y salida: son los pines donde conectar los sensores, componentes y actuadores que necesiten de señales digitales

4)Puerto USB: Utilizado tanto para conectar con un ordenador y transferir o cargar los programas al microcontrolador como para dar electricidad al Arduino.

5)Chip de interfaz USB: es el encargado de controlar la comunicación con el puerto USB.

6)Reloj oscilador: Es el elemento que hace que el Arduino vaya ejecutando las instrucciones. Es el encargado de marcar el ritmo al cual se debe ejecutar cada instrucción del programa.

7)Led de encendido: Es un pequeño LED que se ilumina cuando la placa está correctamente alimentada.

8)Microcontrolador: Este es el cerebro de cualquier placa Arduino. Es el procesador que se encarga de ejecutar las instrucciones de los programas.

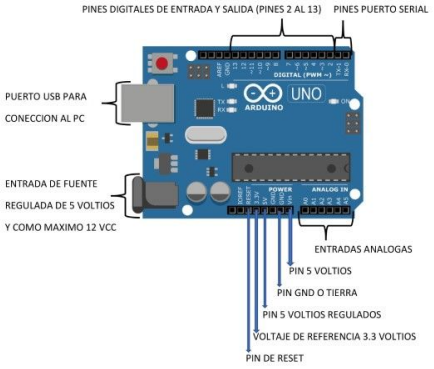
9)Regulador de tensión: Este sirve para controlar la cantidad de electricidad que se envía a los pines, con lo que asegura que no se estropee lo que conectemos a dichos pines.

10)Puerto de corriente continua: Este puerto es el que se usa para darle electricidad a la placa si no se usa alimentación USB.

11)Zócalo de tensión: Aquí estarán los pines con los que alimentaremos nuestro circuito. Entradas analógicas.

12) Zócalo: con distintos pines de entrada analógica que permiten leer entradas analógicas.

3)Entrada y salida, Input and Output



Cada uno de los 14 pines digitales de la Uno puede utilizarse como entrada o salida, utilizando las funciones pinMode(), digitalWrite() y digitalRead(). Funcionan a 5 voltios. Cada clavija puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia pull-up interna (desconectada por defecto) de 20-50 kOhms(McRoberts, M. 2011).Además, algunos pines tienen funciones especializadas:

Serial: 0 (RX) y 1 (TX). Se utiliza para recibir (RX) y transmitir (TX) datos en serie TTL. Estos pines están conectados a los pines correspondientes del chip Serial ATmega8U2 USB-to-TTL.

Interrupciones externas: 2 y 3. Estos pines pueden configurarse para activar una interrupción en un valor bajo, un flanco ascendente o descendente, o un cambio de valor. Vea la función attachInterrupt() para más detalles.

PWM: 3, 5, 6, 9, 10 y 11. Proporciona salida PWM de 8 bits con la función analogWrite().

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estos pines soportan la comunicación SPI utilizando la biblioteca SPI.

LED: 13. Hay un LED incorporado conectado al pin 13 digital. Cuando la clavija es de valor ALTO, el LED se enciende, cuando la clavija es BAJA, se apaga[9]

IV. SOFTWARE

G. SOFTWARE RASPBERRY PI

Existen tres principales lenguajes de programación que se puede ejecutar en la Raspberry Pi:

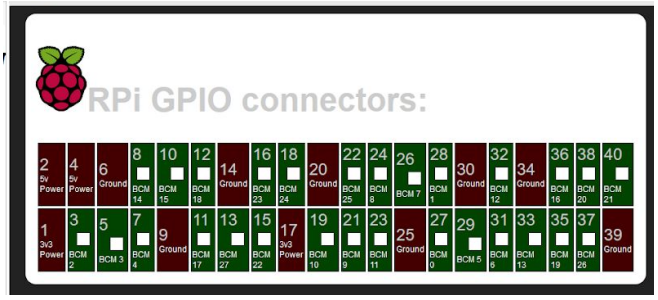
1)Python: Python es un lenguaje de programación interpretado dinámico y multiplataforma, cuya filosofía hace hincapié en la legibilidad de su código. Este soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.

2)Java: Java es un lenguaje de programación de computadora de propósito general que es concurrente, basado en clases, orientado a objetos y específicamente diseñado para tener la menor cantidad posible de dependencias de implementación.

4)JavaScript: Es un lenguaje de programación ligero e interpretado, orientado a objetos con funciones de primera clase. Es un lenguaje script multi paradigma, basado en prototipos, dinámico, soporta estilos orientados a objetos, imperativos y declarativos.

