**Informe del Módulo de Reservas**

**Trabajo Práctico Final – Programación I  
Integrante responsable del módulo de Reservas: Sebastian Marquez  
Grupo: 8**

**1. Introducción**

**En el marco del Trabajo Práctico Final de la materia Programación I, se desarrolló un sistema de gestión de paquetes turísticos dividido en módulos. A mí me correspondió la parte de reservas, que tiene como objetivo registrar, consultar, modificar y eliminar las reservas que realizan los clientes sobre los paquetes disponibles.**

**El módulo se diseñó de forma independiente, pero conectado con los otros dos (Clientes y Paquetes), ya que necesita validar que los IDs existan y que los cupos de los paquetes alcancen.**

**2. Funcionamiento del módulo**

**El menú de Reservas ofrece distintas opciones:**

* **Agregar reserva: solicita el ID del cliente y del paquete, valida que existan, controla que haya cupos disponibles y finalmente guarda la reserva en la matriz. Al mismo tiempo descuenta los cupos del paquete.**
* **Mostrar reservas: lista todas las reservas en forma ordenada, usando funciones lambda para que se pueda ordenar por distintos campos. Además, si se busca una en particular, se muestra el detalle relacionado con los datos del cliente y del paquete.**
* **Modificar reserva: permite cambiar la cantidad de personas o incluso el paquete elegido. Cada cambio actualiza automáticamente los cupos de los paquetes.**
* **Eliminar reserva: elimina una reserva y restaura los cupos correspondientes en el paquete.**
* **Estadísticas: calcula total de reservas, promedio de personas por reserva, reserva con más y menos personas, cantidad por destino, porcentaje por destino y también muestra los destinos únicos utilizando conjuntos.**

**También se integraron validaciones con expresiones regulares para asegurar que los IDs y las cantidades ingresadas sean números enteros positivos. Esto mejora la robustez del sistema y evita errores de carga.**

**3. Estructuras de datos utilizadas**

* **Matriz: la lista de reservas se implementa como una matriz (lista de listas), donde cada fila representa una reserva con sus campos.**
* **Listas: utilizadas en general para manejar clientes, paquetes y reservas.**
* **Diccionarios: aplicados en el cálculo de estadísticas (conteo y porcentajes por destino).**
* **Tuplas: usadas para devolver múltiples valores en funciones de resumen (total, promedio, máximo y mínimo).**
* **Conjuntos (set): empleados para obtener los destinos únicos sin duplicados.**

**4. Problemas encontrados y soluciones aplicadas**

* **Ingreso de datos inválidos: al principio era común que si el usuario escribía letras en lugar de números, el programa se caía. Se solucionó agregando validaciones con expresiones regulares y bucles que obligan a reingresar el dato hasta que sea válido.**
* **Control de cupos: surgieron inconsistencias cuando se eliminaba o modificaba una reserva. La solución fue ajustar los cupos automáticamente cada vez que se crea, modifica o elimina una reserva.**
* **Repetición de destinos en estadísticas: al listar destinos aparecían duplicados. Esto se resolvió usando un conjunto (set) para quedarnos solo con los únicos.**
* **Claridad del código: se organizaron las funciones en archivos separados (agregar, mostrar, modificar, eliminar, estadísticas) y se agregaron comentarios/docstrings para mejorar la lectura.**

**5. Backlog del módulo de Reservas**

* **Crear reservas validando clientes, paquetes y cupos.**
* **Listar reservas con opción de ver detalle (cliente + paquete asociado).**
* **Modificar reservas (cantidad o paquete).**
* **Eliminar reservas restaurando cupos.**
* **Calcular estadísticas básicas (total, promedio, máximo, mínimo, conteos, porcentajes).**
* **Validar entradas con expresiones regulares.**
* **Mostrar destinos únicos utilizando conjuntos.**

**6. Planificación de sprints**

**El desarrollo se dividió en etapas cortas, con entregas parciales al finalizar cada una:**

* **Sprint 1: estructura de la matriz de reservas y función para agregar reservas.**
* **Sprint 2: mostrar reservas con ordenamiento por lambda y detalle relacionado.**
* **Sprint 3: modificar y eliminar reservas, asegurando consistencia de cupos.**
* **Sprint 4: agregar estadísticas, expresiones regulares y conjuntos para finalizar la primera entrega.**