T

Node
- element: T
- next: Node
+ cambiarSiguiente(Node): void
+ darElemento(): T
+ Node(T, Node)
+ siguiente(): T

-first

--next

«interface»
IPunto
+ getLatitud(): double
+ getLongitud(): double
+ setLatitud(double): void
+ setLongitud(double): void

«interface»
IPilaEventosSospec
+ darTamanio(): int
+ estaVacia(): boolean
+ pop(): IEventosVehi
+ push(IEventosVehic

Punto
- latitud: double
- longitud: double
+ getLatitud(): double
+ getLongitud(): double
+ Punto(double, double)
+ setLatitud(double): void
+ setLongitud(double): void

-puntoInicial -punto

«interface»
IEventoVehiculo
+ getEstampillaTiempo(): lon
+ getTipo(): TipoEventoCont
+ setEstampillaTiempo(long)
+ setTipo(TipoEventoControl

Object

NodeLista
- element: T
- next: Node
- prev: Node
+ anterior(): T
+ cambiarAnterior(NodeLista): void
+ cambiarSiguiente(NodeLista): void
+ darElemento(): T
+ NodeLista(T, NodeLista, NodeLista)
+ siguiente(): T

EventoVehiculo
- fecha: long
- punto: Punto
- tipoEvento: TipoEventoVehiculo
~ TipoEventoVehiculo: enum
+ darPunto(): Punto
+ estaEnCuadrado(Punto, Punto, Punto):
+ getEstampillaTiempo(): long
+ getTipo(): TipoEventoControl
+ setEstampillaTiempo(long): void
+ setPunto(Punto): void
+ setTipo(TipoEventoControl): void

-nodo

T

IListaSegmentos
ListaSegmentos
- nodo: NodeLista
- size: int
+ agregarAlFinal(T): void
+ agregarAlPrincipio(T): void
+ agregarAntesDe(T, T): boolean
+ agregarDespuesDe(T, T): boolean
+ buscar(T): T
+ dar(int): T
+ darElemento(): T

Segmento
- horiVert: boolean
- puntoFinal: Punto
- puntoInicial: Punto
+ esHorizontaloVertical(): boolean
+ getPuntoLlegada(): Punto
+ getPuntoSalida(): Punto
+ puntoPerteneceSegmento(Punto): boolean

Vehiculo

- capacidad: int
- colaEventos: Iterable<EventoVehiculo>
- fechaFinRuta: long
- fechaInicioRuta: long
- idVehiculo: int
- modelo: int
- placa: String
- rutaVehiculo: Ruta
- tipoVehiculo: TipoVehiculo

+ darFechaFin(): long
+ darFechaInicio(): long
+ darGasolina(): double
+ darId(): int
+ darModelo(): int
+ darPlaca(): String
+ darPlaca(): int
+ darTipoVehiculo(): TipoVehiculo
+ finalizarRuta(Ruta, EventoVehiculo): void
+ getEventos(): Iterable<EventoVehiculo>
+ getRuta(): IRuta
+ guardarEvento(Iterable<EventoVehiculo>): void
+ iniciarRuta(Ruta, EventoVehiculo): void
+ setRuta(Ruta): void
+ vehiculo(int, TipoVehiculo, String, int, int)

CentralVehiculos

- listaBitacoras: ArrayList
- listaVehiculos: ArrayList
- pilaEventos: PilaEventosSospechos

+ buscarRutaMasAccidentada(): IRuta
+ buscarSegmentoMasAccidentado(): ISegmento
+ buscarVehiculoMasAccidentado(): IVehiculo
+ cargarEventosAutorizadosRuta(IVehiculo, IRuta, Iterable<IEventoVehiculo>): void
+ cargarRutaVehiculo(IVehiculo, IRuta): void
+ darEventosPorArea(IPunto, IPunto, Iterable<IEventoVehiculo>): Iterable<IEventoVehicu
+ darEventosRutaPorTiempo(Iterable<IEventoVehiculo>, Date, Date): Iterable<IEventoVe
+ darEventosSospechososPorPatron(IVehiculo, IRuta, Iterable<IEventoVehiculo>): Iterab
+ darPromedioConsumoCombustiblePorFechas(Date, Date): double
+ finalizarRutaVehiculo(IVehiculo, IRuta, IEventoVehiculo): void
+ iniciarCentralVehiculos(): void
+ iniciarRutaVehiculo(IVehiculo, IRuta, IEventoVehiculo): void
+ recibirEventosControl(File): void
+ unificarEventosRuta(IVehiculo, IRuta): Iterable<IEventoVehiculo>

