

Testing MediCareDesk



Presentado por:

Yamid Alfonso Gonzalez Torres

Jenny Catherine Herrera Garzon

Edwin Andres Marin Vanegas

Diego Steven Pinzon Yossa

Profesor:

Oscar Eduardo Alvarez Rodriguez

Universidad Nacional de Colombia

Facultad de Ingeniería

Ingeniería de software

2025



I. Introducción

Con el fin de asegurar la calidad y correcto funcionamiento de los módulos principales de MediCareDesk, cada integrante del equipo implementó un mínimo de tres pruebas unitarias. Estas pruebas fueron desarrolladas utilizando la herramienta unittest, estándar en el lenguaje Python, y validan funcionalidades esenciales del sistema, evitando pruebas triviales.

II. Herramienta utilizada

Se utilizó el módulo **unittest** de Python, ya que permite estructurar, documentar y ejecutar pruebas de manera sencilla e integrada con el entorno de desarrollo.

III. Ubicación de las pruebas

Las pruebas desarrolladas se encuentran en la carpeta:

Proyecto/MediCareDesk/tests

IV. Desarrollo por estudiante:

Test hechos: 20, entre unitarios y de integración

test

TABLA I.
DESARROLLO POR ESTUDIANTE

| Integrante | Módulo probado | Archivo | Pruebas realizadas |
|-------------------|---|----------------------------|--|
| Catherine Herrera | Autenticación | tests/test_auth.py | test_login_correcto test_login_email_invalido test_login_password_incorrecta |
| Yamid González | Tratamientos y asignación de tratamientos | tests/test_tratamientos.py | test_asignacion_medimento_valida test_asignacion_sin_tratamiento test_asignacion_sin_medimento test_hora_preferida_invalida test_fechas_inconsistentes |

| | | | |
|---------------------|--|--|---|
| Diego pinzón | Pacientes y Medicamentos, UI | test//test_medicamentos test/test_pacientes | test_mostrar_medamentos_crea_interfaz test_cargar_datos_exitoso test_cargar_datos_error_db test_agregar_medimento_valido test_eliminar_medimento_valido test_eliminar_medimento_en_uso test_refrescar_datos |
| Edwin Marin | Registro la toma Registro de historial de tomas Alertas de tomas | tests/test_tomas.py | test_registro_toma_exitoso test_registro_toma_faltan_datos test_historial_tomas test_alertas_tomas |

V. Descripción de las pruebas implementadas

- **test_login_correcto**

Debe autenticar a un cuidador con credenciales válidas.

Valida que un usuario registrado con la contraseña correcta pueda acceder al sistema.

- **test_login_email_invalido**

Debe fallar si el email no está registrado.

Verifica que el sistema rechace credenciales cuyo email no existe en la base de datos.

- **test_login_password_incorrecta**

Debe fallar si la contraseña no es correcta.

Comprueba que un usuario con email válido, pero con contraseña incorrecta, no pueda autenticarse.

- **test_asignacion_medimento_valida**

Debe asignar correctamente un medicamento a un tratamiento

Verifica que se cree la relación tratamiento-medicamento en la base de datos

- **test_asignacion_sin_tratamiento**



Debe fallar si el tratamiento no existe

Comprueba que el sistema rechace asignaciones a tratamientos inexistentes

- **test asignación sin medicamento**

Debe fallar si el medicamento no existe

Valida que el sistema rechace asignaciones de medicamentos no registrados

- **test_hora_preferida_invalida**

Debe fallar con formato de hora incorrecto

Verifica que el sistema rechace horas en formato no HH:MM

- **test_fechas_inconsistentes**

Debe fallar si fecha fin es anterior a inicio

- **test_registro_toma_exitoso**

Verifica que el registro de una toma con todos los datos correctos sea exitoso y retorne True.

- **test_registro_toma_faltan_datos:**

Comprueba que el registro falle (retorne False) si falta algún dato obligatorio.

- **test_historial_tomas:**

Valida que al consultar el historial de tomas de un cuidador se obtenga una lista de tomas (puede estar vacía o tener elementos).

- **test_alertas_tomas:**

Verifica que al consultar las alertas de tomas para un cuidador se obtenga una lista de alertas (puede estar vacía o tener elementos).

