



# Title: Raspberry Pi, conectividad y programación mediante puertos GPIO

**Authors:** CABALLERO-JULIÁN, Franco Gabriel, MORALES-HERNÁNDEZ, Maricela, SILVA-CRUZ, Eric Mario y CABALLERO-CANTARELL, Diego Gabriel

Editorial label ECORFAN: 607-8695  
BCIERMMI Control Number: 2020-04  
BCIERMMI Classification (2020): 211020-0004

Pages: 13  
RNA: 03-2010-032610115700-14

**ECORFAN-México, S.C.**  
143 – 50 Itzopan Street  
La Florida, Ecatepec Municipality  
Mexico State, 55120 Zipcode  
Phone: +52 1 55 6159 2296  
Skype: ecorfan-mexico.s.c.  
E-mail: contacto@ecorfan.org  
Facebook: ECORFAN-México S. C.  
Twitter: @EcorfanC

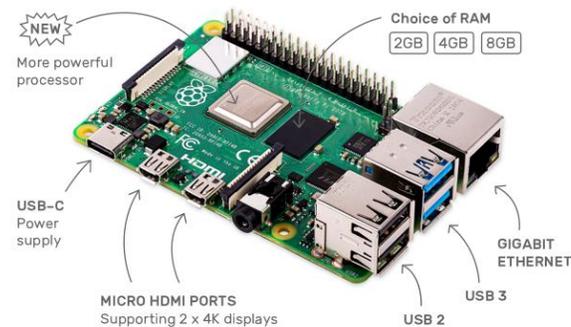
[www.ecorfan.org](http://www.ecorfan.org)

Holdings		
Mexico	Colombia	Guatemala
Bolivia	Cameroon	Democratic
Spain	El Salvador	Republic
Ecuador	Taiwan	of Congo
Peru	Paraguay	Nicaragua

# Introduction

La Raspberry Pi Foundation desarrolló la Raspberry Pi como una computadora de placa única u ordenador de placa simple (SBC) de bajo costo, basada en un microprocesador embebido. El diseño original es un entorno de desarrollo práctico, que permite un sinnúmero de posibilidades para aplicaciones novedosas.

La arquitectura de *Raspberry Pi* (Ray, 2017), está centrada en un procesador multinúcleo *Broadcom* (*BCM 2835*, *BCM2836* y *BCM 2837*), memoria compartida entre la *CPU* y la unidad de procesamiento de gráficos (*GPU*), puertos *USB*, *HDMI*, *RJ45*, 40 pines *GPIO* multiplexados, *miniUSB*, y un conector para cámara, etc.



# The problem

El interés se centra en que el lector arme y configure paso a paso una computadora *Raspberry Pi* y se concentre en el manejo de los puertos de propósito general (*GPIO*). Puertos con los que sea posible desarrollar aplicaciones en las que desde esa computadora *Raspberry Pi*, la misma plataforma o algún dispositivo inteligente; conectado en su red *LAN*, tenga acceso para prender, apagar, programar, leer o controlar otros actuadores y sistemas de comunicación.

# Methodology

## 1.- Material necesario

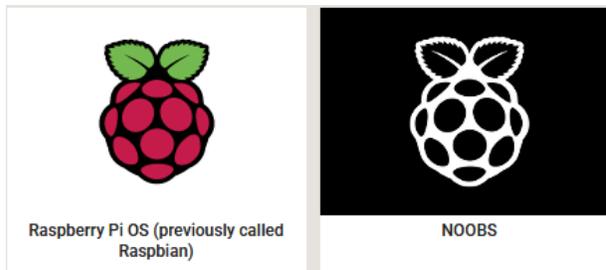
- Una computadora *Raspberry Pi 2, 3* ó su versión más reciente, una tarjeta *micro SD* mínimo de 8G clase 4, considerando de preferencia 16 o 32G de almacenamiento.
- *Mouse*, teclado, un monitor, cable *hdmi* y un convertidor *hdmi-VGA* dependiendo el monitor.
- Una fuente de alimentación dedicada con salida micro USB de 5V a 1.5 Amp, un cable de red RJ-45.
- *Internet* para actualizar el sistema operativo, descargar los controladores y recursos, un *protoboard*, resistencias y *leds*.

