

Informe N° 2: Restaurante

Gianfranco Pupiales

2024-11-05

Tabla de Contenidos

1. Objetivos	1
2. Introducción	1
3. Desarrollo	2
3.1 Enunciado	2
3.1 Implementación	2
4. Conclusiones	8
5. Declaración del Uso de IA	8
6. Referencias Bibliográficas	8

1. Objetivos

- Comprender la Programación Orientada a Objetos en Python.
- Implementar una solución óptima a un problema específico utilizando Python

2. Introducción

La Programación Orientada a Objetos (POO) en Python es un paradigma que organiza el código en objetos, los cuales encapsulan datos y funciones relacionadas. En este enfoque, los objetos son instancias de clases, que actúan como moldes para crearlos. Cada objeto tiene atributos que representan sus datos y métodos que definen su comportamiento. Los pilares fundamentales de la POO son la encapsulación, que protege los datos dentro de un objeto y controla su acceso mediante métodos; la herencia, que permite a las clases heredar atributos y métodos de otras clases, promoviendo la reutilización de código; y el polimorfismo, que permite a los objetos de distintas clases ser tratados de manera uniforme, lo que facilita la flexibilidad y la extensibilidad del código ([«Object-Oriented Programming in Python \(OOP\): Tutorial — datacamp.com»](#)).

3. Desarrollo

3.1 Enunciado

Cree un programa en Python utilizando JupyterLab:

- Escribir un programa para un restaurante que tiene un menú vegetariano y un menú tradicional (no vegetariano). Debes preguntar al cliente si es vegetariano o no para, en base a su respuesta, presentarle el menú. Para el platillo que decides ofrecer, el cliente podría escoger un ingrediente adicional (de dos o tres ingredientes). Presenta también un pequeño menú de bebidas frías y calientes.
- Es posible calcular el costo del menú + ingrediente extra + bebida, de acuerdo con la selección del cliente?

NOTA: puedes mantener valores quemados como el menú de platillos y precios

3.1 Implementación

```
from abc import ABC
import sys
```

```
class Comida:
    def __init__(self, nombre, precio):
        self.nombre = nombre
        self.precio = precio

    def __str__(self):
        nombre_formateado = self.nombre.ljust(25)
        precio_formateado = f"${self.precio:.2f}"
        return f"{nombre_formateado} {precio_formateado}"

class Menu(ABC):
    def __init__(self):
        self.platos = []

    def mostrar_menu(self):
        print(self.__class__.__name__)
        for comida in self.platos:
```

```

        print(comida.__str__())

class MenuTradicional(Menu):
    def __init__(self):
        super().__init__()
        self.platos = [
            Comida("Milanesa", 15),
            Comida("Pizza", 20),
            Comida("Cordon Bleu", 10),
            Comida("Asado", 25),
        ]

class MenuVegetariano(Menu):
    def __init__(self):
        super().__init__()
        self.platos = [
            Comida("Curry de lentejas", 5),
            Comida("Risotto de champiñones", 7),
            Comida("Hamburguesa de garbanzos", 8),
            Comida("Tofu", 10),
        ]

class Cliente:
    def __init__(self, nombre, presupuesto):
        self.nombre = nombre
        self.presupuesto = presupuesto

    def __str__(self):
        return f'Nombre: {self.nombre}\nSaldo: ${self.presupuesto}'

    def comprar(self, precio):
        if self.presupuesto >= precio:
            self.presupuesto -= precio
            print(f"Gracias por su compra, {self.nombre}, su presupuesto disponible es de ${self.presupuesto}")
        else:
            print(f"Lo siento, {self.nombre}, no tienes suficiente dinero para comprar este plato")
            print("Muchas gracias por su visita!")
            sys.exit()

```

```

class Restaurante:
    def __init__(self, nombre):
        self.nombre = nombre
        self.menu_tradicional = MenuTradicional()
        self.menu_vegetariano = MenuVegetariano()
        self.ingredientes_adicionales = [
            Comida("Papas", 2.50),
            Comida("Aguacate", 2.00),
            Comida("Arroz", 2.50)
        ]
        self.bebidas = [
            Comida("Agua", 1.00),
            Comida("Gaseosa", 1.50),
            Comida("Jugo", 2.00),
            Comida("Café", 1.50),
            Comida("Té", 1.00),
            Comida("Cerveza", 2.50)
        ]
        self.clientes = []

    def obtener_platos(self, menu):
        menus = {
            "tradicional": self.menu_tradicional,
            "vegetariano": self.menu_vegetariano
        }

        menu_seleccionado = menus.get(menu)
        if not menu_seleccionado:
            return []

        return menu_seleccionado.platos

    def imprimir_bebidas(self):
        print("Bebidas:")
        for i, bebida in enumerate(self.bebidas, 1):
            print(f"{i}. {bebida}")

    def imprimir_ingredientes(self):
        print("Ingredientes adicionales:")
        for i, ingrediente in enumerate(self.ingredientes_adicionales, 1):
            print(f"{i}. {ingrediente}")

restaurante = Restaurante("Nico's Restaurant")

