

DISEÑO DE LA INTERFAZ DE UNA MÁQUINA EXPENDEDORA DE ALIMENTOS UTILIZANDO LENGUAJE DE PROGRAMACIÓN PYTHON

1 PLANTEAMIENTO DEL PROBLEMA

Se desconoce la interfaz de funcionamiento, control y conexión de una máquina expendedora de alimentos:

- ¿Cuáles son las librerías utilizadas en la interfaz de funcionamiento?
- ¿Cuál es el lenguaje de programación utilizado para controlar una máquina expendedora de alimentos?
- ¿Cómo funciona una máquina expendedora de alimentos?

2 OBJETIVOS

A. Objetivo general

Diseñar la interfaz de una máquina expendedora de alimentos mediante el uso del Lenguaje de programación Python.

B. Objetivos específicos

- Identificar los principales parámetros de la programación orientada a objetos en el diseño de la interfaz.
- Implementar clases, atributos, métodos y funciones en el diseño del código
- Realizar una interfaz amigable con el usuario que permita la interacción directa entre el comprador y la consola.

3 ESTADO DEL ARTE

En 2018, Fangtao Liu y Zhiyong Yang del Department of Information Engineering del Institute of Engineering Chongqing en China, desarrollaron el Diseño del sistema automático de operación y mantenimiento de VMware vSphere basado en Python, un tipo de sistema automático de operación y mantenimiento establecido sobre la base del lenguaje de programación objetivo Python mediante el análisis de los requisitos durante la operación y el mantenimiento de VMware vSphere. El resultado indica que este sistema tiene alta eficiencia, universalidad y expansibilidad. Debido a su código simple, capa clara y período corto, también tiene un amplio potencial de aplicación al utilizar el sistema automático de operación y mantenimiento de VMware vSphere desarrollado por Python (Fangtao Liu, Zhiyong Yang, 2018, p.1) [1].

En 2019, Wook Hyun, Mi Young Huh y Ju Young Park del ETRI (Electronics and Telecommunications Research Institute en Korea, implementaron un protocolo de control de servicio de invernadero usando Python, la tecnología de las TIC se aplica ampliamente a diversas industrias, incluida la agricultura, la mayoría de los dispositivos e invernaderos siguen utilizando tecnologías de comunicación en serie muy antiguas. Hoy en día, muchas tecnologías TIC como ZigBee, Bluetooth e Internet se están difundiendo pero aún no son interoperables. Se está desarrollando un conjunto de estándares en Corea para garantizar la interoperabilidad entre diferentes proveedores.

En particular, la interoperabilidad del núcleo se puede proporcionar mediante la estandarización del protocolo LCP independiente de la infraestructura de red subyacente. En este documento, diseñamos un protocolo de control de servicio basado en el protocolo de Python (Wook Hyun, Mi Young Huh, Ju Young Park, 2019) [2].

En 2016, Ing. Felix Palacios desarrolló un Sistema de identificación de huella dactilares para el acceso a lugares restringidos y control de asistencia con protocolo de comunicación IEEE 802.11. El presente trabajo muestra la implementación de un Sensor Biométrico GT511C3, un Arduino Nano, una Raspberry Pi 2.0, una pantalla táctil LCD 3.5" y un módulo USB Wifi, el entorno de programación para el desarrollo del aplicativo Web, se hizo mediante los módulos Django y Adafruit sobre la base de Python. El sistema tiene la capacidad de registrar huellas de personas, las cuales se les asigna un identificador entero de entre 0 a 199 y almacenarlo en una base de datos instalada como es MySQL, además tiene la capacidad de almacenar nombres asociados al identificador generado, en la base de datos con el cual se puede habilitar el acceso hacia algún lugar específico (Palacios, 2016, p.1) [3].

En mayo del 2016, Matteo Orru, Ewan Tempero, Michele Marchesi y Roberto Tonelli del Dept. of Electrical and Electronic Engineering (DIEE) de la University of Cagliari, Italia, desarrollaron un estudio de replicación basado en la explicación de ¿Cómo utilizan la herencia los programas de Python?. En este trabajo presentamos un estudio empírico sobre el uso de la herencia en un cuerpo curado de sistemas Python. Al replicar un estudio realizado en Java, analizamos una colección de 51 sistemas de software escritos en Python, e investigamos cómo los herederos de Python utilizan efectivamente la herencia en la práctica a través de un conjunto conveniente de métricas de herencia. Nuestros resultados sugieren que, en promedio, menos clases se heredan de otras clases que en Java, pero se heredan más clases. También vemos una especie de simetría que relaciona el número de ancestros y el número de descendientes en cada sistema (Matteo Orru, Ewan Tempero, 2016) [4]

En 2015, Andrej Trost y Andrej Žemva de la Faculty of Electrical Engineering, de la University of Ljubljana, Slovenia desarrollaron un Generador de componentes de hardware configurable en Python, el documento presenta una metodología basada en un lenguaje Python para el diseño de generadores de componentes de hardware en un nivel de abstracción superior. El lenguaje de scripting se usa para producir un ciclo personalizable de comportamiento de hardware preciso y las herramientas de código abierto proporcionan conversión automática para registrar el nivel de transferencia. Un estudio de caso presenta el diseño configurable de componentes de procesamiento de gráficos. La inserción automática de la interfaz, la síntesis de la máquina de estado y la configuración de la tubería proporcionada por la metodología propuesta permite una descripción eficiente del hardware y la exploración del espacio de diseño (Andrej Trost, Andrej Žemva) [5].

4 MARCO TEÓRICO

4.1 Raspberry Pi 3 – modelo B

Raspberry Pi es un ordenador de placa reducida, ordenador de placa única u ordenador de placa simple (SBC) de bajo costo desarrollado en el Reino Unido.

Aunque no se indica expresamente si es hardware libre o con derechos de marca, es compatible con varios Sistemas operativos de libre acceso, siendo su sistema operativo oficial una versión adaptada de Debian, denominada Raspbian, aunque permite usar otros sistemas operativos, promueve principalmente el aprendizaje del lenguaje de programación. Python – IDLE, otros lenguajes también soportados son: Tiny BASIC, 11 C, Perl4 y Ruby 12.

IDLE (Integrated Development and Learning Environment) es un entorno de desarrollo integrado (IDE) para Python. El instalador de Python para Windows contiene el módulo IDLE de manera predeterminada

4.2 Python – IDLE

IDLE se puede usar para ejecutar una sola declaración como Python Shell y también para crear, modificar y ejecutar scripts de Python. IDLE proporciona un editor de texto con todas las funciones para crear scripts de Python que incluye funciones como resaltado de sintaxis, autocompletado y sangría inteligente. También tiene un depurador con características de pasos y puntos de interrupción.

NumPy

Es una extensión de Python, que le agrega mayor soporte para vectores y matrices, constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices. El ancestro de NumPy, Numeric, fue creado originalmente por Jim Hugunin con algunas contribuciones de otros desarrolladores.

4.3 Librerías utilizadas:

`import board`

Esta librería sirve para asignar constantes fijas a los pines del tablero, esto hace que usar el módulo de la placa sea más seguro y confiable.

`import digitalio`

Este módulo sirve utilizar la pantalla LCD con una retroalimentación de un solo color, en el caso que se desee una retroalimentación RGB (varios colores) se debe definir los pines de salida según su ubicación en el board y el color.

`import adafruit_character_lcd.character_lcd as characterlcd`

Este módulo permite escribir fácilmente el código de Python que controla una LCD de caracteres (ya sea con luz de fondo individual o con luz de fondo RGB)

`os.clear`

Cuando se trabaja con el terminal / shell interactivo de Python (no una consola), se termina con una salida desordenada y se desea borrar la pantalla por alguna razón. Si se trabaja en una terminal y se quiere hacer `cls`, se puede hacer `CTR + L` o usando el código de Python : `import os. os.system('agregar el comando clear basado en OS')`

`temps`

‘temps’ es un módulo de Python que contiene administradores de contexto para crear y limpiar archivos y directorios temporales.

Máquinas Expendedoras

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 con tarjeta monedero, tarjeta de crédito

o teléfono móvil.

Tipos de Máquinas Expendedoras.

- Mecánicas: Aquellas en que todo su funcionamiento es mecánico, sin intervención de ningún mecanismo eléctrico o electrónico. Son máquinas sencillas, prácticamente en desuso por las limitaciones que presentan.
- Electrónicas: Cuentan con componentes electrónicos para su funcionamiento y necesitan de energía eléctrica.

Ventajas y Desventajas.

Entre las muchas ventajas de las máquinas expendedoras, destacan el que trabajan las 24 h los 365 días del año, no necesitan personal cualificado ni atención personalizada, no se requiere contratación de personal para su explotación, no requiere largas horas de trabajo, solo unas pocas horas de limpieza, recarga y recaudación, por lo que este negocio es perfectamente compatible con cualquier otro trabajo.

Otra de las ventajas de las máquinas expendedoras es la movilidad. En función de los resultados que se obtenga, la máquina se puede cambiarse de sitio ya que el emplazamiento lo es todo, no hay vendedor y es el consumidor el que decide si en ese momento quiere adquirir un producto.

5 DIAGRAMAS

5.1 DIAGRAMA DE ELECTRÓNICO

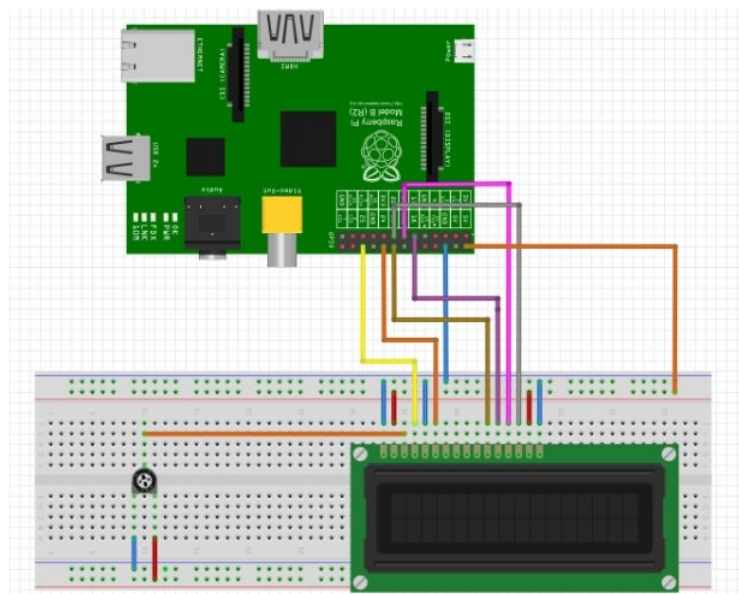


Fig 4. Diagrama Electrónico

6 LISTA DE COMPONENTES

- LAPTOP HP DESKTOP-H07UCK2:
Procesador: AMD E1-6015 APU con Radeon TM R2 graphics 1,40 GHz
RAM: 4 GB, 330 GB de almacenamiento interno
Tipo de Sistema: 64 bits
- Python 32 bits
- Editor de código PyCharm
- Pantalla LCD 16x2
- Potenciómetro 10K
- Pantalla LCD 16x2
- Raspberry Pi 3 model-B
- Proto board
- Micro SD de 8 GB

7 MAPA DE VARIABLES

Clase menú

variable	Función operativa
saludo	Imprime el mensaje inicial “bienvenido”
margenInf	Línea de margen (estético)
advertencia	Devuelve una advertencia al usuario
listaProducto	El arreglo con el que se presenta el menú

Clase operación	
variable	Función operativa
gracias	Imprime el mensaje final de la interfaz
descision1	Condición de bucle, si el usuario ingresa el valor de 0 repite el bucle, si escoge 1 cierra el ciclo
opcionP	permite al usuario escoger el producto que desea consumir.
costo1	Es el valor del producto N.-1 valor: 40 ctvs.
costo2	Es el valor del producto N.-2 valor: 50 ctvs.
costo3	Es el valor del producto N.-3 valor: 35 ctvs.
costo4	Es el valor del producto N.-4 valor: 45 ctvs.
costo5	Es el valor del producto N.-5 valor: 75 ctvs.
costo6	Es el valor del producto N.-6 valor: 70 ctvs.
monedaIn	Es el valor que el usuario ingresará para comprar su producto
valorT	Contador que se ira acumulando hasta que su valor sea igual al coste del producto
cond	Variable que hará cumplir al bucle el ciclo siempre y cuando sea mayor que valorT, y es la diferencia del costo y la moneda ingresada
monedaIn2	Valor que el usuario ingresa después de monedaIn en caso de que esta última no sea igual o mayor al costo del producto elegido
costoInG	Devuelve el valor total ingresado por el usuario
cambio	Devuelve la diferencia entre el valor total ingresado y el costo del producto elegido en caso de que este ultimo sea menor.

8 EXPLICACIÓN DEL CÓDIGO FUENTE

LIBRERÍAS

import numpy as np

Esta libreria permite realizar arreglos bidimensionales matrices sin necesidad de utilizar ciclos de repetición para su impresión

CLASES

clase menu

En esta clase se va a estructurar la matriz de productos o lista de productos indica al usuario las opciones que puede elegir

```
class operacion
```

En esta clase se van a estructurar todas las operaciones e instrucciones que el programa debe seguir para realizar las diferentes validaciones al momento que el usuario ingrese los valores correspondientes.

```
Producto de Unidad 2.py x
```

```
1  
2 #----- LIBRERIAS -----  
3  
4 import numpy as np.# Esta libreria permite realizar arreglos bidimensionales o  
5                     # matrices sin necesidad de utilizar ciclos de repetición  
6                     # para su impresión  
7  
8  
9 #----- CLASES -----  
10  
11 class menu:# En esta clase se va ha estructurar la matriz de productos o lista de productos  
12           # que indica al usuario las opciones que puede elegir  
13  
14     #----Atributos-----  
15     # Los atributos definidos en la clase menu sirven para estructurar el encabezado  
16     # que es una guía visual para el usuario  
17     saludo="\n>>>>>>>>>>>>>>>>>>>> BIENVENIDO <<<<<<<<<<<<<<<<<<<<<"  
18     margenInf=">>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<"  
19     advertencia="!! NO SE ACEPTAN MONEDAS DE 1 ctvs !! \n"  
20  
21     # ----Metodos-----  
22     # Este metodo contiene todas las opciones de productos que puede elegir el usuario  
23     def matrizProducto():...  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40 class operacion:  
41  
42     gracias= "\n                !!! GRACIAS POR SU COMPRA !!!"  
43  
44     def ingresoOpc():...  
45  
46  
47  
48  
49  
50 #-----VOID-----  
51  
52 #----- OBJETOS -----  
53 #Los objetos se utilizan para invocar a las funciones , métodos, atributos  
# e imprimir los procesos efectuados en cada clase segun corresponda
```

Fig 5. Código Fuente

ATRIBUTOS

saludo, margenInf, advertencia, gracias

Estos atributos son los atributos tanto de la clase menu como de la clase operacion, sirven para imprimir en pantalla mensajes específicos sin necesidad de escribir líneas de código, haciendo que el código del programa se vuelva más compacto, debido a que se los puede invocar en cualquier lugar de la interfaz según se los necesite.

MÉTODOS

```
def matrizProducto():
```

Este método contiene todas las opciones de productos que puede elegir el usuario, dentro de este método creamos la matriz (listaProducto:[]) en la cual ingresamos los productos y su costo individual.

```
def ingresoOpc():
```

```
Producto de Unidad 2.py x
```

```
1  
2 #----- LIBRERIAS -----  
3  
4 import numpy as np # Esta libreria permite realizar arreglos bidimensionales o  
5                     # matrices sin necesidad de utilizar ciclos de repetición  
6                     # para su impresión  
7  
8  
9 #----- CLASES -----  
10  
11 class menu: # En esta clase se va ha estructurar la matriz de productos o lista de productos  
12             # que indica al usuario las opciones que puede elegir  
13  
14     #----Atributos-----  
15     # Los atributos definidos en la clase menu sirven para estructurar el encabezado  
16     # que es una guía visual para el usuario  
17     saludo="\n>>>>>>>>>>>>>>>>>>>>>>>>> BIENVENIDO <<<<<<<<<<<<<<<<<<<<<<<"  
18     margenInf=">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<"  
19     advertencia="" !! NO SE ACEPTAN MONEDAS DE 1 ctvs !! \n"  
20  
21     # ----Metodos-----  
22     # Este metodo contiene todas las opciones de productos que puede elegir el usuario  
23     def matrizProducto():  
24         listaProducto = np.array(  
25             [[ " ", " ", " ", " " ],  
26              [ " 1. Festival   ", " 2. Oreo      ", " 3. Nestle    ", " 4. Ricas   " ],  
27              [ " 40 ctvs       ", " 50 ctvs      ", " 35 ctvs     ", " 45 ctvs   " ],  
28              [ " ", " ", " ", " " ],  
29              [ " 5. Kit Kat    ", " 6. Galak     ", " 7. Manicho   ", " 8. Snickers"],  
30              [ " 75 ctvs       ", " 50 ctvs      ", " 45 ctvs     ", " 35 ctvs   " ],  
31              [ " ", " ", " ", " " ],  
32              [ " 9. Doritos    ", " 10. Rufles   ", " 11. Kchitos  ", "12. DeIodito"],  
33              [ " 50 ctvs       ", " 50 ctvs      ", " 35 ctvs     ", " 70 ctvs   " ],  
34              [ " ", " ", " ", " " ]])  
35         return listaProducto #El retorno nos sirve para guardar en una variable la  
36                               # matriz listaProducto y posteriormente invocarla en el
```

OBJETOS

producto=menu

8

instruccion=operacion

Mediante la creación del objeto instrucción podemos invocar en pantalla toda la parte operativa de la interfaz, debido a que en la clase operacion se encuentran todas las instrucciones del programa.

```
249 #-----VOID-----
250
251 #----- OBJETOS -----
252 #Los objetos se utilizan para invocar a las funciones , métodos, atributos
253 # e imprimir los procesos efectuados en cada clase segun corresponda
254
255 #----- ENCABEZADO -----
256 producto=menu
257 print(producto.saludo)
258 print(producto.matrizProducto())
259 print(producto.margenInf)
260 print(producto.advertencia)
261
262
263 #----- EJECUCION -----
264 instruccion=operacion
265 print(instruccion.ingresoOpc())
266 print(instruccion.gracias)
267
```

Fig 5. Código Fuente

8.1 DESCRIPCIÓN DE PRERREQUISITOS Y CONFIGURACIÓN

Tener el sistema operativo Raspbian ya que ahí viene incluido Python el lenguaje de programación utilizado para el desarrollo de la interfaz de la máquina expendedora de alimentos y así presentar su diseño en la pantalla LCD 16x2. En el caso de instalar Python en un PC es de vital importancia instalar la extensión Numpy.

- Una Raspberry Pi 2, 3 o 3+
- Un PC o Mac con Etcher.io instalado para preparar la tarjeta microSD.
- Fichero imagen con la última versión de Raspbian Stretch Lite descargada desde la web de Raspberry Pi.
- Una tarjeta SD de 4GiB, si es más pequeña puede que no funcione correctamente. Es recomendable usar una más grande, ya que con los tamaños de archivos que se manejan actualmente 4GiB (menos lo que ocupe el sistema y las aplicaciones) se quedan en poco.
- Tener conexión a internet no es un requisito fundamental, pero si recomendado, ya que permitirá instalar parte de las actualizaciones durante el proceso de instalación.
- Es esencial el módulo Adafruit CircuitPython CharLCD ya que esta biblioteca es compatible con la LCD de caracteres estándar y Numpy ya que facilitará la creación de matrices en el código fuente sin la necesidad de la creación de ciclos.
- Es esencial el módulo Adafruit CircuitPython CharLCD ya que esta biblioteca es compatible con la LCD de caracteres estándar.

9 APORTACIONES

Aprovechando la interfaz de Python se realizó un programa más amigable con el usuario, donde se puede proyectar varios de los mensajes que aparecen en la consola en una pantalla LCD por medio de una conexión de puertos con una Raspberry Pi 3 modelo B.



