**Informe desafío 2.**

**Elaborado por:**

* **Maria Alejandra Gutierrez Rengifo**
* **Juan Camilo Perez**

**Asignatura: Informática 2**

**Docente: Augusto Enrique Salazar Giménez**

**Universidad de Antioquia**

**Facultad de Ingeniería**

**Programa de Ingeniería de Telecomunicaciones Medellín, 2025-1**

1. **Análisis del problema:**

El proyecto consiste en desarrollar un sistema en C++ utilizando Programación Orientada a Objetos para administrar un mercado de estadías hogareñas. El sistema debe gestionar cuatro entidades principales: alojamientos, anfitriones, huéspedes y reservaciones, y permitir operaciones como búsqueda, reserva, anulación y actualización de datos.

El sistema debe asegurar que los alojamientos estén disponibles para las fechas solicitadas y que los huéspedes no tengan reservas que coincidan en esas fechas. Además, se debe validar el acceso de los usuarios según su perfil (huésped o anfitrión) y controlar que solo ciertos roles puedan ejecutar determinadas acciones (por ejemplo, solo huéspedes pueden reservar).

Los datos se cargarán desde archivos que contienen información tanto histórica como vigente, y se requiere separar las reservas activas de las históricas para mantener una gestión eficiente. También es necesario medir el uso de recursos como memoria y cantidad de iteraciones para garantizar un desempeño óptimo.

* **Entidades y relaciones**

El sistema gira en torno a cuatro entidades principales:

**Alojamiento:** Representa las unidades disponibles para renta, caracterizadas por atributos como nombre, código único, ubicación, tipo, precio, amenidades y disponibilidad en fechas específicas.

**Anfitrión:** Persona responsable de uno o más alojamientos, con atributos que incluyen documento de identidad, antigüedad en la plataforma y una puntuación basada en evaluaciones.

**Huésped:** Usuario que realiza reservas, identificado por documento, antigüedad y puntuación. El huésped debe tener un historial de reservaciones sin conflictos de fechas.

**Reservación:** Acción de reservar un alojamiento para un período definido, con atributos como código único, fecha de entrada, duración, monto pagado y observaciones.

Las relaciones entre estas entidades son: cada anfitrión puede gestionar varios alojamientos; cada huésped puede hacer múltiples reservaciones, siempre que las fechas no se superpongan; y cada reservación está asociada a un único alojamiento y huésped.

1. **Consideraciones para la solución propuesta:**

**Diseño orientado a objetos**: Definir clases claras para cada entidad (Alojamiento, Anfitrión, Huésped, Reservación) que incluyan atributos y métodos para manipular sus datos.

**Gestión de datos mediante archivos:** Cargar y guardar la información desde archivos, con formatos bien definidos para facilitar la actualización y mantenimiento.

**Validación y control de acceso:** Implementar un sistema de inicio de sesión que valide usuarios desde los archivos y restrinja las funcionalidades según el tipo de usuario.

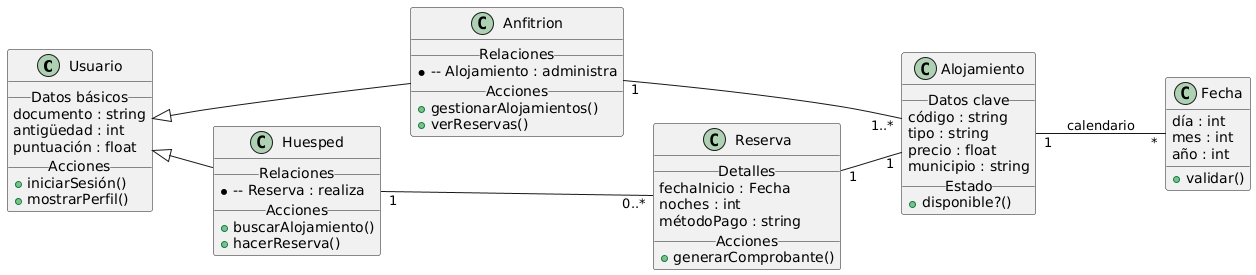
**Control de disponibilidad:** Validar que un alojamiento esté libre para las fechas solicitadas y que el huésped no tenga otra reserva que coincida en ese periodo.

**Optimización de recursos:** Usar memoria dinámica y referencias para evitar copias innecesarias, lo que mejora la eficiencia en tiempo y memoria.

**Modularidad y mantenimiento:** Estructurar el código en módulos y funciones claras para facilitar futuras modificaciones y la lectura del programa.

**Separación de reservas activas e históricas:** Implementar procesos para mover reservas pasadas a un archivo histórico y actualizar la disponibilidad para los próximos 12 meses.

**Interfaz diferenciada por rol:** Mostrar menús y opciones distintas para huéspedes y anfitriones, facilitando el uso y evitando errores.

1. Diagrama de clases: