Proyecto

Sistema Python con Implementación de herencia

Programación orientada a objetos

Docente: José Gabriel Rodríguez Rivas

Alumno: José Eduardo Cabrera Pegueros

Explicación del uso de la herencia y ventajas en mi sistema

En esta versión del sistema para la ferretería, apliqué el concepto de **herencia** de la Programación Orientada a Objetos. Con este principio pude reutilizar atributos y métodos sin tener que repetir el mismo código varias veces, haciendo que el programa quedara más organizado y fácil de mantener.

Para lograrlo, primero creé una **clase base llamada Persona**, donde coloqué los datos que comparten tanto los clientes como los empleados: nombre, correo, dirección y teléfono. Después hice que las clases **Cliente y Empleado** heredaran de esta clase base. Gracias a eso, ambas clases pueden usar directamente los atributos y métodos de Persona sin tener que volver a escribirlos.

En el caso del **cliente**, le agregué un atributo propio llamado rfc, mientras que al **empleado** le añadí los suyos: id_empleado, departamento, usuario y contraseña. De esa forma, cada clase conserva lo que tienen en común, pero también lo que las diferencia.

Además, en esta nueva versión incluí una función de **inicio de sesión**. Cuando se ejecuta el programa, primero se pide un usuario y una contraseña. Si los datos son correctos, se puede acceder al menú principal. Yo creé un **empleado administrador por defecto**, con el usuario admin y la contraseña 1234, que sirve para poder entrar por primera vez al sistema. Después de eso, agregué la opción para **registrar nuevos empleados** que también pueden iniciar sesión.

Otra cosa que añadí fue la opción de **cerrar sesión e iniciar nuevamente**, para que se pueda cambiar de usuario sin cerrar todo el programa. Esto hace que el sistema sea más realista, como si varias personas pudieran usarlo en diferentes turnos.

Comparando esta versión con la anterior, puedo decir que el código quedó más estructurado y profesional. En la versión anterior, los datos personales estaban separados dentro de cada clase, lo que hacía el programa más largo y difícil de modificar. Con la herencia, todo lo común está concentrado en la clase Persona, y cualquier cambio se refleja automáticamente en las clases hijas.

En general, esta versión del sistema me permitió aplicar de forma práctica el concepto de herencia y notar sus ventajas. Mi código ahora es más claro, reutilizable y escalable, lo que significa que en el futuro podría seguir mejorándolo sin tener que volver a escribirlo desde cero.

Programa en ejecución