-vehicu

PilaEventosSospechosos

- first: Node
- rut: Ruta
- size: int
- vehicu: Vehiculo

+ clear(): void
+ contains(Object): boolean
+ isEmpty(): boolean
+ peek(): Object
+ PilaEventosSospechosos(Vehiculo, Ruta): Iterable<T>
+ pop(): Object
+ push(Object): void
+ search(Object): int
+ size(): int
+ toArray(): ArrayList

«interface» ISegmento

+ getPuntoLlegada(): IPunto
+ getPuntoSalida(): IPunto
+ puntoPerteneceSegmento(IPunto): boolean
+ setPuntoLlegada(IPunto): void
+ setPuntoSalida(IPunto): void

(partial left-side boxes)

...EventoVehiculo
Iterable
...chosos
...culo
...ulo): void

...g
...rol
...t void
...): void

...boolean

«interface»
ICentralVehiculos

+ buscarRutaMasAccidentada(): IRuta
+ buscarSegmentoMasAccidentado(): ISegmento
+ buscarVehiculoMasAccidentado(): IVehiculo
+ cargarEventosAutorizadosRuta(IVehiculo, IRuta, Iterable<IEventoVehiculo>): void
+ cargarRutaVehiculo(IVehiculo, IRuta): void
+ darEventosPorArea(IPunto, IPunto, Iterable<IEventoVehiculo>): Iterable<IEventoVehiculo>
+ darEventosRutaPorTiempo(Iterable<IEventoVehiculo>, Date, Date): Iterable<IEventoVehiculo>
+ darEventosSospechososPorPatron(IVehiculo, IRuta, Iterable<IEventoVehiculo>): Iterable<IEventoVehiculo>
+ darPromedioConsumoCombustiblePorFechas(Date, Date): double
+ finalizarRutaVehiculo(IVehiculo, IRuta, IEventoVehiculo): void
+ iniciarCentralVehiculos(): void
+ iniciarRutaVehiculo(IVehiculo, IRuta, IEventoVehiculo): void
+ recibirEventosControl(File): void
+ unificarEventosRuta(IVehiculo, IRuta): Iterable<IEventoVehiculo>

+ darTamanio(): int
+ decolar(): IEvento
+ encolar(T): void
+ estaVacia(): bool
+ getEventoVehicu
+ setEventos(Iterab

IEventoVehiculo

Iterable

«interface»
IColaEventos

...Vehiculo

...ean
...lo(): Iterable<IEventoVehiculo>
...le<IEventoVehiculo>): void

«interface»
IRuta

+ buscarSegmentoPorExtremos(IPunto, IPunto): ISegmento
+ getEventosAutorizados(): Iterable<IEventoVehiculo>
+ getSegmentos(): Iterable<ISegmento>
+ puntoEnSegmentoDeRuta(IPunto): boolean
+ setEventosAutorizados(Iterable<IEventoVehiculo>): void
+ setSegmentos(Iterable<ISegmento>): void

«interface»
IVehiculo

+ getEventos(): Iterable<IEventoVehiculo>
+ getRuta(): IRuta
+ guardarEvento(Iterable<IEventoVehiculo>): void
+ setRuta(IRuta): void

MergeSort

+	main(EventoVehiculo[]): void
-	merge(EventoVehiculo[ ], EventoVehiculo[], int, int, int): void
+	mergeSort(Iterable): void
-	mergeSort(EventoVehiculo[ ], EventoVehiculo[], int, int): void

«enumeration»
TipoEventoVehiculo

INICIAR_RUTA
TERMINAR_RUTA
ENCENDER_VEHICULO
APAGAR_VEHICULO
ABRIR_PUERTA_VEHICULO
CERRAR_PUERTA_VEHICULO
SUBIR_PASAJERO
BAJAR_PASAJERO
ABRIR_PUERTA_CARGA
CERRAR_PUERTA_CARGA
SENSOR_PESO
SENSOR_TEMPERATURA
SENSOR_INCLINACION

«enumeration»
TipoVehiculo

CAMION
FURGON

+ darTamanio(): int
+ eliminar(T): T
+ estaVacia(): boolean
+ ListaSegmentos()

+ Segmento(Punto, Punto)
+ setPuntoLlegada(Punto): void
+ setPuntoSalida(Punto): void