Diagrama de Classes AreaDeRisco - id_areaderisco: Integer - id_localizacao: Integer - classificacao: enum {"Alto", "Médio", "Baixo"} + classificarRisco(): String + verificarRiscoLocalizacao(localizacao: Localização): Boolean 1 define 4 Usuario Localizacao # id_usuario: Integer - cpf: String - id_localizacao: Integer - nome: String - data_nasc: LocalDate - latitude: Double - telefone: String - longitude : Double - email: String - registro_data_hora: LocalDateTime - senha: String - bairro: String + tipo_usuario: Enum {"Passageiro", "Motorista"} - cidade: String avalia 1 - estado: String - pais: String + criarConta(): Boolean + atualizarPerfil(novoNome: String, novoEmail: + registrarLocalizacao(latitude: Double, longitude: String, novoTelefone: String): Boolean Double, bairro: String, cidade: String, estado: String, + deletarConta(): Boolean pais: String): Boolean + removerLocalizacao(): Boolean + login(email: String, senha: String): Boolean + calcularMediaRisco(): Double + logout(): void + avaliarLocalizacao(localizacao: Localizacao, + listarAvaliacoes(): List<Avaliacao> nota: Integer, comentario: String): Boolean + visualizarAvaliacoes(): List<Avaliacao> 1 1 1 Veiculo Motorista Passageiro - Modelo: String - Cor: String - Placa: String + validadeCNH: Date + avaliacaoMedia: Decimal(2,1) cadastra - CNH: String + avaliacaoMedia: Decimal(2,1) + metodoPagamento: String + atualizarDadosVeiculo(novoModelo: + cadastrarVeiculo(veiculo: + solicitarCorrida(origem: Localizacao, String, novaCor: String, novaPlaca: Veiculo): Boolean destino: Localizacao): Boolean String): Boolean + aceitarCorrida(): Boolean + visualizarHistoricoCorridas(): + removerVeiculo(): Boolean + finalizarCorrida(): Boolean List<Corrida>