



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

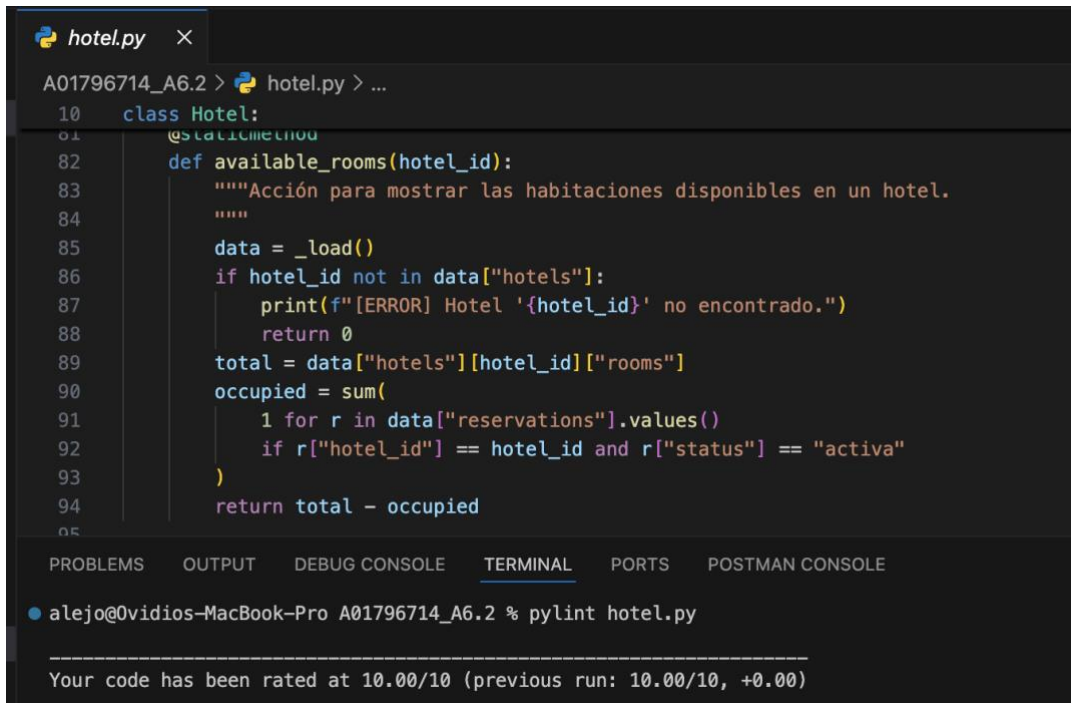
Alumno

A01796714 Ovidio Alejandro Hernández Ruano

Febrero 22, 2026

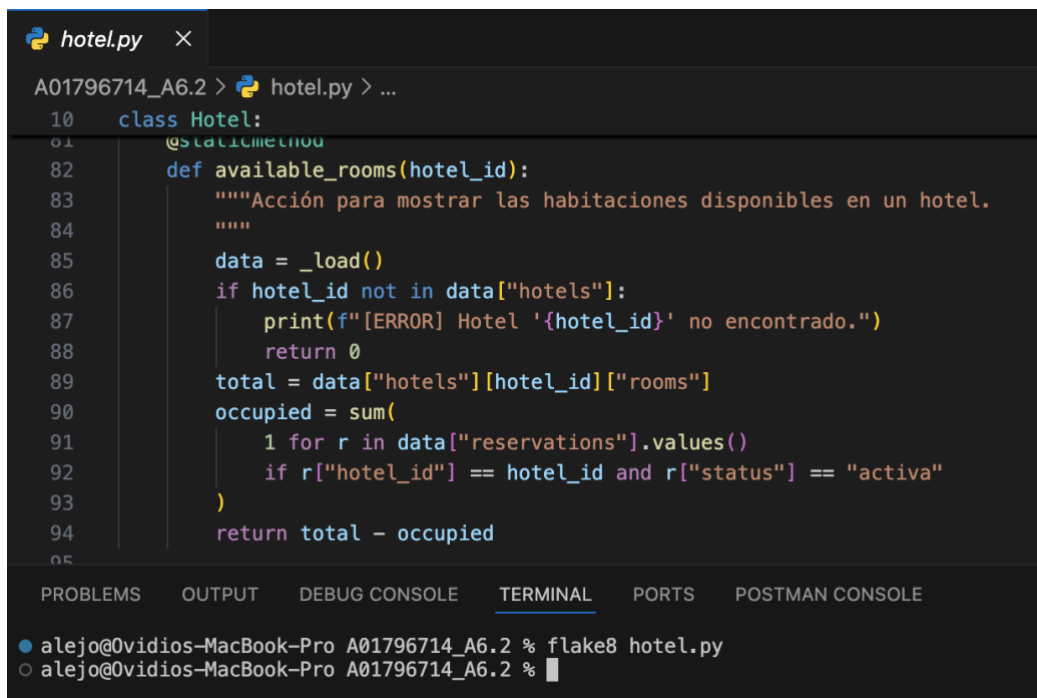
Ejercicio de programación 3

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



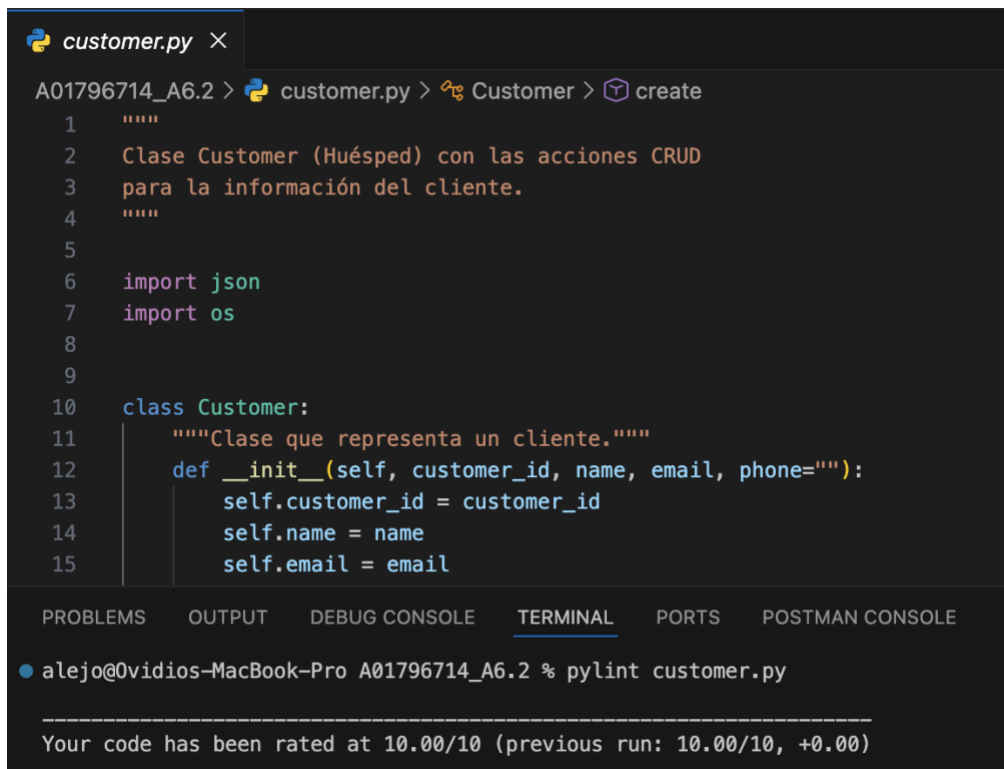
```
hotel.py x
A01796714_A6.2 > hotel.py > ...
10 class Hotel:
11     @staticmethod
12     def available_rooms(hotel_id):
13         """Acción para mostrar las habitaciones disponibles en un hotel.
14         """
15         data = _load()
16         if hotel_id not in data["hotels"]:
17             print(f"[ERROR] 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@0vidios-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.