## 2.- Preparación de la memoria

Para formatear la *SD card* utilizamos la herramienta *SD Card formatter* y para cargar el SO vamos a utilizar *balena etcher* en su versión para *Windows*, este es un programa que nos permite *flashear* memorias.

## 3.- Conexiones de hardware

Con su *micro SD* en la mano, insertarla en la ranura de la plataforma *Raspberry Pi*, y previamente se deber conectar el teclado, *mouse*, el monitor, el cable RJ-45 a la red de internet ( si dispone de la *Raspberry Pi 3 Model B* ya tiene integrado el módulo *wifi*) y finalmente conectar la fuente de alimentación para inicializar la plataforma.



# Methodology

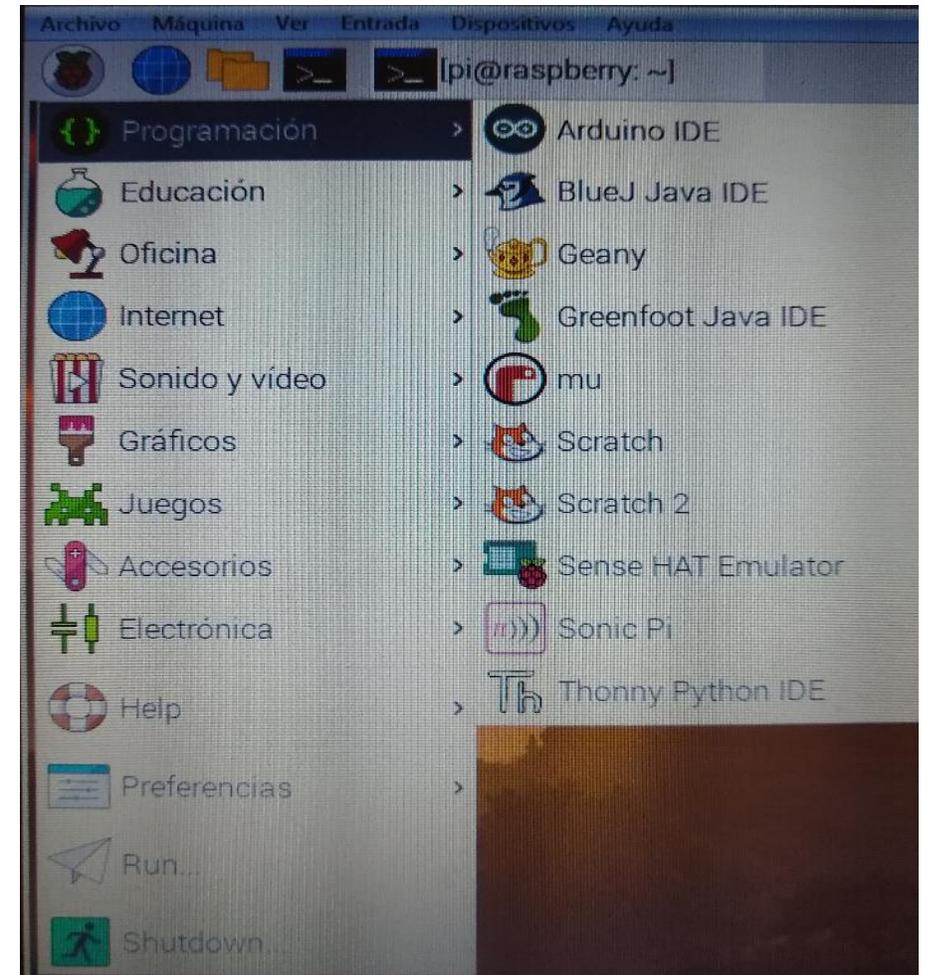
## 4.- Configuración inicial

Al inicio el sistema le da la bienvenida con un

*Welcome to the Raspberry Pi Desktop* con la imagen de la frambuesa, y le indica que hay algunas cosas que configurar antes de poder utilizarla.

## 5.- Configuración del Software y manipulación de archivos y carpetas.

Aunque ya en el paso anterior el sistema se actualizó, por medio de la consola podemos en cualquier momento actualizar y ejecutar instrucciones para manipular archivos y carpetas.



