

Trabajo de promoción

Programación I.

Integrantes:

Franco Tobar,

Castillo Facundo Matias.

Sistema de Reservas de un Hotel.

El presente proyecto consiste en el desarrollo de un sistema de gestión para un hotel, cuya finalidad es administrar reservas, habitaciones, clientes y empleados mediante una interfaz de diálogo utilizando JOptionPane. Se busca simular el funcionamiento real de un hotel, con operaciones separadas por rol (cliente/empleado) y validaciones de identidad para operar en el sistema.

El sistema cuenta con persistencia temporal (en memoria), control de errores, uso de estructuras de array, uso de clases independientes siguiendo principios de la programación orientada a objetos, tratando de optimizar al maximo con herencia y polimorfismo.



1. Roles

El sistema se divide en dos perfiles principales:

Cliente: puede hacer reservas, consultar sus reservas y cancelarlas.

Empleado: accede a todas las operaciones administrativas, previa autenticación con contraseña del sistema (En el caso del main, "admin").

2. Estructura general del sistema

El sistema está compuesto por las siguientes clases principales:

- **Persona (superclass)**
 - **Cliente**
 - **Empleado**
- **Habitacion (superclass)**
 - **HabitacionPremium**
 - **HabitacionEconomica**
- **Reserva (superclass)**
 - **ReservaIndividual**
 - **ReservaBusiness**
- **Hotel:** clase central que gestiona todos los datos y operaciones.
- **main:** clase con el menú principal, funciones de navegación y control de flujo.

3. SEGURIDAD

Dentro de las características que podría tener un programa a nivel de seguridad, no son muchas las que podemos aplicar debido a nuestros conocimientos pero logramos desarrollar una manera que permita solamente acceder al empleado con una determinada contraseña.

*La contraseña en este caso es "admin" (configurable en el constructor de **Hotel** o mediante un set).*

4. MENUS

Menu Cliente

- 1. Hacer una reserva (Individual o Business)***
- 2. Ver mi reserva (por DNI)***
- 3. Cancelar mi reserva (por ID)***
- 4. Salir***

Menu Empleado (con contraseña)

- 1. Alta de cliente***
- 2. Alta de habitación (económica o premium)***
- 3. Alta de reserva***
- 4. Lista de clientes***
- 5. Lista de habitaciones***
- 6. Lista de reservas***

7. *Modificar datos de clientes*

8. *Buscar clientes por DNI*

9. *Ordenar clientes*

10. *Buscar reserva por ID*

11. *Cancelar reserva por ID*

12. *Alta de empleado*

13. *Mostrar empleados*

14. *Buscar empleados por DNI*

15. *Salir*

5. Implementación de Conceptos Clave de Programación Orientada a Objetos (POO)

El desarrollo del sistema estará fundamentado en los principios esenciales de la POO:

Todo el proyecto está estructurado en base a clases que representan entidades del mundo real:

Hotel, Cliente, Empleado, Reserva, Habitación, etc.

Cada clase tiene atributos y comportamientos que definen el estado y las acciones posibles.

Los objetos son instancias de esas clases (por ejemplo, un Cliente específico con nombre y DNI).

Abstracción: Se abstraen las características principales de cada entidad:

- Una **Reserva** sabe sus fechas, habitación, estado, pero no cómo se muestra por pantalla (eso lo hace otra clase).
- La clase **main** maneja el flujo del programa, pero delega todo el trabajo específico a otras clases (buen diseño).

Encapsulamiento: Los atributos están **protegidos** usando modificadores de acceso (**private**), y sólo se accede a ellos a través de métodos públicos (**getters** y **setters**).

Esto permite mantener la integridad de los datos y ocultar la implementación interna:

Herencia: Se reutiliza código y se generalizan comportamientos comunes:

- **Persona** es una clase **abstracta** que sirve de base para **Cliente** y **Empleado**.
- **Habitacion** es la superclase de **HabitacionEconomica** y **HabitacionPremium**.
- **Reserva** es una superclase de **ReservaIndividual** y **ReservaBusiness**.

Esto permite:

- Evitar duplicación de código.
- Tratar diferentes clases como si fueran del mismo tipo base.

.

