

# Programación de una máquina dispensadora con Python

Jorge Cruz, Alison Guachamin, Kevin Mejía  
Universidad de las Fuerzas Armadas “ESPE”, Junio 2019

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Planteamiento del Problema . . . . .	1
1.2. Objetivos . . . . .	1
1.2.1. Objetivos Generales . . . . .	1
1.2.2. Objetivos Específicos . . . . .	1
<b>2. Estado del arte</b>	<b>3</b>
<b>3. Marco Teórico</b>	<b>5</b>
3.1. Diagrama UML . . . . .	5
3.2. ¿Qué es y cómo funciona una máquina expendedora? . . . . .	5
3.3. Funciones en Python . . . . .	5
3.3.1. ¿Qué es una función en Python? . . . . .	5
3.3.2. Funciones Usadas . . . . .	6
<b>4. Diagramas</b>	<b>9</b>
4.1. Diagrama de UML . . . . .	9
<b>5. Explicación</b>	<b>11</b>
5.1. Código Fuente . . . . .	11
5.2. Hardware . . . . .	11
5.3. Configuración y Requisitos . . . . .	11
5.4. Ejemplo . . . . .	12
<b>6. Manual de Usuario</b>	<b>13</b>
<b>7. Conclusiones</b>	<b>15</b>
7.1. Recomendaciones . . . . .	15
<b>8. Bibliografía</b>	<b>17</b>
<b>Bibliografía</b>	<b>59</b>
<b>Lista de Acrónimos y Abreviaturas</b>	<b>61</b>
<b>A. Anexo I</b>	<b>63</b>

# 1. Introducción

Python es un lenguaje de programación muy usado en el campo de la programación, su popularidad se debe a la sencillez que presenta, ya que su estructura de programación y palabras reservadas, son sencillas de entender.

## 1.1. Planteamiento del Problema

¿De qué manera podemos crear un programa en Python que simule ser una máquina expendedora de productos?

## 1.2. Objetivos

### 1.2.1. Objetivos Generales

- Crear un programa en Python el cual realice las funciones de una máquina expendedora de productos.

### 1.2.2. Objetivos Específicos

- Identificar el funcionamiento de los diferentes comandos utilizados en la programación.
- Introducir los conceptos básicos de programación en Python.
- Definir Matrices para hacer la presentación de un menú.

## 2. Estado del arte

El lenguaje de programación Python/The programming language Python

Ivet Challenger-Pérez; Yanet Díaz-Ricardo; Roberto Antonio Becerra-García “El lenguaje de programación Python”, Universidad de Holguín “Oscar Lucero Moya”. Cuba, La Habana, 16/04/2014. RESUMEN: software libre se ha convertido en uno de los movimientos tecnológicos de mayor auge en el siglo XXI. Para su desarrollo ha sido necesario contar con un grupo de herramientas que hagan óptima su utilización y sean fáciles de aprender. Python es un lenguaje de programación que cumple con lo planteado y se viene perfilando como una opción recomendada para el desarrollo de software libre. En este artículo se realizó un análisis de sus características fundamentales, así como de los principales usos que tiene en la actualidad.

### INTRODUCCIÓN

Es el año 2013, y en la Industria Informática ha venido tomando influencia un fenómeno que se conoce como software libre. Este es un movimiento que proclama el acceso al código fuente de un programa, lo que admite ser libre de uso, ejecución, distribución y modificación. Es decir, el nuevo software creado bajo este concepto podría emplearse para cualquier fin, ejecutarse en cualquier ambiente, distribuirse a discreción del propio usuario y modificarse de ser necesario. Durante los años 90 ocurre una serie de eventos que marcan ciertas pautas para el futuro desarrollo del software libre, como es el lanzamiento de la primera versión del núcleo Linux por Linus Torvalds en 1991, y en ese mismo año Guido van Rossum libera la primera versión del lenguaje de programación Python. Los lenguajes de programación son la herramienta básica de construcción de programas, como lo son el machete y el azadón para un campesino, el pico y la pala para un constructor. Python ha ido ganando adeptos en comunidades como la de software libre, científica y educacional, por su sencillez y posibilidad de concentrarse en los problemas actuales. Este artículo hace referencia a las principales características y los diferentes usos de este lenguaje de programación, por lo que se ha tratado de simplificarla parte técnica. No obstante, para su completa comprensión se necesita un nivel básico de conocimientos acerca de programación.

### RESULTADOS DEL TRABAJO

Python cuenta con facilidades para la programación orientada a objetos, imperativa y funcional, por lo que se considera un lenguaje multi-paradigmas. Fue basado en el lenguaje ABC y se dice que fue influenciado por otros como C, Algol 60, Modula-3 e

Icon según su propio autor.

### Programar en PYTHON

Rafael Gonzalez Freites Pepelo. "Programar en PYTHON". Azua Republica Dominicana.

### Resumen

La elección del primer lenguaje de programación es un debate recurrente entre los docentes universitarios de ingenierías informáticas. La Universitat Jaume I ha optado por una solución poco convencional: en el primer curso de dos titulaciones de ingeniería informática se aprende a programar con Python y C. Python es un lenguaje que está en auge en el mundo del software libre y que presenta una serie de características que lo hacen muy atractivo para enseñar a programar. Como material de apoyo hemos escrito un libro de texto (accesible gratuitamente) y desarrollado un sencillo entorno de programación multiplataforma para Python que se distribuye con licencia GPL: el entorno PythonG, formado por un intérprete interactivo, un editor, un depurador sencillo y una ventana con salida gráfica. Con el material docente elaborado se facilita la formación autodidacta para cualquiera que quiera aprender a programar desde cero. En este artículo reflexionamos sobre la idoneidad de Python como primer lenguaje de programación, describimos la experiencia docente de enseñar Python y C en primer curso y presentamos el entorno de programación PythonG.

### Conclusiones

Algunas universidades norteamericanas han empezado a adoptar Python como lenguaje de programación básico. En España, de momento, ya hay una experiencia piloto: la Universidad Jaume I. Durante los cursos 2001/2002 y 2002/2003 se ha puesto en práctica la enseñanza de la programación con Python y C. Creemos que Python es un lenguaje particularmente adecuado como primer lenguaje de programación. Uno de los resultados de la experiencia es el material docente confeccionado: un sencillo entorno de programación, una biblioteca simplificada para la implementación de programas gráficos y un libro de texto. El material se encuentra disponible en Internet y puede utilizarse para el aprendizaje autodidacta de la programación.

La experiencia docente ha sido muy satisfactoria. En un breve plazo de tiempo (un cuatrimestre), los estudiantes aprenden todos los aspectos básicos de la programación. Completamos su formación básica en la materia con la enseñanza del lenguaje C en el segundo semestre. La formación previa con Python es de gran ayuda para asimilar el nuevo lenguaje y entender las cuestiones técnicas que determinan ciertas cuestiones de diseño de C.

---

## **3. Marco Teórico**

### **3.1. Diagrama UML**

UML son las siglas de “Unified Modeling Language” o “Lenguaje Unificado de Modelado”. Se trata de un estándar que se ha adoptado a nivel internacional por numerosos organismos y empresas para crear esquemas, diagramas y documentación relativa a los desarrollos de software (programas informáticos).

Dentro de esta investigación el diagrama de UML es de vital importancia para definir el funcionamiento del programa y además las clases que van intervenir en ella.

### **3.2. ¿Qué es y cómo funciona una máquina expendedora?**

La máquina expendedora es una máquina que proporciona aperitivos, bebidas, golosinas y otros productos a los consumidores. Se trata de vender sin la presencia de un dependiente para cobrar los artículos. Periódicamente un empleado repone el producto y recoge el dinero en forma de monedas o, menos habitualmente, billetes; a veces también se puede pagar. Las posibilidades de las máquinas expendedoras son amplias. Normalmente suelen vender refrescos, café, comida, chucherías, etc. Pero también existen modelos diseñados para vender prensa, libros, sellos de correos, billetes del transporte público, bebidas alcohólicas, cigarrillos de tabaco, también son frecuentes, en las oficinas que atienden al público, las máquinas expendedoras de un impreso pequeño con el número de turno del solicitante.

