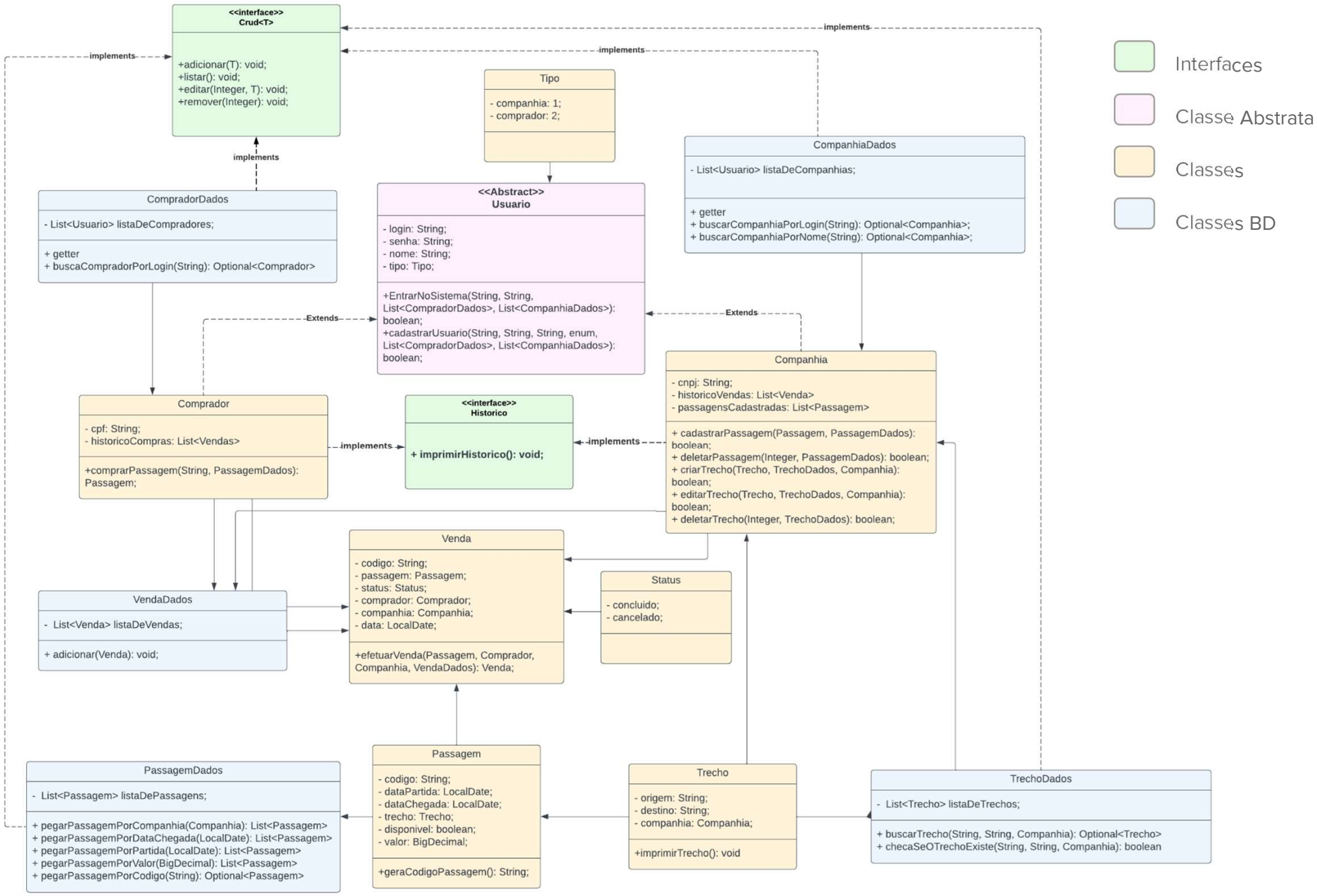
<<interface>>
Crud<T>
+adicionar(T): void;
+listar(): void;
+editar(Integer, T): void;
+remover(Integer): void;
implements
Tipo
- companhia: 1;
- comprador: 2;
CompanhiaDados
- List<Usuario> listaDeCompanhias;
+ getter
+ buscarCompanhiaPorLogin(String): Optional<Companhia>;
+ buscarCompanhiaPorNome(String): Optional<Companhia>;
CompradorDados
- List<Usuario> listaDeCompradores;
+ getter
+ buscaCompradorPorLogin(String): Optional<Comprador>
<<Abstract>>
Usuario
- login: String;
- senha: String;
- nome: String;
- tipo: Tipo;
+EntrarNoSistema(String, String, List<CompradorDados>, List<CompanhiaDados>): boolean;
+cadastrarUsuario(String, String, String, enum, List<CompradorDados>, List<CompanhiaDados>): boolean;
Extends
Comprador
- cpf: String;
- historicoCompras: List<Vendas>
+comprarPassagem(String, PassagemDados): Passagem;
<<interface>>
Historico
+ imprimirHistorico(): void;
Companhia
- cnpj: String;
- historicoVendas: List<Venda>
- passagensCadastradas: List<Passagem>
+ cadastrarPassagem(Passagem, PassagemDados): boolean;
+ deletarPassagem(Integer, PassagemDados): boolean;
+ criarTrecho(Trecho, TrechoDados, Companhia): boolean;
+ editarTrecho(Trecho, TrechoDados, Companhia): boolean;
+ deletarTrecho(Integer, TrechoDados): boolean;
Venda
- codigo: String;
- passagem: Passagem;
- status: Status;
- comprador: Comprador;
- companhia: Companhia;
- data: LocalDate;
+efetuarVenda(Passagem, Comprador, Companhia, VendaDados): Venda;
Status
- concluido;
- cancelado;
VendaDados
- List<Venda> listaDeVendas;
+ adicionar(Venda): void;
PassagemDados
- List<Passagem> listaDePassagens;
+ pegarPassagemPorCompanhia(Companhia): List<Passagem>
+ pegarPassagemPorDataChegada(LocalDate): List<Passagem>
+ pegarPassagemPorPartida(LocalDate): List<Passagem>
+ pegarPassagemPorValor(BigDecimal): List<Passagem>
+ pegarPassagemPorCodigo(String): Optional<Passagem>
Passagem
- codigo: String;
- dataPartida: LocalDate;
- dataChegada: LocalDate;
- trecho: Trecho;
- disponivel: boolean;
- valor: BigDecimal;
+geraCodigoPassagem(): String;
Trecho
- origem: String;
- destino: String;
- companhia: Companhia;
+imprimirTrecho(): void
TrechoDados
- List<Trecho> listaDeTrechos;
+ buscarTrecho(String, String, Companhia): Optional<Trecho>
+ checaSeOTrechoExiste(String, String, Companhia): boolean
Interfaces
Classe Abstrata
Classes
Classes BD