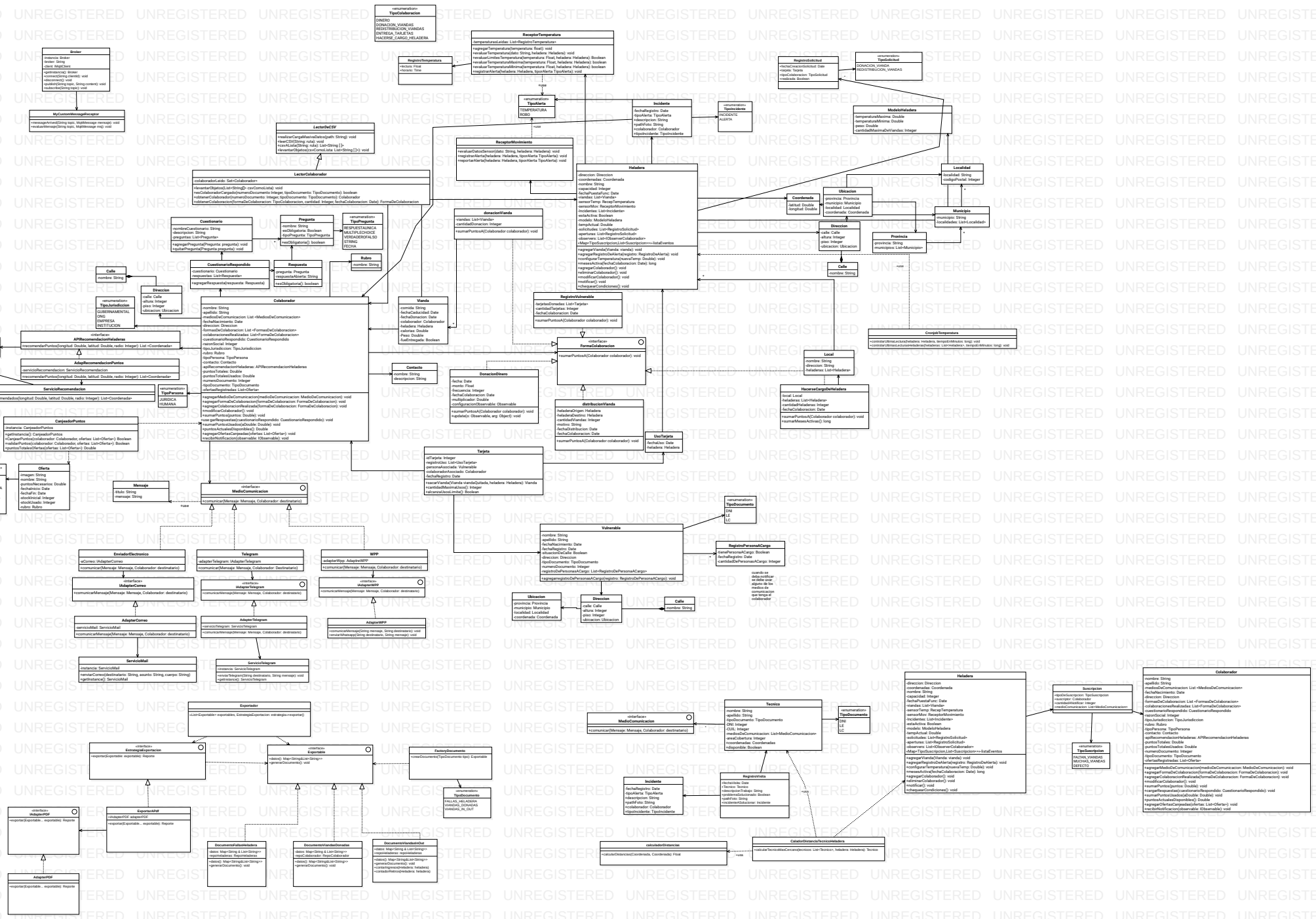
Model: Main


«enumeration»
TipoColaboracion
DINERO
DONACION_VIANDAS
DONACION_VIANDAS
ENTREGA_TARJETAS
HACERSE_CARGO_HELADERA
Broker
-instancia: Broker
-broker: String
-client: MqttClient
+getInstancia(): Broker
+connect(String clientId): void
+disconnect(): void
+publish(String topic, String content): void
+subscribe(String topic): void
MyCustomMessageReceptor
+messageArrived(String topic, MqttMessage message): void
+evaluarMensaje(String topic, MqttMessage mq): void
RegistroTemperatura
-lecturas: Float
-horario: Time
ReceptorTemperatura
-temperaturasLeidas: List<RegistroTemperatura>
+agregarTemperatura(temperatura: float): void
+evaluarTemperatura(date: String, heladera: Heladera): void
+evaluarLimitesTemperatura(temperatura: Float, heladera: Heladera): Boolean
+evaluarTemperaturaMaxima(temperatura: Float, heladera: Heladera): boolean
+evaluarTemperaturaMinima(temperatura: Float, heladera: Heladera): boolean
+registrarAlertaTeladera: Heladera, tipoAlerta TipoAlerta): void
«enumeration»
TipoAlerta
TEMPERATURA
ROBO
RegistroSolicitud
+fechaCreacionSolicitud: Date
+motivo: Texto
+tipoColaboracion: TipoSolicitud
+mediadas: Boolean
«enumeration»
TipoSolicitud
DONACION_VIANDA
REDISTRIBUCION_VIANDAS
Incidente
-fechaRegistro: Date
-tipoAlerta: TipoAlerta
-descripcion: String
-pathFoto: String
-colaborador: Colaborador
+tipoIncidente: TipoIncidente
«enumeration»
TipoIncidente
INCIDENTE
ALERTA
ModeloHeladera
-temperaturaMaxima: Double
-temperaturaMinima: Double
-peso: Double
-cantidadMaximaDeViandas: Integer
LectorDeCSV
+realizarCargaMasivaDatos(path: String): void
+leerCSV(String: ruta): void
+csvALista(String: ruta): List<String []>
+levantarObjetos(csvComoLista: List<String []>): void
ReceptorMovimiento
+evaluarDatosSensor(dato: String, heladera: Heladera): void
+registrarAlerta(heladera: Heladera, tipoAlerta TipoAlerta): void
+reportarAlerta(heladera: Heladera, tipoAlerta TipoAlerta): void
LectorColaborador
-colaboradorLeido: Set<Colaborador>
+levantarObjetos(List<String[]> csvComoLista): void
+esColaboradorCargado(numeroDocumento Integer, tipoDocumento: TipoDocumento): boolean
+obtenerColaborador(numeroDocumento: Integer, tipoDocumento: TipoDocumento): Colaborador
+obtenerColaboracion(formaDeColaboracion: TipoColaboracion, cantidad: Integer, fechaColaboracion: Date): FormaDeColaboracion
Heladera
-direccion: Direccion
-coordenadas: Coordenada
-nombre: String
-capacidad: Integer
-fechaPuestaFunc: Date
-viandas: List<Vianda>
-sensorTemp: RecepTemperatura
-sensorMov: ReceptorMovimiento
-incidentes: List<Incidente>
-estaActiva: Boolean
-modelo: ModeloHeladera
-tempActual: Double
-solicitudes: List<RegistroSolicitud>
-aperturas: List<RegistroSolicitud>
-observers: List<IObserverColaborador>
+Map<TipoSuscripcion,List<Suscripcion>>>listaEventos
+agregarVianda(Vianda: vianda): void