# Methodology

## 6.- Programación

### 6.1.- Instrucciones desde la terminal

Desde la consola, es posible controlar las terminales por medio de instrucciones que declaran la *GPIO* de entrada o salida (Ray,2017); en el caso de salida el valor digital que asume es de 5V o 3.3V dependiendo la tecnología.

En las siguientes líneas se declara la *GPIO26* de uso, configurada como salida y se le envía un valor uno y después un cero lógico.

```
$ echo 26 > /sys/class/gpio/export
```

```
$ echo out > /sys/class/gpio/gpio26/direction
```

```
$ echo 1 > /sys/class/gpio/gpio26/value
```

```
$ echo 0 > /sys/class/gpio/gpio26/value
```

# Methodology

## 6.- Programación

### 6.2 Programación en *Python*

En *Thonny Python IDE*, un recurso precargado en la *Raspberry Pi*, se escriben los programas de aplicación. Aquí se exponen tres ejemplos probados con una entrada y cuatro salidas. Estos ejemplos son muy básicos, a partir de aquí que el lector pueda utilizar los recursos más potentes de programación para hacer programas más complejos que se puedan utilizar en sistemas de aplicación. Se eligen pines de entrada o salida.

Entradas:

*INP0: GPIO23, PIN12*

Salidas:

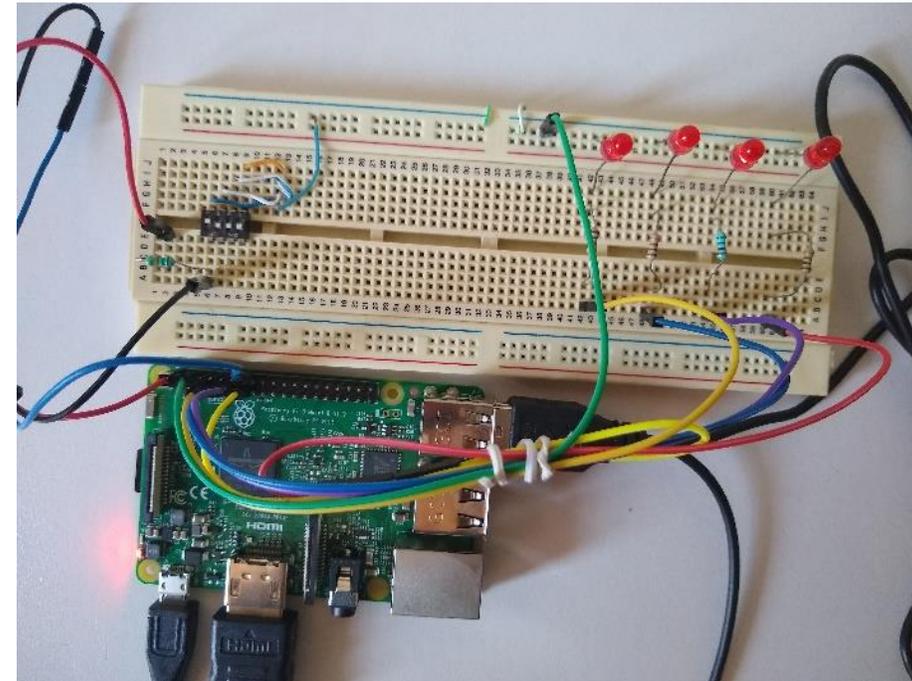
*D0:GPIO22, PIN 15*

*D1:GPIO27, PIN 13*

*D2:GPIO17, PIN 11*

*D3:GPIO4, PIN 7*

El cable de tierra es por medio del PIN 6 de GND.



