

Informe de Análisis y Diseño

Desafío II - Informática II

Semestre 2025-1

Benjamin Ruiz Guarin
Kevin Jimenez Rincón

17 de mayo de 2025

1. Análisis del problema

El objetivo de este desafío es desarrollar un sistema para la administración de un mercado de estadías hogareñas, utilizando POO. El sistema debe permitir la gestión eficiente de los alojamientos, las reservaciones, los anfitriones y los huéspedes, entre otros.

Consideramos que los principales puntos a tener en cuenta en este análisis son los siguientes:

- Tanto los huéspedes, anfitriones, alojamientos y reservas tienen asociados a ellos una serie de datos (códigos, nombres, puntuaciones, etc.) y capacidades (reservar, anular reserva, etc) asociadas a ellos.
- Algunos de los datos como los códigos de reserva, alojamientos, fechas, entre otros, pueden estar asociados a más de una entidad (reservas, huéspedes, etc.) a la vez y debemos evitar la copia innecesaria.
- Los datos de huéspedes, anfitriones, alojamientos y reservas deben ser almacenados en archivos de forma permanente, además de un histórico de reservas. Para esto, debemos escoger formatos y cantidad de archivos.
- Es necesario implementar un método para medir recursos utilizados en la ejecución de las funciones (iteraciones y memoria ocupada).

2. Diseño de la solución

El enfoque que tomaremos para solucionar el problema es el siguiente:

- Como es de esperarse, huéspedes, anfitriones, alojamientos y reservas serán cada uno una clase, con todos sus datos como atributos y sus capacidades como métodos.
- Las clases que se manejarán estarán relacionadas por "pertenencia" de la siguiente manera: Los alojamientos son parte de los anfitriones, las reservas son parte de los alojamientos y las reservas también son parte de los huéspedes. Esto se decidió teniendo en cuenta como son las cosas en la vida real y el buscar la eficiencia, evitando tener que implementar algoritmos de

busqueda (por ejemplo, si los anfitriones pertenecieran a los alojamientos para que el anfitrión conozca sus alojamientos, tendríamos que buscar cuales lugares lo tienen como anfitrión), aprovechando que cada alojamiento tiene un único anfitrión y cada reserva un único alojamiento y un único huésped.

- Se evitará la copia innecesaria haciendo uso de punteros y parametros por referencia.
- Para los archivos se usará el formato .txt. Se usará un archivo por clase y uno para el histórico, con la intención de facilitar la legibilidad de estos. Además se buscará que datos con opciones limitadas (días, meses, años, datos booleanos) y los datos numerico se almacenen y lean de forma binaria (si es un número 32 no se leerá en el archivo "32" si no que se leerá ' ' que es 00100000 en ascii y 32 en binario).
- Para medir las iteraciones se usará un contador que se pasará por referencia para aumentar con cada iteración interna. Para medir el almacenamiento utilizado se usará la función sizeof() para conocer el espacio utilizados por arreglos al finalizar una función.