Universidad del Rosario
Matemáticas aplicadas y ciencias de la computación
Informe
Tatiana Cabrera
2024.1

UNIVERSIDAD DEL ROSARIO

# INFORME

## ANÁLISIS

El código está dividido en clases que representan entidades relacionadas entre sí, como Destino, Actividades, Viaje y Planeacion_viaje. Esto permite una fácil comprensión y mantenimiento del código, ya que cada clase se encarga de una funcionalidad específica.

**Clases y estructuras:**

Interfaz de Usuario: La clase Interfaz maneja la interacción con el usuario a través de la consola. Presenta un menú de opciones y solicita la entrada del usuario para realizar diferentes acciones, como agregar destinos, asignar viajes y actividades, y consultar información.

Agregar Destinos: La clase Destino proporciona un método para agregar destinos a un conjunto (set) de destinos únicos. Antes de agregar un destino, se verifica si ya existe en la lista para evitar duplicados.

Asignar Viajes y Actividades: Las clases Planeacion_viaje y Actividades permiten asignar viajes y actividades a miembros específicos de la familia. Se utilizan mapas para asociar cada miembro con sus asignaciones correspondientes.

Consultar Destinos y Actividades: Se proporcionan funciones para consultar los destinos asignados a un miembro específico de la familia y para mostrar las actividades asignadas a ese miembro.

**Uso de Estructuras de Datos:**

Se utilizan varios `set`, `map` y `vector`, para almacenar y manipular datos de manera eficiente. Por ejemplo, se utiliza un `set` para almacenar destinos únicos y evitar duplicados, un `map` para asociar miembros de la familia con sus asignaciones de viajes y actividades, y un `vector` para almacenar actividades dentro de la clase Actividades.

**Mejoras Potenciales:**

Manejo de Conflictos de Fechas: Aunque se verifica si un miembro ya tiene un viaje asignado para las mismas fechas antes de asignar un nuevo viaje, podría mejorarse para manejar conflictos de manera más precisa, como permitir al usuario elegir entre reprogramar el viaje existente o cancelar el nuevo.

Validación de Entrada del Usuario: Se podría agregar validación de entrada del usuario para garantizar que los datos ingresados sean del tipo esperado y estén en el formato correcto, especialmente para las fechas.

Mejoras en la Interfaz de Usuario: Se podría mejorar la presentación y usabilidad de la interfaz de usuario agregando mensajes más descriptivos, opciones de retroceso y un manejo más robusto de errores del usuario.

En conclusión, el código proporciona una base funcional para la gestión de destinos, viajes y actividades para una familia, pero podría mejorarse agregando características adicionales y puliendo la interfaz de usuario.