### **3.3. Funciones en Python**

#### **3.3.1. ¿Qué es una función en Python?**

Una función es un bloque de código con un nombre asociado, que recibe cero o más argumentos como entrada, sigue una secuencia de sentencias, la cuales ejecuta una operación deseada y devuelve un valor y/o realiza una tarea, este bloque puede ser llamado cuando se necesite. El uso de funciones es un componente muy importante del paradigma de la programación llamada estructurada, y tiene varias ventajas:

- Modularización: permite segmentar un programa complejo en una serie de partes o módulos más simples, facilitando así la programación y el depurado.
- Reutilización: permite reutilizar una misma función en distintos programas.

Python dispone de una serie de funciones integradas al lenguaje, y también permite crear funciones definidas por el usuario para ser usadas en su propios programas.

### 3.3.2. Funciones Usadas

#### Bucle While

Un bucle while permite repetir la ejecución de un grupo de instrucciones mientras se cumpla una condición (es decir, mientras la condición tenga el valor True).

La sintaxis del bucle while es la siguiente:  
while condicion:  
cuerpo del bucle

La ejecución de esta estructura de control while es la siguiente:  
Python evalúa la condición:

- si el resultado es True se ejecuta el cuerpo del bucle. Una vez ejecutado el cuerpo del bucle, se repite el proceso (se evalúa de nuevo la condición y, si es cierta, se ejecuta de nuevo el cuerpo del bucle) una y otra vez mientras la condición sea cierta.
- si el resultado es False, el cuerpo del bucle no se ejecuta y continúa la ejecución del resto del programa.

La variable o las variables que aparezcan en la condición se suelen llamar variables de control. Las variables de control deben definirse antes del bucle while y modificarse en el bucle while.

#### Condicionales if y else

Algunas veces en nuestros programas es necesario que tomemos algunas “decisiones”, esto en el sentido de que necesitamos decidir si ejecutar una pieza en especial de código

---

o no, o tal vez dadas ciertas condiciones tendríamos varias alternativas de código que ejecutar. Para esto tenemos una sentencia llamada `if` y otra llamada `else`.

Sentencia `if` en python

La sentencia `If` evalúa básicamente una operación lógica, es decir una expresión que de como resultado verdadero o falso (`true` o `false`), y ejecuta la pieza de código siguiente siempre y cuando el resultado sea verdadero.

Sintaxis:

La sintaxis va de la siguiente manera:

```
1 if expresion: donde la expresion es verdadera
2   dejar una indentacion
3   codigo_a_ejecutar
4 fin
5 alternativa
6
7 if expresion : codigo_a_ejecutar
8 sin indentacion no se evalua elif
9 fin
```

Ejemplo básico .

```
1 !/usr/bin/python
2 3 a = 7
4 5 if ( a > 5) : print "La variable es mayor a 5!"
6 7 print "fin"
```

Lo que has visto en este ejemplo es una evaluación de  $(a > 5)$  dado que “a” tiene un valor a 7, y 7 es mayor a 5, el resultado de la operación lógica es verdadero, por lo tanto si ejecutas este programa tu resultado será:

```
1 La variable es mayor a 5!
2 fin
```

Si por otro lado cambiáramos el programa y dijéramos que “a” es igual a 3. El resultado sería únicamente:

```
1 fin
```

Pues la condición habrá dejado de ser verdadera.