```
hotel.py x
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@0vidios-MacBook-Pro A01796714_A6.2 % flake8 hotel.py
○ alejo@0vidios-MacBook-Pro A01796714_A6.2 %
```

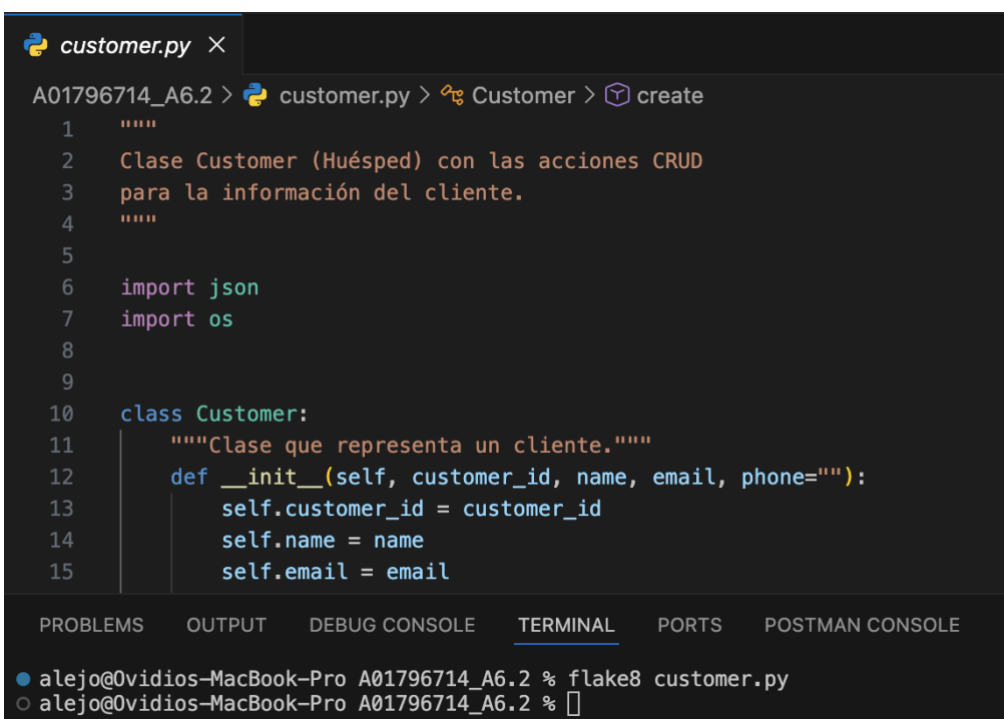
Evidencia de la correcta ejecución de *pylint*, aplicado al archivo customer.py que contiene la clase Customer (Huesped).



```
customer.py X
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)
```

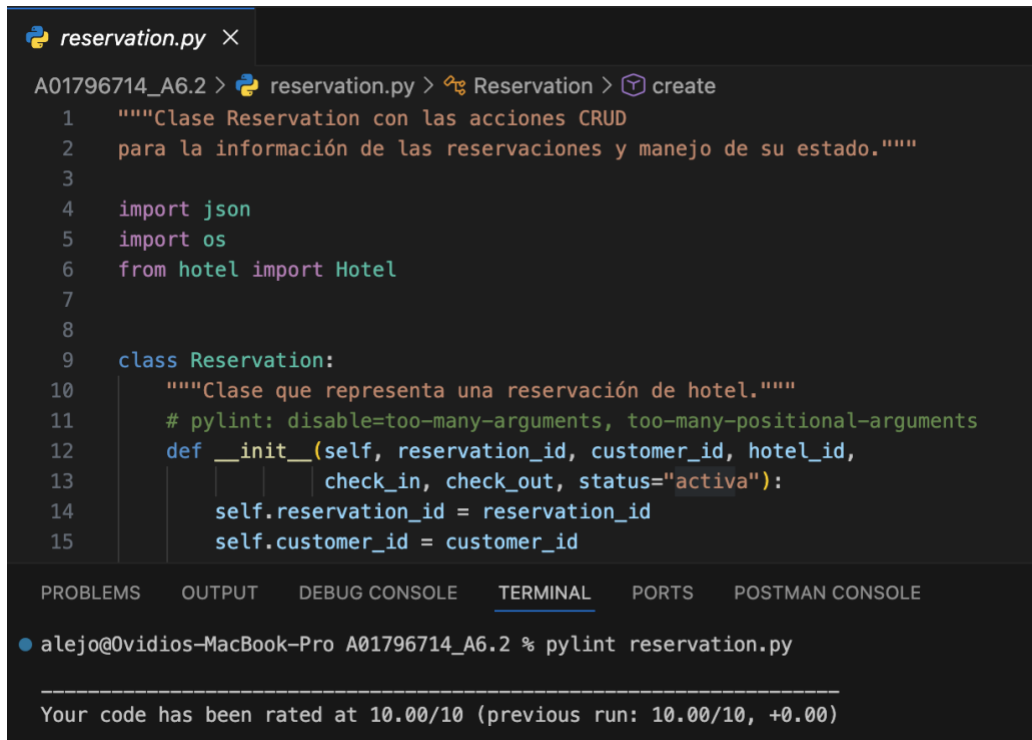
Evidencia de la correcta ejecución de *flake8*, aplicado al archivo customer.py que contiene la clase Customer (Huesped).