+agregarRegistroDeAlerta(registro: RegistroDeAlerta): void
+configurarTemperatura(nueva Temp: Double): void
+mesesActiva(fechaColaboracion: Date): long
+agregarColaborador(): void
+eliminarColaborador(): void
+modificar(): void
+chequeaeCondiciones(): void
Localidad
-localidad: String
-codigoPostal: Integer
Coordenada
-latitud: Double
-longitud: Double
Ubicacion
-provincia: Provincia
-municipio: Municipio
-localidad: Localidad
-coordenada: Coordenada
Municipio
-municipio: String
-localidades: List<Localidad>
Cuestionario
-nombreCuestionario: String
-descripcion: String
-preguntas: List<Pregunta>
+agregarPregunta(Pregunta: pregunta): void
+quitarPregunta(Pregunta: pregunta): void
Pregunta
-nombre: String
-esObligatoria: Boolean
-tipoPregunta: TipoPregunta
+esObligatoria(): boolean
«enumeration»
TipoPregunta
RESPUESTAUNICA
MULTIPLECHOICE
VERDADERO/FALSO
STRING
FECHA
Rubro
-nombre: String
donacionVianda
+viandas: List<Vianda>
-cantidadDonacion: Integer
+sumarPuntosA(Colaborador colaborador): void
Direccion
-calle: Calle
-altura: Integer
-piso: Integer
-ubicacion: Ubicacion
Provincia
-provincia: String
-municipios: List<Municipio>
Calle
-nombre: String
CuestionarioRespondido
-cuestionario: Cuestionario
-respuestas: List<Respuesta>
+agregarRespuesta(respuesta: Respuesta)
Respuesta
-pregunta: Pregunta
-respuestaAbierta: String
+esObligatoria(): boolean
Calle
-nombre: String
Direccion
-calle: Calle
-altura: Integer
-piso: Integer
-ubicacion: Ubicacion
«enumeration»
TipoJurisdiccion
GUBERNAMENTAL
ONG
EMPRESA
INSTITUCION
Colaborador
-nombre: String
-apellido: String
-mediosDeComunicacion: List <MedioDeComunicacion>
-fechaNacimiento: Date
-direccion: Direccion
-formasDeColaboracion: List <FormasDeColaboracion>
-colaboraciones Realizadas: List<FormaDeColaboracion>
-cuestionarioRespondido: CuestionarioRespondido
-razonSocial: Integer
-tipoJurisdiccion: TipoJurisdiccion
-rubro: Rubro
-tipoPersona: TipoPersona
-contacto: Contacto
-apiRecomendacionHeladeras: APIRecomendacionHeladeras
-puntosTotales: Double
-puntosTotalesUsados: Double
-numeroDocumento: Integer
-tipoDocumento: TipoDocumento
-ofertasRegistradas: List<Oferta>
+agregarMedioDeComunicacion(medioDeComunicacion: MedioDeComunicacion): void
+agregarFormaDeColaboracion(formaDeColaboracion: FormaDeColaboracion): void
+agregarColaboracionRealizada(formaDeColaboracion: FormaDeColaboracion): void
+modificarColaborador(): void
+sumarPuntos(puntos: Double): void
+cargarRespuestas(cuestionarioRespondido: CuestionarioRespondido): void
+sumarPuntosUsados(usDouble: Double): void
+puntosActualesDisponibles(): Double
+agregarOfertasCanjeadas(ofertas: List<Oferta>): void
+recibirNotificacion(observable: IObservable): void
Vianda
-comida: String
-fechaCaducidad: Date
-fechaDonacion: Date
-colaborador: Colaborador
-heladera: Heladera
-calorias: Double
-Peso: Double
-fueEntregada: Boolean
Contacto
-nombre: String
