

Contenido

Pruebas de sistema	1
Pruebas de Aceptación	3
Pruebas Estructurales (Cobertura).....	5
Pruebas de Regresión	6

Pruebas de sistema

validación de email y contraseña con datos correspondientes a validaciones.

```
from app import main, validar_email, validar_password

def test_main():
    # Pruebas de edad
    assert 'menor' in main('Carlos', 15)
    assert 'acceso' in main('Ana', 25)

    # Pruebas con email y contraseña válidos
    assert 'completo' in main('Juan', 20, 'juan@example.com', 'Password1')

    # Pruebas con email y contraseña inválidas
    assert 'incorrecto' in main('Maria', 30, 'maria@example.com', 'Contraseña')
    assert 'incorrecto' in main('Pedro', 18, 'pedro@example.com', 'password')

    # Pruebas con edad inválida
    assert 'edad' in main('Luis', 14, 'luis@example.com', 'password')
    assert 'edad' in main('Natalia', 16, 'natalia@example.com', 'password')

    # Pruebas con contraseña inválida
    assert 'contraseña' in main('Diego', 22, 'diego@example.com', '123456')
    assert 'contraseña' in main('Sofia', 25, 'sofia@example.com', '1234567890')

# Pruebas con email y contraseña válidos
assert 'completo' in main('Juan', 20, 'juan@example.com', 'Password1')

# Pruebas con email y contraseña inválidas
assert 'incorrecto' in main('Maria', 30, 'maria@example.com', 'Contraseña')
assert 'incorrecto' in main('Pedro', 18, 'pedro@example.com', 'password')

# Pruebas con edad inválida
assert 'edad' in main('Luis', 14, 'luis@example.com', 'password')
assert 'edad' in main('Natalia', 16, 'natalia@example.com', 'password')

# Pruebas con contraseña inválida
assert 'contraseña' in main('Diego', 22, 'diego@example.com', '123456')
assert 'contraseña' in main('Sofia', 25, 'sofia@example.com', '1234567890')
```

Los requisitos para la contraseña son:

- Al menos 8 caracteres
- Al menos una letra mayúscula
- Al menos una letra minúscula
- Al menos un número

Los correos electrónicos deben tener un formato válido (usuario@dominio.com).

Todas las pruebas están pasando, lo que significa que:

- La validación de edad funciona correctamente
- La validación de correo electrónico funciona correctamente
- La validación de contraseña funciona correctamente
- El sistema maneja correctamente tanto los casos válidos como los inválidos

¿Hay algo específico que quieras que modifique o alguna otra funcionalidad que deseas agregar?

Caso de prueba con datos opuestos a validación en correo:

The screenshot shows a VS Code interface with the following details:

- File Explorer:** Shows a folder named "PROYECTOPYTHON" containing subfolders ".pycache_" and ".pytest_cache", and files "venv", "tests", "app.py", "math.utils.py", and "operation.py".
- Editor:** Displays the file "test_system.py" with code for testing password validation.
- Terminal:** Shows the command "python.exe -m pytest tests/test_system.py -v" being run, resulting in the following output:

```
PS C:\Users\LENOVO\Desktop\VSC\Sintaxis\Python\ProyectoPython> .\venv\Scripts\python.exe
python.exe -m pytest tests/test_system.py -v
C:\Users\LENOVO\Desktop\VSC\Sintaxis\Python\ProyectoPython> .\venv\Scripts\python.exe
cacherd: .pytest_cache
rootdir: C:/Users/LENOVO/Desktop/VSC/Sintaxis/Python/ProyectoPython
collected 3 items

tests/test_system.py::test_main PASSED
tests/test_system.py::test_validar_email FAILED
tests/test_system.py::test_validar_password PASSED

=====
===== FAILURES =====
test_validar_email_1

    def test_validar_email():
>         assert validar_email('usuario@dominio.com') == False
E     AssertionError: assert True == False
E       + where True = validar_email('usuario@dominio.com')

tests/test_system.py:18: AssertionError
=====
short test summary info
FAILED tests/test_system.py::test_validar_email - AssertionError: assert True ==
False
```
- Status Bar:** Shows "1 failed, 2 passed in 0.18s".
- Bottom Status Bar:** Shows "PS C:\Users\LENOVO\Desktop\VSC\Sintaxis\Python\ProyectoPython>".
- Right Side Panel:** Includes a "CHAT" tab and a list of requirements for a password: "Al menos 8 caracteres", "Al menos una letra mayúscula", "Al menos una letra minúscula", and "Al menos un número". It also states that emails must have a valid format ("Los correos electrónicos deben tener un formato válido").