```
customer.py X
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.



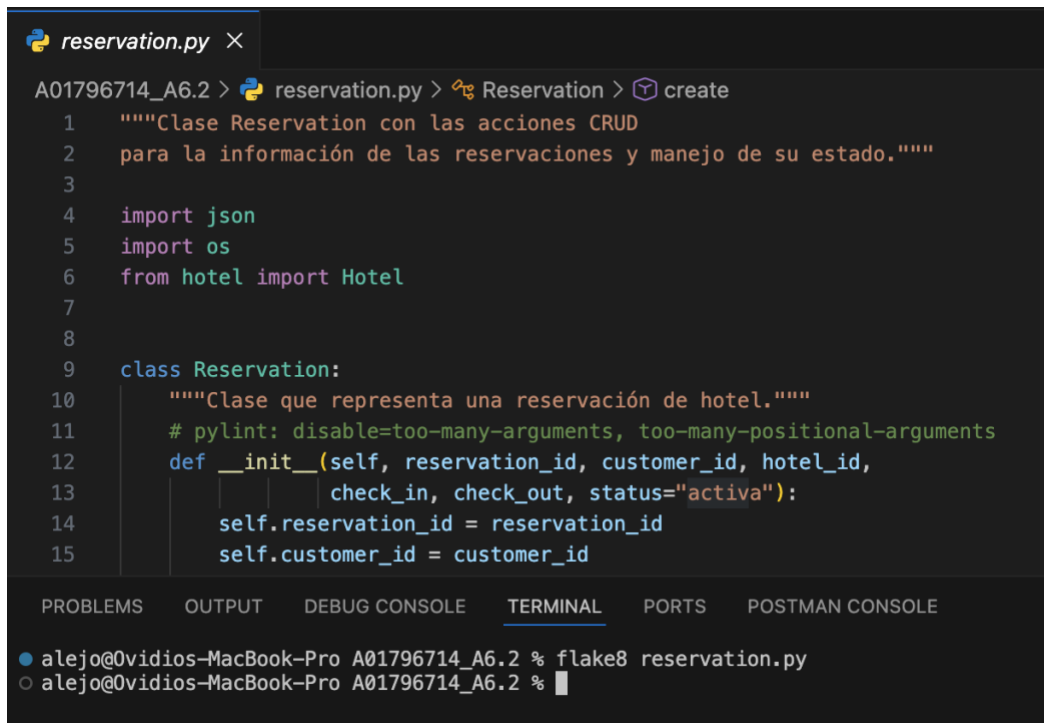
```
reservation.py X
A01796714_A6.2 > reservation.py > Reservation > create
1  """Clase Reservation con las acciones CRUD
2  para la información de las reservaciones y manejo de su estado."""
3
4  import json
5  import os
6  from hotel import Hotel
7
8
9  class Reservation:
10     """Clase que representa 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.



