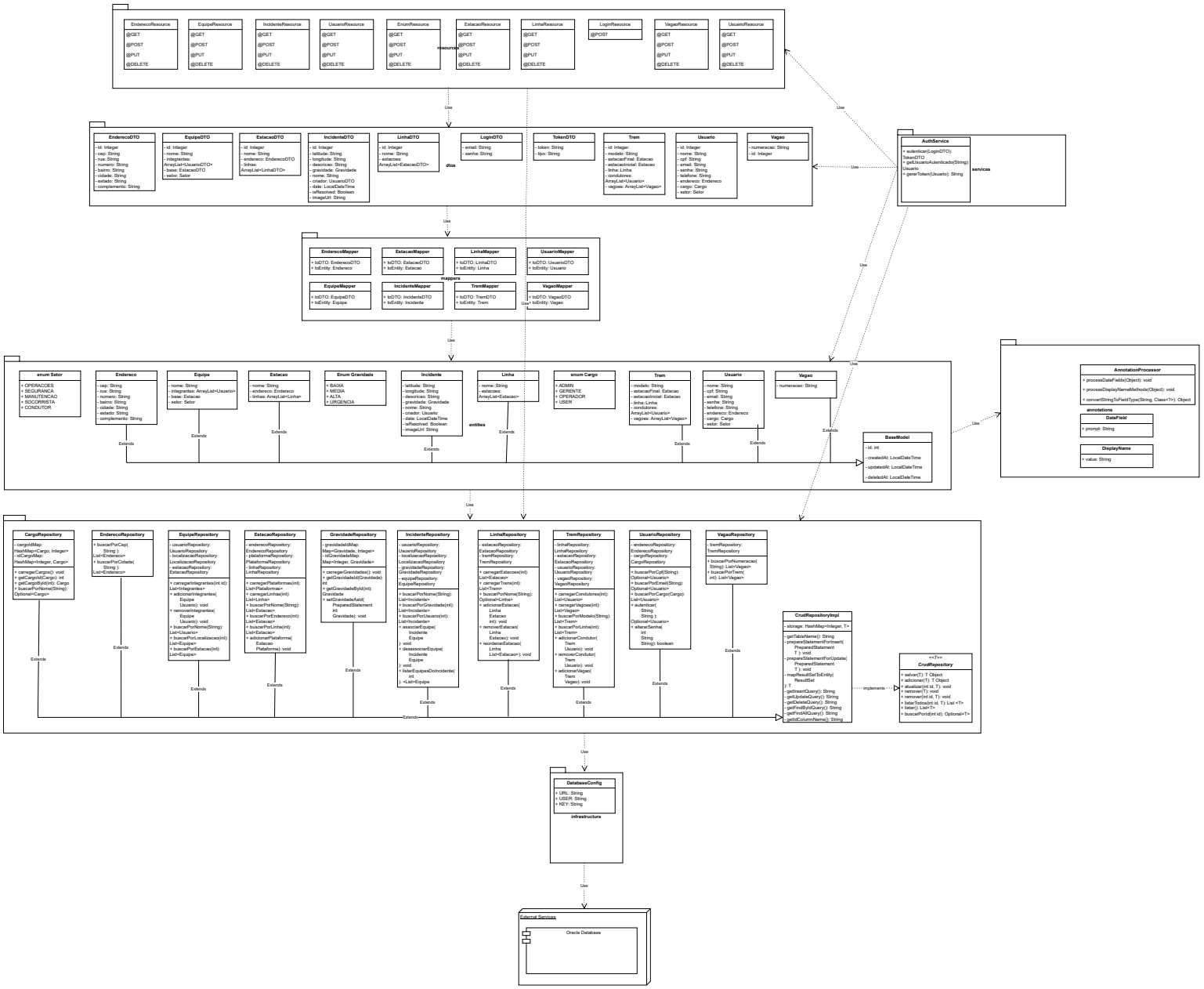
resources
EnderecoResource
@GET
@POST
@PUT
@DELETE
EquipeResource
@GET
@POST
@PUT
@DELETE
IncidenteResource
@GET
@POST
@PUT
@DELETE
UsuarioResource
@GET
@POST
@PUT
@DELETE
EnumResource
@GET
@POST
@PUT
@DELETE
EstacaoResource
@GET
@POST
@PUT
@DELETE
LinhaResource
@GET
@POST
@PUT
@DELETE
LoginResource
@POST
VagaoResource
@GET
@POST
@PUT
@DELETE
UsuarioResource
@GET
@POST
@PUT
@DELETE
Uses
dtos
EnderecoDTO
- id: Integer
- cep: String
- rua: String
- numero: String
- bairro: String
- cidade: String
- estado: String
- complemento: String
EquipeDTO
- id: Integer
- nome: String
- integrantes: ArrayList<UsuarioDTO>
- base: EstacaoDTO
- setor: Setor
EstacaoDTO
- id: Integer
- nome: String
- endereco: EnderecoDTO
- linhas: ArrayList<LinhaDTO>
IncidenteDTO
- id: Integer
- latitude: String
- longitude: String
- descricao: String
- gravidade: Gravidade
- nome: String
- criador: UsuarioDTO
- date: LocalDateTime
- isResolved: Boolean
- imageUrl: String
LinhaDTO
- id: Integer
- nome: String
- estacoes: ArrayList<EstacaoDTO>
LoginDTO
- email: String
- senha: String
TokenDTO
- token: String
- tipo: String
Trem
- id: Integer
- modelo: String
- estacaoFinal: Estacao
- estacaoInicial: Estacao
- linha: Linha
- condutores: ArrayList<Usuario>
- vagoes: ArrayList<Vagao>
Usuario
- id: Integer
- nome: String
- cpf: String
- email: String
- senha: String
- telefone: String
- endereco: Endereco
- cargo: Cargo
- setor: Setor
Vagao
- numeracao: String
- id: Integer
AuthService
+ autenticar(LoginDTO): TokenDTO
+ getUsuarioAutenticado(String): Usuario
+ gerarToken(Usuario): String
services
mappers
EnderecoMapper
+ toDTO: EnderecoDTO
+ toEntity: Endereco
EstacaoMapper
+ toDTO: EstacaoDTO
+ toEntity: Estacao
LinhaMapper
+ toDTO: LinhaDTO
+ toEntity: Linha
UsuarioMapper
+ toDTO: UsuarioDTO
+ toEntity: Usuario
EquipeMapper
+ toDTO: EquipeDTO
+ toEntity: Equipe
IncidenteMapper
+ toDTO: IncidenteDTO
+ toEntity: Incidente
TremMapper
+ toDTO: TremDTO
+ toEntity: Trem
VagaoMapper
+ toDTO: VagaoDTO
+ toEntity: Vagao
entities
enum Setor
+ OPERACOES
+ SEGURANCA
+ MANUTENCAO
+ SOCORRISTA
+ CONDUTOR
Endereco
- cep: String
- rua: String
- numero: String
- bairro: String
- cidade: String
- estado: String
- complemento: String
Equipe
- nome: String
- integrantes: ArrayList<Usuario>
- base: Estacao
- setor: Setor
Estacao
- nome: String
