

Módulo 3 – Proyecto

La HeladerAPI

Objetivo general

El objetivo de este Proyecto es practicar todos los conceptos trabajados en este módulo 3. 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 la creación de repositorios.
2. Implementar un sistema de autenticación en una aplicación Flask
3. Implementar autorización para diferentes vistas en una aplicación Flask
4. Practicar la construcción y consultas de APIS
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 2, puede serle muy útil.

Enunciado

La primera digitalización de la heladería ha sido un completo éxito, y se nos ha pedido mejorar la presencia web de la página. En esta ocasión debemos implementar sistemas de autorización y autenticación. Y construir un API REST completo para que la aplicación sea robusta y utilizable por terceros en un futuro.

Parte 1 | Autenticación

El primer asunto a resolver es la autenticación. Es necesario crear un sistema que permita autenticarse y asignar roles. En primer lugar, vamos a implementar el sistema de autenticación, los permisos los asignaremos después de construir la versión final de la aplicación.

Punto 1 | Crear 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 del módulo 2 o a las diapositivas, si no está seguro de cómo crear el proyecto. **ESTE REPOSITORIO DEBE SER CREADO O QUEDAR CONFIGURADO COMO PRIVADO.**

Punto 2 | Tabla Usuarios

Construya una clase usuario con los atributos id, username y password. Adicionalmente construya una tabla con SQLAlchemy que tenga estas 3 columnas precisamente. Implemente también un método que permita consultar en la tabla si un usuario y contraseña coinciden, de modo que se pueda realizar la autenticación con base al contenido de la tabla.

Punto 3 | Autenticación

Implemente un módulo de autenticación según lo descrito en las diapositivas. Cuando se logre iniciar sesión, muestre un mensaje de saludo como el que aparece en las diapositivas, puede agregarle más elementos a esta vista si lo desea.

Parte 2 | API REST

La segunda parte de este proyecto implica construir el API completo de la heladería, también es necesario crear pruebas unitarias con Postman y modelar un nuevo UML que muestre la nueva estructura del proyecto.

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 | Construir el API Rest

Construya el API completo de la heladería. El API debe permitir las siguientes operaciones:

- Consultar todos los productos
- Consultar un producto según su ID
- Consultar un producto según su nombre
- Consultar las calorías de un producto según su ID
- Consultar la rentabilidad de un producto según su ID
- Consultar el costo de producción de un producto según su ID
- Vender un producto según su ID
- Consultar todos los Ingredientes
- Consultar un ingrediente según su ID
- Consultar un ingrediente según su nombre
- Consultar si un ingrediente es sano según su ID
- Reabastecer un producto según su ID
- Renovar el inventario de un producto según su ID

Tiene libertad en escoger los nombres y arquitectura propuestos. **Sin embargo, es obligatorio DOCUMENTAR COMPLETAMENTE cada uno de los endpoints que usted genere.**

Haga pruebas en Postman que demuestren que el API funciona. Pruebe al menos 2 veces cada uno de los endpoints. Tome capturas de pantalla y guarde el documento en el repositorio.

Parte 3 | Autorización

La última parte del proyecto implica crear el sistema de autorización para la aplicación. Ciertas partes del API no deberían accederse por todo el público, así que es necesario restringir el acceso a las mismas. Haga las pruebas de Postman ANTES de implementar el módulo de autorización.

Punto 5 | 401

Modifique la tabla (Revise la documentación de SQLAlchemy) de los usuarios para agregar 2 columnas: es_admin, es_empleado. En consecuencia, actualice la clase usuario y los métodos que hagan uso de esta. Construya una vista que muestre un mensaje de error “No autorizado”, esta pantalla se mostrará es en caso de que alguien intente acceder a algún recurso prohibido.

Punto 6 | Restringir el paso

Implemente un sistema de autorización y roles similar al visto en clase. Las reglas de acceso son las siguientes:

- El administrador puede acceder a todas las funciones.
- Los empleados pueden acceder a todas las funciones excepto las relacionadas con rentabilidad.
- Los clientes pueden solamente a la función vender, a la lista de productos y a sus calorías.
- Cualquiera (no tiene que estar autenticado) puede ver la lista de productos.

Entrega

Entregue el vínculo a su repositorio con el trabajo desarrollado. Ponga como nombre al repositorio **PROYECTO3-login** donde login es su login de uniandes (Por ejemplo, si su login es p.perez123, el repositorio se llamará PROYECTO3-pperez123). **El repositorio DEBE SER PRIVADO.** Entregue el vínculo al repositorio a través de Bloque Neón en la actividad designada como “Proyecto módulo 3”.