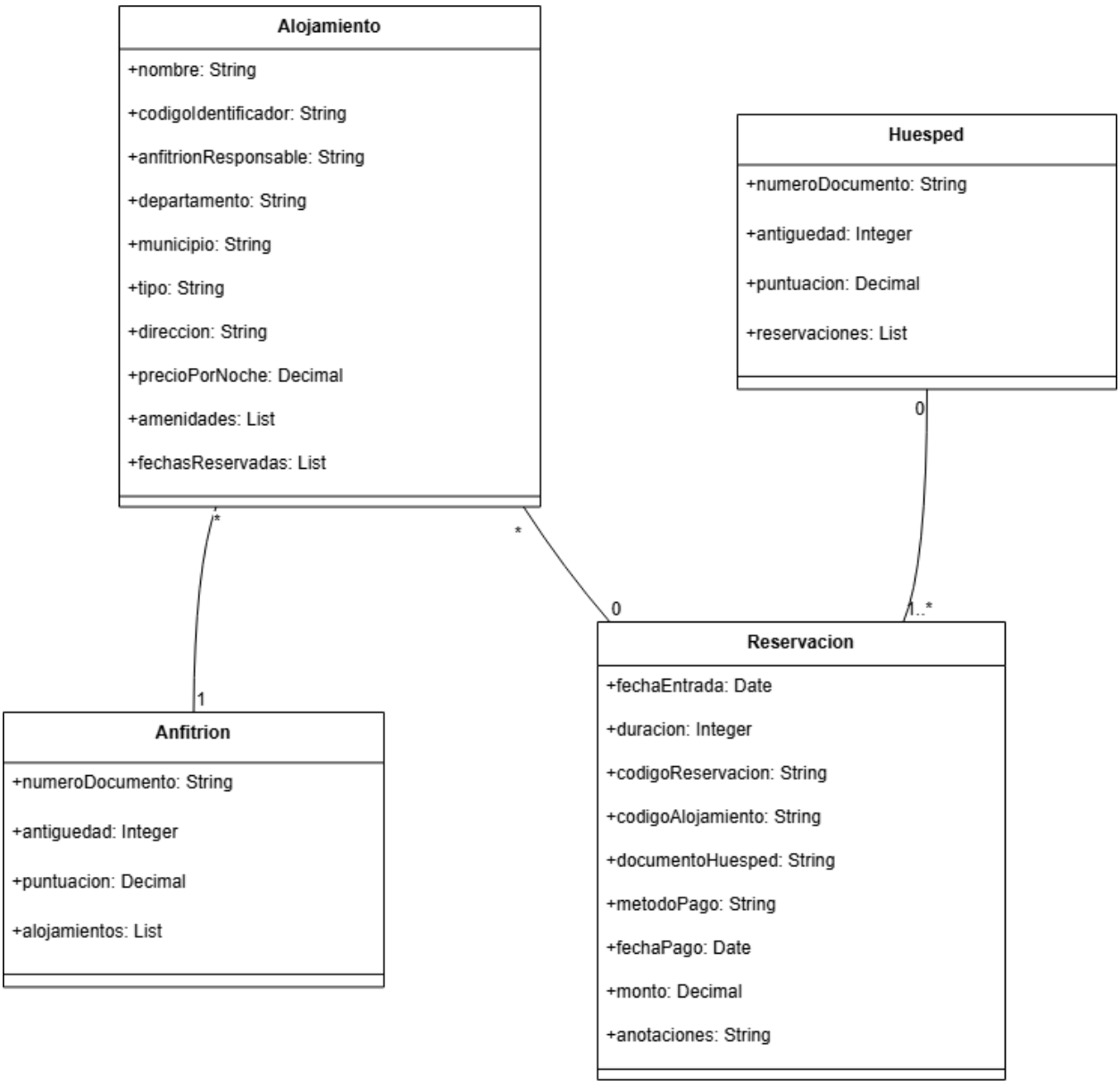
a. Análisis del problema y consideraciones para la alternativa de solución propuesta.
   R// La creación de un sistema llamado UdeAStay, el cual es un mercado que trata de estadías hogareñas para conectar a los huéspedes con los respectivos anfitriones de dichos espacios a ocupar.
   Consideraciones
   El sistema debe manejar clases como Anfitrión, huésped, reservación y alojamiento ya que por el momento son los que mas importan
b. Diagrama de clases de la solución planteada. No debe ser un diagrama trivial que sólo incluya una o dos clases.



c. Descripción en alto nivel la lógica de las tareas que usted definió para aquellos subprogramas cuya solución no sea trivial.
R//Las cuatro clases son importantes ya que si no definen como tal no se podría buscar o añadir reservaciones ya que muchas de las funciones dependen de estas 4 clases

d. Algoritmos implementados debidamente intra-documentados.

R//

e. Problemas de desarrollo que afrontó.

R//Que cuando me aparecía errores me tocaba revisar donde estaba ese error y ver cómo se podía arreglar y aparte organizando lo del archivo que aparecía que no existía

f. Evolución de la solución y consideraciones para tener en cuenta en la implementación.
R// El sistema de UdeAStay es una “plataforma” que hace una conexión entre anfitrión y huésped, así permitiendo la gestión de propiedades, reservaciones y usuarios. Cuenta con lo siguiente en implementación:

1. **Jerarquía de clases** bien definida (Usuario base con derivados anfitrión y Huésped)
2. **Gestión de memoria** con new/delete y destructores adecuados
3. **Persistencia de datos** mediante archivos TXT
4. **Validaciones** para disponibilidad y fechas
5. **Interfaz de usuario** por consola

Consideraciones para la Evolución

1. Mejoras en la Estructura de Datos: Uso de listas enlazadas simples para todo, lo que puede ser ineficiente para búsquedas.

2. Manejo de Fechas Mejorado: Uso de estructuras Fecha simples con comparaciones manuales.

3. Persistencia de Datos Mejorada: Formato TXT plano sin validación de integridad.

4. Gestión de Errores y Excepciones: Manejo básico de errores con cout/cerr.

5. Seguridad y Validaciones: Entradas de usuario con validación mínima.

6. Escalabilidad y Rendimiento: Posibles cuellos de botella con grandes volúmenes de datos.

7. Pruebas Automatizadas: Falta de suite de pruebas.(Esta es una posibilidad para considerarse)

Conclusión

El sistema como se encuentra actualmente proporciona base sólida, pero puede beneficiarse significativamente de modernizaciones en cuanto a estructura de datos, manejo de errores, seguridad y escalabilidad. Si se implementa de manera gradual las mejoras necesarias mantendrá la estabilidad mientras se añaden capacidades profesionales.