- **test_mostrar_medicamentos_crear_interfaz:** Verifica que la función principal crea todos los componentes de la UI correctamente (Frame, Canvas, Treeview, Botones)

- **test_cargar_datos_exitoso:** Comprueba que los datos se cargan correctamente desde la base de datos al Treeview

- **test_cargar_datos_error_db:** Verifica el manejo de errores cuando falla la conexión a la base de datos

- **test_agregar_medicamento_valido:** Prueba el flujo completo de agregar un nuevo medicamento con datos válidos

- **test_eliminar_medicamento_valido:** Verifica la eliminación exitosa de un medicamento sin tratamientos asociados

- **test_eliminar_medicamento_en_uso:** Comprueba que no se permite eliminar un medicamento vinculado a tratamientos

- **test_refrescar_datos:** Verifica que el botón "Refrescar" recarga los datos correctamente

Comprueba validación de rango de fechas



VI. Evidencia de ejecución de los test's:

```
C:\Users\Usuario\Documents\Proyectos Catherine\Proyecto\Proyecto\MediCareDesk>python -m tests.test_auth
...
-----
Ran 3 tests in 0.002s

OK

C:\Users\Usuario\Documents\Proyectos Catherine\Proyecto\Proyecto\MediCareDesk>
```

Figura 1. Resultado de la ejecución de pruebas unitarias en tests/test_auth.py

```
PS D:\ProyectoFinal ingesoft\Proyecto\Proyecto\MediCareDesk> & C:/Users/USUARIO/AppData
owsApps/python3.13.exe "d:/ProyectoFinal ingesoft/Proyecto/Proyecto/MediCareDesk/test

• >>> test_tratamientos.py ejecutándose desde: d:\ProyectoFinal ingesoft\Proyecto\Proye
test_tratamientos.py
>>> modelos.py cargado desde: d:\ProyectoFinal ingesoft\Proyecto\Proyecto\MediCareDes
>>> funciones en modelos: ['__builtins__', '__cached__', '__doc__', '__file__', '__lo
_package__', '__spec__', 'buscar_cuidador_por_credenciales', 'buscar_cuidador_por_ema
ar_tratamiento', 'crear_tratamiento medicamento', 'insertar_cuidador', 'obtener_conex
nto', 'obtener medicamentos', 'obtener pacientes', 'obtener_tratamiento', 'obtener_tr
'obtener tratamientos']
test_1_asignar_medicamento_valido (__main__.TestTratamientos.test_1_asignar_medicamen
Prueba asignación válida de medicamento - Versión corregida ... ok
test_2_generar_tomas_automaticas (__main__.TestTratamientos.test_2_generar_tomas_auto
Prueba generación automática de tomas ... ok

-----
Ran 2 tests in 0.012s

OK
```

Figura 2. Resultado de la ejecución de pruebas unitarias en tests/tests/test_tratamientos.py

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Versión 10.0.19045.6093]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Pc\OneDrive\Desktop\Proyecto-RamaEdwin\Proyecto\MediCareDesk>python -m unittest tests.test_tomas
Alertas de tomas para cuidador 1:
ID: 2, Mensaje: Toma programada para las 12:00
ID: 1, Mensaje: Toma tomada a las 08:00
Total de alertas encontradas: 2
.Historial de tomas para cuidador 1:
ID: 1, Fecha: 2025-07-24, Hora: 08:00, Paciente: Juan, Medicamento: Paracetamol, Estado: tomada
ID: 2, Fecha: 2025-07-24, Hora: 12:00, Paciente: Ana, Medicamento: Ibuprofeno, Estado: programada
Total de tomas encontradas: 2
.Toma registrada: cuidador_id=1, medicamento_id=2, fecha=2025-07-24, hora=08:00
Registro exitoso. Todos los datos fueron proporcionados.
.Error: Faltan datos para registrar la toma.
Datos recibidos: cuidador_id=None, medicamento_id=2, fecha=2025-07-24, hora=08:00
:
-----
Ran 4 tests in 0.002s

OK
```

Figura3. Resultado de la ejecución de pruebas unitarias en tests/test_tomas.py

