



Pruebas de software y aseguramiento de la calidad

TC4017.10

6.2 Ejercicio de programación 3 y pruebas de unidad

Profesor titular: Dr. Gerardo Padilla Zárate

Profesor asistente: Mtra. Viridiana Rodríguez González

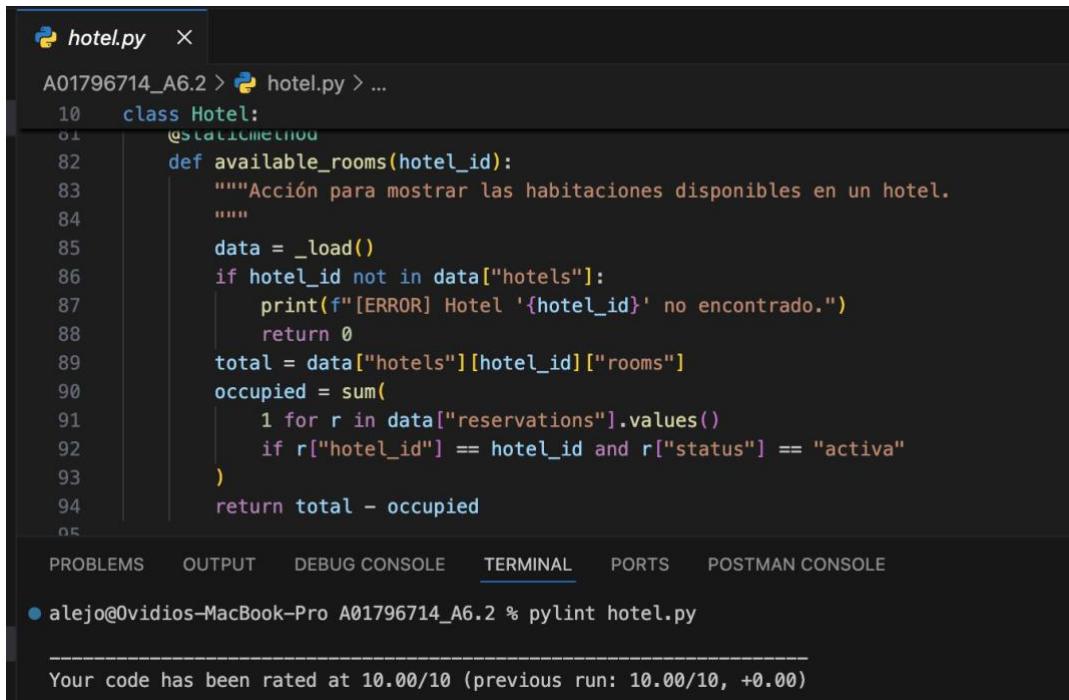
Alumno

A01796714 Ovidio Alejandro Hernández Ruano

Febrero 22, 2026

Ejercicio de programación 3

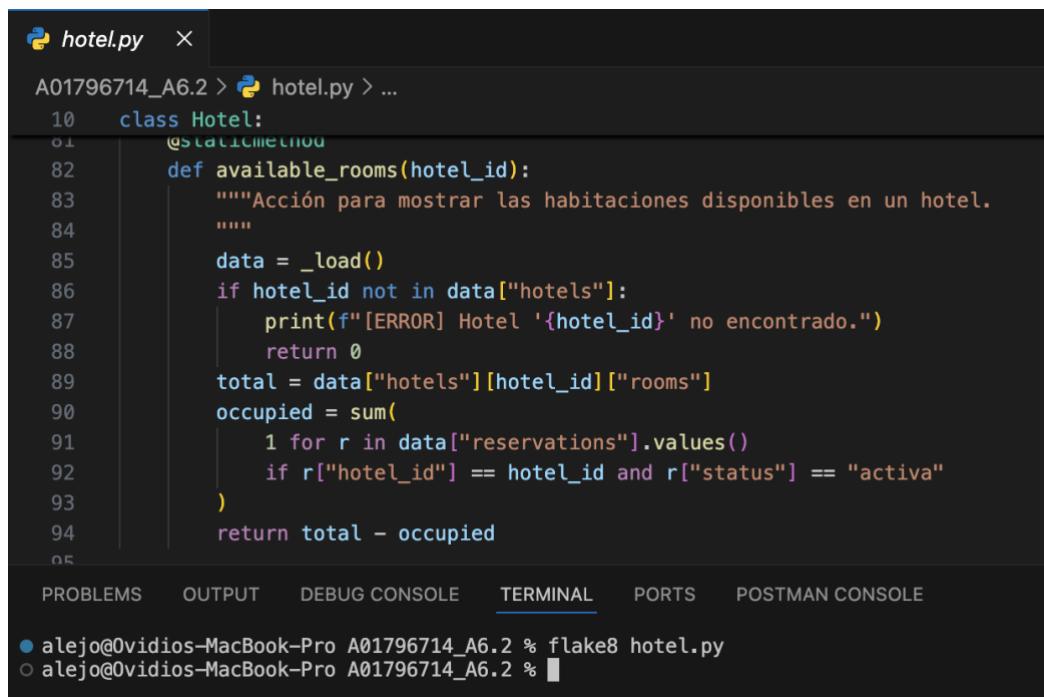
Evidencia de la correcta ejecución de *pylint*, aplicado al archivo hotel.py que contiene la clase Hotel.



```
A01796714_A6.2 > 🐍 hotel.py > ...
10  class Hotel:
11      @staticmethod
12          def available_rooms(hotel_id):
13              """Acción para mostrar las habitaciones disponibles en un hotel.
14              """
15              data = _load()
16              if hotel_id not in data["hotels"]:
17                  print(f"[ERROR] Hotel '{hotel_id}' no encontrado.")
18                  return 0
19              total = data["hotels"][hotel_id]["rooms"]
20              occupied = sum(
21                  1 for r in data["reservations"].values()
22                      if r["hotel_id"] == hotel_id and r["status"] == "activa"
23              )
24              return total - occupied
25

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    POSTMAN CONSOLE
● alejo@Ovidios-MacBook-Pro A01796714_A6.2 % pylint hotel.py
-----
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
```