```
reservation.py X
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"
```

JSON Escenario de prueba TC2

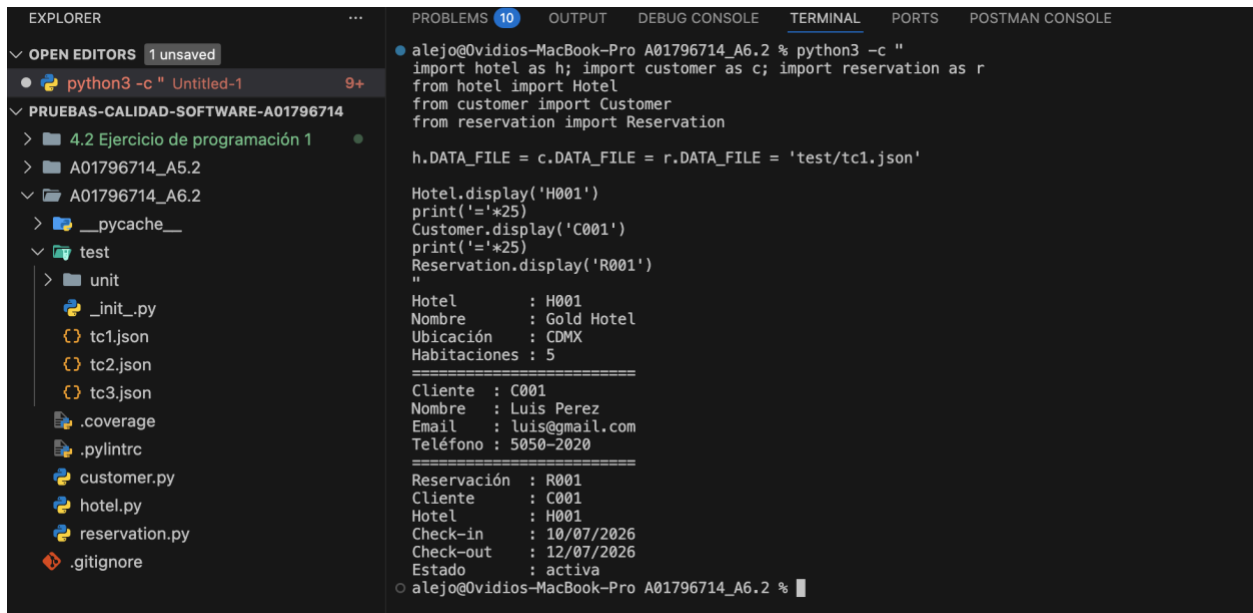
```
tc2.json ×
A01796714_A6.2 > test > tc2.json > {} hotels
1  {
2    "hotels": {
3      "H001": {
4        "hotel_id": "H111",
5        "name": "Hotel Independencia",
6        "location": "CDMX",
7        "rooms": 5
8      },
9      "H002": {
10       "hotel_id": "H450",
11       "name": "Hotel Sultana",
12       "location": "Monterrey",
13       "rooms": 2
14     },
15     "H003": {
16       "hotel_id": "H7346",
17       "name": "Hotel Pacifico",
18       "location": "Baja California",
19       "rooms": 1
20     }
21   },
22   "customers": {
23     "C001": {
24       "customer_id": "C123",
25       "name": "Juan Rojas",
26       "email": "juan.rojas@gmail.com",
27       "phone": "5252-2120"
28     },
29     "C002": {
30       "customer_id": "C985",
31       "name": "Alberto Sanchez",
32       "email": "alberto.sanchez@gmail.com",
33       "phone": "1236-7852"
34     },
35     "C003": {
36       "customer_id": "C875",
37       "name": "Carlos Ruiz",
38       "email": "carlos.ruiz@gmail.com",
39       "phone": "3312-9900"
40     },
41     "C004": {
42       "customer_id": "C004"
```

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"
```

