

INFORME DE PRUEBAS

Sistema de Gestión de Contactos

Introducción

El presente informe detalla las pruebas unitarias realizadas sobre el proyecto Sistema de Gestión de Contactos desarrollado en Python. El objetivo de estas pruebas fue verificar la correcta implementación de las validaciones incluidas en la clase Contacto, así como las funcionalidades de la clase GestorContactos, de manera de garantizar el correcto funcionamiento de este sistema.

Las pruebas se realizaron utilizando el módulo estándar unittest y se ejecutaron con el parámetro verbosity = 2, lo que permite obtener un informe detallado de cada caso de prueba.

Alcance de las pruebas

Las pruebas se enfocaron en los siguientes aspectos:

- Validaciones de datos en la clase Contacto (nombre, teléfono, email).
- Métodos setters y su comportamiento frente a datos inválidos.
- Operaciones CRUD en la clase GestorContactos (agregar, editar, eliminar, buscar y mostrar).
- Formato de salida del método `__str__`.

Entorno de pruebas

- **Lenguaje:** Python 3.10+
- **Framework:** unittest
- **Sistema operativo:** Windows 10/11

- **Estructura del proyecto:**
 - proyecto_contactos/contacto.py
 - proyecto_contactos/gestorcontactos.py
 - proyecto_contactos/test_sistema_contactos.py

Pruebas realizadas

1. Clase Contacto

- ✓ **Creación válida:** Contacto con datos correctos se crea exitosamente.
- ✗ **Nombre vacío:** Se lanza ValueError.
- ✗ **Teléfono inválido:** Se rechazan teléfonos con caracteres no numéricos o longitud < 7.
- ✗ **Email inválido:** Se rechazan emails sin @, sin dominio o sin TLD.
- ✓ **Setters válidos:** Actualizan correctamente los atributos.
- ✗ **Setters inválidos:** Lanzan ValueError cuando se asignan valores inválidos.
- ✓ **Método __str__:** Representación contiene los campos separados por | y con formato alineado.

2. Clase GestorContactos

- ✓ **Agregar contacto válido:** Se agrega correctamente.
- ! **Agregar duplicado:** No se permite agregar un contacto con teléfono ya existente.
- ✓ **Editar contacto existente:** Se modifican nombre, teléfono, email y dirección.
- ✗ **Editar contacto inexistente:** Retorna False.
- ✗ **Editar contacto con datos inválidos:** Arroja ValueError.
- ✓ **Eliminar contacto existente:** Se elimina correctamente.

- ✗ **Eliminar contacto inexistente:** Retorna False.
- ✓ **Buscar por nombre:** Insensible a mayúsculas/minúsculas.
- ✓ **Buscar por teléfono:** Retorna el contacto correcto.
- ✓ **Mostrar contactos:** Devuelve lista formateada.

Resultados

- **Total de pruebas ejecutadas:** 25
- **Pruebas exitosas:** 25
- **Pruebas fallidas:** 0

Todas las pruebas unitarias se ejecutaron correctamente, validando que el sistema cumple con los requisitos de validación y funcionalidad.

⭐ **Estado final de las pruebas:** ✓ Todas las pruebas se ejecutaron exitosamente.

```
test_formato_str_columnas (__main__.TestContactoStr.test_formato_str_columnas) ... ok
test_creacion_valida (__main__.TestContactoValidaciones.test_creacion_valida) ... ok
test_email_sin_arroba (__main__.TestContactoValidaciones.test_email_sin_arroba) ... ok
test_email_sinDominio (__main__.TestContactoValidaciones.test_email_sinDominio) ... ok
test_email_sin_tld (__main__.TestContactoValidaciones.test_email_sin_tld) ... ok
test_nombre_vacio (__main__.TestContactoValidaciones.test_nombre_vacio) ... ok
test_set_email_invalido (__main__.TestContactoValidaciones.test_set_email_invalido) ... ok
test_set_nombre_vacio (__main__.TestContactoValidaciones.test_set_nombre_vacio) ... ok
test_set_telefono_invalido (__main__.TestContactoValidaciones.test_set_telefono_invalido) ... ok
test_setters_validos (__main__.TestContactoValidaciones.test_setters_validos) ... ok
test_telefono_corto (__main__.TestContactoValidaciones.test_telefono_corto) ... ok
test_telefono_no_digitos (__main__.TestContactoValidaciones.test_telefono_no_digitos) ... ok
test_agregar_contacto (__main__.TestGestorContactos.test_agregar_contacto) ... ok
test_agregar_contacto_duplicado_por_telefono (__main__.TestGestorContactos.test_agregar_contacto_duplicado_por_telefono) ... ok
test_buscar_por_nombre_coincide_exactamente (__main__.TestGestorContactos.test_buscar_por_nombre_coincide_exactamente) ... ok
test_buscar_por_nombre_sensible_a_minusculas (__main__.TestGestorContactos.test_buscar_por_nombre_sensible_a_minusculas) ... ok
test_buscar_por_telefono (__main__.TestGestorContactos.test_buscar_por_telefono) ... ok
test_editar_contacto_con_datos_invalidos_lanza_error (__main__.TestGestorContactos.test_editar_contacto_con_datos_invalidos_lanza_error) ... ok
test_editar_contacto_email_valido (__main__.TestGestorContactos.test_editar_contacto_email_valido) ... ok
test_editar_contacto_inexistente (__main__.TestGestorContactos.test_editar_contacto_inexistente) ... ok
test_editar_contacto_nombre (__main__.TestGestorContactos.test_editar_contacto_nombre) ... ok
test_editar_contacto_telefono_valido (__main__.TestGestorContactos.test_editar_contacto_telefono_valido) ... ok
test_eliminar_contacto (__main__.TestGestorContactos.test_eliminar_contacto) ... ok
test_eliminar_contacto_inexistente (__main__.TestGestorContactos.test_eliminar_contacto_inexistente) ... ok
test_mostrar_contactos_formateados (__main__.TestGestorContactos.test_mostrar_contactos_formateados) ... ok

-----
Ran 25 tests in 0.004s
OK
```

Conclusiones Finales

El Sistema de Gestión de Contactos implementa de manera correcta las validaciones de datos y las operaciones de gestión. Las pruebas unitarias confirman que:

- No se permiten nombres vacíos.
- Los teléfonos deben ser numéricos y con longitud mínima de 7 caracteres.
- Los emails deben tener formato válido.
- Las operaciones CRUD funcionan según lo esperado.