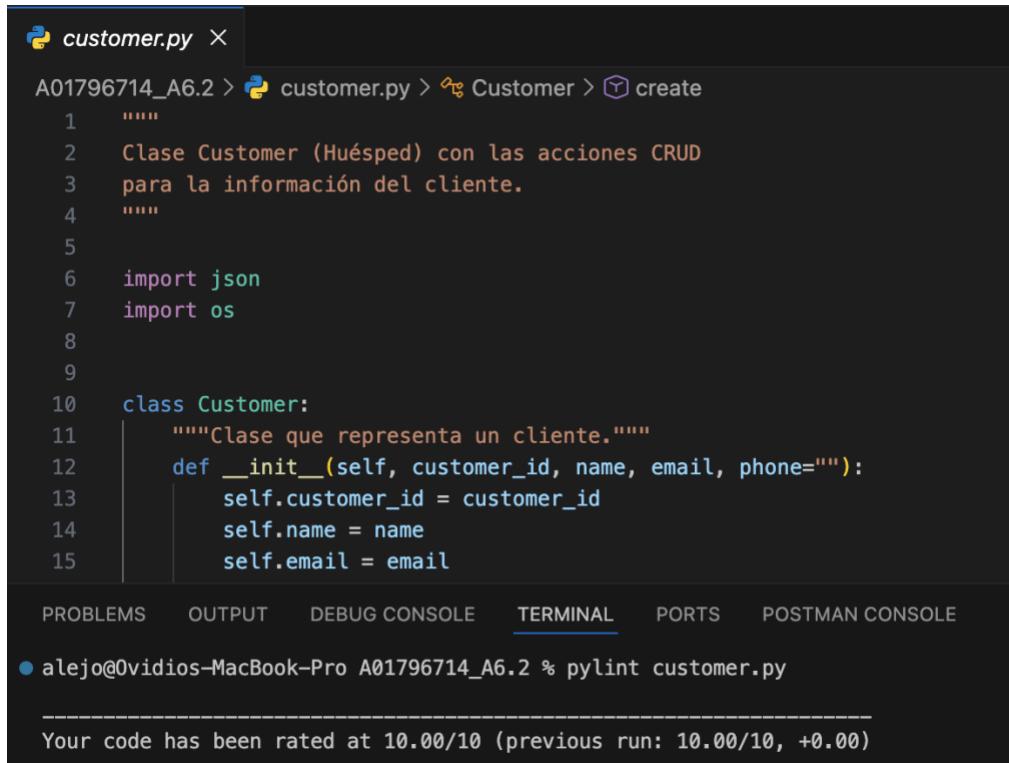
Evidencia de la correcta ejecución de *flake8*, aplicado al archivo hotel.py que contiene la clase Hotel.



```
A01796714_A6.2 > 🐍 hotel.py > ...
10  class Hotel:
11      @staticmethod
12          def available_rooms(hotel_id):
13              """Acción para mostrar las habitaciones disponibles en un hotel.
14              """
15              data = _load()
16              if hotel_id not in data["hotels"]:
17                  print(f"[ERROR] Hotel '{hotel_id}' no encontrado.")
18                  return 0
19              total = data["hotels"][hotel_id]["rooms"]
20              occupied = sum(
21                  1 for r in data["reservations"].values()
22                      if r["hotel_id"] == hotel_id and r["status"] == "activa"
23              )
24              return total - occupied
25

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    POSTMAN CONSOLE
● alejo@Ovidios-MacBook-Pro A01796714_A6.2 % flake8 hotel.py
○ alejo@Ovidios-MacBook-Pro A01796714_A6.2 % █
```

Evidencia de la correcta ejecución de *pylint*, aplicado al archivo customer.py que contiene la clase Customer (Huesped).

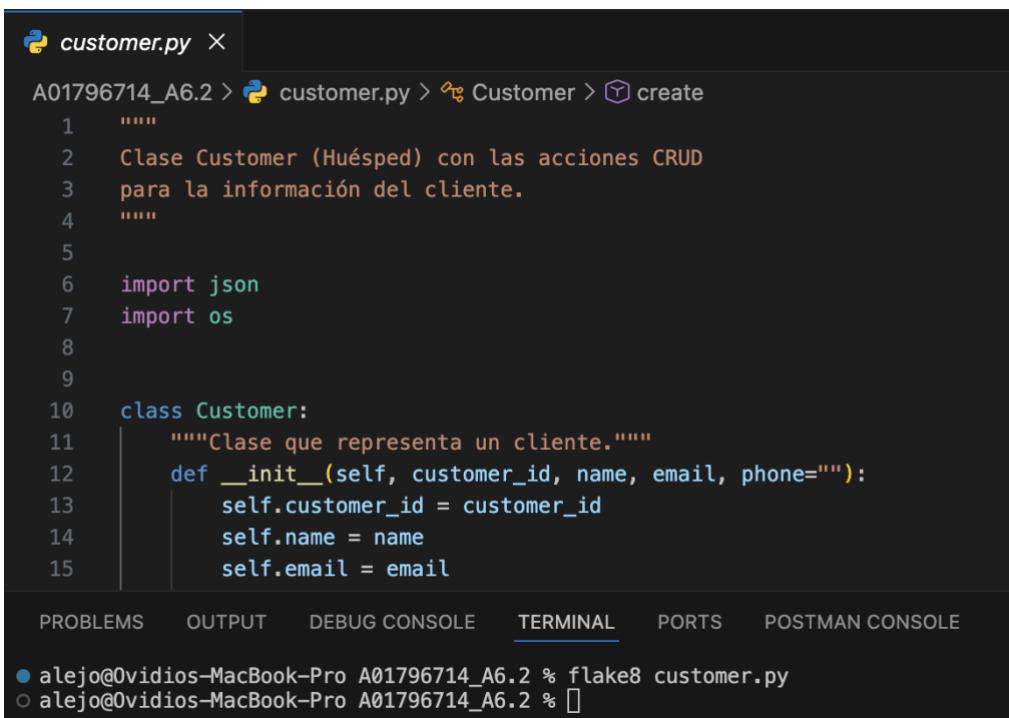


```
A01796714_A6.2 > 🐍 customer.py > ⚙ Customer > ✎ create
1     """
2     Clase Customer (Huésped) con las acciones CRUD
3     para la información del cliente.
4     """
5
6     import json
7     import os
8
9
10    class Customer:
11        """Clase que representa un cliente."""
12        def __init__(self, customer_id, name, email, phone=""):
13            self.customer_id = customer_id
14            self.name = name
15            self.email = email

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    POSTMAN CONSOLE

● alejo@0vidios-MacBook-Pro A01796714_A6.2 % pylint customer.py
-----
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
```

Evidencia de la correcta ejecución de *flake8*, aplicado al archivo customer.py que contiene la clase Customer (Huesped).

