

Función Principal

```
# Función principal
def main():
    """
    Función principal que crea y manipula hoteles, clientes y reservas.
    """

    # Nos aseguramos de que existan los archivos JSON (con diccionarios vacíos)
    for filename in ["hotels.json", "customers.json", "reservations.json"]:
        if not os.path.exists(filename):
            with open(filename, 'w', encoding='utf-8') as f:
                json.dump({}, f)

    # 1. Crea un Hotel
    mi_hotel = Hotel.create_hotel("Grand Plaza", "New York", total_rooms=10)
    print("Hotel creado:", mi_hotel)

    # 2. Muestra la información del Hotel
    hotel_obtenido = Hotel.display_hotel_info(mi_hotel.hotel_id)
    print("Información del Hotel:", hotel_obtenido)

    # 3. Crea un Cliente
    mi_cliente = Customer.create_customer("John Doe", "john@example.com")
    print("Cliente creado:", mi_cliente)

    # 4. Crea una Reservación
    fecha_entrada = datetime(2025, 3, 1)
    fecha_salida = datetime(2025, 3, 5)
    mi_reserva = Reservation.create_reservation(
        customer_id=mi_cliente.customer_id,
        hotel_id=mi_hotel.hotel_id,
        check_in=fecha_entrada,
        check_out=fecha_salida
    )
    print("Reserva creada:", mi_reserva)

    # 5. Cancela la Reservación
    if mi_reserva:
        exito_cancelar = Reservation.cancel_reservation(
            mi_reserva.reservation_id
        )
        print("Reserva cancelada:", exito_cancelar)

    # 6. Modifica el Hotel
    Hotel.modify_hotel_information(
        mi_hotel.hotel_id,
        name="Grand Plaza - Renovado",
        total_rooms=12
    )
    hotel_modificado = Hotel.display_hotel_info(mi_hotel.hotel_id)
    print("Hotel modificado:", hotel_modificado)

    # 7. Elimina el Hotel
    exito_eliminar_hotel = Hotel.delete_hotel(mi_hotel.hotel_id)
    print("Hotel eliminado:", exito_eliminar_hotel)

    # 8. Elimina el Cliente
    exito_eliminar_cliente = Customer.delete_customer(mi_cliente.customer_id)
    print("Cliente eliminado:", exito_eliminar_cliente)

if __name__ == "__main__":
    main()
```

```

PS C:\Users\Oscar\Documents\Universidad Posgrado Inteligencia Artificial Aplicada\Cuarto trimestre\Software> python .\A01323987_A6_2.py
Hotel creado: Hotel(hotel_id=1, name='Grand Plaza', location='New York', total_rooms=10, booked_rooms=0)
{'hotel_id': 1, 'name': 'Grand Plaza', 'location': 'New York', 'total_rooms': 10, 'booked_rooms': 0}
Información del Hotel: Hotel(hotel_id=1, name='Grand Plaza', location='New York', total_rooms=10, booked_rooms=0)
Cliente creado: Customer(customer_id=1, name='John Doe', email='john@example.com')
{'customer_id': 1, 'name': 'John Doe', 'email': 'john@example.com'}
{'hotel_id': 1, 'name': 'Grand Plaza', 'location': 'New York', 'total_rooms': 10, 'booked_rooms': 0}
Reserva creada: Reservation(reservation_id=1, customer_id=1, hotel_id=1, check_in='2025-03-01', check_out='2025-03-05', is_active=True)
Reserva cancelada: True
{'hotel_id': 1, 'name': 'Grand Plaza - Renovado', 'location': 'New York', 'total_rooms': 12, 'booked_rooms': 0}
Hotel modificado: Hotel(hotel_id=1, name='Grand Plaza - Renovado', location='New York', total_rooms=12, booked_rooms=0)
Hotel eliminado: True
Cliente eliminado: True
PS C:\Users\Oscar\Documents\Universidad Posgrado Inteligencia Artificial Aplicada\Cuarto trimestre\Software> |

