# Análisis de la estructura de la solución – Desafío 2

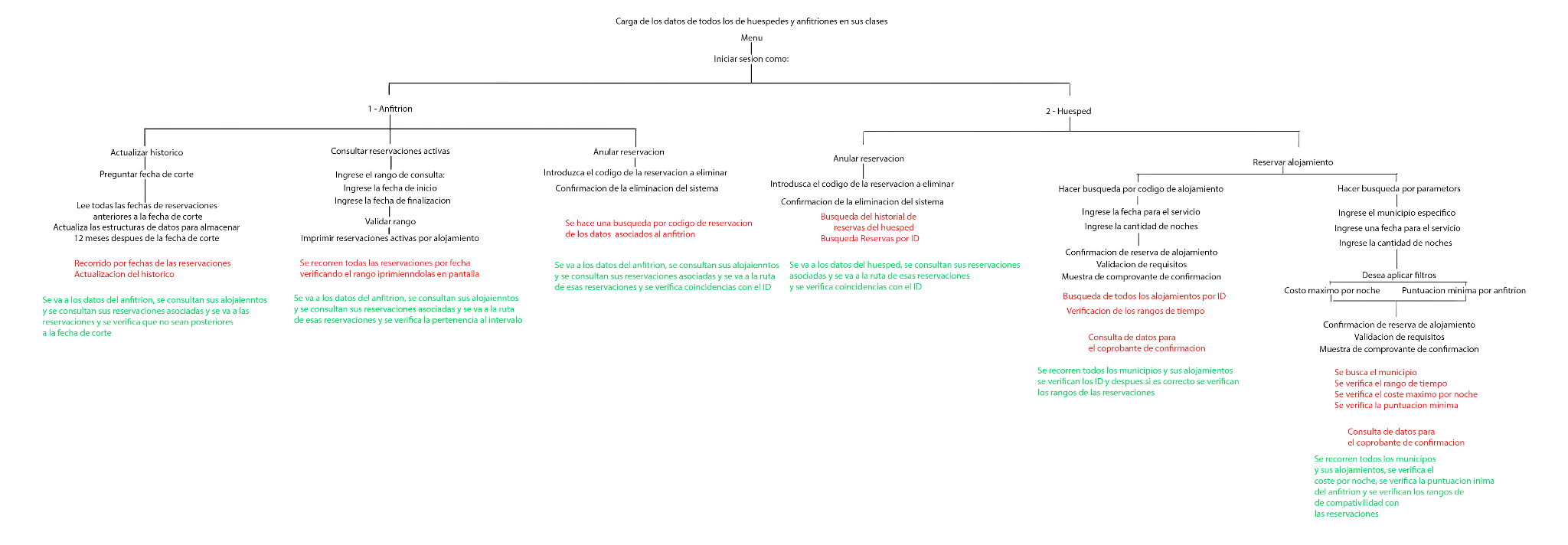
## 1. Objetivo general del sistema

Desarrollar un sistema que permita la gestión de alojamientos, anfitriones, huéspedes y reservas, utilizando archivos `.txt` como medio de almacenamiento persistente. El sistema contará con menús interactivos que permitan registrar, modificar y consultar datos, sin uso de bases de datos ni estructuras avanzadas como `struct` o `vector`.