- endereco: Endereco
- linhas: ArrayList<Linha>
Enum Gravidade
+ BAIXA
+ MEDIA
+ ALTA
+ URGENCIA
Incidente
- latitude: String
- longitude: String
- descricao: String
- gravidade: Gravidade
- nome: String
- criador: Usuario
- date: LocalDateTime
- isResolved: Boolean
- imageUrl: String
Linha
- nome: String
- estacoes: ArrayList<Estacao>
enum Cargo
+ ADMIN
+ GERENTE
+ OPERADOR
+ USER
Trem
- modelo: String
- estacaoFinal: Estacao
- estacaoInicial: Estacao
- linha: Linha
- condutores: ArrayList<Usuario>
- vagoes: ArrayList<Vagao>
Usuario
- nome: String
- cpf: String
- email: String
- senha: String
- telefone: String
- endereco: Endereco
- cargo: Cargo
- setor: Setor
Vagao
- numeracao: String
BaseModel
- id: int
- createdAt: LocalDateTime
- updatedAt: LocalDateTime
- deletedAt: LocalDateTime
AnnotationProcessor
+ processDataFields(Object): void
+ processDisplayNameMethods(Object): void
+ convertString(ToFieldType(String, Class<?>): Object
annotations
DateField
+ prompt: String
DisplayName
+ value: String
CargoRepository
- cargoIdMap: HashMap<Cargo, Integer>
- idCargoMap: HashMap<Integer, Cargo>
+ carregarCargos(): void
+ getCargoId(Cargo): int
+ getCargoById(int): Cargo
+ buscarPorNome(String): Optional<Cargo>
EnderecoRepository
+ buscarPorCep(String): List<Endereco>
+ buscarPorCidade(String): List<Endereco>
EquipeRepository
- usuarioRepository: UsuarioRepository
- localizacaoRepository: LocalizacaoRepository
- estacaoRepository: EstacaoRepository
+ carregarIntegrantes(int id): List<Integrantes>
+ adicionarIntegrantes(Equipe Usuario): void
+ removerIntegrantes(Equipe Usuario): void
+ buscarPorNome(String): List<Usuario>
+ buscarPorLocalizacao(int): List<Equipe>
+ buscarPorEstacao(int): List<Equipe>
EstacaoRepository
- enderecoRepository: EnderecoRepository
- plataformaRepository: PlataformaRepository
- linhaRepository: LinhaRepository
+ carregarPlataformas(int): List<Plataformas>
+ carregarLinhas(int): List<Linha>
+ buscarPorNome(String): List<Estacao>
+ buscarPorEndereco(int): List<Estacao>
+ buscarPorLinha(int): List<Estacao>
+ adicionarPlataforma(Estacao Plataforma): void
GravidadeRepository
- gravidadeIdMap: Map<Gravidade, Integer>
- idGravidadeMap: Map<Integer, Gravidade>
+ carregarGravidades(): void
+ getGravidadeId(Gravidade): int
+ getGravidadeById(int): Gravidade
+ setGravidadeAsId(PreparedStatement int Gravidade): void