```
test_editar_medimento_calls_messagebox (__main__.TestMedicamentosManager.test_editar_medimento_calls_messagebox)
Test que editar_medimento muestra mensaje informativo ... ok
test_eliminar_medimento_calls_messagebox (__main__.TestMedicamentosManager.test_eliminar_medimento_calls_messagebox)
Test que eliminar_medimento muestra mensaje informativo ... ok
test_init_basic (__main__.TestMedicamentosManager.test_init_basic)
Test de inicialización básica de MedicamentosManager ... ok
test_setup_ui_clears_frame (__main__.TestMedicamentosManager.test_setup_ui_clears_frame)
Test que setup_ui limpia el frame dinámico ... ok
test_setup_ui_configures_style (__main__.TestMedicamentosManager.test_setup_ui_configures_style)
Test que setup_ui configura el estilo correctamente ... ok
test_setup_ui_configures_treeview (__main__.TestMedicamentosManager.test_setup_ui_configures_treeview)
Test que setup_ui configura correctamente el treeview ... ok
test_setup_ui_creates_buttons (__main__.TestMedicamentosManager.test_setup_ui_creates_buttons)
Test que setup_ui crea todos los botones necesarios ... ok
test_setup_ui_creates_main_components (__main__.TestMedicamentosManager.test_setup_ui_creates_main_components)
Test que setup_ui crea los componentes principales ... ok
test_mostrar_medamentos_creates_manager (__main__.TestMostrarMedicamentos.test_mostrar_medamentos_creates_manager)
Test que mostrar_medamentos crea una instancia de MedicamentosManager ... ok

-----
Ran 14 tests in 0.057s

OK
```

Figura 4. Resultado de la ejecución de las pruebas de test/test_medamentos (14 test)

VII. Documentación de linter, y analizador estático de código



Para garantizar la calidad, estilo y consistencia del código fuente del proyecto, se aplicaron herramientas de análisis estático y formateo automático de código sobre la carpeta del repositorio de MediCareDesk utilizando dos herramientas complementarias: flake8 (linter) y black (formateador).

Herramientas:

1. Flake8
 - a. Tipo: Analizador estático de código (Linter)
 - b. Versión instalada: flake8 = 7

Se utilizó la configuración aplicada por defecto

2. Black
 - a. Tipo: Formateador de código automático, este se usó para complementar el linter y automatizar la corrección de errores del linter
 - b. Versión instalada: black = 24.4.2

Configuración aplicada:

Black sigue las reglas de estilo PEP8 de manera automática, por lo que no es necesaria su modificación.

3. Resultados

VIII. Evidencia, y resultados de ejecución del linter y del analizador estático de código:

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: black in c:\users\usuario\appdata\local\packages\pythonsoftwarefoundation.python.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (25.1.0)
Requirement already satisfied: flake8 in c:\users\usuario\appdata\local\packages\pythonsoftwarefoundation.python.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (7.3.0)
Requirement already satisfied: pep8-naming in c:\users\usuario\appdata\local\packages\pythonsoftwarefoundation.python.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (0.15.1)
Requirement already satisfied: click>=8.0.0 in c:\users\usuario\appdata\local\packages\pythonsoftwarefoundation.python.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from black) (8.2.1)
Requirement already satisfied: mypy_extensions>=0.4.3 in c:\users\usuario\appdata\local\packages\pythonsoftwarefoundation.python.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from black) (1.1.0)
Requirement already satisfied: packaging>=22.0 in c:\users\usuario\appdata\local\packages\pythonsoftwarefoundation.python.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from black) (25.0)
Requirement already satisfied: pathspec>=0.9.0 in c:\users\usuario\appdata\local\packages\pythonsoftwarefoundation.python.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from black) (0.12.1)
Requirement already satisfied: platformdirs>=2 in c:\users\usuario\appdata\local\packages\pythonsoftwarefoundation.python.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from black) (4.3.8)
Requirement already satisfied: mccabe<0.8.0,>=0.7.0 in c:\users\usuario\appdata\local\packages\pythonsoftwarefoundation.python.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from flake8) (0.7.0)
Requirement already satisfied: pycodestyle<2.15.0,>=2.14.0 in c:\users\usuario\appdata\local\packages\pythonsoftwarefoundation.python.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from flake8) (2.14.0)
Requirement already satisfied: pyflakes<3.5.0,>=3.4.0 in c:\users\usuario\appdata\local\packages\pythonsoftwarefoundation.python.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from flake8) (3.4.0)
Requirement already satisfied: colorama in c:\users\usuario\appdata\local\packages\pythonsoftwarefoundation.python.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from click>=8.0.0->black) (0.4.6)
```