Ejecución por script del escenario de prueba TC1

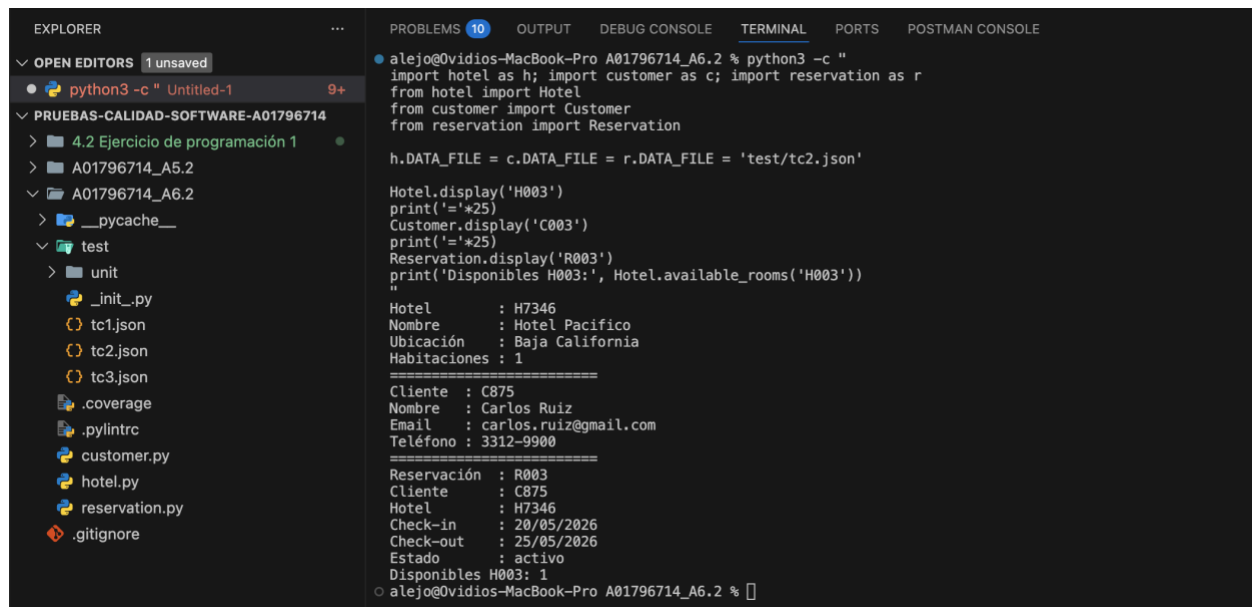
```
python3 -c "  
import hotel as h; import customer as c; import reservation as r  
from hotel import Hotel  
from customer import Customer  
from reservation import Reservation  
  
h.DATA_FILE = c.DATA_FILE = r.DATA_FILE = 'test/tc1.json'  
  
Hotel.display('H001')  
print('='*25)  
Customer.display('C001')  
print('='*25)  
Reservation.display('R001')  
"
```



```
EXPLORER 1 unsaved  
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  
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@gmail.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

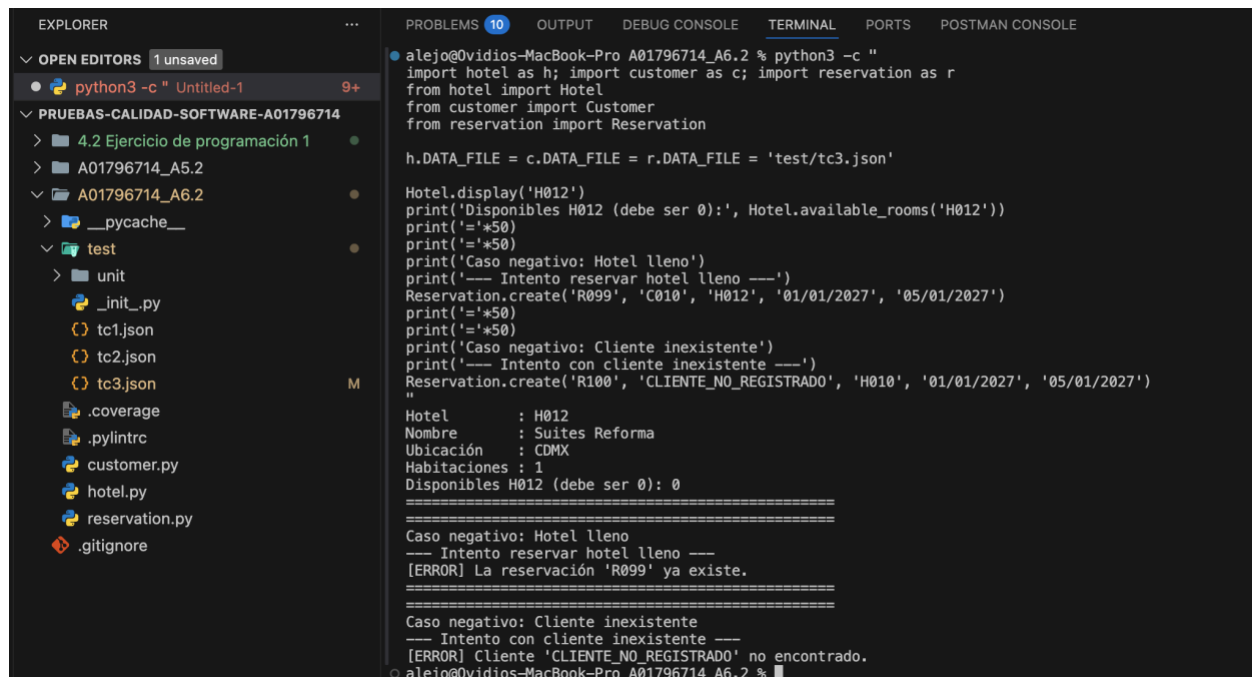
```
python3 -c "  
import hotel as h; import customer as c; import reservation as r  
from hotel import Hotel  
from customer import Customer  
from reservation import Reservation  
  
h.DATA_FILE = c.DATA_FILE = r.DATA_FILE = 'test/tc2.json'  
  