# Methodology

## Ejemplo1.py

```
#####  
from gpiozero import LED  
from time import sleep  
led0=LED(22)  
led1=LED(27)  
led2=LED(17)  
led3=LED(4)  
while True:  
    led0.on()  
    sleep(1)  
    led0.off()  
    sleep(1)  
    led1.on()  
    sleep(1)  
    led1.off()  
    sleep(1)  
    led2.on()  
    sleep(1)  
    led2.off()  
    sleep(1)  
    led3.on()  
    sleep(1)  
    led3.off()  
    sleep(1)  
#####
```

## Ejemplo2.py

```
#####  
import RPi.GPIO as GPIO  
import time  
LEDPin0= 22  
LEDPin1= 27  
LEDPin2= 17  
LEDPin3= 4  
Inicio=0  
Fin=5  
GPIO.setmode (GPIO.BCM)  
GPIO.setup(LEDPin0, GPIO.OUT)  
GPIO.setup(LEDPin1, GPIO.OUT)  
GPIO.setup(LEDPin2, GPIO.OUT)  
GPIO.setup(LEDPin3, GPIO.OUT)  
while (Inicio<= Fin):  
    GPIO.output(LEDPin0,GPIO.HIGH)  
    time.sleep(0.5)  
    GPIO.output(LEDPin0,GPIO.LOW)  
    time.sleep(0.5)  
    GPIO.output(LEDPin1,GPIO.HIGH)  
    time.sleep(0.5)  
    GPIO.output(LEDPin1,GPIO.LOW)  
    time.sleep(0.5)  
    GPIO.output(LEDPin2,GPIO.HIGH)  
    time.sleep(0.5)
```

```
GPIO.output(LEDPin2,GPIO.LOW)  
time.sleep(0.5)  
GPIO.output(LEDPin3,GPIO.HIGH)  
time.sleep(0.5)  
GPIO.output(LEDPin3,GPIO.LOW)  
time.sleep(0.5)  
time.sleep(0.5)  
print("Numero de rutina:", Inicio)  
Inicio = Inicio + 1  
GPIO.cleanup()  
#####
```

# Methodology

## Ejemplo3.py

```
#####
```

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

```
import time
```

```
LEDPin0= 22
```

```
LEDPin1= 27
```

```
LEDPin2= 17
```

```
LEDPin3= 4
```

```
Inicio=0
```

```
Fin=5
```

```
GPIO.setmode (GPIO.BCM)
```

```
GPIO.setup(LEDPin0, GPIO.OUT)
```

```
GPIO.setup(LEDPin1, GPIO.OUT)
```

```
GPIO.setup(LEDPin2, GPIO.OUT)
```

```
GPIO.setup(LEDPin3, GPIO.OUT)
```

```
GPIO.setup(23,GPIO.IN)
```

```
while (Inicio<= Fin):
```

```
    input_state=GPIO.input(23)
```

```
    if input_state == True:
```

```
        GPIO.output(LEDPin0,GPIO.HIGH)
```

```
        time.sleep(0.5)
```

```
        GPIO.output(LEDPin0,GPIO.LOW)
```

```
        time.sleep(0.5)
```

```
        GPIO.output(LEDPin1,GPIO.HIGH)
```

```
        time.sleep(0.5)
```

```
        GPIO.output(LEDPin1,GPIO.LOW)
```

```
        time.sleep(0.5)
```

```
        print("Numero de rutina:", Inicio)
```

```
        Inicio = Inicio + 1
```

```
    else:
```

```
        GPIO.output(LEDPin2,GPIO.HIGH)
```

```
        time.sleep(0.5)
```

```
        GPIO.output(LEDPin2,GPIO.LOW)
```

```
        time.sleep(0.5)
```

```
        GPIO.output(LEDPin3,GPIO.HIGH)
```

```
        time.sleep(0.5)
```

```
        GPIO.output(LEDPin3,GPIO.LOW)
```

```
        time.sleep(0.5)
```

```
        time.sleep(0.5)
```

```
        print("Numero de rutina:", Inicio)
```

```
        Inicio = Inicio + 1
```

```
GPIO.cleanup()
```

```
#####
```