Ejemplo con sintaxis alternativa:

```
1 !/usr/bin/python
2 3 a = 10
4 5 if ( a == 10) :
6   print "La variable es igual a 10!"
7 8 print "fin"
```

Usando `if` y `else` El utilizar `else` lo que hace es que podemos tener la capacidad de ejecutar un código alternativo en caso de que el resultado lógico de la expresión evaluada sea falso. Esta es una forma de tener varias alternativas de código a ejecutar. El uso

---



con un ejemplo seria el siguiente.

```
01 !/usr/bin/python
02 03 a = 10
04 05 if ( a != 10) :
06 print "La variable es diferente de 10!"
07 else:
08 print "La variable es igual a 10!"
09 10 print "fin"
```

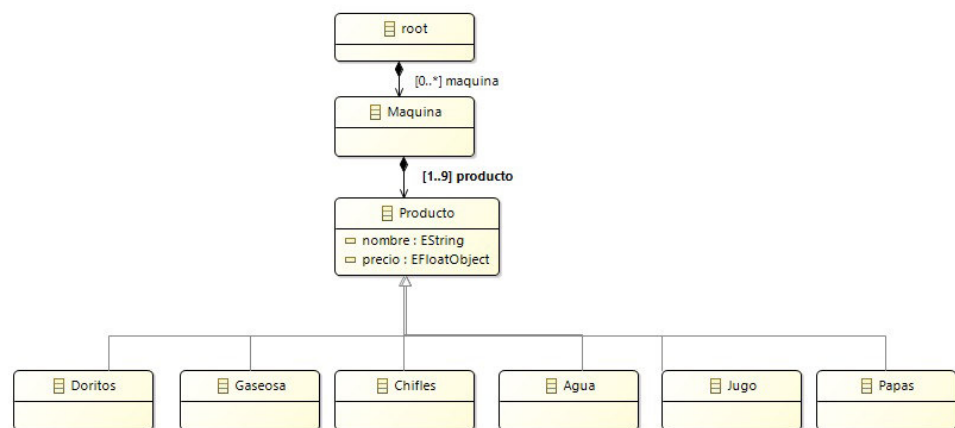
En este ejemplo se puede ver que la expresion solo sera verdadera si la variable es diferente de 10, dado que esto no es cierto, ejecutaremos la alternativa con “else”, el resultado de este programa sera el siguiente.

```
1 La variable es igual a 10!
2 fin
```

## 4. Diagramas

### 4.1. Diagrama de UML

Representación de la máquina dispensadora



## 5. Explicación

### 5.1. Código Fuente

- `import numpy`: este comando nos permite mejorar el aspecto visual en la creación de matrices, ya que nos permite ver filas y columnas. Dentro de la programación de la máquina esto nos permite poner los productos de una forma similar a la que se encuentra en las máquinas de la vida real.
- `while`: este bucle nos permite hacer repeticiones con el objetivo de que el usuario logre ingresar su dinero hasta que alcance el precio indicado.
- `if-else`: este condicional nos permite diferenciar 2 casos que se pueden presentar durante la compra un producto: el primer caso sucede cuando el dinero ingresado es mayor o igual al del producto, en esta condición el programa devuelve un cambio.

El segundo caso es cuando el valor ingresado es menor al del producto, el software te dira la cantidad de dinero que te falta, si es que lo ingresas te hará la entrega de tu producto.

