PERSONA
VisitaHeladera
-fechaVisita: LocalDateTime
- trabajoRealizado: String
- foto: String
- incidenteSolucionado: Boolean
- trabajoCompletado: Boolean
- incidenteASolucionar: Incidente
<<Enum>>
TipoDocumento
DNI
LIBRETA_CIVICA
LIBRETA_DE_ENROLAMIENTO
Documento
-numeroDocumento: String
-tipoDocumento: TipoDocumento
Tecnico
- apellido:String
- cuil: String
- areasDeCobertura: List<Zona>
+ visitar(Incidente)
<<Enum>>
TipoJuridico
GUBERNAMENTAL
ONG
EMPRESA
INSTITUCION
PersonaEnSituacionVulnerable
- fechaNac:LocalDate
-fechaRegistro: LocalDate
- TieneMenores: Boolean
-canMenores: Int
- tarjetaHeladera: TarjetaHeladera
Rol
- estaActivo: Boolean
+ darDeBaja()
PersonaHumana
+ nombre:String
+ apellido:String
+ fechaNacimiento:LocalDate
PersonaJurdica
- razonSocial: String
- tipo: TipoJurdico
- rubro: String
- recomendadorDeUbicaciones: IAdapter
+ obtenerUbicaciones(zona):List<Ubicacion>
<<Interface>>
MedioDeContacto
+ notificar(Mensaje mensaje)
Mensaje
- titulo: String
- contenido: String
- fechaEnviado: LocalDateTime
- destinatario: Persona
Telefono
- telefono: String
+ notificar(Mensaje mensaje)
Whatsapp
- telefono: String
+ notificar(Mensaje mensaje)
CorreoElectronico
- correoElectronico: String
+ notificar(Mensaje mensaje)
(A) Persona
- direccion: String
- rol: Rol
- documento: Documento
- mediosDeContacto: List<MedioDeContacto>
+ cambiarRol(nuevoRol)
+ getNombre(): String
Colaborador
- formasContribucion: List<Contribucion>
- puntaje:float
- ofertasCanjeadas: List<Oferta>
-tarjeta: TarjetaColaborador
+agregarContribucion(contribucion: Contribucion)
+ calcularPuntaje()
+ canjearOferta(oferta: Oferta)
+ puedeCanjearOferta(): Boolean
+ emitirAvisoHeladera(heladera):SolicitudDeApertura
+ reportarFallaTecnica()
+ cantidadDeVisitasRealizadas(): Int
+ asignarTarjetaTarjetaColaborador()
RECOMENDADOR
<<Interface>>
IAdapter
+ consultarUbicaciones(Zona):List<Ubicacion>
APIAdapter
+ consultarUbicaciones(Zona):List<Ubicacion>
Zona
- nombreZona: String
- ubicacion: Ubicacion
- radio: int
MIGRADOR
MigradorContribucion
- contribuciones: List<Colaborador>
- colaboradores: List<PersonaHumana>
+ cargarCSV(FileInputStream)
+ migrarContribucion(PersonaHumana,Contribucion)
+ migrarColaboradores()
CSVColaborador
- doucmento: Documento
- nombre: String
- apellido: String
-mail: String
- fechaColaboracion: LocalDate
- formaDeColaboracion: TipoColaboracion
-cantidad: Int
+ cargarColaborador(filaCSV: String)
EstrategiaDistribuirViandas
+ crearColaboracion(cantidad: Int): Contribucion
EstrategiaEntregarTarjetas
+ crearColaboracion(cantidad: Int): Contribucion
EstrategiaDonarViandas
+ crearColaboracion(cantidad: Int): Contribucion
VerificadorContribucion
VerificadorContribucion
+ puedeRealizarContribucion(Contribucion): Boolean
VerificadorContribucion
VerificadorContribucion
VerificadorContribucion
+ puedeRealizarContribucion(Contribucion): Boolean
VerificadorContribucion
+ puedeRealizarContribucion(Contribucion): Boolean
VerificadorContribucion
+ puedeRealizarContribucion(Contribucion): Boolean
VALIDADORES
VerificadorAperturaHeladera
+ puedeAbrirHeladera(heladera, colaborador): Boolean
VerificadorContribucion
+ puedeRealizarContribucion (Persona, Contribucion): boolean
TARJETA
SolicitudDeApertura
- heladera: Heladera
- tarjeta: TarjetaColaborador
- fechaCreacion: LocalDateTime
- contribucion: Contribucion
TarjetaColaborador
- limiteDeTiempoDeUso: Int
+ usar(duenio, heladera)
(A) Tarjeta
id: String
- usos: List<Uso>
+ usar(duenio, heladera)
AperturaHeladera
- heladera: Heladera
- horaIniAperturaLocalDateTime
- tarjeta: TarjetaColaborador
TarjetaHeladera
- limiteDeUsosDiarios: Int
- cantidadDeUsosEnEsteDia: Int
+ calcularLimiteTarjeta(Int canMenores)
+ usar(duenio, heladera)
COLABORACIONES
<<Enum>>
TipoDePersona
JURIDICA
HUMANA
