

#### **UsuariosSistemaServicio**

- + crear(Persona, username, pass) : Void
- + eliminar(UsuarioSistema) : Void
- + modificar(UsuarioSistema, nombre, pass) : Void
- + login(nombre, pass) : Bool

#### SensoreoServicio

- + ingresarDataMovimiento(Sensor, Data) : Void
- + ingresarDataTemperatura(Sensor, Data) : Void

# ImportadorServicio

+ importar(ArchivoBase64) : Void

# TecnicosServicio

- + crear(Persona, latitud, longitud, radio) : Void
- + eliminar(Tecnico) : Void
- + modificar(Tecnico, latitud, longitud, radio) : Void
- + obtenerTecnicosEnZona(latitud, longitud) : Tecnico[]

# ReportesServicios

+ obtenerReportes() : List<Reporte>

#### ColaboracionesServicio

- + crearAdministracionHeladera(Colaborador, Heladera, FechaContr): FormaContribucion
- + crearDistribucionViandas(HOrigen, HDestino, Cant, Motivo, FechaContr): FormaContribucion
- + crearRegistroPersonaVulnerable(Persona, Menores, Tarjeta, FechaContr): FormaContribucion
- + crearDonacionMonetaria(Monto, FrecDias, Fecha, FechaContr) : FormaContribucion
- + crearDonacionVianda(Heladera, Vianda, FechaContr): FormaContribucion
- + crearOfertaPremio(Premio, FechaContr) : FormaContribucion

## HeladerasServicio

- + crear(nombre, long, lat, Direccion, Modelo, tempMinima, tempMaxima): Void
- + eliminar(Heladera) : Void
- + modificar(heladera, nombre, tMinConf, tMaxConf): Void
- + buscar(Heladera) : Heladera
- + agregarSensor(heladera, Sensor) : Void
- + quitarSensor(heladera, Sensor) : Void
- + obtenerEstado(Heladera) : EstadoHeladera

# PersonasServicio

- + crearHumana(nombre, direccion, documento, medioDeContacto, apellido, sexo) : PersonaHumana
- + crearJuridica(nombre, direccion, documento, medioDeContacto, tipoJuridico, rubro): PersonaJuridica
- + eliminar(idPersona) : Void
- + modificarHumana(persona, nombre, direccion, documento, medioDeContacto, apellido, sexo) : Void
- + modificarJuridica(persona, nombre, direccion, documento, medioDeContacto, tipoJuridico, rubro) : Void

#### **PremiosServicio**

- + canjearPremio(Premio, Colaborador) : Void
- + obtenerPremios() : Premio[]

#### **PersonasVulnerablesServicio**

- + eliminar(PersonaVulnerable) : Void
- + modificar(personaVul, cantidadDeMenores) : Void

## ColaboradoresServicio

- + crear(Persona, TipoContribucion[]) : Void
- + eliminar(Colaborador) : Void
- + modificar(Colaborador, TipoContribucion[]) : Void
- + obtenerTodos() : Colaborador[]
- + buscar(idColaborador) : Colaborador
- + asignarTarjeta(Colaborador, Tarjeta) : Void
- + obtenerPuntos(Colaborador) : Int
- + subscribirseAHeladera(Heladera, Suscripcion): Void
- + desubscribirseAHeladera(Suscripcion) : Void
- + agregarPuntos(Puntos) : Void

# AutorizacionesServicio

- + registrarAcceso(idTarjeta, TipoAcceso, Heladera) : Void
- + crearAutorizacion(idTarjeta, Heladera) : Void

## IncidentesServicio

- + crearIncidente(int idHeladera, Incidente incidente) : Void
- + notificarVisita(int idIncidente, int idTecnico, string foto, DateTime fecha, string com) : Void

## RecomendacionesServicio

+ obtenerPuntosRecomendados(lat, long, radio) : PuntoEstrategico[]

# Infraestructura **Recomendacion Ubicacion Heladeras** Recomendaciones **DatosResponse UbicacionResponse** - longitud: String - data: DatosResponse[] - latitud: String - direccion: DireccionResponse <<instantiate>> DireccionResponse ConsultoraExternaApi - Calle: String - url: readonly String - Numero: String + GetRecomendacion(latitud, longitud, radio): PuntoEstrategico[] - Localidad: String - CodigoPostal: String **Importacion Colaboradores**

