

## Desarrollo de Circuito de Riego – Circuito de Alarma de incendios

**Autores: Paspuel C. Christian A.**

**Topón S. Kevin D.**

**Simba C. Henry J.**

**Estudiantes de la Universidad de las Fuerzas Armadas ESPE Sangolquí/Ecuador.**

**Departamento de Ciencias De la Computación.**

Fax: +593 23989-400. [capaspuel1@espe.edu.ec](mailto:capaspuel1@espe.edu.ec)

**Estudiantes de la Universidad de las Fuerzas Armadas ESPE Sangolquí/Ecuador.**

**Departamento de Ciencias De la Computación.**

Fax: +593 23989-400. [kdtopon@espe.edu.ec](mailto:kdtopon@espe.edu.ec)

**Estudiantes de la Universidad de las Fuerzas Armadas ESPE Sangolquí/Ecuador.**

**Departamento de Ciencias De la Computación.**

Fax: +593 23989-400. [hjsimba@espe.edu.ec](mailto:hjsimba@espe.edu.ec)

### **Resumen**

Durante el semestre hemos adquirido conocimientos de la Materia de arquitectura de computadoras junto a Programacion y en este producto unidad empleamos todas las herramientas necesarias para la realizacion de dos ejercicios muy interesantes lo cual nos ayudara a desarrollar nuestras habilidades como futuros ingenieros.

### **Introducción**

En el presente trabajo de investigación se va hacer una implementación de situaciones reales en las que son la toma decisiones y para llegar a ello procedemos en crear nuestro código en lenguaje de python además de que se tiene que utilizar un emulador de raspberry para llegar al objetivo planteado.

**Palabras Clave:** código, Phyton, emulador, Raspberry Pi.

### **Abstract**

In this research work, an implementation of real situations in which decisions are made will be carried out and to get there we proceed to create our code in Python language in addition to using a raspberry emulator to reach the goal raised.

**Keywords:** code, Python, emulator, Raspberry Pi.

## Python

Según (Rossum, 1995) Python es un lenguaje de programación simple pero potente e interpretado diferente a la programación en C y Shell ideal para la programación desechable y la creación rápida de prototipos. Su sintaxis se elabora a partir de construcciones tomadas de una variedad de otros idiomas, el intérprete de Python se amplía fácilmente con nuevas funciones y tipos de datos implementados en C. Python de igual manera es adecuado como un lenguaje de extensión para aplicaciones C altamente personalizables para varios sistemas operativos como por ejemplos UNIX (incluido Linux), el sistema operativo Apple Macintosh, MSDOS describiéndolo a esta sintaxis y la semántica central del lenguaje como conciso pero intenta ser exacto y completo.

Según (Aguilar, 2010) argumenta que en Python se puede extender fácilmente por medio de módulos escritos en C o C++, proponiendo un énfasis en que el código sea legible (import this).

De igual manera (Duque, 2009) argumenta que 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.

Se trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos.

La ventaja de los lenguajes compilados es que su ejecución es más rápida. Sin embargo, los lenguajes interpretados son más flexibles y más portables. Python tiene, no obstante, muchas de las características de los lenguajes compilados, por lo que se podría decir que es semi interpretado. En Python, como en Java y muchos otros lenguajes, el código fuente se traduce a un pseudo código máquina intermedio llamado bytecode la primera vez que se ejecuta, generando archivos .pyc o .pyo (bytecode optimizado), que son los que se ejecutarán en sucesivas ocasiones.

## Tipado dinámico

La característica de tipado dinámico se refiere a que no es necesario declarar el tipo de dato que va a contener una determinada variable, sino que su tipo se determinará en tiempo de ejecución según el tipo del valor al que se asigne, y el tipo de esta variable puede cambiar si se le asigna un valor de otro tipo.

## Orientado a objetos

La orientación a objetos es un paradigma de programación en el que los conceptos del mundo real relevantes para nuestro problema se trasladan a clases y objetos en nuestro programa. La ejecución del programa consiste en una serie de interacciones entre los objetos. Python también permite la programación imperativa, programación funcional y programación orientada a aspectos.

Python es un lenguaje que todo el mundo debería conocer. Su sintaxis simple, clara y sencilla; el tipado dinámico, el gestor de memoria, la gran cantidad de librerías disponibles y la potencia del lenguaje, entre otros, hacen que desarrollar una aplicación en Python sea sencillo

## Raspberry Pi

Según (Aguilar, 2014) la Raspberry Pi es una computadora en una sola tarjeta (Single-Board Computer) creada por la Raspberry Pi Foundation para promover la enseñanza de la programación en escuelas y países en desarrollo.

### Módulo time.

```
import time
```

`time.time()` regresa el tiempo transcurrido en segundos desde el primero de enero de 1970 como un número de punto flotante

`time.sleep()` Suspende la ejecución del Script por el tiempo especificado como parámetro (número de segundos expresado como número de punto flotante)

### Biblioteca para acceso a GPIO

```
>>> import RPi.GPIO as GPIO #importa la librería de GPIO
```

```
#usar número de terminal no de GPIO
```

```
>>>GPIO.setmode(GPIO.BOARD)
```

```
#configura como salida en bajo a la terminal 11  
nf>>>GPIO.setup(11,GPIO.OUT,GPIO.PUD_OFF,GPIO.LOW)
```

```
>>>GPIO.output(11,GPIO.HIGH) #pone  
la salida en alto
```

### Estructuras de control básicas

Las estructuras de control más comunes son if, if - else y while, que son controladas por una condición lógica, separada del bloque controlado por : El inicio y el fin del bloque controlado depende únicamente de la indentación.

### Tips básicos de Python

- Importante: Python es sensible a mayúsculas y minúsculas
- Numerico: int (1 45 -678), float (12.234 -43.56), complex (-1.23+34.9j 56.1-156j)
- Cadenas de caracteres: “ Hola”, ‘Mundo’
- // división de punto flotante:  
 $3/2=1.5$
- // división entera:  $3//2=1$
- type(variable) regresa el tipo de variable

### Funciones

Una función de un bloque de código organizado, probado y reutilizable. Permiten que los programas sean modulares.

Python incluye muchas funciones internas, tal como printf, pero también se puede construir funciones propias personalizadas, llamadas funciones definidas por el usuario

La definición de una función comienza por la palabra clave def seguida por el nombre de la función y paréntesis Los parámetros de entrada se colocan entre los paréntesis.

### Bibliografía

- Aguilar, M. F. (2014). Programación en Python en la. Obtenido de <http://sagitario.itmorelia.edu.mx/mfraga/materias/soemb/python.pdf>
- Duque, R. G. (2009). Python para todos. España. Obtenido de [http://www.utic.edu.py/citil/images/Manuales/Python\\_para\\_todos.pdf](http://www.utic.edu.py/citil/images/Manuales/Python_para_todos.pdf)
- OLIPHANT, T. E. (15 de 05 de 2007). Python for Scientific Computing. CiSE Computational Physics.
- Rossum, G. (1995). Manual de referencia de Python. *CWI (Centro de Matemáticas e Informática)*.
- Chazallet, S. (2016). *Python 3: los fundamentos del lenguaje*. Ediciones ENI.