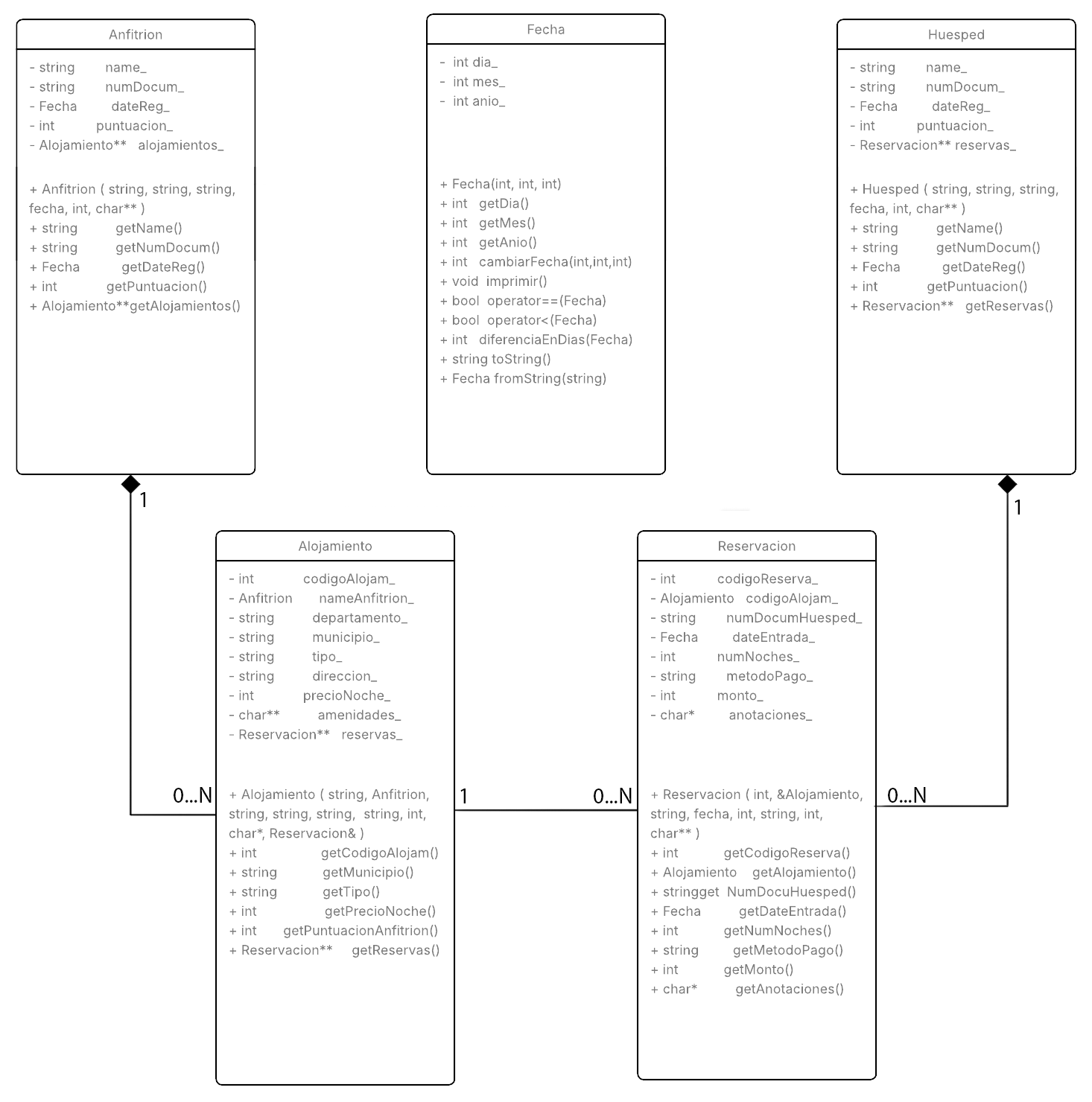
## 2. Estructura general del sistema

El sistema se divide en tres módulos principales:  
- Interfaz principal / Menú de navegación: Presenta las opciones disponibles para el usuario y llama funciones específicas según el rol.  
- Gestión de datos (archivos): Lectura y escritura de archivos `.txt` organizados por entidad.  
- Lógica funcional: Métodos para agregar, editar, eliminar o consultar entidades, con validaciones mínimas.

**3. Diagrama de flujo simplificado del programa**

****

## 4. Diagrama de clases simplificado

****

## 5. Entidades clave y su estructura de almacenamiento

| Entidad | Tipo de archivo | Campos principales | Almacenamiento en memoria |
| --- | --- | --- | --- |
| Anfitrión | anfitriones.txt | ID, nombre, correo | Arreglo de char\* |
| Huésped | huespedes.txt | ID, nombre, correo | Arreglo de char\* |
| Alojamiento | alojamientos.txt | Código, ciudad, precio, ID anfitrión | Arreglo de punteros a strings |
| Reserva | reservas.txt | Código, ID huésped, ID alojamiento, fecha | Arreglo de punteros |

## 6. Gestión de memoria y archivos

Todos los arreglos se implementan usando punteros y memoria dinámica (`new`, `delete`). Se define un límite máximo por tipo de entidad (ej. 100 registros).

## 7. **Manejo de archivos en memoria secundaria**

Para dar soporte a la **Funcionalidad I – Carga y actualización de datos**, el sistema utiliza una estructura jerárquica de directorios y archivos planos diseñada para optimizar lecturas dirigidas y mantener índices ligeros Desafio Info II 2025-1 ….

1. **Directorios principales**
   * reservations/
     1. vigentes/depXX/munYYY/res\_ZZZZZ.txt → Reservas activas, organizadas por departamento (2 dígitos) y municipio (3 dígitos).
     2. historicas/depXX/munYYY/res\_ZZZZZ.txt → Reservas finalizadas, misma jerarquía.
   * hosts/hostID/
     1. profile.txt → Datos del anfitrión.
     2. accommodations.txt → Lista de alojamientos (dep,mun,alojID).
     3. reservas\_activas.txt → Índice local de reservas vigentes (mun,alojID,resID,fechaIni,fechaFin).
   * guests/guestID/profile.txt → Datos de cada huésped.
2. **Índices globales**
   * index\_reservas.txt
     1. Registros <dep>,<mun>,<resID>,<YYYY-MM-DD inicio>,<YYYY-MM-DD fin> ordenados por fecha de inicio.
     2. Permite consultas por rango temporal sin explorar todo el árbol de carpetas.
   * index\_host\_acom.txt
     1. Registros <hostID>,<dep>,<mun>,<alojID> para localizar rápidamente los alojamientos de un anfitrión.

**Formato de los archivos de reserva**

Cada res\_ZZZZZ.txt incluye campos clave en texto plano:  
 reservation\_id: 00012

guest\_id: 54321

accommodation\_id:30001

fecha\_inicio: 2025-05-10

fecha\_fin: 2025-05-15

total\_price: 350000

status: confirmed

1. **Flujo de actualización**
   * Creación:
     1. Escribir res\_ZZZZZ.txt en vigentes/….
     2. Insertar línea ordenada en index\_reservas.txt.
     3. Agregar entrada en hosts/hostID/reservas\_activas.txt.