

Módulo 1 – Proyecto

La heladería

Objetivo general

El objetivo de este Proyecto es practicar todos los conceptos trabajados en este módulo 1. Cada punto le permitirá poner en práctica los conceptos vistos en clase. Recuerde poner en práctica las buenas prácticas de programación discutidas en el curso. Comente y ponga nombres significativos a las funciones y variables para que su código sea claro. **Este proyecto debe ser resuelto de forma individual.**

Objetivos específicos

1. Practicar el manejo de diferentes tipos de datos y de variables.
2. Practicar la construcción de funciones propias para resolver requerimientos.
3. Practicar los conceptos base de la Programación Orientada a Objetos
4. Implementar clases con herencia y clases abstractas.
5. Practicar buenas prácticas de programación

Punto 0 | Preparación

Cree una carpeta llamada **Proyecto** en la cual podrá guardar todos los archivos y diagramas que desarrolle. Recuerde que todos estos serán importantes al momento de la entrega de su proyecto.

Enunciado

Una heladería muy popular está buscando digitalizar su menú y parte de su manejo de inventario. Para esto ha decidido contar con nuestra ayuda. La heladería tiene dos conceptos principales: Productos e Ingredientes. A partir de los ingredientes que pueden ser bases o complementos, es posible armar productos, que pueden ser Copas o Malteadas.

Cada Ingrediente tiene un precio, número de calorías por porción, un nombre, un contador para su inventario y un indicador de si es o no vegetariano. Para cada ingrediente se debe saber si es sano o no, según las regulaciones del departamento de salud. También debe ser posible reabastecerlos cuando sea necesario. Como ya se mencionó en el párrafo anterior, los ingredientes pueden ser bases o complementos. Las bases son las que sirven como ingrediente principal de los productos, así que adicional a lo descrito, tendrán como característica adicional su sabor. Por otra parte, los complementos sufren en algunos casos de baja rotación de inventario, así que debe ser posible bajar su inventario a 0 con una sola instrucción.

Por otra parte, tenemos los productos. Cada producto está compuesto por 3 ingredientes, realmente no tiene relevancia si son bases o complementos. Por regulación del departamento de salud es necesario poder calcular las calorías del producto y tener esta información a la mano. Para la heladería es importante saber el costo de un producto a partir de los ingredientes y la rentabilidad del mismo. Como

ya lo mencionamos hay 2 tipos de productos, copas y malteadas. Si bien ambos productos tendrán su nombre y su precio al público, las copas tendrán un tipo de vaso y las malteadas un volumen en onzas.

Finalmente, la heladería. Por su nivel de operaciones, la heladería tiene solo 4 productos a la vez en cualquier debido momento. Sin embargo, la cantidad de ingredientes en los libros de inventario puede ser ilimitada. Al jefe le interesan realmente solo dos tareas, en primer lugar, saber cuál es el producto que más rentabilidad le genera a la heladería, y una función que le permita vender un producto, revisando previamente si se tienen los ingredientes para armarlo. A parte de sus relaciones y funciones, la heladería tiene un contador que permite llevar la cuenta de las ventas del día.

Parte 1 | Funciones

La primera parte de este proyecto consiste en construir funciones que nos serán útiles para el funcionamiento de nuestra heladería. Cree un archivo llamado **funciones.py** y desarrolle en él las siguientes funciones.

Punto 1 | ¿Esto es Sano?

Construya una función para determinar si un ingrediente es sano, un ingrediente es sano si tiene estrictamente menos de 100 calorías o si es vegetariano. La función debe recibir un número de calorías (int) y un booleano que indique si el ingrediente es o no vegetariano, y retornar un booleano que indique si el ingrediente es sano según las reglas descritas. True en caso de ser sano y False de lo contrario.

Punto 2 | Las Calorías

Construya una función que permita hacer el conteo de calorías de un producto. La función debe recibir una lista de enteros que represente el número de calorías de cada ingrediente y retornar la suma de estos números de calorías multiplicada por 0.95. Este número debe redondearse a dos cifras decimales.

Punto 3 | Costos

Construya una función que permita calcular el costo de producir un producto en particular. Dicha función debe recibir 3 diccionarios, con las llaves “nombre” y “precio”, y sus respectivos valores, que representan la información de los ingredientes. Para calcular el costo, simplemente basta sumar el precio de cada uno de los 3 diccionarios.

Si quiere probar la función, use estos diccionarios:

```
{“nombre”: “Helado de Fresa”, “precio”: 1200}
```

```
{“nombre”: “Chispas de chocolate”, “precio”: 500}
```

```
{“nombre”: “Mani Japonés”, “precio”: 900}
```

Punto 4 | Rentabilidad

Construya una función que calcule la rentabilidad de un producto. La fórmula para saber la rentabilidad de un producto es simplemente la diferencia entre el precio de venta del producto, y el costo de sus ingredientes. La función debe recibir el precio del producto y 3 diccionarios, con las llaves “nombre” y “precio”, y sus respectivos valores, que representan la información de los ingredientes.