The screenshot shows the Thonny Python IDE interface. The top bar indicates the file path: `Thonny - /home/pi/Desktop/Segundo Parcial/ProdUnidad 2_Aportacion.py @ 337 : 1`. The menu bar includes `File Edit View Run Device Tools Help`. The toolbar contains icons for file operations and execution. The editor window displays the file `ProdUnidad 2_Aportacion.py` with the following code:

```
61 def matrizProducto():
62     listaProducto = np.array(
63         [
64             [" 1. Festival ", " 2. Oreo ", " 3. Nestle ", " 4. Ricas "],
65             [" 40 ctvs ", " 50 ctvs ", " 35 ctvs ", " 45 ctvs "],
66             [" 5. Kit Kat ", " 6. Galak ", " 7. Manicho ", " 8. Snickers"],
67             [" 75 ctvs ", " 50 ctvs ", " 45 ctvs ", " 35 ctvs "],
68             [" 9. Doritos ", " 10. Rufles ", " 11. Kchitos ", " 12. DeTodito"],
69             [" 50 ctvs ", " 50 ctvs ", " 35 ctvs ", " 70 ctvs "],
70             [" 50 ctvs ", " 50 ctvs ", " 35 ctvs ", " 70 ctvs "],
71             [" 50 ctvs ", " 50 ctvs ", " 35 ctvs ", " 70 ctvs "],
72             [" 50 ctvs ", " 50 ctvs ", " 35 ctvs ", " 70 ctvs "],
73         ],
74         dtype=object
75     )
76     return listaProducto
77
78 class operacion:
79     gracias=" \n          !!! GRACIAS POR SU COMPRA !!!"
```

The Shell window shows the execution output:

```
----- Retire su producto -----

Digite '0' para realizar una nueva compra o digite '1' para salir
1
None

          !!! GRACIAS POR SU COMPRA !!!

>>> |
```

The bottom status bar shows the Raspberry Pi icon, a file explorer icon, and the text `Segundo Parcial` and `Th Thonny - /home/pi/...`.

Fig 21. Mensaje de final de agradecimiento al usuario.