5)Plataforma: La plataforma Create.withcode.uk, es una herramienta gratuita donde se permite escribir, ejecutar, depurar y compartir programas de Python en el navegador web.

Su ventaja es que no se necesita descargar ni instalar ningún archivo, es así que los programas de Python no pueden acceder a los archivos o dañar la computadora, por lo que es una forma segura de aprender a crear con código[4].



En esta parte se puede editar el código, y se lo ejecuta dando clic en la flecha que se encuentra al lado derecho.En el caso de usar código compatible con la Raspberry, nos mostrará los pines de esta pudiéndose usar como simulador.

H. SOFTWARE MICROBIT

1)Python

Python es un lenguaje de programación creado por Guido van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses "Monty Python". Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible[7].

2)Javascript

JavaScript es un lenguaje de los denominados lenguajes de scripting. Los scripts (script se traduce como guión, literalmente) son archivos de órdenes, programas por lo general simples. Es por esto que no podemos definir JavaScript como un lenguaje de programación en un sentido estricto, pero sin embargo sí nos permite crear páginas dinámicas, con algunos efectos realmente interesantes y que mejoren considerablemente su aspecto.

I. SOFTWARE ARDUINO

1)Tinkercad Simulador

Tinkercad es una herramienta online ofrecida por Autodesk. Se utiliza de forma gratuita y sólo requiere crearse una cuenta de usuario. De entre sus utilidades, probablemente la más conocida es la de diseñar piezas en 3D. Sin embargo, ofrece también una posibilidad realmente interesante y es la de montar, programar y simular circuitos con Arduino.Esta plataforma usa el lenguaje propio de arduino para la elaboración de circuitos.Arduino a su vez está basado en C++ .

C++ es un lenguaje de programación diseñado en 1979 por Bjarne Stroustrup. La intención de su creación fue extender al lenguaje de programación C mecanismos que permiten la manipulación de objetos(Castro Campos, R. A. 2011).

1)Estructura básica para lenguaje Arduino

La estructura básica del lenguaje de programación de Arduino es bastante simple y se compone de al menos dos partes. Estas dos partes necesarias, o funciones, encierran bloques que contienen declaraciones, estamentos o instrucciones[10].

```
void setup()
{

estamentos;
 }
void loop()
{
 estamentos;
 }
```

V. EXPLICACIÓN DE CÓDIGO FUENTE

J.ARDUINO

1)Programa "Teclado de seguridad "

Para realizar este programa debemos ingresar a la plataforma TinkerCad o incluso directamente en la plataforma de Arduino,en cualquier caso y como para este ejemplo usaremos un teclado matricial lo que hacemos es declarar la librería #include <Keypad.h>,posterior a esto se declaran las filas y columnas ,además colocar los pines a los que el teclado ,estas serán las entradas

```
byte rowPins[ROWS]={9,8,7,6}; byte colPins[COLS]={5,4,3,2};
```

Además se establece la contraseña con la que compararemos la contraseña que el usuario ingrese ,a esta la declararemos como un carácter de longitud 7.

```
char password[7];
char rpassword[7]="456786".
```

Dentro del set up pondremos los pines que se ejecutan como las salidas que en este caso será un LED de tipo RGB ,incluimos **Serial**. **begin(9600)**; que es una instrucción le indica al Arduino que inicie comunicación con la computadora y un mensaje en el monitor en serie que indique que se debe ingresar la contraseña

```
Serial.begin(9600);
  Serial.print("INGRESE CONTRASENIA\n");
  pinMode(12,OUTPUT);
```

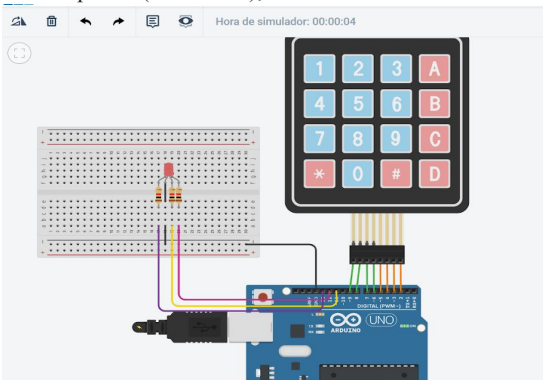
```
 pinMode(13,OUTPUT);
```

Dentro del void loop tenemos las primeras condiciones en las que se verifica si el usuario ha ingresado el dato solicitado en el caso de que no el programa se mantendrá estático caso contrario el datos ingresado será almacenado prar ser comparado

```
void loop(){
 char key=keypad.getKey();
  if(key!= NO_KEY){//no se ha apretado ninguna tecla
  Serial.println(key);
  password[index]=key;
  index++;
 }
```

