

Análisis y diseño de solución del Desafío II

Lady Vanessa Montenegro Alarcón
Steven Hernández Quintero Facultad de Ingeniería,

Universidad de Antioquia

259852 informática II

José Aníbal Guerra Soler

12 de abril de 2025

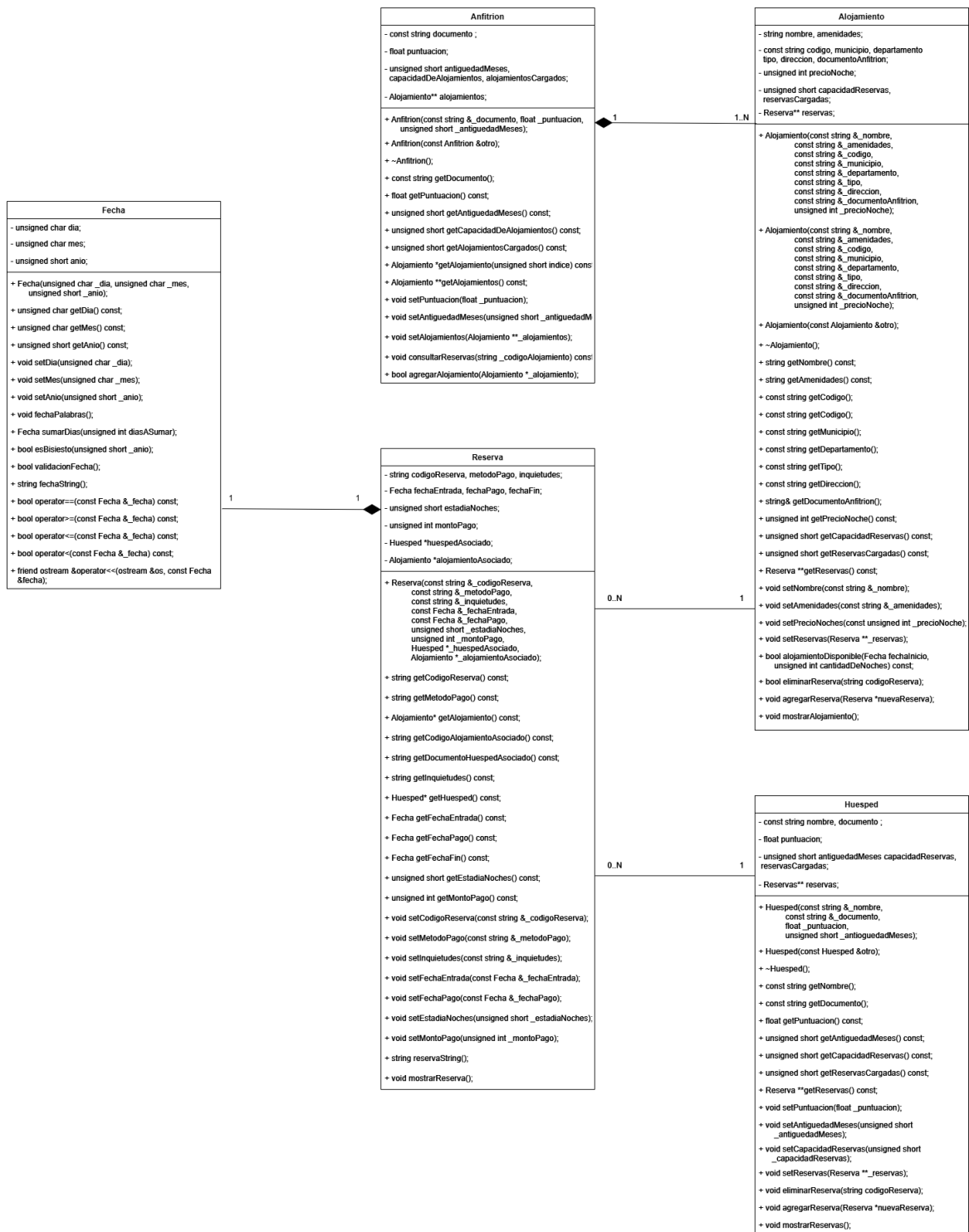
Análisis General del Problema

El sistema se estructura alrededor de cinco clases interconectadas: Anfitrión, Huésped, Reserva, Alojamiento y Fecha, diseñadas para modelar las entidades clave y sus interacciones. La clase Anfitrión gestiona los datos personales y los alojamientos asociados, almacenando estos últimos en un arreglo de objetos Alojamiento. Cada Alojamiento, a su vez, contiene detalles como ubicación, precio y un arreglo de reservas vigentes, vinculadas mediante punteros a objetos Reserva. Por su parte, los Huéspedes mantienen un historial de reservas propias, representadas como arreglos dinámicos que referencian las instancias de Reserva correspondientes. Las reservas son el elemento central del sistema: guardan las fechas (usando la clase Fecha para manejar días y verificarlos) y conectan directamente los alojamientos con los huéspedes, sin repetir información innecesaria.

La carga inicial de datos se realiza leyendo archivos de texto, donde cada línea del archivo se analiza y convierte en instancias de las clases correspondientes, para luego ser almacenadas en sus respectivos arreglos. La lógica y flujo del programa se gestiona mediante funciones globales que operan sobre los arreglos dinámicos principales, mientras que las validaciones se encapsulan en métodos de las clases afectadas.