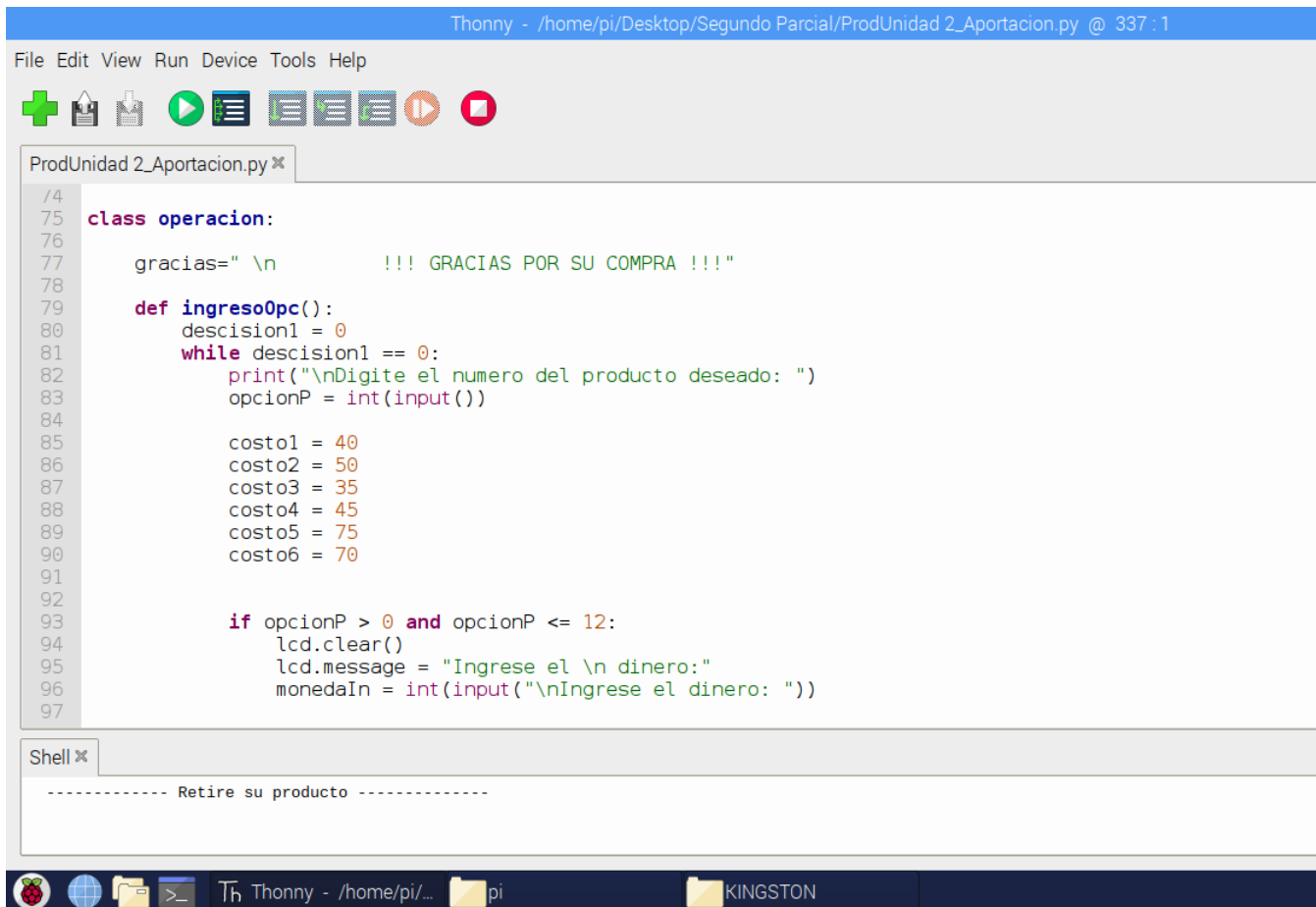


Fig 21.Mensaje de final de agradecimiento al usuario.



Fig 21.Mensaje de final de agradecimiento al usuario.

10 CONCLUSIONES

En conclusión:

- La programación orientada a objetos es de vital importancia ya que está basada en el modo de pensar del ser humano y en el modo de trabajar de la máquina expendedora de alimentos, el elemento básico de esta programación no es solo la función sino un ente denominado objeto.
- El uso de las clases con sus respectivos atributos y métodos junto con funciones y objetos en el diseño de la interfaz de la máquina expendedora de alimentos establecen una estructura y orden al momento de escribir las líneas de código, facilitando su comprensión y corrección en el caso de que exista un error.
- Para generar una interfaz amigable con el usuario es necesario que el programador tenga una visión externa de una máquina expendedora de alimentos, de esta forma el programa que se estructure será de fácil comprensión para el usuario a este principio se lo conoce como abstracción.

11 RECOMENDACIONES

- Se recomienda utilizar un potenciómetro de 10K para controlar el contraste de la pantalla LCD, debido a que si se conecta directamente a la Raspberry Pi se corre el riesgo de quemar los leds de la pantalla.
- Es recomendable utilizar mínimo 4 pines de los 8 pines de Datos de la pantalla LCD, para establecer una óptima comunicación entre los dispositivos.
- Al momento de escribir los mensajes de salida de la máquina expendedora que se va a proyectar en la pantalla, se debe tener en cuenta que la LCD solo soporta dos filas de 16 caracteres cada una.

12 CRONOGRAMA

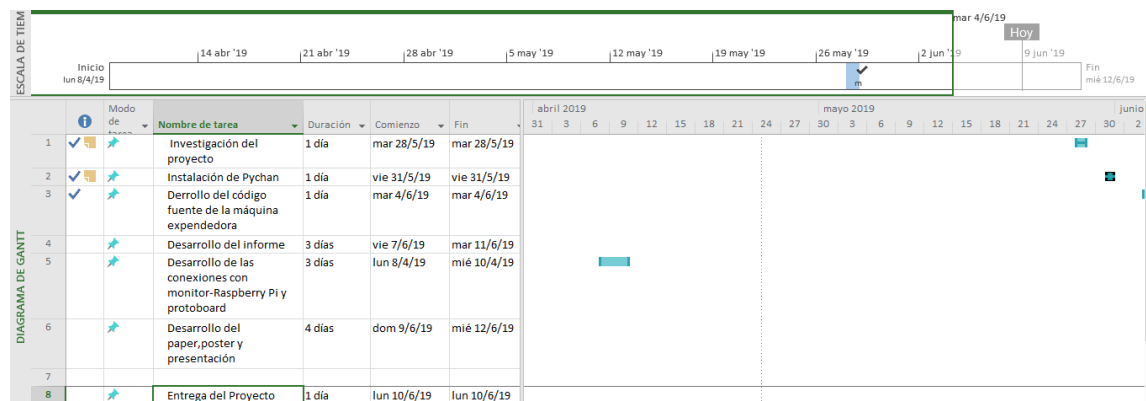


Fig 1. Cronograma de actividades

References

- [1] Liu, F., y Yang, Z. (2018). Diseño del sistema automático de operación y mantenimiento de VMware vSphere basado en Python. 2018 Conferencia internacional sobre sistemas mecatrónicos avanzados (ICAMEchS). doi: 10.1109 / icamechs.2018.8506789

- [2] Hyun, W., Huh, MY y Young Park, J. (2019). Implementación del protocolo de control de servicio de invernadero utilizando Python en Raspberry PI. 2019 21ª Conferencia Internacional sobre Tecnología de Comunicación Avanzada (ICACT). doi: 10.23919 / icact.2019.8701890
- [3] Palacios, F. (2016). Sistema de identificación de huella dactilares para el acceso a lugares restringidos y control de asistencia con protocolo de comunicación IEEE 802.11. ISBN-13: 978-0-13-235613-8
- [4] Orru, M., Tempero, E., Marchesi, M., y Tonelli, R. (2015). ¿Cómo utilizan la herencia los programas de Python? Un estudio de replicación. Conferencia de Ingeniería de Software Asia-Pacífico 2015 (APSEC). doi: 10.1109 / apsec.2015.51
- [5] Trost, A., y Zemva, A. (2015). Generador de componentes de hardware configurable en Python. 2015 4ª Conferencia Mediterránea de Informática Integrada (MECO). doi: 10.1109 / meco.2015.7181876
- [6] Tutorials Teacher. (2018). Python-IDLE. Blog de Tecnologías de la Información. Recuperado el 7 de mayo del 2019 de: <https://www.tutorialsteacher.com/python/python-idle>
- [7] Shead, S. (2012). Raspberry Pi delivery delays leave buyers hungry (and angry). Hardware. Recuperado el 8 de mayo del 2019 de: <https://www.zdnet.com/article/raspberry-pi-delivery-delays-leave-buyers-hungry-and-angry/>
- [8] Raspberry Shop. (2019). Hardware y Accesorios. Raspberry Pi 3 modelo B. Recuperado el 8 de mayo del 2019 de: <https://www.raspberrypi-shop.es/raspberry-pi-3.php>

