Problema

- estampilla: long
- lugar: Punto
- tipo: TipoProblema

- + getEstampillaTiempo(): long
- + getPunto(): Punto
- + getTipoProblema(): TipoProblema
- + setEstampillaTiempo(long): void
- + setPunto(Punto): void
- + setTipo(TipoProblema): void

Cliente

- apellido: String
- casos: Iterable<Caso>
- nombre: String
- pedidos: Iterable<PedidoDespacho>

- + agregarCasoAHistorico(Caso): void
- + agregarPedidoDespacho(PedidoDespacho): void
- + darCasosAbiertos(): Iterable<Caso>
- + darHistoricoCasos(): Iterable<Caso>
- + darPedidosDespacho(): Iterable<PedidoDespacho>
- + getApellido(): String
- + getNombre(): String
- + setApellido(String): void
- + setNombre(String): void

ListaAgentes

serializable
ILista

- elementos: Object ([])
- tamanioEst: int = 0

- + agrandarArreglo(): void
- + agregarAgente(T): void
- + atenderAgente(): T
- + eliminarAgente(T): T
- + ListaAgentes()

-listaAgentes

Llamada

- atendiendo: Agente
- cliente: Cliente
- estampilla: long
- lio: Problema

- + getAgente(): Agente
- + getCliente(): Cliente
- + getEstampillaTiempo(): long
- + getProblema(): Problema
- + Llamada()
- + setAgente(Agente): void
- + setCliente(Cliente): void
- + setEstampillaTiempo(long): void
- + setProblema(Problema): void

PedidoDespacho

- cliente: Cliente
- puntoDes: Punto
- puntoOri: Punto

- + getCLiente(): Cliente
- + getPuntoDestino(): Punto
- + getPuntoOrigen(): Punto
- + setCliente(Cliente): void
- + setPuntoDestino(Punto): void
- + setPuntoOrigen(Punto): void

CentralLlamadas

- listaAgentes: ListaAgentes
- listaCasos: ArrayList

- + activarAgente(Agente): void
- + agregarAgente(Agente): void
- + darAgenteActivo(): Agente
- + darAgenteMasCasosCierraPrimeraLlamada(): A
- + darAgentesMasAtiendenPorFecha(int, Date, Da
- + darCasosNoSolucionadosPorFecha(Date, Date
- + darCasosSospechaMalServicio(): Iterable<Cas
- + darCategoriaPromedioMasDemorado(): Catego
- + darPromedioCierreCasosAgentePorFecha(Age
- + darPromedioCierreCasosPorFecha(Date, Date
- + desactivarAgente(Agente): void
- + finalizarCaso(Caso): void
- + iniciarCaso(Cliente, Llamada, CategoriaCaso):
- + iniciarCentralLlamadas(): void
- + procesarLlamada(Llamada): void
- + recibirEventosControl(File): void
- + unificarLlamadas(Iterable<Iterable<Llamada>>

-cliente   -clienteCreador

-pedido

-llamada

Caso

iCaso

- categoria: CatergoriaCaso
- clienteCreador: Cliente
- estado: EstadoCaso
- llamadas: Array
- pedido: PedidoDespacho

- + asignarLlamada(Llamada): void
- + Caso(CategoriaCaso, PedidoDespacho, Cliente, Array)
- + Caso(CategoriaCaso, Cliente, Llamada)
- + getCategoriaCaso(): CategoriaCaso
- + getCliente(): boolean
- + getEstado(): EstadoCaso
- + getHistorialLlamadas(): Array<Llamada>
- + getPedidoDespacho(): PedidoDespacho
- + setCategoriaCaso(CategoriaCaso): void
- + setCliente(Cliente): void
- + setEstado(EstadoCaso): void
- + setPedidoDespacho(PedidoDespacho): void

-atendiendo

Agente

- apellido: String
- lista: iterable<Llamadas>
- llamada: Llamada
- llamadasRecibidas: ArrayList
- nombre: String

- + Agente(String, String)
- + atenderLlamada(Llamada): void
- + darLlamadasAtendidas(): Iterable<Llamada>
- + getApellido(): String

-lio

T

AnalisisUnificado

+ darEventosGeneranMasPQRPorRuta(Ruta): Iterable<EventoVehiculo>
+ darEventosSospechososAsociadosAPQR(Cliente, Ruta): Iterable<EventoVehiculo>
+ darRutaPQRAgrupadoPorSegmentos(): Iterable<Iterable<EventoVehiculo>>

ProcessJSON

+ readAsParser(File): void
+ readFromLine(): void

T

serializab...
IC...

QueueLlamadas

- elementos: Object ([])
- tamanioEst: int = 0
+ agrandarArreglo(): void
+ agregarLlamada(T): void
+ atenderLlamada(): T
+ mostrarPrimerLlamada(): T
+ QueueLlamadas()

ICentralLlamdas

Agente
ate): Iterable<Agente>
e): Iterable<Caso>
o>
oriaCaso
ente, Date, Date): double
): double
Caso
): Iterable<Llamada>

+ getNombre(): String
+ setApellido(String): void
+ setNombre(String): void