Informe de Análisis y Diseño - UdeAStay

# Análisis del problema

El desafío UdeAStay plantea el diseño de un sistema orientado a objetos para gestionar un mercado de estadías hogareñas (tipo Airbnb). [...]

# Diagrama de clases UML

El diagrama de clases incluye Usuario como clase abstracta base, de la cual derivan Huésped y Anfitrión. [...] Este modelo UML garantiza una diseño modular y eficiente, evitando redundancias.

# Lógica de subprogramas clave

Para las funcionalidades no triviales describimos a alto nivel su lógica principal:  
- Ingreso a plataforma (login): solicitar credenciales, recorrer la lista de usuarios [...]  
- Filtrado de alojamientos: dado una fecha inicio, cantidad de noches [...]  
- Validación de fechas: al solicitar reservar se comprueba que la fecha de inicio tenga formato válido [...]  
- Reserva de alojamiento: una vez seleccionado el alojamiento válido, se crea un nuevo objeto Reservación [...]  
- Anulación de reservación: dado un código de reserva, localizar el objeto [...]  
- Actualización del histórico: periódicamente se recorre la lista de reservas activas [...]

# Problemas potenciales y estrategias

Durante el desarrollo pueden surgir varios retos:  
- Gestión de memoria dinámica: sin STL, es fácil olvidar delete [...]  
- Validación de fechas y solapamiento: implementar correctamente el cálculo de fechas [...]  
- Generación y unicidad de códigos: asegurar que cada alojamiento y reservación tenga un código único [...]  
- Estructuras de datos eficientes: sin STL, el acceso es O(n) [...]  
- Manejo de archivos: definir claramente el formato de entrada/salida [...]  
- Interfaz de usuario (menú) ordenada: presentar menús de forma modular [...]

# Evolución esperada y futuras consideraciones

En versiones futuras el sistema podría ampliarse o migrarse de estas restricciones académicas:  
- Escalabilidad: en un escenario real sería conveniente sustituir las estructuras caseras [...]  
- Nuevas funcionalidades: se podrían incluir sistemas de calificación [...]  
- Interfaz avanzada: se podría desarrollar una interfaz gráfica o basada en web [...]  
- Mantenimiento y colaboración: el diseño claro y la documentación con UML facilitarán la comprensión del sistema [...]  
En resumen, la solución propuesta respeta las restricciones y se basa en OOP para modelar el dominio de UdeAStay.