(A) Contribucion
- tiposDePersona: List<TipoDePersona>
- coeficientePuntaje: float
- fecha: LocalDate
- roColaborador: Colaborador
+ ejecutarContribucion()
+ obtenerPuntaje()
OfrecerProductos
-ofertas: List<Oferta>
+ ejecutarContribucion()
+ obtenerPuntaje()
+agregarOferta(oferta: Oferta)
HacerseCargoDeUnaHeladera
-heladerasColocadas: List<Heladera>
+ ejecutarContribucion()
+ obtenerPuntaje()
+ agregarHeladera(heladera: Heladera)
RegistrarPersonasEnSituacionVulnerable
- personasRegistradas: List<Persona>
- tarjetasAEntregar: List <TarjetaHeladera>
- tarjetasEntregadas: Int
+ ejecutarContribucion()
+ obtenerPuntaje()
+ enviarTarjetasViaMail()
+agregarPersona(Persona)
DonacionDeDinero
- monto: int
- frecuencia: int
+ ejecutarContribucion()
+ puedeRealizarContribucion()
+ obtenerPuntaje()
DistribucionDeViandas
- origen: Heladera
- destino: Heladera
- cantidad: int
- motivo: String
+ ejecutarContribucion()
+ obtenerPuntaje()
DonacionDeViandas
- heladera: Heladera
- viandas: List<Vianda>
- cantidadDeViantas: int
+ ejecutarContribucion()
+ obtenerPuntaje()
+agregarVianda(vianda: Vianda)
Oferta
-nombre: String
-puntosNecesarios: Int
-imagenURL: String
Uso
- fechaHoraDeUso: LocalDateTime
- heladera: Heladera
MovimientoViandas
- cantidadDeViandasIntroducidas: int
- cantidadDeViandasSacadas: Int
- fechaMovimiento: LocalDateTime
Heladera
- ubicacion: Ubicacion
-direccion: String
-nombre: String
- capacidadEnViandas: int
- viandasEnHeladera: Int
- fechaInicioFuncionamiento: LocalDate
-estadoHeladeraActual: EstadoHeladera
- historialEstadosHeladeras: List<EstadoHeladera>
-temperaturaDeFuncionamiento: TemperaturaHeladera
- temperatura: Int
- historialMovimientos: List<MovimientoViandas>
- publisherViandasDisponibles: PublisherViandasDisponibles
- publisherViandasFaltantes: PublisherViandasFaltantes
- publisherDesperfecto: PublisherDesperfecto
+actualizarEstadoHeladera(nuevoEstado)
+ obtenerMesesActivos(): Int
+estaActiva(): boolean
+ notificarCambioVianda(int viandas, int capacidad)
+ verificarTemperatura()
Ubicacion
- latitud: float
-longitud: float
Vianda
- descripcion:String
- fechaProduccion:LocalDate
- fechaCaducidad:LocalDate
- colaboradorColaborador
- heladeraHeladera
- calorias:Int
- peso:Float
-entregaEntrega
Entrega
- estadoEntrega: EstadoEntrega
- fechaEntrega: LocalDate
<<Enum>>
EstadoEntrega
PENDIENTE
ENTREGADO
(A) Sensor
-heladera: Heladera
+ agregar(Heladera heladera)
+ quitar(Heladera heladera)
+notificar()
+ getEstadoHeladeras(): boolean
+ estaActiva()
SensorDeTemperatura
- temperatura: Int
+ getEstadoHeladera(): boolean
SensorDeMovimiento
-estaActivado: Bool
+ getEstadoHeladera(): boolean
TemperaturaHeladera
- temperaturaMinima: Integer
- temperaturaMaxima: Integer
EstadoHeladera
-estaActiva: bool
- fechaHoraInicio: LocalDateTime
- fechaHoraFin: LocalDateTime
mesesActivos(): int
SUSCRIPCIONES HELADERAS
(A) PublisherHeladera
+ void notificarATodos(Heladera heladera)
+ void suscribir(SubscripcionHeladera)
(A) SubscripcionHeladera
-subscriptor: Persona
+ notificar (Heladera)
PublisherViandasDisponibles
- subscripciones: List<SubscripcionViandasDisponibles>
+ void suscribir(SubscripcionViandasDisponibles)
SubscripcionViandasDisponibles
- cantidadDeViandas: int
+ void notificar(Heladera)
PublisherViandasFaltantes
- subscripciones: List<SubscripcionViandasFaltantes>
+ void suscribir(SubscripcionViandasFaltantes)
SubscripcionViandasFaltantes
- cantidadDeViandas: int
+ void notificar(Heladera)
Publisher/Desperfecto
- subscripciones: List<SubscripcionDesperfecto>
+ void suscribir(SubscripcionDesperfecto)
SubscripcionDesperfecto
+ void notificar(Heladera)
REPORTES
ItemReporteHeladera
- cantidad: Int
- heladera: Heladera
ItemReporteColaborador
- cantidad: Int
- colaborador: Colaborador
GeneradorDeReportes
