Classe UML

Enums

<<enum>> Estados
Todos estados do Brasil

<<enum>> Tipo
- <<enum>> RESIDENCIAL
- <<enum>> COMERCIAL
- + ofTipo(tipo: Integer): Tipo

<<enum>> TipoUsuario
- <<enum>> CLIENTE
- <<enum>> ESPECIALISTA
- <<enum>> ADMIN
- + ofTipo(tipo: Integer): TipoMuda

<<enum>> StatusEntrega
- <<enum>> RECEBIDO
- <<enum>> ENVIADO
- <<enum>> ENTREGUE
- + ofTipo(tipo: Integer): TamanhoMuda

<<enum>> TipoMuda
- <<enum>> PLANTA
- <<enum>> ARVORE
- + ofTipo(tipo: Integer): TipoMuda

Classes

Usuario
- - id: int
- - nome: String
- - email: String
- - senha: String
- - tipoUsuario: TipoUsuario
- + Usuario()
- + Usuario(nome: String, email: String, senha: String, tipoUsuario: TipoUsuario)
- + getters e setters
- + toString(): String

Admin
- + Admin(nome: String, email: String, senha: String)
- + getters e setters
- + toString(): String

Especialista
- - contato: Contato
- - documento: String
- - especializacao: String
- - regiaoResponsavel: Estados
- + Especialista()
- + Especialista(nome: String, email: String, senha: String, cpf: String, contato: Contato, documento: String, especializacao: String, regiaoResponsavel: Estados)
- + getters e setters
- + toString(): String

Cliente
- - cpf: String
- - endereco: ArrayList<Endereco>
- - contatos: ArrayList<Contato>
- - mudas: ArrayList<Muda>
- - entregas: ArrayList<Entrega>
- + Cliente()
- + Cliente(nome: String, email: String, senha: String, cpf: String)
- + getters e setters
- + imprimirContatos(): void
- + imprimirEnderecos(): void
- + imprimirMudas(): void
- + imprimirEntregas(): void
- + adicionarContato(contato: Contato): void
- + adicionarEndereco(endereco: Endereco): void
- + adicionarMuda(muda: Muda): void
- + adicionarEntrega(entrega: Entrega): void
- + toString(): String

Relatorio
- - id: int
- - dono: Cliente
- - avaliador: Especialista
- - muda: Muda
- - estadoMuda: String
- - sugestoes: String
- - cidade: String
- - avaliacaoEspecialista: double
- + Relatorio(dono: Cliente, muda: Muda, numero: String, estadoMuda: String, sugestoes: String)
- + getters e setters
- + toString(): String

Contato
- - id: int
- - descricao: String
- - numero: String
- - tipo: Tipo
- + Contato()
- + Contato(descricao: String, numero: String, tipo: String)
- + getters e setters
- + toString(): String

Muda
- - id: int
- - tipo: TipoMuda
- - nome: String
- - nomeCientifico: String
- - porte: TamanhoMuda
- - ambienteIdeal: String
- - descricao: String
- - especialistasDeReparo: List<Especialista>
- + Muda(tipo: int, nome: String, nomeCientifico: String, porte: int, ambienteIdeal: String, descricao: String)
- + getters e setters
- + toString(): String

Endereco
- - id: int
- - cep: String
- - logradouro: String
- - complemento: String
- - cidade: String
- - estado: String
- - tipo: Tipo
- + Endereco()
- + Endereco(cep: String, logradouro: String, numero: String, complemento: String, cidade: String, estado: String, tipo: Tipo)
- + getters e setters
- + toString(): String

Entrega
- - id: int
- - mudas: List<Muda>
- - status: StatusEntrega
- - cliente: Cliente
- - enderecoDeEntrega: Endereco
- + Entrega()
- + Entrega(mudas: List<Muda>, status: StatusEntrega, cliente: Cliente)
- + getters e setters
- + atualizarEntrega(status: StatusEntrega): void
- + consultarMudaEntrega(): void
- + toString(): String