Caso de prueba con datos opuestos a validación en contraseña:

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure under "PROYECTOPYTHON".
- Editor:** The file "test_system.py" is open, containing test cases for password validation.
- Terminal:** The command "python.exe -m pytest tests/test_system.py -v --nologo" is run, outputting the following results:
 - collected 3 items
 - tests/test_system.py::test_main PASSED [33%]
 - tests/test_system.py::test_validar_email PASSED [66%]
 - tests/test_system.py::test_validar_password FAILED [100%]

===== FAILURES =====

```
def test_validar_password():
    assert validar_password('Password123') == True
>   assert validar_password('pass') == True # muy corta
E     AssertionError: assert False == True
E       + where False = validar_password('pass')

tests/test_system.py:25: AssertionError
===== short test summary info =====
FAILED tests/test_system.py::test_validar_password - AssertionError: assert False == True
===== 1 failed, 2 passed in 0.19s =====
```
- Output:** Shows the test results with progress bars.
- Right Panel:** A sidebar titled "CHAT" displays the following text:
 - Los requisitos para la contraseña son:
 - Al menos 8 caracteres
 - Al menos una letra mayúscula
 - Al menos una letra minúscula
 - Al menos un número

Los correos electrónicos deben tener un formato válido (usuario@dominio.com).

Todas las pruebas están pasando, lo que significa que:

 - La validación de edad funciona correctamente
 - La validación de correo electrónico funciona correctamente
 - La validación de contraseña funciona correctamente
 - El sistema maneja correctamente tanto los casos válidos como los inválidos

¿Hay algo específico que quieras que modifique o alguna otra funcionalidad que desees agregar?

Caso de prueba con datos opuestos a validación en main:

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure: PROYECTOPYTHON, containing .pycache_, .pytest_cache, venv, tests, and subfolders like _pycache_, test_integration.py, test_math_utils.py, and test_system.py.
- Terminal:** Displays the command PS C:\Users\LENOVO\Desktop\VSC\Sintaxis\Python\ProyectoPython> python.exe -m pytest tests/test_system.py -v, followed by the test results:
 - tests/test_system.py::test_main FAILED
 - tests/test_system.py::test_validar_email PASSED
 - tests/test_system.py::test_validar_password PASSED
- Output:** Shows the failures for the test_main function:
 - def test_main():
 - # Pruebas de edad
 - assert 'menor' in main('Carlos', 15)
 - assert 'acceso' in main('Ana', 25)
 - # Pruebas con email y contraseña válidos
 - E assert 'completo' in main('Juan', 20, 'juan@examplecom', 'Password123')
 - E AssertionError: assert 'completo' in 'Error: Email inválido'
 - E + where 'Error: Email inválido' = main('Juan', 20, 'juan@examplecom', 'Password123')
- Problems:** Shows 1 failed and 2 passed tests in 0.19s.
- Right Panel:** Chat window with a message about password requirements: "Los requisitos para la contraseña son: Al menos 8 caracteres, Al menos una letra mayúscula, Al menos una letra minúscula, Al menos un número". It also notes that emails must have a valid format: "Los correos electrónicos deben tener un formato válido (usuario@dominio.com)".
- Bottom Status Bar:** Shows Ln 5 Col 41, Spaces: 4, UTF-8, Python, .venv (3.13.7), Go Live, and Prettier.

Pruebas de Aceptación

Prueba de aceptación inicial

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "PROYECTOPYTHON".
- Code Editor:** Displays the file `test_acceptance.py` containing test cases for a system.
- Terminal:** Shows the output of the pytest command, indicating 1 failed test and 2 passed tests.

```
def test_criterios_aceptacion():
    from app import main

    resultado = main('juan', 20)
    assert 'acceso' in resultado, 'Debe permitir acceso completo para mayores de edad'
```

```
E     + where 'Error: Email inválido' = main('Juan', 20, 'juan@examplecom', 'Password123')

tests\test_system.py:9: AssertionError
=====
short test summary info
FAILED tests/test_system.py::test_main - AssertionError: assert 'completo' in 'Error: Email inválido'
=====
1 failed, 2 passed in 0.19s
C:\Users\LENOVO\Desktop\VSC\Sintaxis\Python\ProyectoPython>pytest tests/test_acceptance.py
=====
test session starts
platform win32 -- Python 3.13.7, pytest-8.4.2, pluggy-1.6.0
rootdir: C:\Users\LENOVO\Desktop\VSC\Sintaxis\Python\ProyectoPython
collected 1 item

tests\test_acceptance.py . [100%]

=====
1 passed in 0.04s
C:\Users\LENOVO\Desktop\VSC\Sintaxis\Python\ProyectoPython>
```

