Universidad de Antioquia
Ingeniería Electrónica
Informática II
Desafío II: Mercado de estadías hogareñas UdeAStay
Andres Camilo Rosero Villarreal
Santiago Ortiz Vargas
28/05/2025

Descripción General
Se debe desarrollar un sistema llamado UdeAStay para gestionar un mercado de estadías hogareñas. Este sistema permite a huéspedes y anfitriones interactuar con alojamientos, hacer o administrar reservaciones, y mantener un control de la información mediante archivos.

Funciones Principales del Sistema
- Carga automática de datos desde archivos (no visible en el menú).
- Ingreso por perfil (huésped o anfitrión) con menú personalizado.
- Reservar alojamiento, aplicando filtros por municipio, precio y puntuación (huéspedes).
- Anular reservación (huésped o anfitrión).
- Consultar reservaciones activas por parte de anfitriones.
- Actualizar histórico para mover reservas antiguas.
- Medición de recursos: memoria usada y número de iteraciones

Modelado de Datos
Para el desarrollo del sistema utilizando el paradigma de programación orientada a objetos (POO), se definieron cinco clases principales que representan las entidades fundamentales del dominio del problema: **Huesped, Anfitrion, Fecha, Alojamiento** y **Reservacion.**
Con el fin de almacenar de manera permanente la información correspondiente, se diseñaron cuatro archivos de texto que contienen los datos relevantes del sistema, además de un archivo adicional denominado **historico.txt** , que almacena las reservaciones antiguas utilizando la misma estructura del archivo de reservaciones.

A continuación, se describe la estructura de cada archivo:

- **huespedes.txt**

  Contiene los datos de los huéspedes registrados.

  Campos:
  documento, nombre, correo, teléfono, antigüedad, puntuación

- **anfitriones.txt**

  Incluye la información de los anfitriones y los códigos de los alojamientos que administran.

  Campos:
  documento, nombre, correo, teléfono, antigüedad, puntuación, códigosAlojamientos

- **alojamientos.txt**

  Registra todos los alojamientos disponibles en la plataforma.

Campos:
código, nombre, documentoAnfitrión, departamento, municipio, tipo, dirección, precio, amenidades(...), fechasReservadas

- **reservaciones.txt**

  Guarda las reservaciones activas realizadas por los huéspedes.

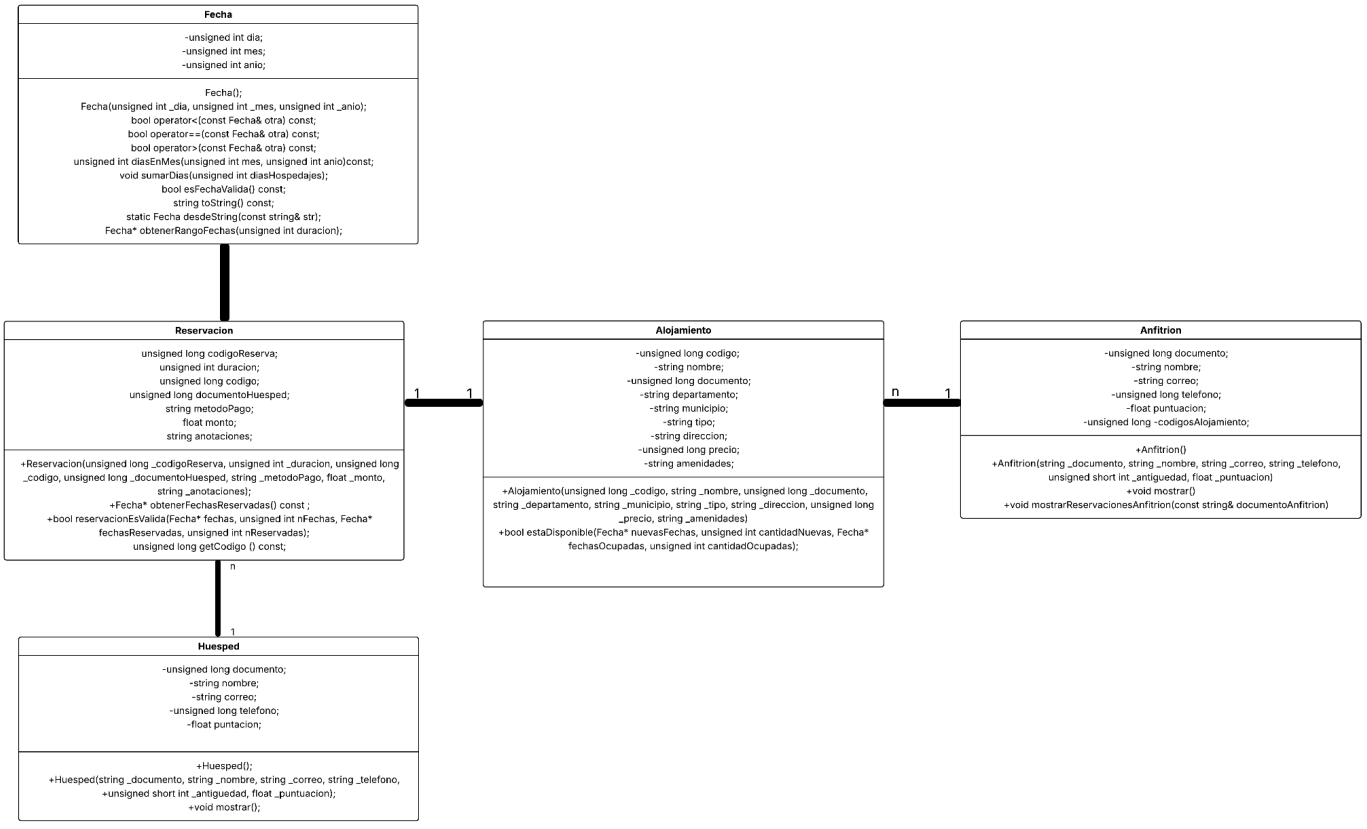
  Campos:
  códigoReserva, fechaEntrada, duración, códigoAlojamiento, documentoHuésped, métodoPago, fechaPago, monto, anotación

