

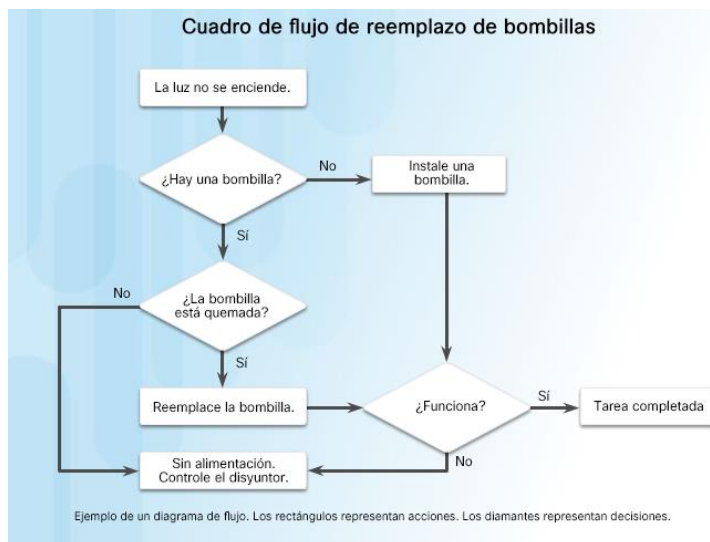


Capítulo 2: Secciones y objetivos

- 2.1 Aplicación de la programación básica para el soporte de dispositivos de IoT
 - Use Python para crear programas que acepten las entradas del usuario, y la lectura y escritura en archivos externos.
 - Describa las variables de programación básicas y aspectos esenciales.
 - Aplique las variables de programación básicas y los aspectos esenciales en Blockly.
 - Aplique las variables de programación básicas y los aspectos esenciales con Python
- 2.2 Creación de un prototipo de su idea
 - Explique la creación de prototipos y su propósito
 - Describa la creación de prototipos.
 - Describa las diversas herramientas y materiales que se usan para crear un prototipo.

2.1 Aplicación de la programación básica para el soporte de dispositivos de IoT

Conceptos básicos de programación Diagramas de flujo



Diagramas de flujo:

- Son diagramas que se utilizan para representar estos procesos o flujos de trabajo.
- Ilustran cómo debe funcionar un proceso.
- Muestran los estados de entrada, las decisiones tomadas y los resultados de estas.

Conceptos básicos de programación

Software del sistema, software de aplicaciones y lenguajes informáticos

- Dos tipos comunes de software de computadora: software del sistema y el software de aplicaciones.
 - Los programas para software de aplicaciones se crean con el fin de realizar una tarea determinada o un conjunto de tareas.
 - El software del sistema funciona entre el hardware de la computadora y el programa de aplicaciones.
 - El software del sistema y el software de aplicaciones se crean con un lenguaje de programación.
 - Python es un ejemplo de un lenguaje de programación interpretado o interpretativo.

Programa para verificar los años bisiestos en Python

```

year = int(input("Enter a year to check if it is a leap year\n"))
if (year % 4) == 0:
    if (year % 100) == 0:
        if (year % 400) == 0:
            print("{0} is a leap year".format(year))
        else:
            print("{0} is not a leap year".format(year))
    else:
        print("{0} is a leap year".format(year))
else:
    print("{0} is not a leap year".format(year))

```



© 2016 Cisco y/o sus filiales. Todos los derechos reservados. Información confidencial de Cisco. 6

Conceptos de programación básica

Variables de programación

- Los lenguajes de programación utilizan variables para alojar frases, números u otra información importante que pueda utilizarse en la codificación.
 - Las variables pueden contener el resultado de un cálculo, el resultado de una consulta en una base de datos o algún otro valor.
 - $x = y + z$
 - Aquí, "x", "y" y "z" son las variables que pueden representar caracteres, cadenas de caracteres, valores numéricos o direcciones de memorias
 - $a = 10$
 - asocia el valor de 10 a la variable "a"
- Las variables permiten que los programadores creen rápidamente una amplia variedad de programas simples o complejos que le indiquen a la computadora que se comporte de manera predefinida.



© 2016 Cisco y/o sus filiales. Todos los derechos reservados. Información confidencial de Cisco. 7

Conceptos de programación básica

Estructuras básicas de programas

```
IF (value1 > value2) THEN print_on_the_screen "Value1 is greater than Value2"
```

El código anterior emite "Value1 is greater than Value2" (El valor 1 es mayor que el valor 2) en la pantalla si la expresión Valor 1 > Valor 2 es verdadera.

```
FOR (i=0; i < 100; i++) {
  print_on_the_screen "counter =" + i
}
```

El código anterior emite "Counter = N" (Contador = N), donde N es el valor del contador variable "i". El mensaje se publica 100 veces en la pantalla.

```
WHILE (value < 10) {
  print_on_the_screen "Value is still less than 10"
  value = value + 1
}
```

El código anterior muestra "Value is still less than 10" (El valor sigue siendo inferior a 10) en la pantalla, si Valor < 10. Tenga en cuenta que el programa también incrementa el valor cada vez que el bucle WHILE se ejecuta.



- Las estructuras lógicas más comunes son las siguientes:

- **IF – THEN:** permite que la computadora tome una decisión según el resultado de una expresión.
 - myVar > 0
 - Es verdadera si el valor almacenado en la variable myVar es mayor que cero.
 - Si es falsa, la computadora continúa con la siguiente estructura.
 - Si es verdadera, la computadora ejecuta la acción asociada antes de pasar a la siguiente instrucción del programa.
- Los **bucles FOR** ejecutan un conjunto específico de instrucciones una cantidad de veces específica según una expresión.
 - Una variable actúa como un contador dentro de un rango de valores que se identifica con un valor mínimo y un valor máximo. Cada vez que se ejecuta el bucle, aumenta la variable del contador. Cuando el contador es igual al valor máximo definido, se abandona el bucle y la ejecución avanza a la siguiente instrucción.
- Los **bucles WHILE** ejecutan un conjunto específico de instrucciones mientras que una expresión sea verdadera.