```

Unit Test

```

Windows PowerShell
PS C:\Users\Oscar\Documents\Universidad Posgrado Inteligencia Artificial Aplicada\Cuarto trimestre\Software> python .\A01323987_A6_2.py
Hotel creado: Hotel(hotel_id=1, name='Grand Plaza', location='New York', total_rooms=10, booked_rooms=0)
{'hotel_id': 1, 'name': 'Grand Plaza', 'location': 'New York', 'total_rooms': 10, 'booked_rooms': 0}
Información del Hotel: Hotel(hotel_id=1, name='Grand Plaza', location='New York', total_rooms=10, booked_rooms=0)
Cliente creado: Customer(customer_id=1, name='John Doe', email='john@example.com')
{'customer_id': 1, 'name': 'John Doe', 'email': 'john@example.com'}
{'hotel_id': 1, 'name': 'Grand Plaza', 'location': 'New York', 'total_rooms': 10, 'booked_rooms': 0}
Reserva creada: Reservation(reservation_id=2, customer_id=1, hotel_id=1, check_in='2025-03-01', check_out='2025-03-05', is_active=True)
Reserva cancelada: True
{'hotel_id': 1, 'name': 'Grand Plaza - Renovado', 'location': 'New York', 'total_rooms': 12, 'booked_rooms': 0}
Hotel modificado: Hotel(hotel_id=1, name='Grand Plaza - Renovado', location='New York', total_rooms=12, booked_rooms=0)
Hotel eliminado: True
Cliente eliminado: True
PS C:\Users\Oscar\Documents\Universidad Posgrado Inteligencia Artificial Aplicada\Cuarto trimestre\Software> python .\unit_test.py
test_create_customer (__main__.TestCustomer.test_create_customer)
Prueba la creación de un cliente y verifica sus atributos. ... ok
test_delete_customer (__main__.TestCustomer.test_delete_customer)
Prueba la eliminación de un cliente y verifica que no exista luego. ... None
ok
test_display_customer_info (__main__.TestCustomer.test_display_customer_info)
Prueba la obtención de la información de un cliente recién creado. ... {'customer_id': 1, 'name': 'John', 'email': 'john@example.com'}
ok
test_modify_customer_information (__main__.TestCustomer.test_modify_customer_information)
Prueba la modificación de la información de un cliente. ... {'customer_id': 1, 'name': 'NombreNuevo', 'email': 'nuevo@correo.com'}
ok
test_cancel_room_reservation (__main__.TestHotel.test_cancel_room_reservation)
Prueba la cancelación de una habitación reservada. ... {'hotel_id': 1, 'name': 'Hotel Cancelación', 'location': 'Ciudad Cancelación', 'total_rooms': 2, 'booked_rooms': 1}
ok
test_create_hotel (__main__.TestHotel.test_create_hotel)
Prueba la creación de un hotel y verifica sus atributos por defecto. ... ok
test_delete_hotel (__main__.TestHotel.test_delete_hotel)
Prueba la eliminación de un hotel ... None
ok
test_display_hotel_info (__main__.TestHotel.test_display_hotel_info)
Prueba la función de mostrar información para un hotel recién creado. ... {'hotel_id': 1, 'name': 'Hotel para Mostrar', 'location': 'Ciudad Muestra', 'total_rooms': 10, 'booked_rooms': 0}
ok
test_modify_hotel_information (__main__.TestHotel.test_modify_hotel_information)
Prueba la modificación de la información de un hotel existente. ... {'hotel_id': 1, 'name': 'Nombre Nuevo', 'location': 'Ciudad Nueva', 'total_rooms': 12, 'booked_rooms': 0}
ok
test_reserve_room (__main__.TestHotel.test_reserve_room)
Prueba la acción de reservar habitaciones ... ok
test_cancel_reservation (__main__.TestReservation.test_cancel_reservation)
Prueba la cancelación de una reservación activa, ... {'customer_id': 1, 'name': 'CancelarRes User', 'email': 'cancelres@example.com'}

```

```
Windows PowerShell x + -
test_reserve_room (__main__.TestHotel.test_reserve_room)
Prueba la acción de reservar habitaciones ... ok
test_reserve_room_invalid_hotel_id (__main__.TestHotel.test_reserve_room_invalid_hotel_id)
Intenta reservar una habitación usando un hotel_id inexistente (9999). ...
::: Se espera un mensaje de error a continuación: hotel_id inexistente :::
ok
test_cancel_reservation (__main__.TestReservation.test_cancel_reservation)
Prueba la cancelación de una reservación activa, ... {'customer_id': 1, 'name': 'CancelarRes User', 'email': 'cancelres@example.com'}
{'hotel_id': 1, 'name': 'CancelarRes Hotel', 'location': 'CancelarRes City', 'total_rooms': 1, 'booked_rooms': 0}
{'hotel_id': 1, 'name': 'CancelarRes Hotel', 'location': 'CancelarRes City', 'total_rooms': 1, 'booked_rooms': 0}
ok
test_create_reservation (__main__.TestReservation.test_create_reservation)
Prueba la creación de una reservación válida y verifica que el hotel ... {'customer_id': 1, 'name': 'ResTest User', 'email': 'resuser@example.com'}
{'hotel_id': 1, 'name': 'ResTest Hotel', 'location': 'ResTest City', 'total_rooms': 2, 'booked_rooms': 0}
{'hotel_id': 1, 'name': 'ResTest Hotel', 'location': 'ResTest City', 'total_rooms': 2, 'booked_rooms': 1}
ok
test_create_reservation_invalid_customer (__main__.TestReservation.test_create_reservation_invalid_customer)
Prueba la creación de una reservación con un cliente inexistente. ... ::: Se espera un mensaje de error a continuación, debido a un cliente inexistente :::
None
{'hotel_id': 1, 'name': 'Solo Hotel', 'location': 'Ciudad Sola', 'total_rooms': 1, 'booked_rooms': 0}
Error: Cliente o hotel inválido.
ok
test_create_reservation_invalid_hotel (__main__.TestReservation.test_create_reservation_invalid_hotel)
Prueba la creación de una reservación con un hotel inexistente. ... ::: Se espera un mensaje de error a continuación, debido a un hotel inexistente :::
{'customer_id': 1, 'name': 'SinHotel User', 'email': 'nohotel@example.com'}
None
Error: Cliente o hotel inválido.
ok
test_create_reservation_no_rooms_available (__main__.TestReservation.test_create_reservation_no_rooms_available)
Prueba la creación de reservaciones en un hotel con solo 1 habitación. ... ::: Se espera un mensaje de error a continuación, por falta de habitaciones :::
{'customer_id': 1, 'name': 'Usuario Uno', 'email': 'uno@example.com'}
{'hotel_id': 1, 'name': 'Pequeño Hotel', 'location': 'Mini Ciudad', 'total_rooms': 1, 'booked_rooms': 0}
{'customer_id': 1, 'name': 'Usuario Uno', 'email': 'uno@example.com'}
{'hotel_id': 1, 'name': 'Pequeño Hotel', 'location': 'Mini Ciudad', 'total_rooms': 1, 'booked_rooms': 1}
Error: No hay habitaciones disponibles.
ok

-----
Ran 17 tests in 0.204s

OK

-----
PRUEBAS EXITOSAS: 17 de 17

-----
PS C:\Users\Oscar\Documents\Universidad Posgrado Inteligencia Artificial Aplicada\Cuarto trimestre\Software> Z
```