Interfaces

<<interface>> IEntrega
- + atualizarEntrega(status: StatusEntrega): void
- + consultarMudaEntrega(): void

<<interface>> IService<E>
- + adicionar(e: E): void
- + deletar(id: int): void
- + editar(id: int, e: E): boolean
- + procurarPorID(id: int): E
- + deletar(id: int): void
- + listarTodos(): List<E>
- + procurar(id: int): Optional<E>

<<interface>> IServiceUsuario
- + procurarPorEmail(email: Status): Optional<Usuario>

Services

ServiceCliente
- - servicoContato: ServiceContato
- - servicoEndereco: ServiceEndereco
- + adicionar(cliente: Cliente): void
- + deletar(id: int): void
- + editar(id: int, clienteEditado: Cliente): boolean
- + editarContato(idCliente: int, idContato: int, contatoEditado: Contato): boolean
- + editarEndereco(idCliente: int, idEndereco: int, enderecoEditado: Endereco): boolean
- + procurarPorID(id: int): Cliente
- + listarTodos(): List<Cliente>
- + procurar(id: int): Optional<Cliente>
- + procurarPorEmail(email: String): Optional<Cliente>
- + adicionarContato(cliente: Cliente, contato: Contato): void
- + adicionarEndereco(cliente: Cliente, endereco: Endereco): void
- + adicionarMuda(cliente: Cliente, muda: Muda): void
- + adicionarEntregas(cliente: Cliente, entrega: Entrega): void
- + imprimirCliente(cliente: Cliente): void
- + imprimirMudasCliente(cliente: Cliente): void

ServiceEndereco
- + adicionar(contato: Endereco): void
- + deletar(id: int): void
- + editar(id: int, enderecoEditado: Endereco): boolean
- + procurarPorID(id: int): Endereco
- + listarTodos(): List<Endereco>
- + procurar(id: int): Optional<Endereco>

ServiceEntrega
- + adicionar(entrega: Entrega): void
- + deletar(id: int): void
- + editar(id: int, entregaAtualizada: Entrega): Entrega
- + procurarPorID(id: int): Entrega
- + listarTodos(): List<Entrega>
- + procurar(id: int): Optional<Entrega>

ServiceContato
- + adicionar(contato: Contato): void
- + deletar(id: int): void
- + editar(id: int, contatoEditado: Contato): boolean
- + procurarPorID(id: int): Contato
- + listarTodos(): List<Contato>
- + procurar(id: int): Optional<Contato>

ServiceRelatorio
- + adicionar(relatorio: Relatorio): void
- + deletar(id: int): void
- + editar(id: int, relatorio: Relatorio): boolean
- + procurarPorID(id: int): Relatorio
- + listarTodos(): List<Relatorio>
- + procurar(id: int): Optional<Relatorio>

ServiceAdmin
- + adicionar(admin: Admin): void
- + deletar(id: int): void
- + editar(id: int, adminEditado: Admin): boolean
- + procurarPorID(id: int): Admin
- + listarTodos(): List<Admin>
- + procurar(id: int): Optional<Admin>
- + procurar(id: int): Optional<Admin>
- + procurarPorEmail(email: String): Optional<Admin>

ServiceEspecialista
- + adicionar(especialista: Especialista): void
- + deletar(id: int): void
- + editar(id: int, especialistaEditado: Especialista): boolean
- + procurarPorID(id: int): Especialista
- + listarTodos(): List<Especialista>
- + procurarPorEmail(email: String): Optional<Especialista>
- + procurarRelatorioPorEmail(email: String): List<Relatorio>

ServiceMudas
- + adicionar(muda: Muda): void
- + deletar(id: int): void
- + editar(id: int, muda: Muda): boolean
- + procurarPorID(id: int): Muda
- + listarTodos(): List<Muda>
- + procurar(id: int): Optional<Muda>

ServiceUsuario
- - servicoCliente: ServiceCliente
- - servicoEspecialista: ServiceEspecialista
- + logar(email: String, senha: String): Usuario
- + procurarPorEmail(email: String): Optional<Usuario>