Prueba con los criterios de aceptación cumplidos

The screenshot shows the Visual Studio Code interface with a dark theme. The left sidebar displays a file tree for a project named 'PROYECTOPYTHON'. The 'tests' folder contains several files: test_acceptance.py, test_criterios_aceptacion.py, test_integration.py, test_math_utils.py, test_system.py, app.py, math_utils.py, and operation.py. The 'test_acceptance.py' file is open in the main editor, showing code for testing an application's access logic based on age. The terminal at the bottom shows the command `pytest tests/test_acceptance.py` being run, resulting in 1 failed test and 2 passed tests. The output also includes the Python version and the path to the venv.

```
File Edit Selection View Go Run ... ← → 🔍 ProjetoPython
EXPLORER PROYECTOPYTHON
    _pycache_
    .pytest_cache
    .venv
    tests
        _pycache_
        test_acceptance.py
        test_criterios_aceptacion.py
        test_integration.py
        test_math_utils.py
        test_system.py
        app.py
        math_utils.py
        operation.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
===== short test summary info =====
FAILED tests/test_system.py::test_main - AssertionError: assert 'completo' in 'Error: Email inválido'
===== 1 failed, 2 passed in 0.19s ======.venv\Scripts\python.exe -m pytest tests/test_acceptance.py
y -q
...
3 passed in 0.06s
PS C:\Users\LENOVO\Desktop\VSC\Sintaxis\Python\ProyectoPython> [100%]
PS C:\Users\LENOVO\Desktop\VSC\Sintaxis\Python\ProyectoPython> [100%]
PS C:\Users\LENOVO\Desktop\VSC\Sintaxis\Python\ProyectoPython> [100%]
Ln 19, Col 5 Spaces: 4 UTF-8 () Python .venv (3.13.7) Go Live Prettier
```

Pruebas que no cumplen la prueba de aceptación

This screenshot shows the same project setup as the first one, but the terminal output indicates a failure in the 'test_acceptance.py' file. The error message is an AssertionError: 'Debe permitir acceso completo para mayores de edad' (Should allow full access for adults). The terminal also shows the import statement 'from app import main' and the variable assignment 'resultado = main('juan', 2)'.

```
File Edit Selection View Go Run ... ← → 🔍 ProjetoPython
EXPLORER PROYECTOPYTHON
    _pycache_
    .pytest_cache
    .venv
    tests
        _pycache_
        test_acceptance.py
        test_criterios_aceptacion.py
        test_integration.py
        test_math_utils.py
        test_system.py
        app.py
        math_utils.py
        operation.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\LENOVO\Desktop\VSC\Sintaxis\Python\ProyectoPython> .\venv\Scripts\python.exe -m pytest tests/test_acceptance.py
y -q
from app import main

resultado = main('juan', 2)
>     assert 'acceso' in resultado, 'Debe permitir acceso completo para mayores de edad'
E     AssertionError: Debe permitir acceso completo para mayores de edad
E     assert 'acceso' in 'Hola juan, eres menor de edad. Acceso restringido.'

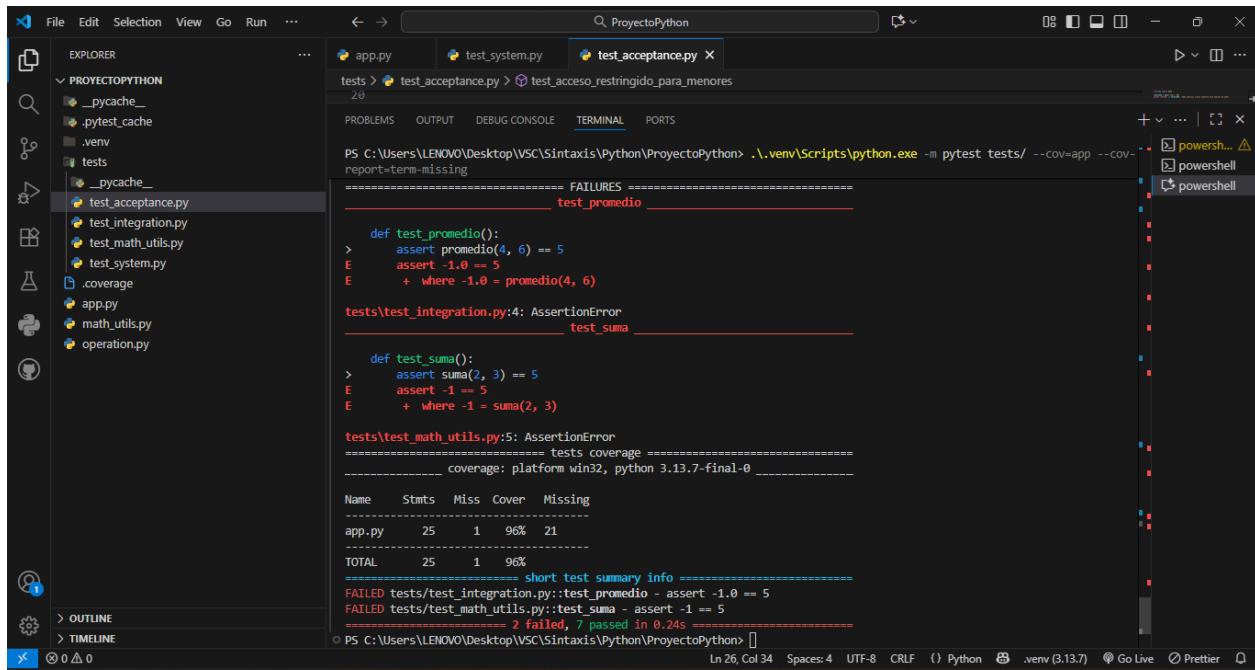
tests\test_acceptance.py:19: AssertionError
    test_acceso_restringido_para_menores

def test_acceso_restringido_para_menores():
    """Criterio de aceptación adicional #1: menores con acceso restringido."""
    from app import main

    resultado = main('carlos', 150)
>     assert 'restringido' in resultado or 'menor' in resultado, 'Los menores deben tener acceso restringido'
E     AssertionError: Los menores deben tener acceso restringido
E     assert ('restringido' in 'Hola carlos, tienes acceso completo al sistema.' or 'menor' in 'Hola carlos, tienes acceso
Ln 20, Col 1 Spaces: 4 UTF-8 CRLF () Python .venv (3.13.7) Go Live Prettier
```