# Methodology

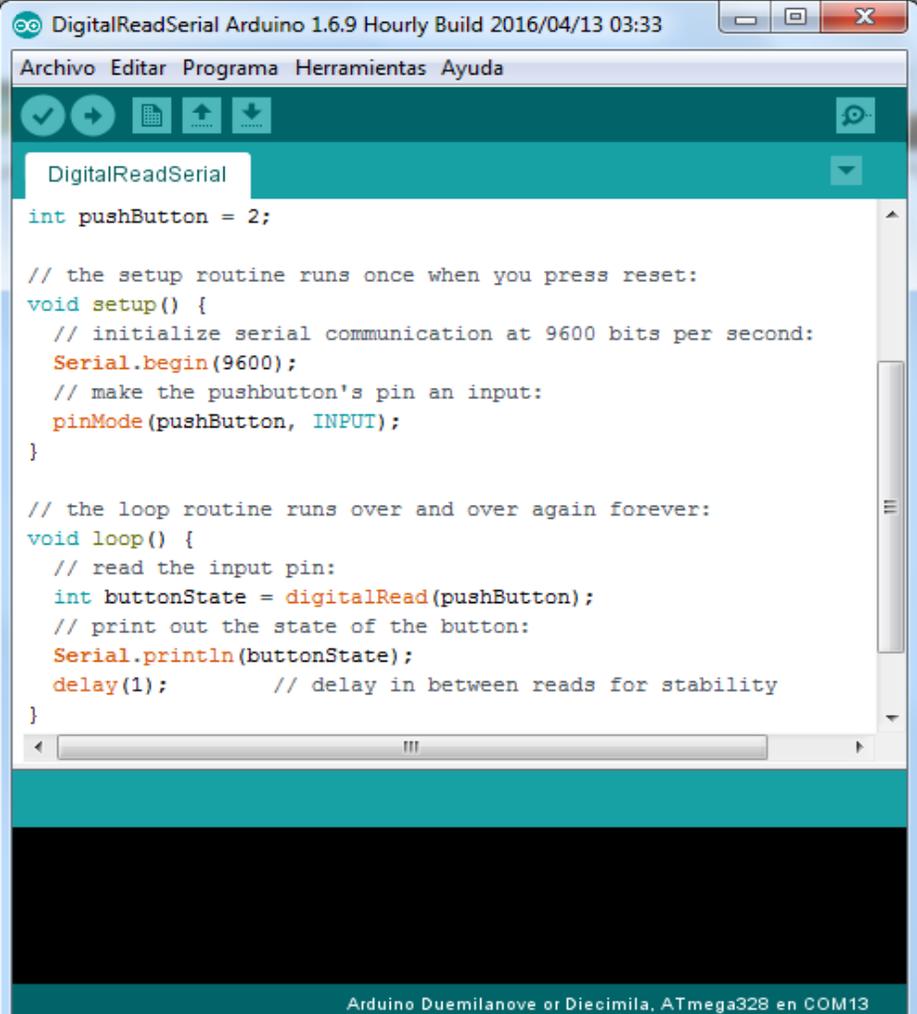
## 6.- Programación

### 6.3 Programación de Arduino

*Arduino* es muy versátil, y presenta una variedad de modelos, con distintos recursos, así ha ganado el interés de muchos desarrolladores en todo el mundo; también ha logrado que se posicione en el mercado por ser de bajo costo y muy fácil de usar (Rumberg, 2015). Entre muchas de sus posibilidades, una placa *Arduino* se le puede dejar en modo esclavo para transmitir o recibir señales de voltaje mediante comunicación serial y realizar interacción con entornos de programación como son: *Matlab*, *LabVIEW*, *Processing*, esto mediante el puerto *USB Ethernet* de la computadora *Raspberry Pi*.

Instalación de *Arduino*

**`$sudo apt-get install arduino`**



```
Arduino IDE: DigitalReadSerial Arduino 1.6.9 Hourly Build 2016/04/13 03:33
Archivo  Editar  Programa  Herramientas  Ayuda

DigitalReadSerial

int pushButton = 2;

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
  // make the pushbutton's pin an input:
  pinMode(pushButton, INPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input pin:
  int buttonState = digitalRead(pushButton);
  // print out the state of the button:
  Serial.println(buttonState);
  delay(1); // delay in between reads for stability
}

Arduino Duemilanove or Diecimila, ATmega328 en COM13
```

# Methodology

## 6.- Programación

### 6.4 Por medio de una página Web

Entre los recursos desarrollados por terceros está *WebIOPi* que es de fácil instalación y muy intuitivo. De antemano, nuestra computadora *Raspberry Pi* ya tiene acceso a *Internet*, entonces abrir el navegador y en *The Raspberry Internet of Things Toolkit* (2020), vamos al apartado de descargas y hacer la descarga que por defecto se aloja en la carpeta *Downloads*. O desde la terminal de comandos hacemos también la descarga de la última versión de la aplicación.