13 ANEXOS

A. MANUAL DE USUARIO

1. Se deben conectar los cables macho-hembra y macho-macho entre la LCD y los pines de la Raspberry, verificamos que el circuito se encuentre correctamente armado tomando como guía la siguiente imagen:

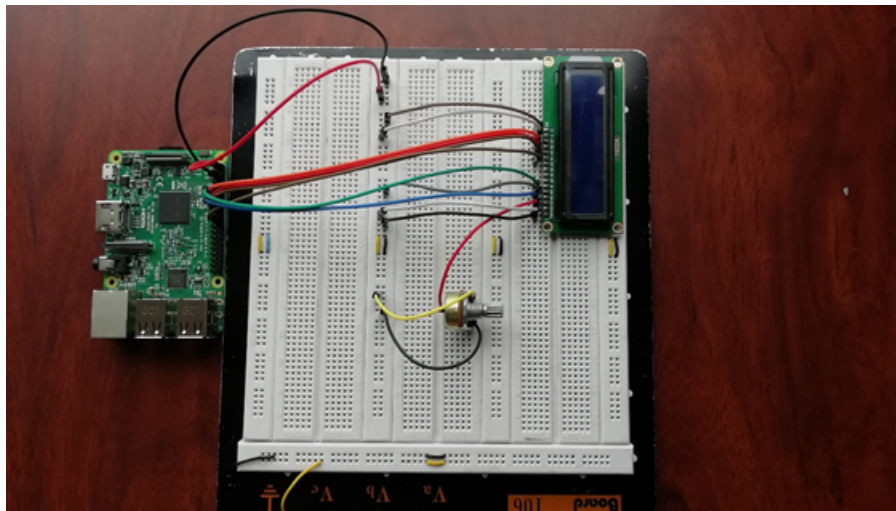


Fig 16. Implementación del Circuito

2. Para encender la Raspberry conectamos un cargador de 5 V y mínimo 1.5 A de alimentación., se conecta el cable HDMI a la Raspberry Pi y al monitor o televisor, de esta forma podremos visualizar la ventana principal del sistema operativo.


```
200 elif opcionP==12:...
201
202
203
204
205
206
207
208
209
210
211 else:
212     print("\nOpcion incorrecta \n\n")
213
214     print("\n\nDigite '0' para realizar una nueva compra o digite '1' para salir\n ")
215     decision1 = int(input())
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
```

operation > ingresoOpc() > while decision1 == 0

Fig 21. Código fuente

4. Se observa que en la pantalla LCD 16X2 se proyecta el mensaje “BIENVENIDO”, mientras que en la ventana principal de Python se imprime el mensaje “ la lista de productos y Digite el número del producto deseado”