Los huéspedes interactúan mediante un menú que permite buscar alojamientos filtrando por fechas (usando métodos de Fecha para calcular disponibilidad), ubicación, precio máximo o puntuación del anfitrión. Los resultados se ordenan combinando los atributos, como precio y calidad, implementados mediante algoritmos básicos de ordenamiento. Las reservas nuevas generan códigos únicos mediante un contador incremental y se añaden a los arreglos correspondientes, actualizando los archivos tras cada operación. Los anfitriones, por su parte, acceden a un menú para consultar o cancelar reservas en sus propiedades, validando que las fechas de cancelación sean posteriores a la fecha actual.

Entre las consideraciones del programa se hallan el manejo manual de memoria, que exige destructores en cada clase para liberar arreglos y punteros, y la validación de datos, como garantizar que una reserva no coincida con otras en el mismo alojamiento.



Las tareas definidas por el programa se encuentran categorizadas de la siguiente forma:

♣ Cargar información desde archivos

1. Leer secuencialmente los archivos correspondientes para cada línea

del archivo:

- a. - Validar formato y consistencia de datos (longitud de documentos o códigos)
 - b. - Crear objetos en memoria usando asignación dinámica.
2. Establecer relaciones entre entidades:
 - a. Asignar alojamientos a sus anfitriones, de la misma forma que las reservas a sus respectivos alojamientos y huéspedes.
 - b. Almacenar todo en arreglos dinámicos gestionados manualmente.

♣ Autenticación de Usuario

1. Solicitar documento
2. Búsqueda en arreglos correspondientes:
 - a. Acceder de forma secuencial a los objetos del arreglo y acceder al atributo de documento por medio de su método

♣ Menú de Huésped

1. Generar reserva:
 - a. Búsqueda de alojamientos:
 1. Datos iniciales obligatorios:
 - Fecha de inicio
 - Lugar (Municipio)
 - Cantidad de noches
 - b. Crear reserva:
 1. Generar código de referencia
 2. Añadir al arreglo de reservas vigentes y actualizar archivo
2. Cancelar reserva:
 - a. Buscar por código en reservas del huésped

♣ Menú de Anfitrión

1. Consulta de reservas activas
 - a. Búsqueda del arreglo de reservas por código de su alojamiento
 - b. Filtrar por rango de fecha
2. Cancelación de reservación
 - a. Búsqueda del arreglo de reservas por código de su alojamiento
 - b. Búsqueda de la reserva por código, asignación a nulo de puntero de la reserva
 - c. Reorganización del arreglo

♣ Actualización Reservas Históricas

1. Modificación de arreglo:
 - a. Recorrer el arreglo de reservas vigentes
 - b. Comparar fecha_fin de cada reserva con fecha actual
 - c. Si fecha_fin < fecha_actual;
 - Copiar reserva al arreglo de reservas históricas
 - Eliminar del arreglo de reservas vigentes (reorganizar)

elementos)

2. Actualizar archivo

Durante el desarrollo inicial del programa, que se llevó a cabo utilizando un enfoque basado en ramas. A continuación, se detallan los problemas más significativos encontrados en cada fase:

Rama "desarrolloClases": Esta rama se dedicó a la definición e implementación de todas las clases, sus atributos y métodos iniciales. El principal inconveniente surgió por el análisis inicial, donde no se previeron algunas funcionalidades. Esto llevó a la necesidad de modificaciones e implementaciones de las características y funcionalidades de las clases.

Rama "funcionesCargaDatosPrograma": El objetivo de esta rama era desarrollar las funciones encargadas de la lectura de datos y la creación de objetos a partir de estos. El mayor problema que se presentó fue la segmentación de memoria. Esto requirió una revisión profunda y la optimización de las estructuras de datos utilizadas durante el proceso de carga.

Rama "funcionesCargaDatosArchivos": Esta rama se centró en las funciones de actualizar los datos en los archivos de texto. El mayor problema encontrado fue la organización de los datos en los arreglos utilizados para el almacenamiento. Esto implicó la necesidad de analizar de nuevo las funciones para asegurar la integridad de la información.

Ramas "MenuAnfitrión" y "MenuHuesped": Estas ramas contenían la lógica específica para la interfaz de usuario de los menús de Anfitrión y Huésped, respectivamente. La principal dificultad fue la correcta integración de las clases y funciones previamente desarrolladas con la lógica de los menús. Esto se manifestó en errores en la invocación de métodos y funciones. Esto implicó una nueva revisión de estos.

Al realizar la integración de todas las ramas en la rama "master", surgieron varios problemas. Se encontraron y corrigieron algunos errores sintácticos, pero la mayor problemática estuvo en los errores semánticos. Se identificó que algunas funciones no estaban implementadas de manera correcta y se hicieron las modificaciones necesarias.

EVOLUCIÓN DE LA SOLUCIÓN

El primer análisis que se tuvo fue utilizar arreglos paralelos para así relacionar los datos, sin embargo, se decidió que esta no era la mejor solución y se pensó en los arreglos dinámicos, después de esto el desarrollo del programa comenzó dividiendo las tareas principales por las ramas mencionadas, cuando se terminaron estas y se

unificó todo en la rama master, en ese momento fue donde encontramos varios errores y se corrigieron quedando así el programa actual.

CONSIDERACIONES PARA TENER EN CUENTA EN LA IMPLEMENTACIÓN

Se tuvo en consideración manejar las listas por índices y no solo por punteros, pero al no tener del todo claro esta lógica se continuó con la idea general.