### 5.2. Hardware

Además debe tener el siguiente Hardware para que todo logre funcionar

- Raspberry Pi 3 Modelo B+
- Monitor
- mando a distancia

### 5.3. Configuración y Requisitos

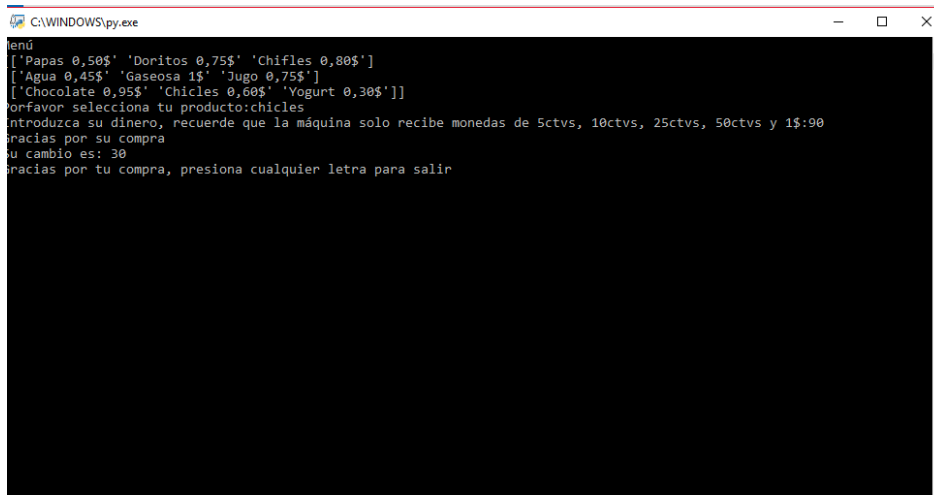
El programa de la máquina expendedora funciona si se tiene los siguientes requisitos en nuestro ordenador:

- Instalación completa de Python: para usar la libreria numpy se debe tener instalado de forma completa python, esto se logra en el momento de la instalación, en la primera ventana que aparece se deben marcar las 2 casillas que se encuentran en la parte inferior izquierda

- En la Raspberry se debe usar los programas predeterminados actualizados o de acuerdo a la versión que sea compatible a con el Python que se tiene instalado.

## 5.4. Ejemplo

La siguiente imagen presenta el ejemplo de una compra en la que se dio un vuelto



```
C:\WINDOWS\py.exe
menú
['Papas 0,50$' 'Doritos 0,75$' 'Chifles 0,80$']
['Agua 0,45$' 'Gaseosa 1$' 'Jugo 0,75$']
['Chocolate 0,95$' 'Chicles 0,60$' 'Yogurt 0,30$']
porfavor selecciona tu producto:chicles
introduzca su dinero, recuerde que la máquina solo recibe monedas de 5ctvs, 10ctvs, 25ctvs, 50ctvs y 1$:90
gracias por su compra
tu cambio es: 30
gracias por tu compra, presiona cualquier letra para salir
```

## 6. Manual de Usuario

Será necesario que este instalado una librería para que el programa pueda correr. El nombre de esta librería es “numpy”, en caso de no tenerla instalada será necesario hacerlo a partir de símbolo del sistema (cmd) utilizando el siguiente comando: `pip install numpy`. Una vez sabiendo que esta todo instalado correctamente procedemos a iniciar nuestro programa en Python. Luego seguimos las órdenes y recomendaciones que el programa brinda. Finalmente se adquiere el producto con el cambio correspondiente. En caso de necesitar adquirir otro producto será necesario reiniciar el programa.

## 7. Conclusiones

- Mediante el lenguaje de programación de Python podemos crear una simulación de una máquina expendedora.
- La ayuda de las librerías, permite mejorar el aspecto visual al ejecutar la programación
- Mediante conceptos básicos de la programación se puede realizar diferentes programas.

### 7.1. Recomendaciones

- Revisar que versión de Python se tienes instalada, para poder instalar la librería que funciones en dicha versión
- Iniciar las aplicaciones correctas en la Raspberry para poder ejecutar el programa.

## 8. Bibliografía

- Introducción a la programación con Python. Publicación de la Universidad Jaume, 12071 Castelló de la Plana. Andrés Marzal Isabel Gracia. Departamento de lenguajes y sistemas informáticos.
- El lenguaje de programación Python. Cuba, [Documento recuperado de internet disponible en <http://www.redalyc.org/html/1815/181531232001/>], Sistema de Información Científica Red de Revistas Científicas de América Latina y el Caribe, España y Portugal.
- Bucle while. Marzo 20, 2019. [Documento recuperado de internet disponible en <http://www.mclibre.org/consultar/python/lecciones/python-while.html>].
- Sentencia def. 2014 - 2018, Covantec R.L. [Documento recuperado de internet disponible en <https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion5/funciones.ht>].