#### WebIOPi Main Menu

##### [GPIO Header](#)

Control and Debug the Raspberry Pi GPIO with a display which looks like the physical header.

##### [GPIO List](#)

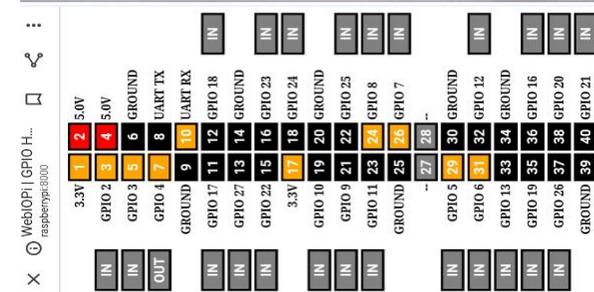
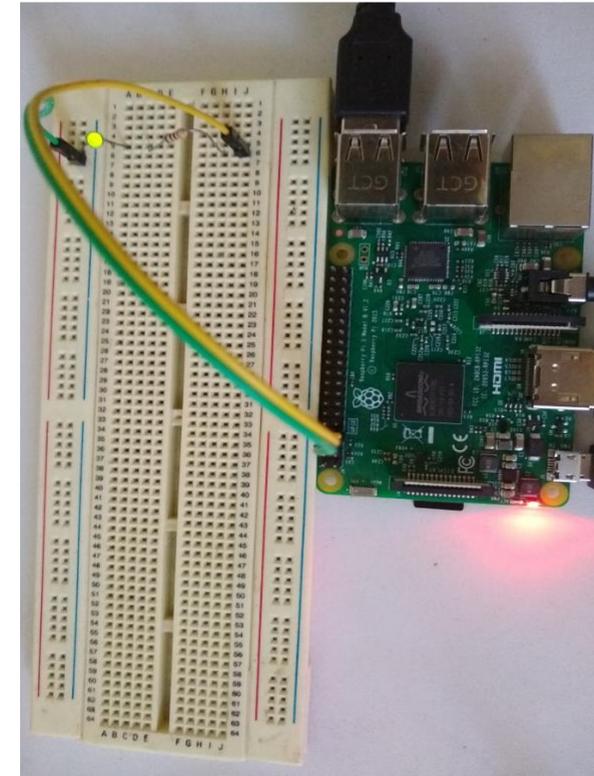
Control and Debug the Raspberry Pi GPIO ordered in a single column.

##### [Serial Monitor](#)

Use the browser to play with Serial interfaces configured in WebIOPi.

##### [Devices Monitor](#)

Control and Debug devices and circuits wired to your Pi and configured in WebIOPi.



# Results

- 1) Disponemos de una computadora muy completa con acceso a internet, con opciones de programación, con office, juegos y muchos otros recursos ya incluidos.
- 2) De manera muy directa, se muestran cuatro formas de acceder al accionamiento de los GPIO, podemos elegir cualquiera y mediante lenguaje Python se puedan crear aplicaciones de interés, o también por medio de la plataforma Arduino, y así mismo esta opción que considera la obtención de información y con otra aplicación sea posible mostrarla en pantalla y darle el tratamiento adecuado, como un trabajo a futuro.
- 3) Por medio de WebIOPi, disponemos de una herramienta Web muy práctica para el acceso de lectura o escritura de los puertos GPIO.
- 4) Para construir interfaces que nos permitan crear soluciones reales de proyectos más complejos, incluso en el modo headless se puedan manipular vía remota usando SSH y VNC Viewer.

# Conclusions

En este artículo muy concreto y simple tenemos una guía de instalación, configuración y programación de una computadora Raspberry Pi, mediante la cual los autores pretenden contribuir con su experiencia a los interesados en desarrollador y crear sistemas y aplicaciones mediante este sistema de cómputo.