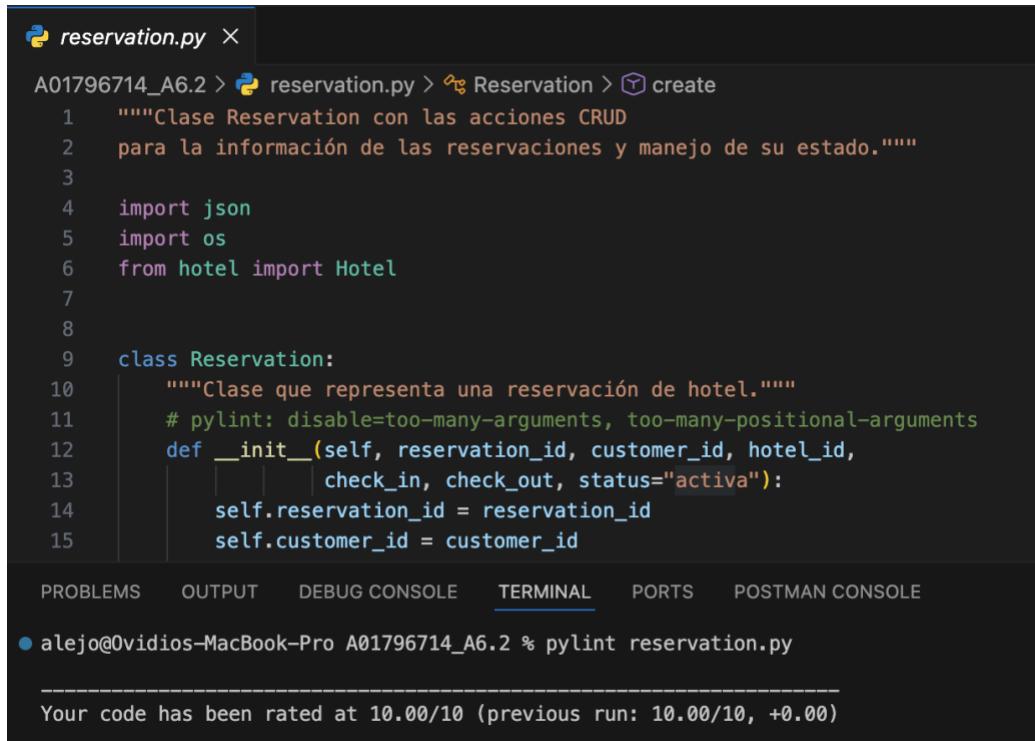


```
A01796714_A6.2 > 🐍 customer.py > ⚙ Customer > ✎ create
1     """
2     Clase Customer (Huésped) con las acciones CRUD
3     para la información del cliente.
4     """
5
6     import json
7     import os
8
9
10    class Customer:
11        """Clase que representa un cliente."""
12        def __init__(self, customer_id, name, email, phone=""):
13            self.customer_id = customer_id
14            self.name = name
15            self.email = email

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    POSTMAN CONSOLE

● alejo@0vidios-MacBook-Pro A01796714_A6.2 % flake8 customer.py
○ alejo@0vidios-MacBook-Pro A01796714_A6.2 % [ ]
```

Evidencia de la correcta ejecución de *pylint*, aplicado al archivo reservation.py que contiene la clase Reservation.



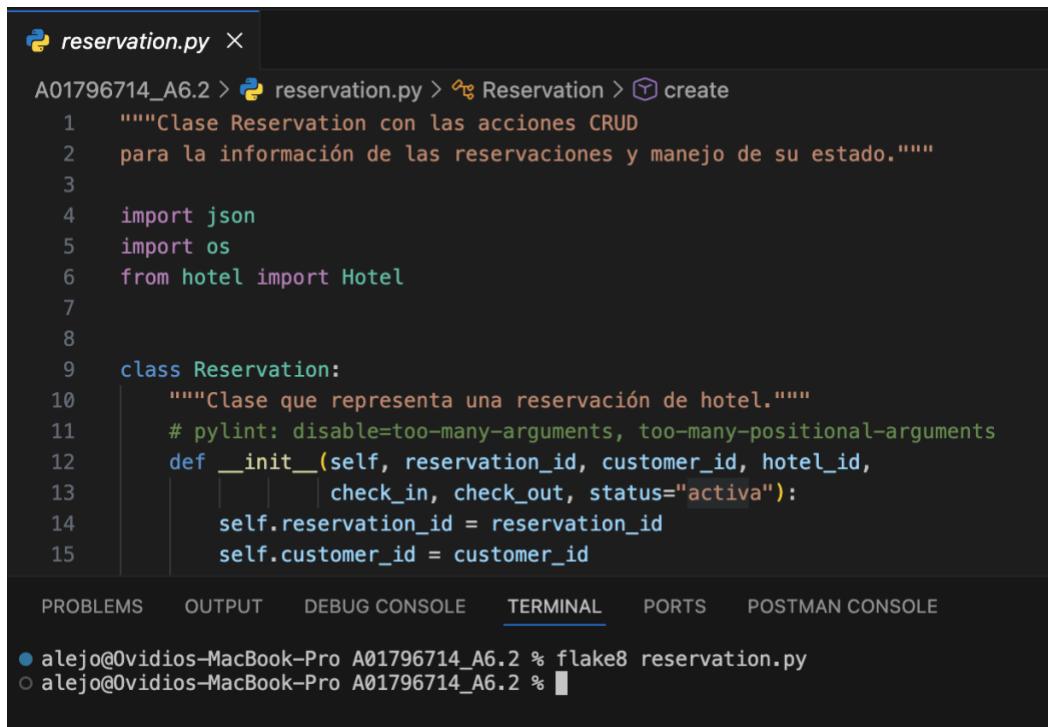
```
A01796714_A6.2 > 🐍 reservation.py > ⚙ Reservation > ⚡ create
1     """Clase Reservation con las acciones CRUD
2     para la información de las reservaciones y manejo de su estado."""
3
4     import json
5     import os
6     from hotel import Hotel
7
8
9     class Reservation:
10         """Clase que representa una reservación de hotel."""
11         # pylint: disable=too-many-arguments, too-many-positional-arguments
12         def __init__(self, reservation_id, customer_id, hotel_id,
13                      check_in, check_out, status="activa"):
14             self.reservation_id = reservation_id
15             self.customer_id = customer_id

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    POSTMAN CONSOLE

● alejo@Ovidios-MacBook-Pro A01796714_A6.2 % pylint reservation.py

-----
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
```