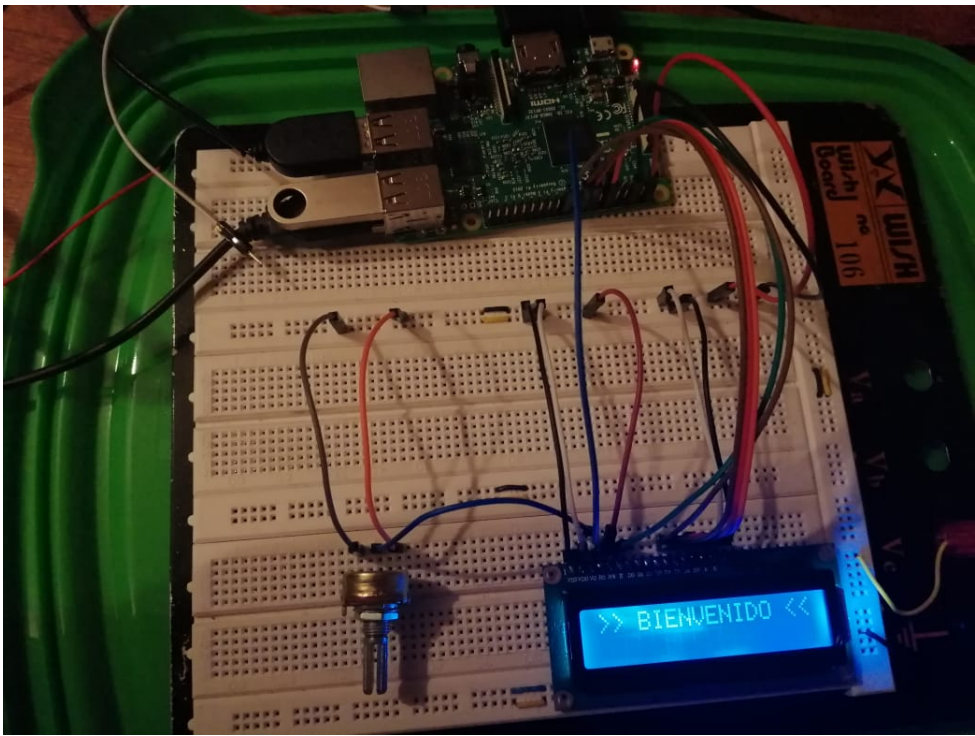


Fig 21. Código fuente

[illegible]

Fig 21. Impresión del mensaje en la ventana principal

6. Al ingresar el costo completo y correcto se proyecta el mensaje “gracias por su compra”, mientras que en la ventana se da la opción de digitar 0 para realizar una nueva compra o digitar 1 para salir quedando a decisión del usuario.



Fig 21.Mensaje de final de agradecimiento al usuario.

B. FOTOGRAFÍAS



Fig 21. Análisis de la aportación

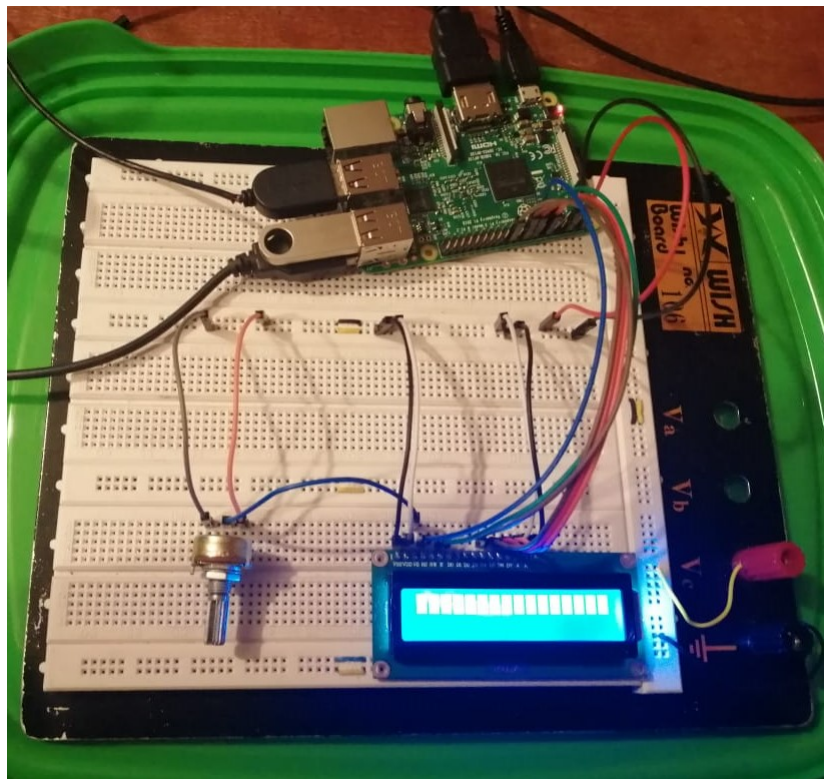


Fig 2. Conexiones LCD-Raspberry Pi

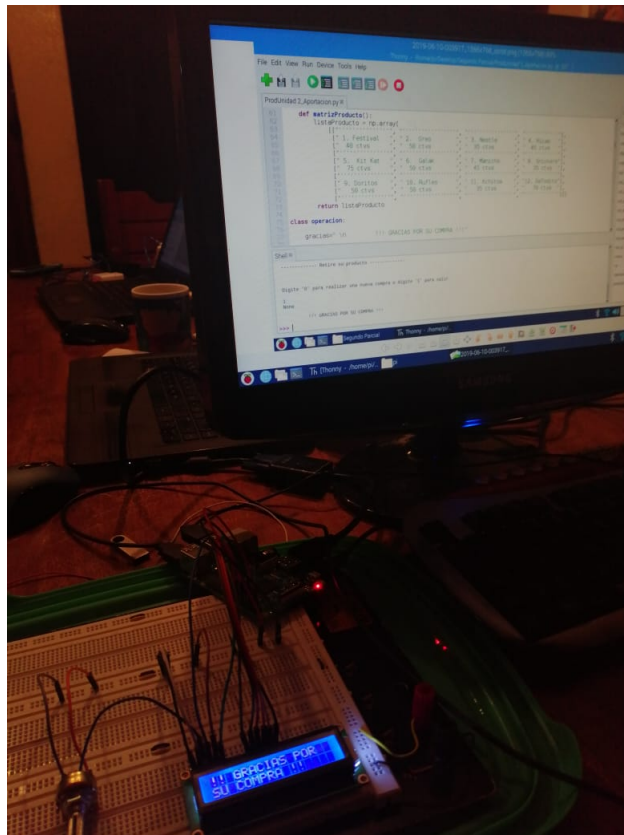


Fig 2. Estructuración del código fuente