Los ejemplos presentados tienen un grado de complejidad simple, lo cual deja oportunidad de explorar los innumerables recursos de los que dispone Raspberry Pi, para así realizar interfaces de lectura y escritura de señales digitales y analógicas, todo depende del entusiasmo y magnitud del problema que se presente.

El reto para el futuro desarrollador consiste en encontrar el problema y proponer una solución con la computadora Raspberry Pi mediante el manejo de los puertos GPIO.

# References

- Amaya, A. (2020). Desarrollo de un Sistema de Comunicación LP-WAN para transmitir señales de geolocalización de personas dentro de entornos naturales. Tesis Profesional. Instituto Tecnológico de Oaxaca-CICESE.
- Arostegui Gallardo, C.I., Mata Cruz, F.A.C., & Romero Rodríguez, R.O. (2019). Accesos controlados al cuarto de racks transaccionales financieros aplicando tecnología de corto alcance. Reporte Técnico. ESIME, IPN.
- Asadi, A., Bagheri, S., Imam, A. (2016). A data acquisition system based on Raspberry Pi. Design, construction and evaluation. IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON).
- Bernal, C., (2010). Metodología de la Investigación. Tercera edición. Pearson-Prentice Hall.
- Falcone, F., Matías, I., Miltino, J., Gabilondo, A., (2015). Domótica e Inmótica. Instalaciones de telecomunicaciones para edificios. Alfaomega-marcombo.
- Grey, G.(2018). RASPBERRY PI, Guía paso a paso para principiantes de Raspberry Pi. Gabriel Grey.
- Guzdial, M., Ericson, B., (2013). Introducción a la computación y programación con Python. PEARSON.
- Magpi(s.f). Beginner's book. The official Raspberry Pi magazine.
- Magpi(2019). The 50 best tips. Issue 80. April 2019. The official Raspberry Pi magazine.
- Mocq, F., (2020). Raspberry Pi 3 o Pi Zero. Explote todo el potencial de su nano-ordenador. <https://www.ediciones-eni.com/open/mediabook.aspx?idR=0d1aa94cd5698d4846e4519ae4030923>
- Oshana, R. y Kraeling, M. (2013). Software Engineering for Embedded Systems: Methods, Practical Techniques, and Applications. USA: Elsevier, Inc.
- Programo Ergo Sum(2020). Control de Raspberry pi con Python en Raspberry Pi. <https://www.programoergosum.com/cursos-online/raspberry-pi/238-control-de-gpio-con-python-en-raspberry-pi/ques-gpio>
- Ray, R. (2017). Raspberry PI, Guía paso a paso para dominar Hardware y Software Raspberry PI 3. Ranny Ray.
- Runberg, D.(2015). The Sparkfun guide to processing. No Starch Press, Inc.
- Sesma Martínez, J. (2015). Monitorización y detección de fallos en una instalación solar fotovoltaica mediante sistema remoto. Trabajo fin de Máster. Escuela Politécnica Superior de Orihuela.
- Suehle, R., Callaway, T. (2014). Raspberry Pi Hacks. Tips & Tools for Making Things with the Inexpensive Linux Computer. O'REILLY. USA.
- The Raspberry Internet of Things Toolkit (2020). The Raspberry Internet of Things Toolkit- Now in two flavors: Cayenne-the Spicy one, WebIOPi- the Original one. Recuperado de <http://webiopi.trouch.com/>.



**ECORFAN®**

© ECORFAN-Mexico, S.C.

No part of this document covered by the Federal Copyright Law may be reproduced, transmitted or used in any form or medium, whether graphic, electronic or mechanical, including but not limited to the following: Citations in articles and comments Bibliographical, compilation of radio or electronic journalistic data. For the effects of articles 13, 162,163 fraction I, 164 fraction I, 168, 169,209 fraction III and other relative of the Federal Law of Copyright. Violations: Be forced to prosecute under Mexican copyright law. The use of general descriptive names, registered names, trademarks, in this publication do not imply, uniformly in the absence of a specific statement, that such names are exempt from the relevant protector in laws and regulations of Mexico and therefore free for General use of the international scientific community. BCIERMMI is part of the media of ECORFAN-Mexico, S.C., E: 94-443.F: 008- ([www.ecorfan.org/](http://www.ecorfan.org/) booklets)