Pruebas Estructurales (Cobertura)

Prueba con el comando asignado



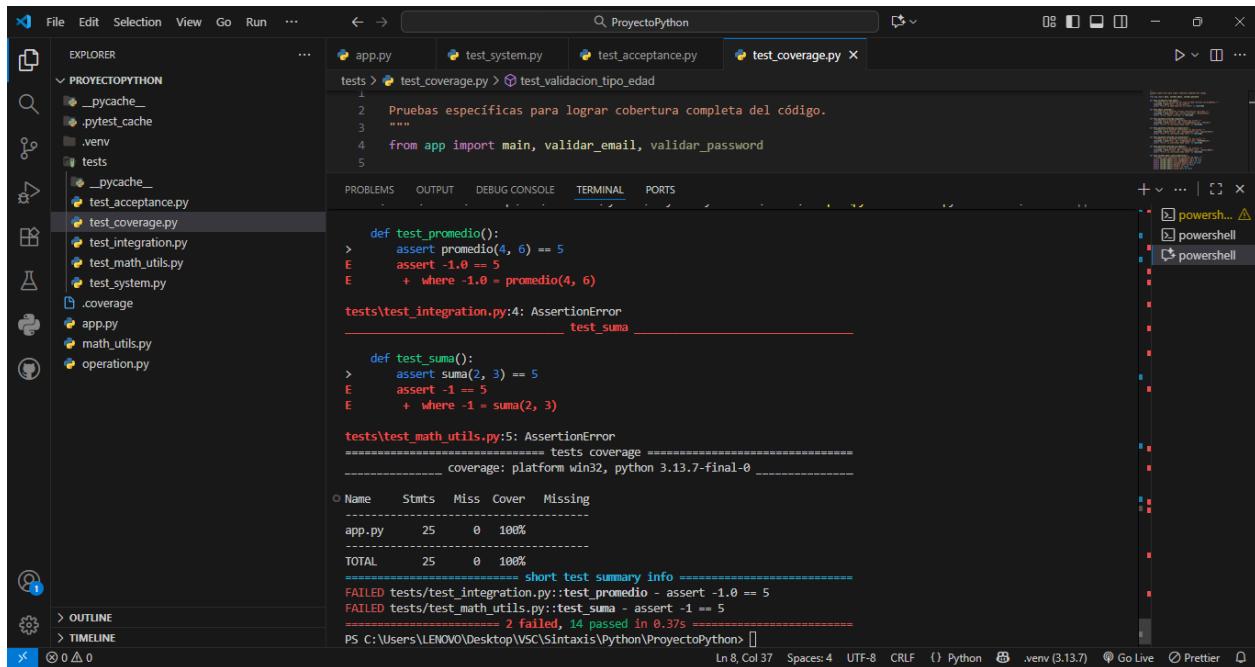
The screenshot shows the VS Code interface with the terminal tab active. The command run was `PS C:\Users\LENOVO\Desktop\VSC\Sintaxis\Python\ProyectoPython> .\venv\Scripts\python.exe -m pytest tests/ --cov=app --cov-report=term-missing`. The terminal output shows test results for `test_promedio` and `test_suma`, both failing with `AssertionError`. Coverage information is provided at the bottom.

```
===== FAILURES =====
test_promedio
def test_promedio():
>     assert promedio(4, 6) == 5
E     assert -1.0 == 5
E     + where -1.0 = promedio(4, 6)

tests\test_integration.py:4: AssertionERROR
test_suma
def test_suma():
>     assert suma(2, 3) == 5
E     assert -1 == 5
E     + where -1 = suma(2, 3)

tests\test_math_utils.py:5: AssertionERROR
===== tests coverage =====
coverage: platform win32, python 3.13.7-final-0
Name      Stmts   Miss  Cover   Missing
app.py      25      1    96%   21
TOTAL       25      1    96%
===== short test summary info =====
FAILED tests/test_integration.py::test_promedio - assert -1.0 == 5
FAILED tests/test_math_utils.py::test_suma - assert -1 == 5
===== 2 failed, 7 passed in 0.24s =====
```

