

Desafío - Métodos y atributos

En este desafío validaremos nuestros conocimientos en clases, métodos, atributos estáticos y no estáticos.

Lee todo el documento antes de comenzar el desarrollo **individual o grupal**, para asegurarte de tener el máximo de puntaje y enfocar bien los esfuerzos.

Descripción

Una cadena de pizzerías desea crear una aplicación para que los clientes puedan autogestionar sus pedidos. Por ahora, se te solicita crear un prototipo que resuelva el algoritmo que permita a un usuario ordenar una pizza de 3 ingredientes, y escoger el tipo de masa. Para ello, debes utilizar el lenguaje Python y las características de la Programación Orientada a Objetos.

En esta cadena, una pizza puede tener 2 tipos de ingredientes, vegetales y proteicos, y su masa puede ser tradicional o delgada. Dentro de los vegetales, las posibilidades son tomate, aceitunas y champiñones. Dentro de los proteicos, las posibilidades son pollo, vacuno o carne vegetal. Sin embargo, los ingredientes posibles pueden variar debido a stock y/o estacionalidad, por lo que se debe considerar que no siempre serán estas las alternativas posibles. Cualquier pizza ordenada debe tener 1 ingrediente proteico y 2 vegetales. Todas las pizzas tienen un precio de \$10.000 y tamaño familiar.

Para desarrollar este desafío debes haber revisado los contenidos de Clase y Objeto, así como los de Métodos y Atributos no estáticos y estáticos.

Requerimientos

1. En el archivo `pizza.py`, crear una clase que permita crear objetos de tipo Pizza. Considerar qué atributos de clase debe contener la clase, según la descripción del problema.
(1 Punto)
2. En el mismo archivo y clase del requerimiento anterior, agregar un método que permita validar un elemento dentro de una lista de casos posibles. Este método se debe poder utilizar sin necesidad de crear un objeto de la clase, y debe recibir 2 argumentos:
 - a. El elemento a validar (un texto).
 - b. Los valores posibles a considerar para el elemento ingresado (una lista de textos).

En caso de que el elemento ingresado como primer argumento se encuentre entre la lista de valores posibles ingresada como segundo argumento, el método debe retornar `True`. En caso contrario, debe retornar `False`.

(2 Puntos)

3. En el mismo archivo y clase del requerimiento anterior, agregar un método que permita realizar un pedido. Para ello, dentro de la definición de este método, debe solicitar al usuario ingresar el ingrediente proteico, luego el primer ingrediente vegetal, luego el segundo ingrediente vegetal, y finalmente el tipo de masa. Cada ingreso debe almacenarse (o añadirse, si corresponde) en un atributo de la instancia.
(2 Puntos)
4. Dentro del mismo método del requerimiento 3, una vez asignados los valores a los atributos de la instancia, debe evaluarse si se trata de un ingreso válido (considerar la información de la descripción), usando el método del requerimiento 2. Si los 3 ingredientes y el tipo de masa son válidos, deberá almacenar en un atributo el valor `True`. En caso contrario, deberá almacenar el valor `False`. Este atributo permitirá saber si una instancia específica es o no una Pizza válida.
(2 Puntos)

Tip: Si lo deseas, puedes crear un archivo `ingredientes.py` que contenga 3 variables de tipo lista, una con los valores de ingredientes cárnicos posibles, otra con los valores de ingredientes de tipo vegetales, y otra con los los valores posibles del tipo de masa, e importar este archivo en el de la clase.

5. En un archivo llamado `evaluaciones.py`, importe la clase creada en el requerimiento 1 (conteniendo los requerimientos 2, 3 y 4), y realice lo siguiente:
(3 Puntos)

- a. Usar la función `print()`, para que al ejecutar el script se muestre en pantalla los valores de los atributos de clase de la clase importada, sin crear una instancia de ella.
- b. Usar la función `print()`, para que al ejecutar el script se muestre en pantalla si el elemento "salsa de tomate" se encuentra presente en la lista `["salsa de tomate", "salsa bbq"]`. Para ello use el método creado en el requerimiento 2, sin crear una instancia de la clase importada.
- c. Crear una instancia de la clase importada, y luego llamar al método del requerimiento 3, para que al ejecutar el script se solicite ingredientes y tipo de masa al usuario.
- d. Usar la función `print()`, para que al ejecutar el script, luego de que el usuario haya ingresado los ingredientes y tipo de masa, se muestre en pantalla los ingredientes vegetales, el ingrediente proteico y el tipo de masa de la instancia creada en el paso anterior, y si esa instancia es una pizza válida o no.
- e. Usar la función `print()`, para mostrar en pantalla si la clase Pizza es una pizza válida o no, haciendo uso del atributo creado en el requerimiento 4, sin crear una instancia de la clase. En este punto, la ejecución del script debe mostrar un error (todos los pasos anteriores se deben haber ejecutado correctamente).



¡Mucho éxito!