

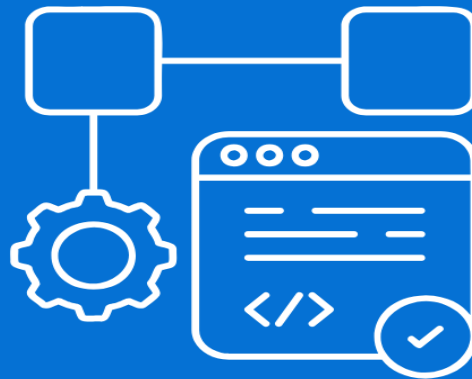


Tecnológico  
de Monterrey

# PRUEBAS DE SOFTWARE Y ASEGURAMIENTO DE LA CALIDAD

TC4017.10

PRUEBAS DE UNIDAD



Jhamyr Arnulfo Alcalde Oballe

A01795401

## Sistema Hotel

1. Pruebas de funcionamiento del sistema realizado en python.

### Ejecuciones:

main\_1.py

```
na 6/Tarea/hotel_system ➤ python3 main_1.py

✖ Creando hoteles en Perú...
✖ Creando clientes peruanos...
✖ Realizando reservas...

♦ Lista de Hoteles después de reservas:
1: JW Marriott - Lima (18 habitaciones)
2: Belmond Miraflores Park - Lima (12 habitaciones)

♦ Lista de Clientes:
101: Luis Rodríguez - luis.rodriguez@email.com
102: María Pérez - maria.perez@email.com

♦ Lista de Reservas:
Reserva 2001: Cliente 101 - Hotel 1
Reserva 2002: Cliente 102 - Hotel 2
```

main\_2.py

```
na 6/Tarea/hotel_system ➤ python3 main_2.py

✖ Creando hoteles en Perú...
✖ Creando clientes peruanos...
✖ Modificando el hotel 'Casa Andina Premium' en Arequipa...
✖ Eliminando hotel 'Hotel Libertador' en Cusco...
✖ Eliminando cliente Andrea Torres...
101: Luis Rodríguez - luis.rodriguez@email.com
102: María Pérez - maria.perez@email.com
103: Carlos Gómez - carlos.gomez@email.com
104: Andrea Torres - andrea.torres@email.com
104: Andrea Torres - andrea.torres@email.com
101: Luis Rodríguez - luis.rodriguez@email.com
102: María Pérez - maria.perez@email.com
103: Carlos Gómez - carlos.gomez@email.com
104: Andrea Torres - andrea.torres@email.com
104: Andrea Torres - andrea.torres@email.com
103: Carlos Gómez - carlos.gomez@email.com

♦ Lista de Hoteles después de eliminación:
1: JW Marriott - Lima (18 habitaciones)
2: Belmond Miraflores Park - Lima (12 habitaciones)
3: Casa Andina Arequipa - Centro Histórico, Arequipa (20 habitaciones)

♦ Lista de Clientes después de eliminación:
101: Luis Rodríguez - luis.rodriguez@email.com
102: María Pérez - maria.perez@email.com
103: Carlos Gómez - carlos.gomez@email.com
104: Andrea Torres - andrea.torres@email.com
104: Andrea Torres - andrea.torres@email.com
103: Carlos Gómez - carlos.gomez@email.com
```

## main\_3.py

```
na 6/Tarea/hotel_system ➤ python3 main_3.py
✖ Creando hoteles en Perú...
✖ Intentando reservar más habitaciones de las disponibles...
Error: No hay suficientes habitaciones disponibles.
✖ Corrector No se permitió la reserva.
✖ Intentando reservar en un hotel que no existe...
Traceback (most recent call last):
  File "/Users/jhamyr/calde/Library/CloudStorage/GoogleDrive-jhamyr/calde@gmail.com/My Drive/TEC - IA APLICADA/TEC - MAESTRIA - IA/Pruebas de software y aseguramiento de calidad/Semana 6/Tarea
/hotel_system/main_3.py", line 19, in <module>
    if not Hotel.reserve_room(99, 2): # No existe hotel 99
  File "/Users/jhamyr/calde/Library/CloudStorage/GoogleDrive-jhamyr/calde@gmail.com/My Drive/TEC - IA APLICADA/TEC - MAESTRIA - IA/Pruebas de software y aseguramiento de calidad/Semana 6/Tarea
/hotel_system/hotel_system.py", line 86, in reserve_room
    raise ValueError(f'Error: Hotel con ID {hotel_id} no encontrado.') # Asegurar que se lanza el ValueError
ValueError: Error: Hotel con ID 99 no encontrado.
```

## main\_4.py

```
na 6/Tarea/hotel_system ➤ python3 main_4.py
✖ Creando hoteles en Perú...
✖ Creando clientes peruanos...
✖ Realizando reservas en Inkaterria Machu Picchu...
♦ Lista de Reservas antes de la cancelación:
Reserva 2001: Cliente 101 - Hotel 1
Reserva 2002: Cliente 102 - Hotel 2
Reserva 3001: Cliente 999 - Hotel 5
Reserva 4001: Cliente 105 - Hotel 6
Reserva 4002: Cliente 106 - Hotel 6
✖ Cancelando la reserva 4001 y 4002...
✔ Reserva 4001 cancelada correctamente.
✔ Reserva 4002 cancelada correctamente.
♦ Lista de Reservas después de la cancelación:
Reserva 2001: Cliente 101 - Hotel 1
Reserva 2002: Cliente 102 - Hotel 2
Reserva 3001: Cliente 999 - Hotel 5
♦ Lista de Hoteles después de cancelaciones (verificar habitaciones disponibles):
1: JW Marriott - Lima (18 habitaciones)
2: Belmond Miraflores Park - Lima (12 habitaciones)
3: Casa Andina Arequipa - Centro Histórico, Arequipa (20 habitaciones)
5: Tambo del Inka - Urubamba (5 habitaciones)
6: Inkaterria Machu Picchu - Machu Picchu (5 habitaciones)
```

## 2. Ejecución de pruebas con unittest.

```
na 6/Tarea/hotel_system ➤ python3 -m unittest test_hotel_system.py
✔ Reserva 3001 cancelada correctamente. Habitaciones restauradas: 1
.....
Ran 10 tests in 0.029s
OK
jhamyr/calde@Jhamyr-MacBook-Air ~/Library/CloudStorage/GoogleDrive-
na 6/Tarea/hotel_system ➤
```

## 3. Aplicamos coverage.

```
na 6/Tarea/hotel_system ➤ python3 -m coverage run --source=hotel_system -m unittest test_hotel_system.py
⚠ No se encontró la reserva 999.
✔ Reserva 3001 cancelada correctamente. Habitaciones restauradas: 1
.....Error: No hay suficientes habitaciones disponibles.
...
Ran 15 tests in 0.045s
OK
```

Reporte:

```
na 6/Tarea/hotel_system python3 -m coverage report -m
Name           Stmts  Miss  Cover   Missing
-----
hotel_system.py 151    15    90%    91-93, 128-129, 151-153, 188-189, 203, 222, 233-235
TOTAL           151    15    90%
jhamyralcalde@Jhamyrs-MacBook-Air ~/Library/CloudStorage/GoogleDrive-jhamyralcalde@gmail.com/My Drive/
na 6/Tarea/hotel_system
```