En la segunda parte de las condiciones se visualiza la comparación entre el dato ingresado y la contraseña almacena ,si son iguales se enciende el LED en un tono verde y de no serlo en color rojo:

```
 if(check == 6){
    digitalWrite(12,HIGH);
    digitalWrite(13,LOW);
    Serial.println("GREEN");
```



K. RASPBERRY

2)Programa “Alarma”:

Para poder realizar el programa nos dirigimos a la plataforma Create.withcode.uk en donde se ejecutarán los códigos en el lenguaje python.

Lo primero que debemos hacer para el control de los pines de la tarjeta con el lenguaje mencionado es importar librería, a más de eso también importamos a la librería tiempo, para poder controlarlo. estas son:

```
import RPi.GPIO as GPIO
import time
```

Configuramos el pin que queremos que envíe la señal

```
pin= 24
```

A continuación asignamos el modo en el que vamos a llamar a los pines, este puede ser BOARD (conteo a partir del número impreso en la placa) o BCM(por el número del "canal Broadcom SOC"). En esta parte también preparamos el pin para el comienzo de su salida, en este caso será en alto.

```
GPIO.setmode (GPIO.BCM) ,
GPIO.setup (pin,GPIO.OUT)
GPIO.output(pin,GPIO.HIGH)
```

Declaramos variables nuevamente para que nuestra alarma tenga sus condiciones

```
print ("    ALARMA DE VEHICULO")
print("Ingrese 1=si, 0=no:")
puerta= int(input("¿La puerta esta abierta?: \n"))
motor= int(input("¿El motor esta encendido?: \n"))
luces= int(input("¿La luces estan encendidas?: \n"))
```

Aquí creamos las condiciones y utilizamos un bucle for para asignar los valores iniciales y finales en los que se va a encender el pin en este caso va del 0 al 10. Llamamos dos veces al GPIO.output debido a que en high se va a prender y en low a apagar y le damos el tiempo que queremos que espere antes de continuar con la siguiente línea, en este caso 0.3 segundos. La función GPIO.cleanup() devuelve los pines al estado inicial.

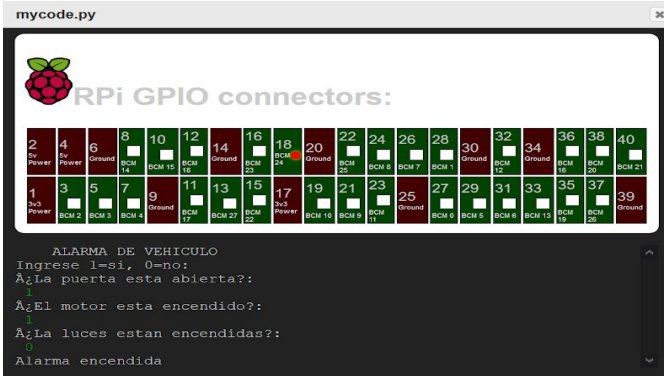
```
if (luces==1 and motor==1):
        print("Alarma encendida")
        for i in range (0,10):
        GPIO.output(pin, GPIO.LOW)
        time.sleep(0.3)
        GPIO.output(pin, GPIO.HIGH)
        time.sleep(0.3)
        GPIO.cleanup()
```

De igual manera será la segunda condición, con distintas variables

```
elif (puerta==1 and motor==1):
        print("Alarma encendida")
        for i in range (0,10):
        GPIO.output(pin, GPIO.LOW)
        time.sleep(0.3)
        GPIO.output(pin, GPIO.HIGH)
        time.sleep(0.3)
        GPIO.cleanup()
```

Y en nuestra última condición haremos que se prenda el pin 3 veces (0,3), con un tiempo de 0.1 segundos.

```
else:
        print("Alarma apagada")
        for i in range (0,3):
        GPIO.output(pin, GPIO.LOW)
        time.sleep(0.1)
        GPIO.output(pin, GPIO.HIGH)
        time.sleep(0.1)
        GPIO.cleanup()
```



L. MICRO BIT

1)Programa “Piedra papel y tijera”