Polimorfismo: Se pueden usar objetos de distintas subclases como si fueran del tipo padre. Por ejemplo:

Reserva nueva = new ReservaBusiness(); // o ReservaIndividual

Y luego se usa `nueva.mostrarInfo()` sin importar cuál sea la subclase. Además, métodos como `mostrarInfo()` están **sobrescritos** en cada subclase para mostrar diferente comportamiento.

6. Otras características importantes

- Uso de `try-catch` para manejar errores de entrada.
- Validación de repetición de DNI en clientes.
- Validación de habitaciones duplicadas por número.
- Habitaciones marcadas como disponibles/ocupadas.
- Las reservas pueden involucrar múltiples habitaciones o clientes según el tipo.
- Utilización de `JOptionPane` junto a clases personalizadas `Ingreso` y `Salida` para entrada y salida de datos.
- Métodos organizados por responsabilidad dentro de `main`.

7. Opinión de los alumnos.

A lo largo del desarrollo de este proyecto pudimos profundizar y afianzar nuestros conocimientos sobre programación orientada a objetos, aplicando conceptos como herencia, polimorfismo, encapsulamiento y abstracción de manera práctica. Aprendimos a organizar un sistema completo dividiendo las responsabilidades en distintas clases, lo cual me permitió mantener un código más ordenado y reutilizable. También mejoramos la lógica de programación al implementar validaciones, menús dinámicos y manejo de excepciones con `try-catch`. Nos resultó muy útil trabajar con estructuras como

arreglos y métodos personalizados de entrada/salida, lo que me ayudó a entender mejor la interacción con el usuario. En general, considero que este trabajo fue una excelente oportunidad para aplicar lo aprendido en clase y desarrollar un proyecto que simula un caso real, lo que nos motivó aún más a seguir aprendiendo.

Por otro lado, es fantástica la experiencia de codear con un compañero en simultaneo, distribuyendonos tareas y arreglando errores del otro, optimizando código etc.

Franco y Facundo.

Manual de Usuario – Hotel Las Flores

Este manual está diseñado para guiar al usuario en el uso del sistema del Hotel Las Flores. Existen dos tipos de usuario:

- Cliente
- Empleado

Al iniciar el programa, se presenta un menú donde se debe seleccionar el rol correspondiente.

MENÚ INICIAL (Selección de rol)

Al ejecutar el sistema, aparece el siguiente menú:

1. Cliente
2. Empleado
3. Salir

Manual para Clientes

Paso 1: Ingresar como Cliente

Selecciona la opción 1. Esto lo llevará al siguiente menú:

Bienvenido al menú cliente

1. Hacer una reserva

Permite reservar una habitación. El sistema te pedirá tus datos. Al finalizar, confirmará la reserva.

2. Ver mi reserva

Muestra los detalles de tu reserva actual (si existe).

3. Cancelar mi reserva

Eliminar una reserva activa.

4. Salir

Manual para empleados

Paso 1: Ingresar la contraseña para el acceso a la modalidad empleado.

Paso 2: Ingresar como Empleado

Seleccioná la opción 2. Empleado. Verás el siguiente menú:

1. **Alta de Cliente**

Registrar a un nuevo cliente en el sistema.

2. **Alta de Habitación**

Agregar una nueva habitación al hotel.

3. **Alta de Reserva**

Crear una reserva para un cliente desde el sistema.

4. **Lista Clientes**

Mostrar un listado completo de todos los clientes registrados.

5. **Lista Habitaciones**

Ver todas las habitaciones disponibles y sus características.

6. **Lista de Reservas**

Visualizar todas las reservas activas.

7. **Modificar datos de Clientes**

Editar la información de un cliente (por ejemplo, nombre o DNI).

8. **Buscar Clientes por DNI**

Buscar un cliente específico utilizando su número de documento.

9. **Ordenar Clientes**

Ordena los clientes por su apellido

10. **Buscar Reserva**

Buscar reservas existentes por nombre o número de reserva.

11. **Cancelar Reserva**

Eliminar una reserva activa.

12. **Alta empleado**

Crea un empleado para el sistema

13. **Mostrar empleados**

Muestra a los empleados en el sistema del hotel

14. **Buscar empleado**

Busca un empleado en específico a través de su DNI

15. **Salir**

Finaliza la sesión del empleado y vuelve al menú principal.