Hotel.display('H003')  
print('='*25)  
Customer.display('C003')  
print('='*25)  
Reservation.display('R003')  
print('Disponibles H003:', Hotel.available_rooms('H003'))  
"
```



```
EXPLORER  
...  
OPEN EDITORS 1 unsaved  
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  
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/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')  
"
```

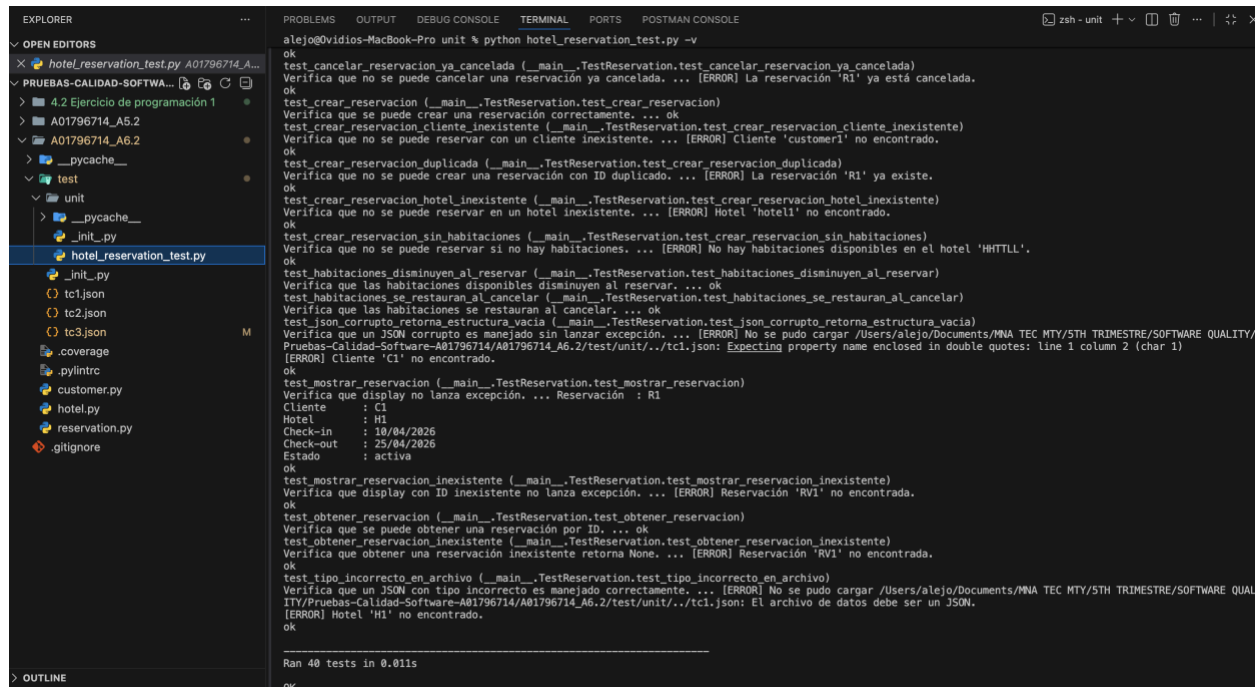


```
EXPLORER 1 unsaved  
python3 -c " Untitled-1 9+  
PRUEBAS-CALIDAD-SOFTWARE-A01796714  
4.2 Ejercicio de programación 1  
A01796714_A5.2  
A01796714_A6.2  
__pycache__  
test  
unit  
_init_.py  
tc1.json  
tc2.json  
tc3.json M  
.coverage  
.pylintrc  
customer.py  
hotel.py  
reservation.py  
.gitignore  
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/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')  
"  
  
Hotel : H012  
Nombre : Suites Reforma  
Ubicación : CDMX  
Habitaciones : 1  
Disponibles H012 (debe ser 0): 0  
=====
```

```
Caso negativo: Hotel lleno  
--- Intento reservar hotel lleno ---  
[ERROR] La reservación 'R099' ya existe.  
=====
```

```
Caso negativo: Cliente inexistente  
--- Intento con cliente inexistente ---  
[ERROR] Cliente 'CLIENTE_NO_REGISTRADO' no encontrado.  
alejo@Ovidios-MacBook-Pro A01796714_A6.2 %
```

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



```
alejo@Ovidios-MacBook-Pro unit % 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 'hotel1' 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 'HHTTLL'.
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/Pruebas-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      : C1
Hotel       : H1
Check-in    : 10/04/2026
Check-out   : 25/04/2026
Estado      : activo
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/./tc1.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 'hotel1' 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 'hotel1' 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 'hotel1' 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 'hotel1' 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 'hotel1' 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 'RV1' 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 'hotel1' 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 'HHTTLL'.
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/Pruebas-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 : 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/./tc1.json: El archivo de datos debe ser un JSON.