```

```

nombre = input("Ingrese su nombre: ")
presupuesto = float(input("Ingrese su presupuesto: "))
cliente = Cliente(nombre, presupuesto)
monto_total = 0

while True:
    # Menú principal
    print("¿Es vegetariano? (Sí: 1, No: 2, Terminar: 0)")
    op = input("Ingrese su opción: ")

    if op == "0":
        break

    menu = "vegetariano" if op == "1" else "tradicional"
    print(f"Bienvenido al menú {menu.capitalize()}")

    # Presentar platos principales numerados
    print("Platos principales:")
    platos = restaurante.obtener_platos(menu)
    for i, plato in enumerate(platos, 1):
        print(f"{i}. {plato}")

    # Elegir plato principal
    while True:
        plato_elegido = int(input("Elija el número del plato que desea (0 para terminar): "))
        if plato_elegido == 0:
            break

        cliente.comprar(platos[plato_elegido - 1].precio)
        monto_total += platos[plato_elegido - 1].precio

    # Presentar ingredientes adicionales
    restaurante.imprimir_ingredientes()

    # Elegir ingrediente adicional
    ingrediente_elegido = int(
        input("Elija el número del ingrediente adicional que desea (0 para terminar): ")
    )
    if ingrediente_elegido == 0:
        break

    cliente.comprar(restaurante.ingredientes_adicionales[ingrediente_elegido - 1].precio)
    monto_total += restaurante.ingredientes_adicionales[ingrediente_elegido - 1].precio

```

```

    # Presentar bebidas
    restaurante.imprimir_bebidas()

    # Elegir bebida
    bebida_elegida = int(input("Elija el número de la bebida que desea: "))
    if bebida_elegida == "0":
        break

    cliente.comprar(restaurante.bebidas[bebida_elegida - 1].precio)
    monto_total += restaurante.bebidas[bebida_elegida - 1].precio

    break

# Menú de opciones después de la compra
print("\n¿Qué desea hacer a continuación?")
print("1. Realizar otra compra")
print("2. Terminar el programa")
op_menu = input("Ingrese su opción: ")
if op_menu == "2":
    print(f"El monto total de su compra es de ${monto_total:.2f}")
    break

print("Gracias por visitar Nico's Restaurant. ¡Que tengas un buen día!")

```

```

Ingrese su nombre: Paula
Ingrese su presupuesto: 100
Ingrese su opción: 1
Elija el número del plato que desea (0 para terminar): 2
Elija el número del ingrediente adicional que desea (0 para terminar): 3
Elija el número de la bebida que desea: 3
Ingrese su opción: 1
Ingrese su opción: 2
Elija el número del plato que desea (0 para terminar): 2
Elija el número del ingrediente adicional que desea (0 para terminar): 1
Elija el número de la bebida que desea: 5
Ingrese su opción: 2

```

```

¿Es vegetariano? (Sí: 1, No: 2, Terminar: 0)
Bienvenido al menú Vegetariano
Platos principales:
1. Curry de lentejas          $5.00

```

2. Risotto de champiñones \$7.00
3. Hamburguesa de garbanzos \$8.00
4. Tofu \$10.00

Gracias por su compra, Paula, su presupuesto disponible es de \$93.00

Ingredientes adicionales:

1. Papas \$2.50
2. Aguacate \$2.00
3. Arroz \$2.50

Gracias por su compra, Paula, su presupuesto disponible es de \$90.50

Bebidas:

1. Agua \$1.00
2. Gaseosa \$1.50
3. Jugo \$2.00
4. Café \$1.50
5. Té \$1.00
6. Cerveza \$2.50

Gracias por su compra, Paula, su presupuesto disponible es de \$88.50

¿Qué desea hacer a continuación?

1. Realizar otra compra
2. Terminar el programa

¿Es vegetariano? (Sí: 1, No: 2, Terminar: 0)

Bienvenido al menú Tradicional

Platos principales:

1. Milanesa \$15.00
2. Pizza \$20.00
3. Cordon Bleu \$10.00
4. Asado \$25.00

Gracias por su compra, Paula, su presupuesto disponible es de \$68.50

Ingredientes adicionales:

1. Papas \$2.50
2. Aguacate \$2.00
3. Arroz \$2.50

Gracias por su compra, Paula, su presupuesto disponible es de \$66.00

Bebidas:

1. Agua \$1.00
2. Gaseosa \$1.50
3. Jugo \$2.00
4. Café \$1.50
5. Té \$1.00
6. Cerveza \$2.50

Gracias por su compra, Paula, su presupuesto disponible es de \$65.00

¿Qué desea hacer a continuación?

1. Realizar otra compra
2. Terminar el programa

El monto total de su compra es de \$35.00

Gracias por visitar Nico's Restaurant. ¡Que tengas un buen día!

4. Conclusiones

- La Programación Orientada a Objetos facilita la reutilización y comprensión del código, alineándose con las buenas prácticas de programación y posibilitando el desarrollo de aplicaciones más complejas.
- El sistema implementado estructura los elementos del restaurante, como menús, clientes y platos, mediante una jerarquía de clases. Se emplean métodos especiales como **init** y **str** para inicializar objetos y representarlos de manera legible, respectivamente. Además, se utiliza la herencia, las clases MenuTradicional y MenuVegetariano heredan de la clase abstracta Menu, y se implementa el polimorfismo con métodos mostrar_menu distintos para cada tipo de menú.

5. Declaración del Uso de IA

En este informe se utilizó IA para mejorar la coherencia, cohesión y sintaxis de las conclusiones.

6. Referencias Bibliográficas

«Object-Oriented Programming in Python (OOP): Tutorial — datacamp.com». <https://www.datacamp.com/tutorial/python-oop-tutorial>.