Si quiere probar la función, use este precio y estos diccionarios:

7500

{“nombre”: “Helado de Fresa”, “precio”: 1200}

{“nombre”: “Chispas de chocolate”, “precio”: 500}

{“nombre”: “Mani Japonés”, “precio”: 900}

Punto 5 | El mejor producto

Finalmente, construya una función que encuentre el producto más rentable. El producto más rentable es aquel cuya rentabilidad es mayor que las demás. La función debe recibir 4 diccionarios con las llaves “nombre” y “rentabilidad”, y sus respectivos valores, que representan 4 diferentes productos. Se debe retornar el nombre del producto más rentable.

Si quiere probar la función, use estos diccionarios:

{“nombre”: “Samurai de fresas”, “rentabilidad”: 4900}

{“nombre”: “Samurai de mandarinas”, “rentabilidad”: 2500}

{“nombre”: “Malteda chocoespacial”, “rentabilidad”: 11000}

{“nombre”: “Cupihelado”, “rentabilidad”: 3200}

Parte 2 | Objetos

La segunda parte de este proyecto consiste en implementar el sistema de la Heladería en un sistema de Objetos que representen el problema descrito en el enunciado. Las funciones definidas en la parte 1 le serán de mucha utilidad para construir este mundo, pero deberá modificarlas ligeramente para que sean coherentes con el sistema propuesto.

Punto 6 | Construir el UML

A partir del enunciado construya un nuevo UML que refleje la construcción del mundo propuesta en el enunciado. Considere las relaciones de herencia y de asociación entre las diferentes clases. Para hacer un UML puede usar herramientas específicas como [Lucidchart](#) o [draw.io \(diagrams.net\)](#), o simplemente hacerlas en Powerpoint o Paint.

Punto 7 | Ingrediente

Construya la clase abstracta **Ingrediente** según la especificación del enunciado. En este caso la clase debe tener los atributos de precio, calorías, nombre, inventario y es_vegetariano. También debe tener la función de **es_sano** (Punto 1) y la función abstracta de **abastecer**. No olvide colocar todos los getters y los setters de los atributos.

Punto 8 | Base y Complemento

Construya las clases **Base** y **Complemento** que heredan de la clase abstracta **Ingrediente**. Recuerde que en este caso la base tendrá el atributo extra **sabor**, mientras que los complementos tendrán la función extra **renovar_inventario**. Para las bases la función **abastecer** sumará 5 en el inventario, mientras que para los complementos se aumentará en 10. No olvide colocar el getter y el setter para el atributo **sabor**.

Punto 9 | IProducto

Construya la interfaz **Producto** según la especificación del enunciado. En este caso la interfaz solamente tendrá las funciones abstractas **calcular_costo**, **calcular_rentabilidad** y **calcular_calorias**.

Punto 10 | Copa y Malteada

Construya las clases **Copa** y **Malteada** que implementan la interfaz **IProducto**. Tanto **Copa** como **Malteada** tienen los atributos **nombre** y **precio_publico**. Tenga en cuenta las siguientes diferencias entre la implementación de ambas clases:

- Calcular el costo de una copa se hace simplemente sumando los ingredientes (Punto 3). Para las malteadas se sumarán 500 pesos por el uso de vasos plásticos.
- Calcular las calorías de una copa, es simplemente seguir la fórmula de sumar los ingredientes y multiplicar por 0.95 (Punto 2). Para las malteadas, se debe sumar y agregar 200 calorías, pues todas utilizan crema chantilly. No es necesario multiplicar por 0.95
- Las malteadas tienen un atributo de **volumen**, mientras que las copas tienen uno de **tipo_vaso**.

Adicional a esto, calcular la rentabilidad (Punto 4) se implementa de la misma manera en ambas clases. No olvide agregar los getters y setters en ambas clases.

Punto 11 | Heladería

Construya la clase **Heladería** con todos los atributos, relaciones y funciones descritas en el enunciado. No olvide colocar la función que le permite determinar el producto más rentable (Punto 5).

Punto 12 | Vender

Finalmente construya la función **vender** en la clase **heladería**, esta función debe seguir los siguientes pasos:

- Recibe el nombre de un producto.
- Verifica que dicho nombre de un producto sea igual a uno de los cuatro productos disponibles.

- Si lo anterior es correcto, verifica que haya existencias de los ingredientes para hacer dicho producto. Se necesita 1 de cada complemento y 0.2 de cada base para vender.
- Si sí hay existencia, resta de cada uno de los productos lo necesario para armar el producto.
- Si se vendió el producto, suma a las ventas del día el precio del producto.
- Retorna True si fue posible venderlo, False de lo contrario.

Entrega

Cree un archivo comprimido .zip con los archivos correspondientes a las funciones y diagramas que escribió anteriormente y sus respectivas pruebas. Ponga como nombre al archivo **PROY-1-login.zip**, donde login es su login de uniandes (Por ejemplo, si su login es p.perez123, entregue el archivo PROY-1-p.perez123.zip). Entregue el archivo comprimido a través de Bloque Neón en la actividad designada como “Proyecto módulo 1”.