Evidencia de la correcta ejecución de *flake8*, aplicado al archivo reservation.py que contiene la clase Reservation.



```
A01796714_A6.2 > 🐍 reservation.py > ⚙ Reservation > ⚡ create
1     """Clase Reservation con las acciones CRUD
2     para la información de las reservaciones y manejo de su estado."""
3
4     import json
5     import os
6     from hotel import Hotel
7
8
9     class Reservation:
10         """Clase que representa una reservación de hotel."""
11         # pylint: disable=too-many-arguments, too-many-positional-arguments
12         def __init__(self, reservation_id, customer_id, hotel_id,
13                      check_in, check_out, status="activa"):
14             self.reservation_id = reservation_id
15             self.customer_id = customer_id

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    POSTMAN CONSOLE

● alejo@Ovidios-MacBook-Pro A01796714_A6.2 % flake8 reservation.py
○ alejo@Ovidios-MacBook-Pro A01796714_A6.2 % █
```

JSON Escenario de prueba TC1

```
tc1.json  ×  
A01796714_A6.2 > test > tc1.json > ...  
1 {  
2     "hotels": {  
3         "H001": {  
4             "hotel_id": "H001",  
5             "name": "Gold Hotel",  
6             "location": "CDMX",  
7             "rooms": 5  
8         },  
9         "H002": {  
10            "hotel_id": "H002",  
11            "name": "Hotel Boutique",  
12            "location": "Guadalajara",  
13            "rooms": 2  
14        }  
15    },  
16    "customers": {  
17        "C001": {  
18            "customer_id": "C001",  
19            "name": "Luis Perez",  
20            "email": "luis@gmail.com",  
21            "phone": "5050-2020"  
22        },  
23        "C002": {  
24            "customer_id": "C002",  
25            "name": "Maria Lopez",  
26            "email": "maria@gmail.com",  
27            "phone": "4040-5678"  
28        }  
29    },  
30    "reservations": {  
31        "R001": {  
32            "reservation_id": "R001",  
33            "customer_id": "C001",  
34            "hotel_id": "H001",  
35            "check_in": "10/07/2026",  
36            "check_out": "12/07/2026",  
37            "status": "activa"  
38        },  
39        "R002": {  
40            "reservation_id": "R002",  
41            "customer_id": "C002",  
42            "hotel_id": "H002"  
43        }  
44    }  
45}
```

JSON Escenario de prueba TC2

```
tc2.json  ×  
A01796714_A6.2 > test > tc2.json > {} hotels  
1 {  
2     "hotels": {  
3         "H001": {  
4             "hotel_id": "H111",  
5             "name": "Hotel Independencia",  
6             "location": "CDMX",  
7             "rooms": 5  
8         },  
9         "H002": {  
10            "hotel_id": "H450",  
11            "name": "Hotel Sultana",  
12            "location": "Monterrey",  
13            "rooms": 2  
14        }  
15    },  
16    "customers": {  
17        "C001": {  
18            "customer_id": "C123",  
19            "name": "Juan Rojas",  
20            "email": "juan.rojas@gmail.com",  
21            "phone": "5252-2120"  
22        },  
23        "C002": {  
24            "customer_id": "C985",  
25            "name": "Alberto Sanchez",  
26            "email": "alberto.sanchez@gmail.com",  
27            "phone": "1236-7852"  
28        },  
29        "C003": {  
30            "customer_id": "C875",  
31            "name": "Carlos Ruiz",  
32            "email": "carlos.ruiz@gmail.com",  
33            "phone": "3312-9900"  
34        },  
35        "C004": {  
36            "customer_id": "C004",  
37            "name": "Daniela Hernandez",  
38            "email": "daniela.hernandez@gmail.com",  
39            "phone": "4444-5555"  
40        }  
41    }  
42}
```

JSON Escenario de prueba TC3

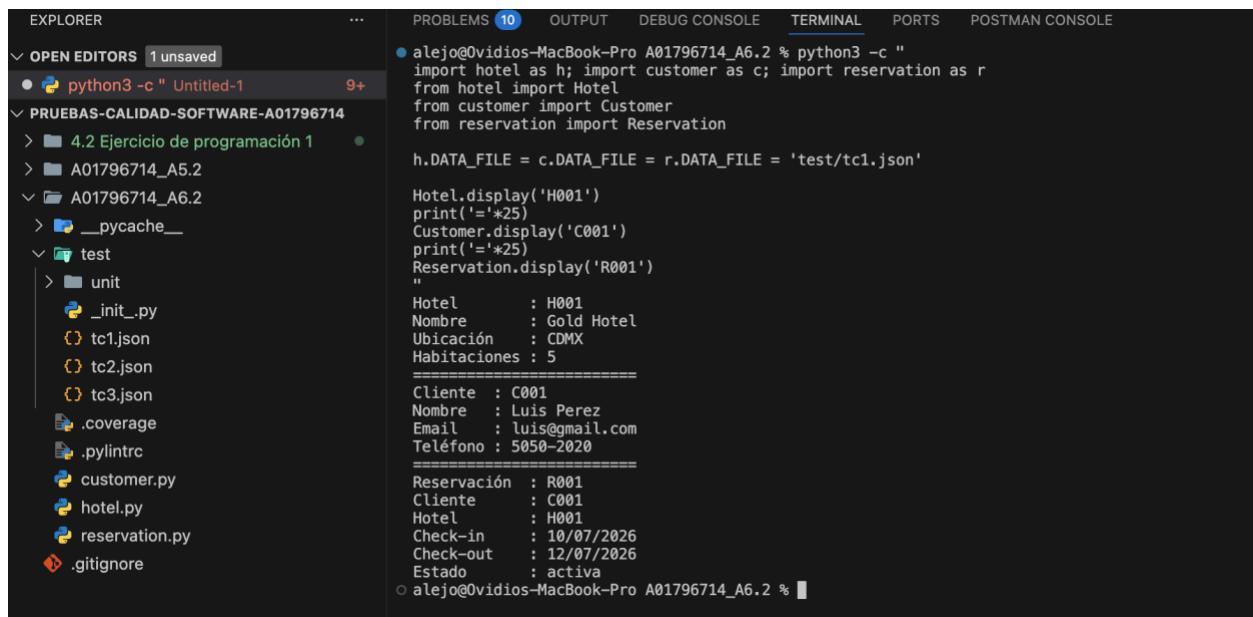
```
tc3.json  X  
A01796714_A6.2 > test > tc3.json > {} hotels > {} H013 > abc name  
1  {  
2      "hotels": {  
3          "H010": {  
4              "hotel_id": "H010",  
5              "name": "Torre Ejecutiva",  
6              "location": "Monterrey",  
7              "rooms": 4  
8          },  
9          "H011": {  
10             "hotel_id": "H011",  
11             "name": "Posada del Sol",  
12             "location": "Merida",  
13             "rooms": 2  
14         },  
15         "H012": {  
16             "hotel_id": "H012",  
17             "name": "Suites Reforma",  
18             "location": "CDMX",  
19             "rooms": 1  
20         },  
21         "H013": {  
22             "hotel_id": "H013",  
23             "name": "Grand Caribe",  
24             "location": "Cancun",  
25             "rooms": 3  
26         }  
27     },  
28     "customers": {  
29         "C010": {  
30             "customer_id": "C010",  
31             "name": "Fernando Castillo",  
32             "email": "fcastillo@proton.com",  
33             "phone": "8112-4455"  
34         },  
35         "C011": {  
36             "customer_id": "C011",  
37             "name": "Gabriela Vega",  
38             "email": "gvega@yahoo.com",  
39             "phone": "9991-6677"  
40         },  
41         "C012": {  
42             "customer_id": "C012",  
43         }  
44     }  
45 }
```