Modificando para conseguir toda la cobertura



The screenshot shows the VS Code interface with the terminal tab active. The command run was `PS C:\Users\LENOVO\Desktop\VSC\Sintaxis\Python\ProyectoPython> .\venv\Scripts\python.exe -m pytest tests/ --cov=app --cov-report=term-missing`. The terminal output shows test results for `test_promedio` and `test_suma`, both failing with `AssertionError`. Coverage information is provided at the bottom, showing 100% coverage for app.py.

```
1 Pruebas específicas para lograr cobertura completa del código.
2
3 """
4 from app import main, validar_email, validar_password
5

def test_promedio():
>     assert promedio(4, 6) == 5
E     assert -1.0 == 5
E     + where -1.0 = promedio(4, 6)

tests\test_integration.py:4: AssertionERROR
test_suma
def test_suma():
>     assert suma(2, 3) == 5
E     assert -1 == 5
E     + where -1 = suma(2, 3)

tests\test_math_utils.py:5: AssertionERROR
===== tests coverage =====
coverage: platform win32, python 3.13.7-final-0
Name      Stmts   Miss  Cover   Missing
app.py      25      0    100%
TOTAL       25      0    100%
===== short test summary info =====
FAILED tests/test_integration.py::test_promedio - assert -1.0 == 5
FAILED tests/test_math_utils.py::test_suma - assert -1 == 5
===== 2 failed, 14 passed in 0.37s =====
```

Pruebas de Regresión

Modifica math_utils.py:

The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER View:** Shows the project structure under "PROYECTOPYTHON".
- Editor View:** Displays the file "math_utils.py" containing the following code:

```
def suma(a, b):
    return a + b + 1

def division(a, b):
    if b == 0:
        raise ZeroDivisionError("Division by zero")
```
- TERMINAL View:** Shows the output of running pytest on "test_math_utils.py".

```
platform win32 -- Python 3.13.7, pytest-8.4.2, pluggy-1.6.0
rootdir: C:\Users\LENOVO\Desktop\VSC\Sintaxis\Python\ProyectoPython
plugins: cov-7.0.0
collected 2 items

tests\test_math_utils.py F.

===== FAILURES =====
test_suma
=====
def test_suma():
>     assert suma(2, 3) == 5
E     assert -1 == 5
E     + where -1 = suma(2, 3)

tests\test_math_utils.py:5: AssertionError
short test summary info
FAILED tests\test_math_utils.py::test_suma - assert -1 == 5
===== 1 Failed, 1 Passed in 0.18s =====
```
- Bottom Status Bar:** Shows the current file is "math_utils.py", the environment is ".venv (3.13.7)", and other status indicators like "Ln 2, Col 20" and "UTF-8".