

# DISEÑO DE LA INTERFAZ DE UNA MÁQUINA EXPENDEDORA DE ALIMENTOS

Bryan V. Alvarado <sup>1\*</sup>, Evelin E. Hidalgo <sup>1</sup>, Carlos E. Orellana <sup>1</sup>, Carlos S. Veloz <sup>1</sup>

*Departamento de Eléctrica y Electrónica, Universidad de las Fuerzas Armadas ESPE, Sangolguí, Ecuador*

bvalvarado@espe.edu.ec

## I. 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?

## II. ABSTRACT

The vending machine, as the name implies, represents a machine that provides snacks, beverages, sweets and other products, capable of providing the consumer with a certain product. The fundamental characteristic of a vending machine is the independence of personnel that fulfills the function of purchase and sale being up to a certain autonomous point, making only necessary the recharge of the product offered and the withdrawal of money collected from sale.

## III. INTRODUCCIÓN

La máquina expendedora como su nombre lo indica representa una máquina que proporciona aperitivos, bebidas, golosinas y otros productos, capaz de proveer al consumidor determinado producto. La característica fundamental de una máquina expendedora es la independencia de personal que cumpla la función de compra y venta siendo hasta determinado punto autónoma, haciendo únicamente necesaria la recarga del producto ofrecido y el retiro del dinero recolectado de venta.

El funcionamiento de la máquina se realiza de forma programada manualmente pero todos los datos visualizados en la pantalla son almacenados y programados en este caso con el lenguaje de programación Python, implementando parámetros de la programación orientada a objetos como clases padre, clases hija con sus respectivos atributos y funciones en el diseño del código fuente del programa, realizando una interfaz amigable con el usuario.

## IV. 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 Zemva) [5].

## V. MARCO TEÓRICO

### V-A. *Python – IDLE*

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

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.

### V-B. *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.

### V-C. *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.

## VI. PROPUESTA

### *LIBRERÍAS*

`import numpy as np`

Esta librería 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

*clase 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.

Fig 1. 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, para tener una apreciación visual.

```
def ingresoOpc():
```

Este es el método más importante de la interfaz, aquí se encuentran estructuradas todas las instrucciones, variables y ciclos que debe seguir el programa para establecer una interacción directa con el usuario.

Fig 2. Código Fuente

## OBJETOS

Los objetos se utilizan para invocar a las funciones , métodos, atributos e imprimir los procesos efectuados en cada clase segun corresponda

*producto=menu*

Este objeto es el encargado de imprimir en pantalla todo el encabezado de nuestra interfaz, el mismo que se encuentra en la clase menu.

*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.

Fig 3. Código Fuente

## VII. RESULTADOS

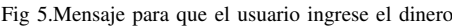
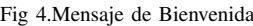




Fig 6.Mensaje final

## VIII. 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.

## REFERENCIAS

- [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 (ICTACT). doi: 10.23919 / ictact.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.raspberrishop.es/raspberry-pi-3.php>