

Agenda de Contactos en Python (POO)

Proyecto Módulo 2

Yoseph Barrera

Especialización en Ciencia de Datos

January 4, 2026

Objetivo y alcance

- **Objetivo:** construir una agenda de contactos en Python aplicando POO y estructuras de datos.
- **Alcance (CRUD):** crear, listar, buscar, editar y eliminar contactos.
- **Campos:** nombre, teléfono, correo y dirección.
- **Entregables:** código en GitHub, documentación, demo, informe de pruebas y presentación.

Clases principales

- **Contacto:** representa un contacto individual.
 - Encapsulación con atributos internos y @property para controlar asignación/validación.
- **AgendaContactos:** administra el conjunto de contactos y expone operaciones CRUD + búsquedas.

Ventaja del diseño: separa datos (Contacto) de la lógica de gestión (Agenda), facilitando mantenimiento y testeo.

Estructuras usadas

- **Lista** para almacenar y listar contactos.
- **Diccionarios** para acceso/búsqueda rápida y control de consistencia (por ID y/o teléfono).

Validaciones básicas

- Teléfono: normalización (quita espacios/guiones), permite +, y largo razonable.
- Correo: validación mínima (contiene @ y texto a ambos lados).
- Nombre: largo mínimo; Dirección: no vacía.

Demo: ejecución en Colab (interfaz por consola)

```
[2] ejecutar_menu()
...
=== Agenda de Contactos ===
1) Agregar
2) Listar
3) Buscar por nombre
4) Buscar por teléfono
5) Editar
6) Eliminar
0) Salir
Opción: 
```

(1) Menú principal

```
6) Eliminar
0) Salir
Opción: 1
Nombre: yoseph
Teléfono: 979452321
Correo: yoseph@uc.cl
Dirección: valdenegro 7762
✅ Agregado con ID 1

=== Agenda de Contactos ===
1) Agregar
2) Listar
3) Buscar por nombre
4) Buscar por teléfono
5) Editar
6) Eliminar
0) Salir
Opción: 
```

(2) Registro exitoso

```
=== Agenda de Contactos ===
1) Agregar
2) Listar
3) Buscar por nombre
4) Buscar por teléfono
5) Editar
6) Eliminar
0) Salir
Opción: 0
Chao 🐼 ...
```

(3) Salida del sistema

Flujo mostrado: agregar → confirmar ID → seguir operando / salir.

Pruebas unitarias: enfoque y resultados

Herramienta: `unittest` (librería estándar de Python)

- Elegida por reproducibilidad y cero dependencias externas.
- Permite validar automáticamente lo crítico del sistema después de cambios.

Casos testeados (mínimos esenciales)

- Registro de contacto (ID y almacenamiento)
- Búsqueda por nombre (substring)
- Búsqueda por teléfono (recuperación correcta)

Resultado: ejecución con estado OK (evidencia en `docs/evidencia_tests.txt`).

Desafíos, soluciones y próximos pasos

Desafíos abordados

- Consistencia del dato (teléfono/correo) → validaciones y normalización.
- Evitar fallos por input del usuario → manejo de errores en menú.
- Confiabilidad del sistema → pruebas unitarias con `unittest`.

Próximos pasos (mejoras)

- Ampliar cobertura de tests: duplicados, edición/eliminación, casos borde.
- Persistencia: guardar/cargar contactos (CSV/JSON) si se requiere.
- Mejoras UX: mensajes más guiados y validaciones más detalladas.

Repositorio: (<https://github.com/Diosephe/agenda-contactos-poo>)