Figura 3. Instalación de black y flake8 según pep8-naming

```
D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk>flake8 .
.\Scripts\borrar_tabla.py:6:80: E501 line too long (83 > 79 characters)
.\Scripts\borrar_tabla.py:9:80: E501 line too long (96 > 79 characters)
.\Scripts\datos_de_prueba.py:4:80: E501 line too long (104 > 79 characters)
.\Scripts\datos_de_prueba.py:6:1: E402 module level import not at top of file
.\Scripts\datos_de_prueba.py:7:1: E402 module level import not at top of file
.\Scripts\datos_de_prueba.py:10:1: E402 module level import not at top of file
.\Scripts\datos_de_prueba.py:29:80: E501 line too long (87 > 79 characters)
.\Scripts\datos_de_prueba.py:34:80: E501 line too long (92 > 79 characters)
.\Scripts\datos_de_prueba.py:36:80: E501 line too long (93 > 79 characters)
.\Scripts\datos_de_prueba.py:53:80: E501 line too long (91 > 79 characters)
.\Scripts\datos_de_prueba.py:73:80: E501 line too long (136 > 79 characters)
.\Scripts\datos_de_prueba.py:89:80: E501 line too long (115 > 79 characters)
.\Scripts\datos_de_prueba.py:103:80: E501 line too long (132 > 79 characters)
.\Scripts\datos_de_prueba.py:123:80: E501 line too long (104 > 79 characters)
.\Scripts\limpiar_tablas.py:41:44: W292 no newline at end of file
.\Scripts\reconstruir_base_datos.py:2:80: E501 line too long (80 > 79 characters)
.\Scripts\reconstruir_base_datos.py:12:80: E501 line too long (112 > 79 characters)
.\app\db\conexion.py:7:80: E501 line too long (85 > 79 characters)
.\app\db\conexion.py:11:1: E302 expected 2 blank lines, found 1
.\app\db\modelos.py:23:80: E501 line too long (128 > 79 characters)
.\app\db\modelos.py:24:80: E501 line too long (99 > 79 characters)
.\app\db\modelos.py:25:80: E501 line too long (99 > 79 characters)
.\app\db\modelos.py:35:80: E501 line too long (82 > 79 characters)
.\app\db\modelos.py:41:80: E501 line too long (96 > 79 characters)
.\app\db\modelos.py:42:1: E302 expected 2 blank lines, found 0
.\app\db\modelos.py:51:80: E501 line too long (91 > 79 characters)
.\app\db\modelos.py:53:80: E501 line too long (124 > 79 characters)
.\app\db\modelos.py:54:80: E501 line too long (95 > 79 characters)
.\app\db\modelos.py:56:80: E501 line too long (95 > 79 characters)
.\app\db\modelos.py:63:80: E501 line too long (97 > 79 characters)
.\app\db\modelos.py:67:1: E302 expected 2 blank lines, found 0
.\app\db\modelos.py:70:80: E501 line too long (127 > 79 characters)
.\app\db\modelos.py:84:1: E302 expected 2 blank lines, found 0
.\app\db\modelos.py:88:76: W291 trailing whitespace
```

Figura 4. Ejercicio de flake8



```
D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk>python -m black .
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\app\session.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\app\db\conexion.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\app\ui\alertas.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\app\logic\auth.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\main.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\Scripts\borrar_tabla.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\app\ui\historial_tomas.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\app\ui\registro.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\app\ui\tomas.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\app\ui\login.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\Scripts\limpiar_tablas.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\app\logic\tratamientos.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\tests\test_customtkinter.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\tests\test_auth.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\app\ui\base_view.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\app\ui\pacientes.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\Scripts\reconstruir_base_datos.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\tests\test_validacion_db.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\tests\test_conexion.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\tests\test_modelos.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\Scripts\datos_de_prueba.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\tests\test_tablas_vistas.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\app\ui\menu_lateral.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\tests\test_tratamientos.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\app\db\modelos.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\app\ui\medicamentos.py
reformatted D:\Rama edwin, ingesoft 1\Proyecto\Proyecto\MediCareDesk\app\ui\tratamientos.py

All done! 27 files reformatted, 16 files left unchanged.
```

Figura 4. Ejecución de black