



SEGUNDO PARCIAL

PATAGONIA: OASIS Y OCIO

HOTEL POO

Grupos de Trabajo

Deberán formar grupos de cuatro (4) personas, los grupos deberán ser formados por los estudiantes e informarlos a la Cátedra vía mail.

Fecha de Entrega del Temario: 2 de Octubre de 2023 a las 17:00 hrs.

Fecha límite de Entrega: 13 de Noviembre de 2023 a las 23:59 hrs.

Sistema a Implementar

Tu y tu equipo de emprendedores desean crear el hotel “POO” (“Patagonia: Oasis y Ocio”), que estará ubicado en la Patagonia Argentina. Desean crear un sistema informático para la gestión y administración integral del mismo. Los aspectos de la administración incluyen:

1. Gestión del personal:
 - a. Inventario del Personal: Administrativo, Mantenimiento, Limpieza
 - b. Altas y bajas
 - c. Asignación de tareas
 - d. Control de ingreso y egreso
2. Gestión de los clientes:
 - a. Nómina de Clientes del hotel
 - b. Historial de sus reservas
3. Gestión de las reservas:

Asignación de las habitaciones: debe haber 3 tipos de habitaciones con características distintivas: Al menos precio, capacidad y tipo (simples, dobles, Suite y familiar). Con/Sin baño privado. Con/Sin ventana balcón. Los clientes pueden elegir las habitaciones de acuerdo a uno o más criterios.

Informes estadístico:

% de ocupación del hotel.

% de ocupación de acuerdo al tipo de habitación.

Cantidad de clientes por tipo: Debe permitir categorizar al cliente en 3 categorías de acuerdo a su inversión (cuanto gasta entre alojamiento y buffet) en el hotel.

Recaudación diaria

Creación del Hotel:

Deberá modelar el Hotel "POO" que desea crear:

Deberá crear varias habitaciones de distintas categorías y comodidades.

Deberá establecer horarios de check-in y de check-out y la política de cobro.

Deberá crear un buffet. Los clientes deberán poder realizar una orden simple (por ejemplo: "Desayuno", "Bebida", "Refrigerio"). Estas compras son ingresos para el hotel

Requerimientos del Parcial:

Para implementar el programa se debe:

1. Deberá implementar la totalidad de las clases que representen los objetos tangibles e intangibles que modelan el funcionamiento de su hotel. Cada clase deberá tener los atributos y los métodos que les son propios. Deberá establecer el vínculo adecuado entre las distintas clases del sistema..
2. Deberá utilizar todas las estructuras de datos vistas en la materia. Esto es, deberá haber al menos: Una lista enlazada, una pila, una cola, un set y un diccionario. Las estructuras utilizadas deberán ser las adecuadas para cada caso.
3. Deberá hacer uso de archivos de texto para obtener informes. Puede usar archivos CSV siempre que se trate de información que no necesite ser leída por una persona.
4. Podrá realizar la interfaz a buen criterio: Podrá utilizar la consola de salida (línea de comandos) para representar la interacción con los usuarios (mostrar habitaciones a los clientes, hacer preguntas, mostrar opciones, información, etc.). El uso de una interfaz de tipo gráfica (GUI) será optativo.
Si lo implementa por consola, deberá plantearse la lógica mediante un menú orientado a cada tipo de usuario. Cada usuario deberá poder realizar las acciones pertinentes dentro de su submenú.
5. A los fines de esta implementación, se requiere acceso de tres tipos de usuario: El usuario Administrador (quien administra el hotel), Personal administrativo y Clientes.

Para ingresar, todos los usuarios deberán registrarse la primera vez (sign in) e iniciar sesión (log-in). Deberá verificarse que no haya Usuarios repetidos (deberá validarlo mediante los atributos, acordes al tipo de usuario).

6. Hacer las validaciones correspondientes para evitar que el programa salga abruptamente.
- 7.

Criterios de evaluación y aprobación.

Debe hacer las validaciones correspondientes para evitar que el programa salga en forma abrupta

Si al seleccionar alguna opción del Menú, no se tienen resultados, la aplicación lo deberá informar con un mensaje al usuario y seguir funcionando correctamente.

Deberá comentar todos los métodos de las distintas clases creadas. Es recomendable, pero no obligatorio, el uso de “docstrings” (se recomienda investigar).

Además, se valorará que cumpla con las características de eficiencia y estilo. Un programa es eficiente cuando usa los recursos en forma adecuada.

Un programa tiene estilo cuando está escrito de manera clara y simple. Esto incluye respetar las partes del programa, usar variables con nombres significativos, no repetir código innecesariamente, etc.

Legibilidad del código (separación en funciones / archivos, nomenclatura de variables y funciones, etc.).

El código principal debería ser lo más reducido posible, ya que la mayor parte del código se debería encapsular en las clases correspondientes.

Reutilización de código.

Armado de clases.

Observaciones

- Las funciones de validaciones generales pueden estar en un archivo aparte, el cuál se importará cuando fuera necesario.
- No utilizar tildes en el código ni en los datos de entrada en absoluto, para prevenir conflictos de formatos en el código fuente y archivos de datos al pasar la información de un sistema operativo al otro.
- Debe hacer las validaciones correspondientes para evitar que el programa salga en forma abrupta
- Respetar los formatos de entrada y salida especificados.
- Recomendamos testear su programa a conciencia.