Ejecución por script del escenario de prueba TC1

```
python3 -c "
import hotel as h; import customer as c; import reservation as r
from hotel import Hotel
from customer import Customer
from reservation import Reservation

h.DATA_FILE = c.DATA_FILE = r.DATA_FILE = 'test/tc1.json'

Hotel.display('H001')
print('*25)
Customer.display('C001')
print('*25)
Reservation.display('R001')
"
```



```
EXPLORER ... PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE
OPEN EDITORS 1 unsaved alejo@Ovidios-MacBook-Pro A01796714_A6.2 % python3 -c "
python3 -c " Untitled-1 9+
PRUEBAS-CALIDAD-SOFTWARE-A01796714
> 4.2 Ejercicio de programación 1
> A01796714_A5.2
A01796714_A6.2
> __pycache__
test
> unit
_init_.py
tc1.json
tc2.json
tc3.json
.coverage
.pylintrc
customer.py
hotel.py
reservation.py
.gitignore
alejo@Ovidios-MacBook-Pro A01796714_A6.2 %

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE
alejo@Ovidios-MacBook-Pro A01796714_A6.2 % python3 -c "
import hotel as h; import customer as c; import reservation as r
from hotel import Hotel
from customer import Customer
from reservation import Reservation

h.DATA_FILE = c.DATA_FILE = r.DATA_FILE = 'test/tc1.json'

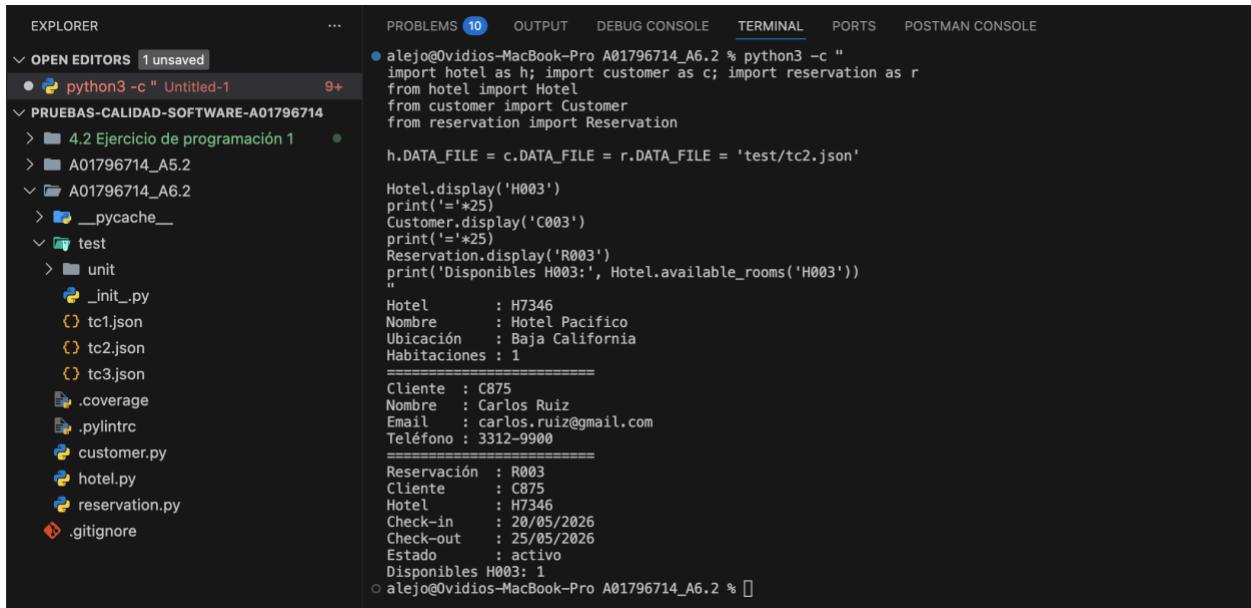
Hotel.display('H001')
print('*25)
Customer.display('C001')
print('*25)
Reservation.display('R001')
"
=====
Hotel      : H001
Nombre    : Gold Hotel
Ubicación  : CDMX
Habitaciones : 5
=====
Cliente   : C001
Nombre    : Luis Perez
Email     : luis@mail.com
Teléfono  : 5050-2020
=====
Reservación : R001
Cliente   : C001
Hotel     : H001
Check-in   : 10/07/2026
Check-out  : 12/07/2026
Estado    : activa
alejo@Ovidios-MacBook-Pro A01796714_A6.2 %
```