-descripcion: String
«interface»
FormaDeColaboracion
+sumarPuntosA(Colaborador colaborador): void
RegistroVulnerable
-tarjetasEntregadas: List<Tarjeta>
-cantidadTarjetas: Integer
-fechaColaboracion: Date
+sumarPuntosA(Colaborador colaborador): void
CronjobTemperatura
+controlarUltimaLecturaHeladera: Heladera, tiempoEnMinutos: long): void
+controlarUltimasLecturasHeladera(heladeras : List<Heladera> , tiempoEnMinutos: long): void
Local
-nombre: String
-direccion: String
-heladeras: List<Heladera>
HacerseCargoDeHeladera
-local: Local
-heladeras: List<Heladera>
-cantidadHeladeras: Integer
-fechaColaboracion: Date
+sumarPuntosA(Colaborador colaborador): void
+sumarMesesActivas(): long
DonacionDinero
-fecha: Date
-monto: Float
-frecuencia: Integer
-fechaColaboracion: Date
-multiplicador: Double
-configuracionObservable: Observable
+sumarPuntosA(Colaborador colaborador): void
+update(o: Observable, arg: Object): void
DistribucionVianda
-heladeraOrigen: Heladera
-heladeraDestino: Heladera
-cantidadViandas: Integer
-motivo: String
-fechaDistribucion: Date
-fechaColaboracion: Date
+sumarPuntosA(Colaborador colaborador): void
UsoTarjeta
-fechaUso: Date
-heladera: Heladera
«enumeration»
TipoPersona
JURIDICA
HUMANA
«interface»
APIRecomendacionHeladeras
+recomendarPuntos(longitud: Double, latitud: Double, radio: Integer): List <Coordenada>
Coordenada
-latitud: Double
-longitud: Double
AdapRecomendacionPuntos
-servicioRecomendacion: ServicioRecomendacion
+recomendarPuntos(longitud: Double, latitud: Double, radio: Integer): List<Coordenada>
ServicioRecomendacion
+puntosRecomendados(longitud: Double, latitud: Double, radio: Integer): List<Coordenada>
CanjeadorPuntos
-instancia: CanjeadorPuntos
+getInstancia(): CanjeadorPuntos
+CanjearPuntos(colaborador: Colaborador, ofertas: List<Oferta>): Boolean
+validarPuntos(colaborador: Colaborador, ofertas: List<Oferta>): Boolean
+puntosTotalesOfertas(ofertas: List<Oferta>): Double
Tarjeta
-idTarjeta: Integer
-registroUso: List<UsoTarjeta>
-personaAsociada: Vulnerable
-colaboradorAsociado: Colaborador
-fechaRegistro: Date
+sacarVianda(Vianda viandaQuitada, heladera: Heladera): Vianda
+cantidadMaximaUsos(): Integer
+alcanzaUsosLimite(): Boolean
«enumeration»
Rubro
SALUD
EDUCACION
GASTRONOMIA
HOGAR
ELECTRONICA
BIENESTAR
INTERIOR
TECNOLOGIA
Oferta
-imagen: String
-nombre: String
-puntosNecesarios: Double
-fechaInicio: Date
-fechaFin: Date
-stockInicial: Integer
-stockUsado: Integer
-rubro: Rubro
Mensaje
-titulo: String
-mensaje: String
«interface»
MedioComunicacion
+comunicar(Mensaje: Mensaje, Colaborador: destinatario)
«enumeration»
TipoDocumento
DNI
LE
LC
Vulnerable
-nombre: String
-apellido: String
-fechaNacimiento: Date
-fechaRegistro: Date
-situacionDeCalle: Boolean
-direccion: Direccion
-tipoDocumento: TipoDocumento
-numeroDocumento: Integer
-registroDePersonasACargo: List<RegistroDePersonaACargo>