[ERROR] Hotel 'H1' no encontrado.

ok

Ran 40 tests in 0.011s

OK

Cobertura de líneas por clase

Esta captura de pantalla demuestra el porcentaje de cobertura de líneas por cada una de las clases desarrolladas, con promedio de 97%.

Clase Customer: 92%

Clase Hotel: 98%

Clase Reservación: 95%

EXPLORER

...

OPEN EDITORS

hotel_reservation_test.py A01796714_A...

PRUEBAS-CALIDAD-SOFTWA...

4.2 Ejercicio de programación 1

A01796714_A5.2

A01796714_A6.2

__pycache__

Documentation

test

unit

__pycache__

init.py

.coverage

hotel_reservation_test.py

init.py

tc1.json

tc2.json

tc3.json

.coverage

.pylintrc

customer.py

hotel.py

reservation.py

.gitignore

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

POSTMAN CONSOLE

zsh - A01796714_A6.2

+ v

🗑

...

🔍

✕

alejo@Ovidios-MacBook-Pro A01796714_A6.2 % coverage run -m unittest discover -s test/unit -p "hotel_reservation_test.py"

..[ADVERTENCIA] El campo 'campo_invalido' no es modificable.

..[ERROR] Cliente 'customer1' no encontrado.

..Cliente : C1

Nombre : Mario Jimenez

Email : mjim@gmail.com

Teléfono : ..

..[ERROR] Cliente 'customer1' no encontrado.

..[ERROR] Cliente 'customer1' no encontrado.

..[ERROR] El hotel 'H1' ya existe.

..[ERROR] Hotel 'H1' no encontrado.

..[ERROR] Hotel 'hotel1' no encontrado.

..[ERROR] Hotel 'hotel1' no encontrado.

..[ADVERTENCIA] El campo 'campo_invalido' no es modificable.

..[ERROR] Hotel 'hotel1' no encontrado.

..Hotel : H1

Nombre : Te big apple

Ubicación : NYC

Habitaciones : 10

..[ERROR] Hotel 'hotel1' no encontrado.

..[ERROR] Hotel 'hotel1' no encontrado.

..[ERROR] Reservación 'RV1' no encontrada.

..[ERROR] La reservación 'R1' ya está cancelada.

..[ERROR] Cliente 'customer1' no encontrado.

..[ERROR] La reservación 'R1' ya existe.

..[ERROR] Hotel 'hotel1' no encontrado.

..[ERROR] No hay habitaciones disponibles en el hotel 'HMTLLI'.

...[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: Expecting property name enclosed in double quotes: line 1 column 2 (char 1)

..[ERROR] Cliente 'C1' no encontrado.

..Reservación : R1

Cliente : C1

Hotel : H1

Check-in : 18/04/2026

Check-out : 25/04/2026

Estado : activa

..[ERROR] Reservación 'RV1' no encontrada.

..[ERROR] Reservación 'RV1' no encontrada.

..[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.j

..son: El archivo de datos debe ser un JSON.

..[ERROR] Hotel 'H1' no encontrado.

..

Ran 40 tests in 0.012s

OK

alejo@Ovidios-MacBook-Pro A01796714_A6.2 % coverage report -m

Name	Stmts	Miss	Cover	Missing
customer.py	74	6	92%	93, 95-97, 105-106
hotel.py	83	2	98%	120-121
reservation.py	78	4	95%	102, 107, 119-120
test/unit/hotel_reservation_test.py	163	1	99%	300
TOTAL	398	13	97%	

Evidencia de Commits realizados

Commits

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

All usersAll time

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

e7c2f44

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

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

Commits on Feb 18, 2026

feat(Hotel)!: Implementación de la clase Hotel.

OvidioHernandezA01796714 committed 4 days ago

2cb45d4

End of commit history for this file