

Sistema gestión de contactos en Python

PROGRAMACIÓN ORIENTADA A OBJETOS Y BUENAS PRÁCTICAS

ALUMN@: FRANCISCA ROJAS GONZÁLEZ

DOCENTE: INGRID SOLIS

CURSO: CIENCIA DE DATOS - MÓDULO 2: PROGRAMACIÓN EN PYTHON

¿Qué es este proyecto?

- Desarrollo de un sistema de gestión de contactos en Python para organización y almacenaje de información de clientes.
- Aplicación permite registrar, editar, eliminar y buscar contactos de manera eficiente.
- Diseñado para trabajar desde la consola con un menú interactivo, de forma fácil e intuitiva.

Aspectos técnicos y Arquitectura

- Lenguaje: Python
- Programación Orientada a Objetos (POO).
 - Módulo Clases:
 - Clase Contacto:
 - Encapsula los datos del contacto.
 - Incluye validaciones mediante getters y setters.
 - Clase AgendaContactos:
 - Gestiona el diccionario de contactos.
 - Controla el registro, edición, eliminación y búsqueda.
 - Módulo funcional:
 - Implementa el menú interactivo.
 - Maneja la interacción con el usuario.
 - Módulo pruebas unitarias

Desafíos enfrentados

- Aprender y entender bien la OOP para aplicarla correctamente, y aprovechar sus funcionalidades.
- Aprender y entender bien unittest para integrar correctamente las pruebas unitarias.
- Manejo correcto de validaciones de forma eficiente y sin alargar el código.
- Ajustes en la edición de datos y búsqueda (especialmente el teléfono).
- Ejecutar el programa de forma correcta para validar el uso mediante interacción con el usuario en consola.

Soluciones implementadas

- Estudio y ayuda con la IA para comprender y aplicar mejoras.
- Uso de @property para validar datos automáticamente dentro de las clases en vez de hacer funciones externas con for.
- Reestructuración del código para mejorar encapsulación, utilizando las propiedades de las clases, para un código más limpio y compacto.
- Preferir el uso de diccionarios para hacer más eficiente las búsquedas, usando como llave el teléfono.
- Corrección manual de errores detectados en pruebas de consola.

Pruebas realizadas

- Pruebas unitarias con unittest:
 - Registro de contactos.
 - Búsqueda.
 - Edición.
 - Validación de errores.
- Pruebas manuales:
 - Uso del menú desde consola.
 - Ingreso de datos válidos e inválidos. Y todas las funcionalidades del menú interactivo.
- Resultados:
 - Todas las pruebas fueron exitosas.
 - El sistema respondió correctamente ante errores de entrada.

Aprendizajes

- Importancia de la Programación Orientada a Objetos, y su correcta aplicación.
- Ventajas de usar diccionarios para eficiencia.
- Uso correcto de unittest.
- Encapsulación y validación de datos en clases.
- Probar el código constantemente para corregir y tener un buen código final.
- El apoyo con la IA para generar soluciones eficientes.
- La validación del código a detalle de forma manual es muy importante.
- Generar código limpios y comentados para su entendimiento.