- **historico.txt**

  Archivo adicional que almacena las reservaciones pasadas. Tiene la misma estructura que el archivo de reservaciones.

  Campos:
  códigoReserva, fechaEntrada, duración, códigoAlojamiento, documentoHuésped, métodoPago, fechaPago, monto, anotación

Diagrama de Clases



Desarrollo de la solución

Este sistema fue desarrollado con una estructura de menú que permite una navegación clara y diferenciada según el tipo de usuario. Al iniciar el programa, se realiza automáticamente la carga de todos los archivos necesarios para la correcta ejecución y funcionamiento del sistema. Esta carga inicial se realiza de manera interna y **no es visible para el usuario**, tal como se especifica en el enunciado del desafío.

Una vez cargados los datos, se muestra en pantalla un menú principal que permite al usuario escoger entre dos perfiles:

- **Usuario Anfitrión**
- **Usuario Huésped**

Cada perfil accede a un menú diferente, adaptado a sus funcionalidades específicas:

- El **usuario Anfitrión** cuenta con las siguientes opciones en su menú:
    1. Consultar reservaciones activas para los alojamientos que administra.
    2. Anular una reservación.
    3. Actualizar el histórico de reservaciones, moviendo aquellas cuyas fechas ya han pasado.
- El **usuario Huésped** dispone de las siguientes funcionalidades:
    1. Realizar una nueva reservación, con filtros opcionales según el criterio del usuario.
    2. Anular una reservación existente.

Este diseño modular y orientado al rol del usuario permite una experiencia de uso más organizada, y facilita el cumplimiento de los requisitos establecidos en el enunciado del proyecto.

Detalles de la implementación

El sistema requiere múltiples validaciones relacionadas con fechas para garantizar el correcto funcionamiento de diversas funcionalidades. Estas validaciones son fundamentales, por ejemplo, en los siguientes casos:

- Al crear una nueva reservación, se deben validar las fechas ingresadas, calcular automáticamente la fecha de salida a partir de la fecha de entrada y la duración, y verificar la disponibilidad del alojamiento durante ese periodo.

- Al seleccionar una “fecha de corte” para actualizar el histórico, es necesario comparar fechas y mover las reservaciones que hayan finalizado.

- Para eliminar reservaciones pasadas o evitar duplicidad de reservas en los mismos días, también se deben realizar comparaciones precisas entre fechas.

Debido a estas necesidades, se implementó una **clase auxiliar llamada Fecha**, la cual centraliza todas las operaciones y validaciones necesarias sobre fechas. Esta clase permite:

- Comparar fechas (anterior, igual o posterior).

- Sumar días a una fecha dada.

- Validar el formato correcto de la fecha ingresada.

- Calcular el rango de días cubiertos por una reservación.

- Evaluar si una fecha está dentro de los próximos 12 meses desde una fecha de referencia, entre otras funciones.

El uso de esta clase permite mantener el código modular y reutilizable, lo cual facilita el mantenimiento del sistema y mejora significativamente su legibilidad.

Asimismo, todas las funcionalidades ofrecidas tanto al **usuario anfitrión** como al **usuario huésped** se implementan mediante **métodos específicos en las respectivas clases**, asegurando una adecuada distribución de responsabilidades dentro del diseño orientado a objetos.