Para la realización de este programa contamos con una estructura de codificación en bloques la cual cuenta con una interfaz bastante agradable al usuario.En primer lugar determinamos en qué condiciones las acciones se van a desarrollar para lo cual generamos un bloque con la condición que si se agita este se va a ejecutar la creación de

una variable llamada “Mano” la cual va a tomar un valor al azar entre el 1 y el 3 mediante la función randint(1, 3)

```
let Mano = 0
input.onGesture(Gesture.Shake, function () {
    Mano = randint(1, 3)
```

a partir de esto crearemos tres condiciones una anidada a otra la cual tomará las acciones de falso de la anterior mediante la condición if else y por último a partir de este número aleatoria mediante la función basic.showLeds mostraremos en la matriz de leds integrada en nuestra placa microbit los leds que deseemos ordenándolos a nuestro gusto

```
if (Mano == 1) {
    basic.showLeds(`
        # # . . #
        # # . # .
        . # # . .
        # # . # .
        # # . . #
        `)
```

VI. Metodología

La revisión bibliográfica fue el principal método empleado en la realización de este artículo, ya que nos centramos en el análisis de material bibliográfico y de consulta de información respecto al tema estudiado, lo que nos permitió afianzar los conceptos necesarios para el desarrollo del código de los programas realizados en los emuladores. La investigación fue posible por medio de herramientas utilizadas, así como el los emuladores empleados para cada tarjeta ,los cuales nos permitieron programar y visualizar el circuito , la herramienta Github y del mismo modo la información recolectada nos permitió encontrar las múltiples estructuras a estudiar.

VII. Resultados

A partir del análisis y revisión de información se logró desarrollar y ejecutar a través de los emuladores antes mencionados ,en dichos programas se visualizan diversas estructuras correspondientes al lenguaje de programación de cada emulador y placa .Los emuladores descritos representan una gran ventaja ya que por un lado permiten ahorrar en componentes físicos y por otro como se ha mencionado proporcionan un entorno visual amigable,siendo más fácil su comprensión.

VIII. CONCLUSIONES

-Se cumplió nuestro tercer objetivo de realizar ejemplos básicos en los que se empleen las tarjetas de desarrollo,gracias a la revisión bibliográfica de su concepto, funcionamiento y principales características .Es así que los ejemplos se encuentran en un buen estado y listas para su ejecución.

-Las plataformas usadas para realizar los ejemplos básicos son de gran ayuda ya que permiten ahorrar en hardware,además,nos ofrecen un entorno visual que es amigable para el usuario.

-Las PBC constituyen una forma eficiente y concreta de realizar o desarrollar un circuito,ya que gracias a sus diversos pines permiten que se conecte a ellos una variedad de elementos.

IX.RECOMENDACIONES

Tener conocimientos básicos de electrónica y electricidad para las conexiones residenciales, es importante para la manipulación correcta de los componentes del mismo modo tener conocimientos de programación en lenguaje javascript y python es esencial para el desarrollo de la lógica de encendido y apagado de las luces y del mismo modo de los programas a ejecutar que interpretaran las placas Arduino,Raspberry y MicroBit. del mismo modo verificar el voltaje y amperaje con el cual pondremos a funcionar nuestras placas debido a que es muy común que los integrados se quemen o dañen superados los 5 voltios.

X.BIBLIOGRAFÍA

[1]Monk, S. (2016). *Raspberry Pi cookbook: Software and hardware problems and solutions*. " O'Reilly Media, Inc.".

[2]Upton, E., & Halfacree, G. (2014). Raspberry Pi user guide. John Wiley & Sons.

[3]Gay, W. (2014). Raspberry Pi hardware reference. Apress.

[4]Hernández, M. F. D., Montenegro, J. L. G., Beleño, R. D. H., García, J. D., & Sánchez, N. S. (2018). Raspberry PI3 y Pc DUINO: Tutorial de instalación y configuración.

[5]Ferko,(2017) Manual de Javascript :bytehost31

[6]Van Rossum, G., & Drake Jr, F. L. (1995). *Python tutorial* (Vol. 620). Amsterdam: Centrum voor Wiskunde en Informatica.

[7]Cooper, M. (2019). Meet the BBC Micro: Bit. *ITNOW*, *61*(3), 14-15.

[8]Arduino, S. A. (2015). Arduino. *Arduino LLC*.

[9]Evans, B. (2011). *Beginning Arduino Programming* (Vol. 6). New York, NY, USA:: Apress.

[10]Lequerica, J. R. (2016). *Arduino para jóvenes y no tan jóvenes*. Anaya Multimedia.