Ejecución por script del escenario de prueba TC2

```
python3 -c "
import hotel as h; import customer as c; import reservation as r
from hotel import Hotel
from customer import Customer
from reservation import Reservation

h.DATA_FILE = c.DATA_FILE = r.DATA_FILE = 'test/tc2.json'

Hotel.display('H003')
print('*25)
Customer.display('C003')
print('*25)
Reservation.display('R003')
print('Disponibles H003:', Hotel.available_rooms('H003'))
"
```



The screenshot shows a code editor interface with the following details:

- EXPLORER** pane on the left displaying project files and folders.
- PROBLEMS** tab in the top bar showing 10 issues.
- OUTPUT** tab in the top bar.
- DEBUG CONSOLE** tab in the top bar.
- TERMINAL** tab in the top bar.
- PORTS** tab in the top bar.
- POSTMAN CONSOLE** tab in the top bar.
- Terminal Content:**

```
alejo@Ovidios-MacBook-Pro A01796714_A6.2 % python3 -c "
import hotel as h; import customer as c; import reservation as r
from hotel import Hotel
from customer import Customer
from reservation import Reservation

h.DATA_FILE = c.DATA_FILE = r.DATA_FILE = 'test/tc2.json'

Hotel.display('H003')
print('*25)
Customer.display('C003')
print('*25)
Reservation.display('R003')
print('Disponibles H003:', Hotel.available_rooms('H003'))
"

Hotel : H7346
Nombre : Hotel Pacifico
Ubicación : Baja California
Habitaciones : 1
=====
Cliente : C875
Nombre : Carlos Ruiz
Email : carlos.ruiz@gmail.com
Teléfono : 3312-9900
=====
Reservación : R003
Cliente : C875
Hotel : H7346
Check-in : 20/05/2026
Check-out : 25/05/2026
Estado : activo
Disponibles H003: 1
alejo@Ovidios-MacBook-Pro A01796714_A6.2 %
```

Ejecución por script del escenario de prueba TC3

```
python3 -c "
import hotel as h; import customer as c; import reservation as r
from hotel import Hotel
from customer import Customer
from reservation import Reservation

h.DATA_FILE = c.DATA_FILE = r.DATA_FILE = 'test/tc3.json'

Hotel.display('H012')
print('Disponibles H012 (debe ser 0):', Hotel.available_rooms('H012'))
print('*'*50)
print('*'*50)
print('Caso negativo: Hotel lleno')
print('--- Intento reservar hotel lleno ---')
Reservation.create('R099', 'C010', 'H012', '01/01/2027', '05/01/2027')
print('*'*50)
print('*'*50)
print('Caso negativo: Cliente inexistente')
print('--- Intento con cliente inexistente ---')
Reservation.create('R100', 'CLIENTE_NO_REGISTRADO', 'H010', '01/01/2027', '05/01/2027')
"
"
```

The screenshot shows a code editor interface with two main panes: EXPLORER and TERMINAL.

EXPLORER: Shows the project structure. It includes a folder named "PRUEBAS-CALIDAD-SOFTWARE-A01796714" containing a subfolder "A01796714_A6.2" and several files: ".coverage", ".pylintrc", "customer.py", "hotel.py", "reservation.py", "tc1.json", "tc2.json", "tc3.json", and ".gitignore". There are also "unit" and "test" folders.

TERMINAL: Displays the output of a Python script named "tc3.py". The script performs the following actions:

- Imports hotel, customer, and reservation modules.
- Defines DATA_FILE variables for hotel, customer, and reservation.
- Prints the number of available rooms for Hotel H012 (expecting 0).
- Tries to create a reservation R099 for Hotel H012, which fails because it's full.
- Tries to create a reservation R100 for a non-existent client, which fails because the client does not exist.

The terminal output ends with an error message about a reservation already existing.

Evidencia de ejecución satisfactoria de pruebas unitarias y evidencia del resultado en consola al ejecutar las pruebas unitarias.

```
alejo@Ovidios-MacBook-Pro:~/Documents/MNA TEC MTY/5TH TRIMESTRE/SOFTWARE QUALITY/Pruebas-Calidad-Software-A01796714/A01796714_A6.2/test/unit/ alejo$ python hotel_reservation_test.py -v
ok
test_cancelar_reservacion_ya_cancelada (__main__.TestReservation.test_cancelar_reservacion_ya_cancelada)
Verifica que no se puede cancelar una reservación ya cancelada. ... [ERROR] La reservación 'R1' ya está cancelada.
ok
test_crear_reservacion (__main__.TestReservation.test_crear_reservacion)
Verifica que se puede crear una reservación correctamente. ... ok
test_crear_reservacion_cliente_inexistente (__main__.TestReservation.test_crear_reservacion_cliente_inexistente)
Verifica que no se puede reservar con un cliente inexistente. ... [ERROR] Cliente 'customer1' no encontrado.
ok
test_crear_reservacion_duplicada (__main__.TestReservation.test_crear_reservacion_duplicada)
Verifica que no se puede crear una reservación con ID duplicado. ... [ERROR] La reservación 'R1' ya existe.
ok
test_crear_reservacion_hotel_inexistente (__main__.TestReservation.test_crear_reservacion_hotel_inexistente)
Verifica que no se puede reservar en un hotel inexistente. ... [ERROR] Hotel 'hotell' no encontrado.
ok
test_crear_reservacion_sin_habitaciones (__main__.TestReservation.test_crear_reservacion_sin_habitaciones)
Verifica que no se puede reservar si no hay habitaciones. ... [ERROR] No hay habitaciones disponibles en el hotel 'HHTLL'.
ok
test_habitaciones_disminuyen_al_reservar (__main__.TestReservation.test_habitaciones_disminuyen_al_reservar)
Verifica que las habitaciones disponibles disminuyen al reservar. ... ok
test_habitaciones_se_restauran_al_cancelar (__main__.TestReservation.test_habitaciones_se_restauran_al_cancelar)
Verifica que las habitaciones se restauran al cancelar. ... ok
test_mostrar_reservacion_vacia (__main__.TestReservation.test_mostrar_reservacion_vacia)
Verifica que un JSON corrupto es manejado sin lanza excepción. ... [ERROR] No se pudo cargar /Users/alejo/Documents/MNA TEC MTY/5TH TRIMESTRE/SOFTWARE QUALITY/Pruebas-Calidad-Software-A01796714/A01796714_A6.2/test/unit/../.tcl.json: Expecting property name enclosed in double quotes: line 1 column 2 (char 1)
[ERROR] Cliente 'C1' no encontrado.
ok
test_mostrar_reservacion (__main__.TestReservation.test_mostrar_reservacion)
Verifica que display no lanza excepción. ... Reservación : R1
Cliente : C1
Hotel : H1
Check-in : 10/04/2026
Check-out : 25/04/2026
Estado : activa
ok
test_mostrar_reservacion_inexistente (__main__.TestReservation.test_mostrar_reservacion_inexistente)
Verifica que display con ID inexistente no lanza excepción. ... [ERROR] Reservación 'RV1' no encontrada.
ok
test_obtener_reservacion (__main__.TestReservation.test_obtener_reservacion)
Verifica que se puede obtener una reservación por ID. ... ok
test_obtener_reservacion_inexistente (__main__.TestReservation.test_obtener_reservacion_inexistente)
Verifica que obtener una reservación inexistente retorna None. ... [ERROR] Reservación 'RV1' no encontrada.
ok
test_tipo_incorrecto_en_archivo (__main__.TestReservation.test_tipo_incorrecto_en_archivo)
Verifica que un JSON con tipo incorrecto es manejado correctamente. ... [ERROR] No se pudo cargar /Users/alejo/Documents/MNA TEC MTY/5TH TRIMESTRE/SOFTWARE QUALITY/Pruebas-Calidad-Software-A01796714/A01796714_A6.2/test/unit/../.tcl.json: El archivo de datos debe ser un JSON.
[ERROR] Hotel 'H1' no encontrado.
ok

Ran 40 tests in 0.011s
OK
```