+agregaregistroDePersonasACargo(registro: RegistroDePersonaACargo): void
RegistroPersonaACargo
-fenePersonaACargo: Boolean
-fechaRegistro: Date
-cantidadDePersonasACargo: Integer
cuando se deba notificar se debe usar el medio de comunicacion que tenga el colaborador
EnviadorElectronico
-eCorreo: IAdapterCorreo
+comunicar(Mensaje: Mensaje, Colaborador: destinatario)
Telegram
-adapterTelegram: IAdapterTelegram
+comunicar(Mensaje: Mensaje, Colaborador: Destinatario)
WPP
-adapterWpp: AdaptreWPP
+comunicar(Mensaje: Mensaje, Colaborador: destinatario)
«interface»
IAdapterCorreo
+comunicarMensaje(Mensaje: Mensaje, Colaborador: destinatario)
«interface»
IAdapterTelegram
+comunicarMensaje(Mensaje: Mensaje, Colaborador: destinatario)
«interface»
IAdapterWPP
+comunicarMensaje(Mensaje: Mensaje, Colaborador: destinatario)
Ubicacion
-provincia: Provincia
-municipio: Municipio
-localidad: Localidad
-coordenada: Coordenada
Direccion
-calle: Calle
-altura: Integer
-piso: Integer
-ubicacion: Ubicacion
Calle
-nombre: String
AdaptorCorreo
-servicioMail: ServicioMail
+comunicarMensaje(Mensaje: Mensaje, Colaborador: destinatario)
AdaptorTelegram
-servicioTelegram: ServicioTelegram
+comunicarMensaje(Mensaje: Mensaje, Colaborador: destinatario)
AdaptorWPP
+comunicarMensaje(String mensaje, String destinatario): void
+enviarWhatsapp(String destinatario, String mensaje): void
ServicioMail
-instancia: ServicioMail
+enviarCorreo(destinatario: String, asunto: String, cuerpo: String)
+getInstance(): ServicioMail
ServicioTelegram
-instancia: ServicioTelegram
+enviarTelegram(String destinatario, String mensaje): void
+getInstancia(): ServicioTelegram
Exportador
+List<Exportable> exportables, EstrategiaExportacion: estrategia=+exportar()
«interface»
EstrategiaExportacion
+exportar(Exportable: exportable): Reporte
«interface»
Exportable
+datos(): Map<String,List<String>>
+generarDocumento(): void
FactoryDocumento
+crearDocumento(TipoDocumento tipo): Exportable
«enumeration»
TipoDocumento
FALLAS_HELADERA
VIANDAS_DONADAS
VIANDAS_IN_OUT
«interface»
IAdapterPDF
+exportar(Exportable... exportable): Reporte
ExportarAPdf
+IAdapterPDF: adapterPDF
+exportar(Exportable... exportable): Reporte
AdapterPDF
+exportar(Exportable... exportable): Reporte
DocumentoFallasHeladera
-datos: Map<String,List<String>>
-repoHeladeras: RepoHeladeras
+datos(): Map<String,List<String>>
+generarDocumento(): void
DocumentoViandasDonadas
-datos: Map<String,List<String>>
-repoColaborador: RepoColaborador
+datos(): Map<String,List<String>>
+generarDocumento(): void
DocumentoViandasEnOut
-datos: Map<String,List<String>>
-repoHeladeras: repoHeladeras
+datos(): Map<String,List<String>>
+generarDocumento(): void
+contabilizarMovs(heladera: heladera)
+contabilizarMovs(heladera: heladera)
«interface»
MedioComunicacion
+comunicar(Mensaje: Mensaje, Colaborador: destinatario)
Tecnico
-nombre: String