IncidenteRepository
- usuarioRepository: UsuarioRepository
- localizacaoRepository: LocalizacaoRepository
- gravidadeRepository: GravidadeRepository
- equipeRepository: EquipeRepository
+ buscarPorNome(String): List<Incidente>
+ buscarPorGravidade(int): List<Incidente>
+ buscarPorUsuario(int): List<Incidente>
+ associarEquipe(Incidente Equipe): void
+ desassociarEquipe(Incidente Equipe): void
+ listarEquipesDoIncidente(int): <ListEquipe
LinhaRepository
- estacaoRepository: EstacaoRepository
- TremRepository: TremRepository
+ carregarEstacoes(int): List<Estacao>
+ carregarTrens(int): List<Trem>
+ buscarPorNome(String): Optional<Linha>
+ adicionarEstacao(Linha Estacao int): void
+ removerEstacao(Linha Estacao): void
+ reordenarEstacao(Linha List<Estacao>): void
TremRepository
- linhaRepository: LinhaRepository
- estacaoRepository: EstacaoRepository
- usuarioRepository: UsuarioRepository
- vagaoRepository: VagaoRepository
+ carregarCondutores(int): List<Usuario>
+ carregarVagoes(int): List<Vagao>
+ buscarPorModelo(String): List<Trem>
+ buscarPorLinha(int): List<Trem>
+ adicionarCondutor(Trem Usuario): void
+ removerCondutor(Trem Usuario): void
+ adicionarVagao(Trem Vagao): void
UsuarioRepository
- enderecoRepository: EnderecoRepository
- cargoRepository: CargoRepository
+ buscarPorCpf(String): Optional<Usuario>
+ buscarPorEmail(String): Optional<Usuario>
+ buscarPorCargo(Cargo): List<Usuario>
+ autenticar(String String): Optional<Usuario>
+ alterarSenha(int String String): boolean
VagaoRepository
- tremRepository: TremRepository
+ buscarPorNumeracao(String): List<Vagao>
+ buscarPorTrem(int): List<Vagao>
CrudRepositoryImpl
- storage: HashMap<Integer, T>
- getTableName(): String
- prepareStatementForInsert(PreparedStatement T): void
- prepareStatementForUpdate(PreparedStatement T): void
- mapResultSetToEntity(ResultSet): T
- getInsertQuery(): String
- getUpdateQuery(): String
- getDeleteQuery(): String
- getFindByIdQuery(): String
- getFindAllQuery(): String
- getIdColumnName(): String
implements
<<T>>
CrudRepository
+ salvar(T): T Object
+ adicionar(T): T Object
+ atualizar(int id, T): void
+ remover(T): void
+ remover(int id, T): void
+ listarTodos(int id, T): List <T>
+ listar(): List<T>
+ buscarPorId(int id): Optional<T>
DatabaseConfig
+ URL: String
+ USER: String
+ KEY: String
Infrastructure
External Services
Oracle Database