`test_crear_cliente (__main__.TestCustomer.test_crear_cliente)`

Verifica que se puede crear un cliente correctamente. ... ok

`test_crear_cliente_duplicado (__main__.TestCustomer.test_crear_cliente_duplicado)`

Verifica que no se puede crear un cliente con ID duplicado. ... [ERROR] El cliente 'C1' ya existe.

ok

`test_eliminar_cliente (__main__.TestCustomer.test_eliminar_cliente)`

Verifica que se puede eliminar un cliente existente. ... [ERROR] Cliente 'C1' no encontrado.

ok

`test_eliminar_cliente_inexistente (__main__.TestCustomer.test_eliminar_cliente_inexistente)`

Verifica que eliminar un cliente inexistente retorna False. ... [ERROR] Cliente 'customer1' no encontrado.

ok

`test_modificar_campo_no_permitido_cliente`

`(__main__.TestCustomer.test_modificar_campo_no_permitido_cliente)`

Verifica que modificar un campo no permitido no falla. ... [ADVERTENCIA] El campo 'campo_invalido' no es modificable.

ok

`test_modificar_cliente (__main__.TestCustomer.test_modificar_cliente)`

Verifica que se pueden modificar campos del cliente. ... ok

`test_modificar_cliente_inexistente (__main__.TestCustomer.test_modificar_cliente_inexistente)`

Verifica que modificar un cliente inexistente retorna False. ... [ERROR] Cliente 'customer1' no encontrado.

Ok

test_mostrar_cliente (__main__.TestCustomer.test_mostrar_cliente)
Verifica que display no lanza excepción. ... Cliente : C1
Nombre : Mario Jimenez
Email : mjim@gmail.com
Teléfono :
ok

test_mostrar_cliente_inexistente (__main__.TestCustomer.test_mostrar_cliente_inexistente)
Verifica que display con ID inexistente no lanza excepción. ... [ERROR] Cliente 'customer1' no encontrado.
ok

test_obtener_cliente (__main__.TestCustomer.test_obtener_cliente)
Verifica que se puede obtener un cliente por ID. ... ok

test_obtener_cliente_inexistente (__main__.TestCustomer.test_obtener_cliente_inexistente)
Verifica que obtener un cliente inexistente retorna None. ... [ERROR] Cliente 'customer1' no encontrado.
ok

test_crear_hotel (__main__.TestHotel.test_crear_hotel)
Verifica que se puede crear un hotel correctamente. ... ok

test_crear_hotel_duplicado (__main__.TestHotel.test_crear_hotel_duplicado)
Verifica que no se puede crear un hotel con ID duplicado. ... [ERROR] El hotel 'H1' ya existe.
ok

test_eliminar_hotel (__main__.TestHotel.test_eliminar_hotel)
Verifica que se puede eliminar un hotel existente. ... [ERROR] Hotel 'H1' no encontrado.
ok

test_eliminar_hotel_inexistente (__main__.TestHotel.test_eliminar_hotel_inexistente)
Verifica que eliminar un hotel inexistente retorna False. ... [ERROR] Hotel 'hotell' no encontrado.
ok

test_habitaciones_disponibles (__main__.TestHotel.test_habitaciones_disponibles)
Verifica el conteo de habitaciones disponibles. ... ok

test_habitaciones_disponibles_hotel_inexistente (__main__.TestHotel.test_habitaciones_disponibles_hotel_inexistente)
Verifica que available_rooms retorna 0 para hotel inexistente. ... [ERROR] Hotel 'hotell' no encontrado.
ok

test_modificar_campo_no_permitido_hotel (__main__.TestHotel.test_modificar_campo_no_permitido_hotel)
Verifica que modificar un campo no permitido no falla. ... [ADVERTENCIA] El campo 'campo_invalido' no es modificable.
ok

test_modificar_hotel (__main__.TestHotel.test_modificar_hotel)
Verifica que se pueden modificar campos del hotel. ... ok

test_modificar_hotel_inexistente (__main__.TestHotel.test_modificar_hotel_inexistente)
Verifica que modificar un hotel inexistente retorna False. ... [ERROR] Hotel 'hotell' no encontrado.
ok

test_mostrar_hotel (__main__.TestHotel.test_mostrar_hotel)
Verifica que display no lanza excepción. ... Hotel : H1
Nombre : Te big apple
Ubicación : NYC
Habitaciones : 10
Ok

`test_mostrar_hotel_inexistente (__main__.TestHotel.test_mostrar_hotel_inexistente)`
Verifica que display con ID inexistente no lanza excepción. ... [ERROR] Hotel 'hotell1' no encontrado.
`ok`

`test_obtener_hotel (__main__.TestHotel.test_obtener_hotel)`
Verifica que se puede obtener un hotel por ID. ... `ok`