-apellido: String
-tipoDocumento: TipoDocumento
-DNI: Integer
-CUIL: Integer
-mediosDeComunicacion: List<MedioDeComunicacion>
-areaCobertura: Integer
+coordenadas: Coordenadas
+disponible: Boolean
«enumeration»
TipoDocumento
DNI
LE
LC
Heladera
-direccion: Direccion
-coordenadas: Coordenada
-nombre: String
-capacidad: Integer
-fechaPuestaFunc: Date
-viandas: List<Vianda>
-sensorTemp: RecepTemperatura
-sensorMov: ReceptorMovimiento
-incidentes: List<Incidente>
-estaActiva: Boolean
-modelo: ModeloHeladera
-tempActual: Double
-solicitudes: List<RegistroSolicitud>
-aperturas: List<RegistroSolicitud>
-observers: List<IObserverColaborador>
+Map<TipoSuscripcion,List<Suscripcion>>>listaEventos
+agregarVianda(Vianda: vianda): void
+agregarRegistroDeAlerta(registro: RegistroDeAlerta): void
+configurarTemperatura(nueva Temp: Double): void
+mesesActiva(fechaColaboracion: Date): long
+agregarColaborador(): void
+eliminarColaborador(): void
+modificar(): void
+chequeaeCondiciones(): void
Suscripcion
-tipoDeSuscripcion: TipoSuscripcion
-suscriptor: Colaborador
-cuantidadNotificar: Integer
-medioComunicacion: List<MedioComunicacion>
«enumeration»
TipoSuscripcion
FALTAN_VIANDAS
MUCHAS_VIANDAS
DEFECTO
Colaborador
-nombre: String
-apellido: String
-mediosDeComunicacion: List <MedioDeComunicacion>
-fechaNacimiento: Date
-direccion: Direccion
-formasDeColaboracion: List <FormasDeColaboracion>
-colaboraciones Realizadas: List<FormaDeColaboracion>
-cuestionarioRespondido: CuestionarioRespondido
-razonSocial: Integer
-tipoJurisdiccion: TipoJurisdiccion
-rubro: Rubro
-tipoPersona: TipoPersona
-contacto: Contacto
-apiRecomendacionHeladeras: APIRecomendacionHeladeras
-puntosTotales: Double
-puntosTotalesUsados: Double
-numeroDocumento: Integer
-tipoDocumento: TipoDocumento
-ofertasRegistradas: List<Oferta>
+agregarMedioDeComunicacion(medioDeComunicacion: MedioDeComunicacion): void
+agregarFormaDeColaboracion(formaDeColaboracion: FormaDeColaboracion): void
+agregarColaboracionRealizada(formaDeColaboracion: FormaDeColaboracion): void
+modificarColaborador(): void
+sumarPuntos(puntos: Double): void
+cargarRespuestas(cuestionarioRespondido: CuestionarioRespondido): void
+sumarPuntosUsados(usDouble: Double): void
+puntosActualesDisponibles(): Double
+agregarOfertasCanjeadas(ofertas: List<Oferta>): void
+recibirNotificacion(observable: IObservable): void
RegistroVista
+fechaVista: Date
+Tecnico: Tecnico
+descripcionTrabajo: String
+fueRealizadoSatisfactorio: Boolean
+pathFoto: String
+incidenteASolucionar: incidente
Incidente
-fechaRegistro: Date
-tipoAlerta: TipoAlerta
-descripcion: String
-pathFoto: String
-colaborador: Colaborador
+tipoIncidente: TipoIncidente
calculadorDistancias
+calcularDistancias(Coordenada, Coordenada): Float
CalcularDistanciaTecnicoHeladera
+calcularTecnicosMasCercanos(tecnicos: List<Tecnico>, heladera: Heladera): Tecnico

Model::Main