

DOCUMENTACIÓN TÉCNICA

Sistema de Gestión de Contactos

Introducción

Este documento corresponde a la documentación técnica asociada al proyecto Sistema de Gestión de Contactos, el cual fue desarrollado en Python a través de Programación Orientada a Objetos (POO). Este sistema utiliza clases para representar contactos y gestionar la información, aplicando principios de encapsulación.

Objetivo del Proyecto

El proyecto Sistema de Gestión de Contactos tiene como objetivo almacenar y gestionar información personal de forma segura y organizada (nombre, teléfono, email y dirección). La aplicación permite el registro, la edición y la eliminación de contactos, además de la búsqueda de contactos, tanto por nombre como por número de teléfono, y la visualización del detalle de todos los contactos almacenados en el sistema.

Estructura del Proyecto

El proyecto Sistema de Gestión de Contactos presenta una estructura modular organizada que facilita la comprensión, al tiempo que resulta mantenible y escalable.

Dentro de proyecto_contactos se encuentran los archivos que contienen el código para la ejecución del programa, así como las pruebas unitarias de validación. Por su parte, en la carpeta Documentos se encuentra la documentación técnica relacionada al proyecto, el informe de pruebas, el prototipo funcional y una presentación final.

```
 proyecto_contactos/
    ├── contacto.py
    ├── gestorcontactos.py
    ├── main.py
    ├── test_sistema_contactos.py
    ├── README.md
    └── Documentos/
        ├── Documentacion_Tecnica.pdf
        ├── Informe_de_Pruebas.pdf
        ├── Prototipo_Funcional.pdf
        └── Presentacion_Final.pdf
```

Clases y Módulos utilizados

- **contacto.py:**

Define la clase Contacto con validaciones:

- **Nombre:** no vacío.
- **Teléfono:** sólo dígitos y longitud mínima de 7 caracteres.
- **Email:** formato con @ y dominio válido.

El objeto Contacto corresponde a un contacto individual dentro del sistema y considera los siguientes atributos: nombre, teléfono, email y dirección.

Se aplica encapsulación mediante el uso de atributos privados y la incorporación de métodos get y set con validación, ayudando a organizar el comportamiento interno del objeto y permitiendo el almacenamiento y gestión de la información de forma segura y organizada.

- **re:**

Se usa para validar el formato del email con una expresión regular.

- **gestorcontactos.py:**

Define la clase GestorContactos y se implementan métodos para:

- **Agregar contactos:** evita duplicados por número de teléfono.
- **Editar contactos:** actualiza campos con validación.
- **Eliminar contactos:** seleccionando por nombre.
- **Buscar contactos:** por nombre y por número de teléfono.
- **Listar los contactos:** representación formateada.

- **main.py:**

Corresponde al menú por consola que utiliza el usuario, el que cuenta con mensajes e íconos.

- Éxito
- Advertencia
- Error

Arquitectura del Sistema

El Sistema de Gestión de Contactos está modelado a través de la Programación Orientada a Objetos. La arquitectura de este sistema considera las siguientes capas y responsabilidades:

- **Capa de Datos (contacto):**

- **Contacto:** encapsula datos y reglas de validación.
- **Responsabilidad:** garantiza que cualquier instancia sea válida.

- **Capa Lógica (gestorcontactos):**

- **GestorContactos:** articula operaciones CRUD sobre una lista en memoria.
- **Responsabilidad:** coordinar, evitar duplicados y ofrecer búsquedas.

- **Capa de Presentación (main):**

- **Arranque:** inicializa el menú y mantiene el bucle de interacción.
- **Menú interactivo:** recoge entradas del usuario, captura errores y muestra resultados con íconos.
- **Responsabilidad:** experiencia de uso clara y segura.

Flujo Principal

1. El usuario selecciona una opción en el menú interactivo.
2. main solicita datos y los pasa a gestorcontactos.
3. gestorcontactos crea/edita un contacto. Si los datos son inválidos arroja ValueError.
4. main captura la excepción y muestra un mensaje con un ícono relacionado.

Instrucciones para ejecutar la aplicación en un entorno local

Requisitos Previos:

- Python 3.10 o superior.
- Terminal (PowerShell, CMD, bash, zsh)

Instalación:

1. Clonar el repositorio desde GitHub.
2. Navegar al directorio del proyecto: cd proyecto_contactos
3. Ejecutar el sistema con el siguiente comando: python main.py
4. Interacción básica:
 - Seleccionar una opción del menú (1 – 7).
 - Ingresar los datos solicitados.
 - Observar los mensajes con íconos que indican el resultado.

Pruebas Unitarias

Se realizaron pruebas unitarias sobre el Sistema de Gestión de Contactos con el framework unittest, de manera de verificar la correcta implementación de las validaciones incluidas en la clase Contacto, así como las funcionalidades de la clase GestorContactos.

Todas las pruebas se ejecutaron correctamente. Los resultados se encuentran detallados en el documento Informe de Pruebas.

Para revisar las pruebas realizadas, dentro de la carpeta proyecto_contactos se debe ejecutar: python test_sistema_contactos.py

Conclusiones Finales

El Sistema de Gestión de Contactos desarrollado cumple con todos los requerimientos generales y técnicos especificados, implementando de manera correcta todas las funcionalidades requeridas.

Presenta una interfaz de uso sencilla y organizada, que permite el uso intuitivo por parte del usuario final.

Por último, su estructura y organización de clases, métodos y atributos, garantizan una aplicación sólida y escalable.