`test_obtener_hotel_inexistente (__main__.TestHotel.test_obtener_hotel_inexistente)`
Verifica que obtener un hotel inexistente retorna None. ... [ERROR] Hotel 'hotell1' no encontrado.
`ok`

`test_cancelar_reservacion (__main__.TestReservation.test_cancelar_reservacion)`
Verifica que se puede cancelar una reservación activa. ... `ok`

`test_cancelar_reservacion_inexistente (__main__.TestReservation.test_cancelar_reservacion_inexistente)`
Verifica que cancelar una reservación inexistente retorna False. ... [ERROR] Reservación 'RVI' no encontrada.
`ok`

`test_cancelar_reservacion_ya_cancelada (__main__.TestReservation.test_cancelar_reservacion_ya_cancelada)`
Verifica que no se puede cancelar una reservación ya cancelada. ... [ERROR] La reservación 'R1' ya está cancelada.
`ok`

`test_crear_reservacion (__main__.TestReservation.test_crear_reservacion)`
Verifica que se puede crear una reservación correctamente. ... `ok`

`test_crear_reservacion_cliente_inexistente (__main__.TestReservation.test_crear_reservacion_cliente_inexistente)`
Verifica que no se puede reservar con un cliente inexistente. ... [ERROR] Cliente 'customer1' no encontrado.
`ok`

`test_crear_reservacion_duplicada (__main__.TestReservation.test_crear_reservacion_duplicada)`
Verifica que no se puede crear una reservación con ID duplicado. ... [ERROR] La reservación 'R1' ya existe.
`ok`

`test_crear_reservacion_hotel_inexistente (__main__.TestReservation.test_crear_reservacion_hotel_inexistente)`
Verifica que no se puede reservar en un hotel inexistente. ... [ERROR] Hotel 'hotell1' no encontrado.
`ok`

`test_crear_reservacion_sin_habitaciones (__main__.TestReservation.test_crear_reservacion_sin_habitaciones)`
Verifica que no se puede reservar si no hay habitaciones. ... [ERROR] No hay habitaciones disponibles en el hotel 'HHTLL'.
`ok`

`test_habitaciones_disminuyen_al_reservar (__main__.TestReservation.test_habitaciones_disminuyen_al_reservar)`
Verifica que las habitaciones disponibles disminuyen al reservar. ... `ok`

`test_habitaciones_se_restauran_al_cancelar (__main__.TestReservation.test_habitaciones_se_restauran_al_cancelar)`
Verifica que las habitaciones se restauran al cancelar. ... `ok`

`test_json_corrupto_retorna_estructura_vacia`
(`__main__.TestReservation.test_json_corrupto_retorna_estructura_vacia`)
Verifica que un JSON corrupto es manejado sin lanzar excepción. ... [ERROR] No se pudo cargar
`/Users/alejo/Documents/MNA TEC MTY/5TH TRIMESTRE/SOFTWARE QUALITY/Pruuebas-Calidad-Software-A01796714/A01796714_A6.2/test/unit/../tc1.json: Expecting property name enclosed in double quotes: line 1`
`column 2 (char 1)`
[ERROR] Cliente 'C1' no encontrado.
`Ok`

```
test_mostrar_reservacion (__main__.TestReservation.test_mostrar_reservacion)
Verifica que display no lanza excepción. ... Reservación : R1
Cliente    : CI
Hotel      : HI
Check-in   : 10/04/2026
Check-out  : 25/04/2026
Estado     : activa
ok

test_mostrar_reservacion_inexistente (__main__.TestReservation.test_mostrar_reservacion_inexistente)
Verifica que display con ID inexistente no lanza excepción. ... [ERROR] Reservación 'RVI' no encontrada.
ok

test_obtener_reservacion (__main__.TestReservation.test_obtener_reservacion)
Verifica que se puede obtener una reservación por ID. ... ok
test_obtener_reservacion_inexistente (__main__.TestReservation.test_obtener_reservacion_inexistente)
Verifica que obtener una reservación inexistente retorna None. ... [ERROR] Reservación 'RVI' no encontrada.
ok

test_tipo_incorrecto_en_archivo (__main__.TestReservation.test_tipo_incorrecto_en_archivo)
Verifica que un JSON con tipo incorrecto es manejado correctamente. ... [ERROR] No se pudo cargar
/Users/alejo/Documents/MNA TEC MTY/5TH TRIMESTRE/SOFTWARE QUALITY/Pruebas-Calidad-Software-
A01796714/A01796714_A6.2/test/unit/../tc1.json: El archivo de datos debe ser un JSON.
[ERROR] Hotel 'HI' no encontrado.
ok
```

Ran 40 tests in 0.011s

OK

Evidencia de Commits realizados

Commits

History for Pruebas-Calidad-Software-A01796714 / A01796714_A6.2 on main

All users All time

-o- Commits on Feb 21, 2026

- test: Se ejecutan pruebas unitarias y se agrega evidencia de resultados.
OvidioHernandezA01796714 committed 1 minute ago d74bbc7
- style(tests): Se agregan archivos tc1, tc2, tc3 json para realizar pruebas, en donde también se incluyen pruebas con valores negativos.
OvidioHernandezA01796714 committed 14 hours ago c93907f
- style(tests): Implementación de correcciones de pylint y flake8 en .py de pruebas unitarias.
OvidioHernandezA01796714 committed 15 hours ago e7c2144

-o- Commits on Feb 20, 2026

- style!: Aplica correcciones para cumplir con estándares de pylint y flake8. Requeridos para el cumplimiento de la actividad 6.2
OvidioHernandezA01796714 committed 2 days ago 6f1e545
- docs: Se agregan Docstrings y documentación para cada clase y para cada método de las clases.
OvidioHernandezA01796714 committed 2 days ago d973c2b

-o- Commits on Feb 19, 2026

- test: Definición de estructura de pruebas y pruebas unitarias ...
OvidioHernandezA01796714 committed 3 days ago 058072e
- feat(Customer & Reservation)!: Implementación de las clases Customer y Reservation.
OvidioHernandezA01796714 committed 3 days ago 48cefa3

-o- Commits on Feb 18, 2026

- feat(Hotel)!: Implementación de la clase Hotel.
OvidioHernandezA01796714 committed 4 days ago 2cb45d4

-o- End of commit history for this file