- reporteCantidadDeFallasPorHeladera: List<ItemReporteHeladera>
- reporteCantidadDeViandasDoncadasPorHeladera: List<ItemReporteHeladera>
- reporteCantidadDeViandasRetiradasPorHeladera: List<ItemReporteHeladera>
- reporteCantidadDeViandasDistribuidasPorColaborador: List<ItemReporteColaborador>
+ generarReporteDeCantidadDeFallasPorHeladera(RepoFallaTecnica): void
+ generarReporteCantidadDeViandasColocadasPorHeladera(RepoHeladera): void
+ generarReporteCantidadDeViandasRetiradasPorHeladera(RepoHeladera): void
+ generarReporteCantidadDeViandasDistribuidasPorColaborador(RepoPersona): void
+ generarReporteSemanalmente(): void
REPOSITORIOS
RepoTarjetaHeladera
-tarjetas: List<TarjetaHeladera>
+ buscarTarjetaPorId(String id): TarjetaHeladera
+ agregarTarjeta(TarjetaHeladera)
RepoPersona
- personas: List<Persona>
+ agregar(Persona)
+ eliminar(Persona)
+tecnicoMasCercanoAHeladera(heladera): Tecnico
+ buscarPorNombre(String nombre): Persona
+ buscarPorNombreYRol(String nombre, Class rol): List<Persona>
RepoHeladera
- heladeras: List<Heladera>
+buscarHeladerasActivas(): List<Heladera>
+agregar(Heladera)
+ eliminar(Heladera)
+buscarHeladerasCercanasA(heladera): List<Heladera>
RepoSolicitudApertura
- solicitudesDeApertura: List<SolicitudDeApertura>
+ agregar(SolicitudDeApertura)
+ eliminar(SolicitudDeApertura)
+buscarSolicitudDeAperturaDeTarjeta(tarjeta): List<Aviso>
+ existeSolicitudDeAperturaDeTarjetaParaHeladera(tarjeta, heladera): boolean
RepoContribucion
- contribuciones: List<Contribucion>
+ agregar(Contribucion)
+ eliminar(Contribucion)
RepoMensaje
- mensajes: List<Mensaje>
+ agregar(Mensaje)
+ eliminar(Mensaje)
RepoVistaHeladera
- visitas: List<VistaHeladera>
+ agregar(VistaHeladera)
+ eliminar(VistaHeladera)
RepoUsuario
-usuarios: List<Usuario>
+ agregarUsuario(usuarioNuevo: Usuario): void
+existeNombreUsuarioRegistrado(nombreUsuario: String): boolean
+ obtenerUsuarioPorDocumento(documento: Documento): Usuario
+obtenerUsuarioPorNombreDeUsuarioYContrasena(nombreUsuario: String, contrasenia: String)
RepositorioAperturasHeladera
- aperturas: List<AperturaHeladera>
+ agregar(AperturaHeladera)
+ eliminar(AperturaHeladera)
RepoOferta
- ofertas: List<Oferta>
+ agregar(FallaTecnica)
+ eliminar(FallaTecnica)
RepoIncidente
- incidentes: List<Incidente>
+ agregar(Incidente)
+ eliminar(Incidente)
INCIDENTES
Alerta
- tipoDeAlerta: String
(A) Incidente
- heladera: Heladera
- tipoDeIncidente: String
- fechaEmision: LocalDateTime
+ avisarTecnico()
FallaTecnica
- colaborador: Persona
- descripcion: String
- foto: String
AUTENTICACION
RegistrarUsuario
-registrados: Registrados
+ registrarUsuario(String, Documento, String)
Usuario
- documento: Documento
- nombreUsuario
- colaborador: Persona
- fechaExpiracionContrasenia:LocalDateTime
-estanDatosCorrectosbool
- estanDatosCompletos: bool
+ cambiarContrasenia(Usuario, String)
IniciarSesion
- intentosRealizados: Int
- agregarSegundos: Int
+iniciarSesion(String, String):boolean
ValidarUsuario
+validarUsuarioIngresado(String,String):Usuario
VALIDADORES
VerificadorContrasenia
- validaciones: List<ValidacionContrasenia>
+ validarContraseña(contrasena: String): void
(A) ValidacionContrasenia
-mensaje: String
+ validar(contrasena: String)
+ condicion(contrasena: String): boolean
LongitudMinimaContrasenia
+ condicion(contrasena: String): boolean
CaracterRepetido
+ condicion(contrasena: String): boolean
ValidacionTopNuevesContrasenia
-contrasenias: List<String>
-nombreDeArchivo: String
+ condicion(contrasena: String): boolean
-cargarPeoresContraseniasDesdeArchivo()
FormatoContrasenia
+ condicion(contrasena: String): boolean
broker
Broker
- gestionarSolicitudAperturaSolicitudDeApertura(): Void
+ mandarTemperaturaHeladera(heladera heladera, int temperatura): Void
- gestionarAlerta(Alerta alerta): Void