Módulo 2 – Proyecto
La Heladería Web

Objetivo general

El objetivo de este Proyecto es practicar todos los conceptos trabajados en este módulo 2. 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 las bases del framework Flask aplicado en Python.
2. Practicar la construcción y manejo de bases de datos utilizando el framework SQLAlchemy.
3. Practicar el uso de pruebas y el manejo de errores enfocado en la Programación Orientada a Objetos.
4. Utilizar repositorios web en github para el manejo de software.
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. Retome su trabajo realizado en el proyecto 1, puede serle muy útil.

Enunciado

La primera digitalización de la heladería ha sido un completo éxito, y se nos ha pedido mejorar un poco el sistema. La heladería ahora va a utilizar una plataforma web en flask para hacer las consultas y realizar las ventas. Conserve la estructura propuesta en el nivel 1, se harán unos cambios, pero la esencia del proyecto seguirá siendo la misma.

Parte 1 | Web

La primera parte de este proyecto consiste en transportar nuestra heladería a un entorno Web.

Punto 1 | Una pequeña aplicación Web

Prepare una aplicación Web en Flask para desplegar la información de la heladería en un ambiente web. Cree un nuevo entorno virtual y configúrelo tal como se trabajó en clase y en los talleres.

Punto 2 | Controlador

Construya un controlador que permita la interacción con el modelo del mundo. Dicho controlador debe poder acceder a **todos los métodos** de la heladería. Respete la arquitectura de MVC que utiliza Flask.

Punto 2 | Mostrar el menú

Edite el index de la aplicación en Flask de modo que se muestre en un formato relativamente bonito los 4 productos disponibles en el menú de la heladería. Puede guiarse con el trabajo realizado en el taller 2 de este módulo. El App de Flask solamente debe incluir esta información por el momento

Parte 2 | Base de datos

La segunda parte de este proyecto consiste en implementar el sistema de base de datos de nuestra heladería. En este caso solo vamos a contar con dos tablas diferentes. INGREDIENTES y PRODUCTOS.

Punto 4 | Construir el UML

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. No olvide incluir la estructura de la base de datos. Para hacer un UML puede usar herramientas específicas como Lucidchart o draw.io (diagrams.net), o simplemente hacerlas en Powerpoint o Paint.

Punto 5 | Base de datos

Construya con SQLAlchemy las dos tablas a utilizar en nuestro proyecto. Recupere los atributos y conviértalos en las diferentes columnas de la base de datos. Sea exhaustivo con las columnas, de modo que ambas tablas sirvan para sus 2 tipos de implementaciones. Para los ingredientes de cada producto, puede utilizar 3 llaves foráneas a la tabla de ingredientes. Edite la clase Heladería para que esta no solo tenga una lista de ingredientes sino también una lista de productos.

Punto 6 | Operaciones de Bases de datos

Construya con SQLAlchemy operaciones de consulta que permitan cargar la información en el modelo de mundo a partir de la base de datos. **Es importante que no traduzca todas las funciones a SQLAlchemy sino solo el proceso de carga de los datos al modelo.** Todas las demás consultas deben conservarse en el formato en que ya se encuentran y hacerse a partir de ingredientes y productos ya cargados en el modelo.

Parte 3 | Repositorio y Pruebas

La última parte del proyecto consiste en montar un repositorio en Github con el trabajo del proyecto, y desarrollar un pequeño módulo de pruebas unitarias. También se modificará el método de vender para aplicar los conceptos de manejo de errores.

Punto 7 | Crear el repositorio

Cree un repositorio en Github y cargue en este su trabajo de este proyecto. El proyecto debe tener como nombre **PROYECTO2-LOGIN** donde login es su login uniandes. Remítase al taller 3 o a las diapositivas, si no está seguro de cómo crear el proyecto. **ESTE REPOSITORIO DEBE SER CREADO O QUEDAR CONFIGURADO COMO PRIVADO.**

Punto 8 | Ventas fallidas

Modifique el método de vender, de modo que, si se intenta vender un producto y no se cumple con la regla de venta, se lance un ValueError, cuyo mensaje debe decir el nombre del ingrediente que no estaba disponible para vender el producto. El retorno ante una venta exitosa será la frase "¡Vendido!".

A continuación, desde el controlador, implemente una estructura try-catch que permita recibir este error y retornar el mensaje de "¡Vendido!" si se ha vendido exitosamente. Y en caso de que la función genere el error se retorne un mensaje con el formato: "¡Oh no! Nos hemos quedado sin XXXX 😳". Donde XXXXX es el nombre del ingrediente faltante en la receta.

Punto 9 | Pruebas Unitarias

Construya un módulo de pruebas unitarias que permita probar los diferentes métodos de la heladería. Recuerde que estos métodos son:

- Probar si un Ingrediente es sano.
- Abastecer un Ingrediente.
- Renovar Inventario para los complementos.
- Calcular las calorías tanto en copas como en malteadas.
- Calcular el costo de producción.
- Calcular la rentabilidad de un producto.
- Encontrar el producto más rentable.
- Vender un producto.

Tenga en cuenta que vender un producto puede generar excepciones y es importante que las pruebas contemplen esta situación y prueben que el método falle cuando debe fallar.

Entrega

Entregue el vínculo a su repositorio con el trabajo desarrollado. Ponga como nombre al repositorio **PROYECTO2-login** donde login es su login de uniandes (Por ejemplo, si su login es p.perez123, el repositorio se llamará PROYECTO2-pperez123). **El repositorio DEBE SER PRIVADO.** Entregue el vínculo al repositorio a través de Bloque Neón en la actividad designada como "Proyecto módulo 2".