**Informe final del Desafío II: UdeAStay.**

El proyecto consistió en crear un sistema de reservas, llamado UdeAStay, donde usuarios registrados como huéspedes o anfitriones pueden gestionar reservas de alojamiento. Desde el inicio, el sistema carga datos desde archivos, permite a los usuarios iniciar sesión, mostrar alojamientos disponibles según diferentes filtros, crear reservaciones automáticas, consultar reservas activas, cancelarlas y actualizar automáticamente un histórico de reservas pasadas. El objetivo principal fue desarrollar una aplicación en C++ que gestione usuarios, alojamientos y reservaciones usando memoria dinámica, sin apoyarse en la STL, y cumpliendo todos los requisitos especificados. En el diseño, creé varias clases principales: fecha (para gestionar entradas y salidas), huésped (información del usuario), anfitrión (datos y puntuación), alojamiento (detalles y disponibilidad), reservación (completo control de reservas) y sistemaUdeaStay (clase central que coordina todo). La memoria se maneja con arreglos de punteros de hasta 100 elementos. Además, todos los datos se cargan desde archivos .txt al iniciar y se guardan al terminar. Entre las funcionalidades implementadas, se incluyen: cargar datos desde archivos, inicio de sesión según tipo de usuario, mostrar alojamientos filtrados (por municipio, fechas, precio, puntuación), crear reservas automáticas con códigos únicos, consultar reservas activas, cancelar reservas y actualizar reservas pasadas en un archivo histórico. Enfrenté problemas para la creación de la función que le permiten ver a los anfitriones sus reservas activas y la función en la que se reservan las fechas para que no se puedan entrecruzar con otras. También manejé manualmente la memoria con el uso de char[] y usé arreglos de caracteres en lugar de cadenas para cumplir con las restricciones. Además, implementé funciones para validar disponibilidad y evitar la superposición de fechas. El proceso de desarrollo inició con clases básicas, y fue evolucionando mediante pruebas por módulos. Finalmente, toda la funcionalidad se integró en la clase sistemaUdeaStay, asegurando modularidad y encapsulación. En conclusión, UdeAStay cumple con la mayoría de requisitos del desafío, ofreciendo una solución casi completa, eficiente y estructurada que permite gestionar reservas de alojamiento desde una interfaz en consola y mantiene la información persistentemente en archivos.