

ISIS1206 - Estructuras de Datos

Proyecto #2

Definición de los datos de Entrada y Datos de Salida de los requerimientos

Primera versión: Geovanny Gonzalez

Revision: Fernando De la Rosa

Fecha: 12 Marzo 2019

Requerimiento	Entradas	Salidas
0 - Carga de datos	int numeroSemestre: El número del semestre. Validez: numeroSemestre es 1 o 2	EstadisticasCargaInfracciones Incluye información de zona Minimax (Xmin, Ymin), (Xmax, Ymax) para localizacion (X, Y)
1A	int N: Número de franjas horarias que tienen más infracciones. Validez: $1 \leq N \leq 24$ Se realiza usando una Cola Prioridad / Heap	IQueue<InfraccionesFranjaHoraria> con top N
2A	double Xcoor : Coordenada X de la posición de la zona de infracción double Ycoor: Coordenada Y de la posición de la zona de infracción. Validez: $Xmin \leq Xcoor \leq Xmax$ $Ymin \leq Ycoor \leq Ymax$ Se realiza usando una tabla Hash	InfraccionesLocalizacion
3A	Las fechas siguen el formato año-mes-día, e.g: 2018-01-01 LocalDate fechaInicial: Fecha inicial del rango de búsqueda LocalDate fechaFinal: Fecha final del rango de búsqueda Validez: $2018-01-01 \leq fechaInicial < fechaFinal \leq 2018-12-31$ Se realiza usando un Arbol Balanceado	IQueue<InfraccionesFecha>
1B	int N: Numero de los tipos de	IQueue<InfraccionesViolationCo

	<p>ViolationCode con más infracciones.</p> <p>Validez: $1 \leq N$</p> <p>Se realiza usando una Cola Prioridad / Heap</p>	de> con top N
2B	<p>double Xcoor : Coordenada X de la posición de la zona de infracción</p> <p>double Ycoor: Coordenada Y de la posición de la zona de infracción</p> <p>Validez: $X_{min} \leq X_{coor} \leq X_{max}$ $Y_{min} \leq Y_{coor} \leq Y_{max}$</p> <p>Se realiza usando un Arbol Balanceado</p>	InfraccionesLocalizacion
3B	<p>double valorInicial: Valor mínimo acumulado de las infracciones</p> <p>double valorFinal: Valor máximo acumulado de las infracciones.</p> <p>Validez: $0 \leq \text{valorInicial} < \text{valorFinal}$</p> <p>Se realiza usando un Arbol Balanceado</p>	IQueue<InfraccionesFechaHora>
1C	<p>int addressID: Localización de la consulta.</p> <p>Validez: addressId válido</p>	InfraccionesLocalizacion
2C	<p>Las horas siguen el formato HH-MM-SS, e.g: 11:15:00</p> <p>LocalTime horaInicial: Hora inicial del rango de búsqueda</p> <p>LocalTime horaFinal: Hora final del rango de búsqueda</p> <p>Validez: $00:00:00 \leq \text{horaInicial} < \text{horaFinal} \leq 23:59:59$</p>	InfraccionesFranjaHorarioViolationCode
3C	<p>int N: Numero de las localizaciones con mayor número de infracciones</p>	IQueue<InfraccionesLocalizacion>

	Validez: 1 <= N	
4C	Ninguna	El requerimiento debe mostrar la gráfica. Se espera utilizar una Contenedora<InfraccionesViolationCode> ordenados de mayor a menor por número de infracciones

Clases VO:

Revisar la definición de estas clases en el paquete model.vo del proyecto