



**UNIVERSIDADE ESTADUAL DA PARAÍBA**

**CAMPUS DE CAMPINA GRANDE  
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

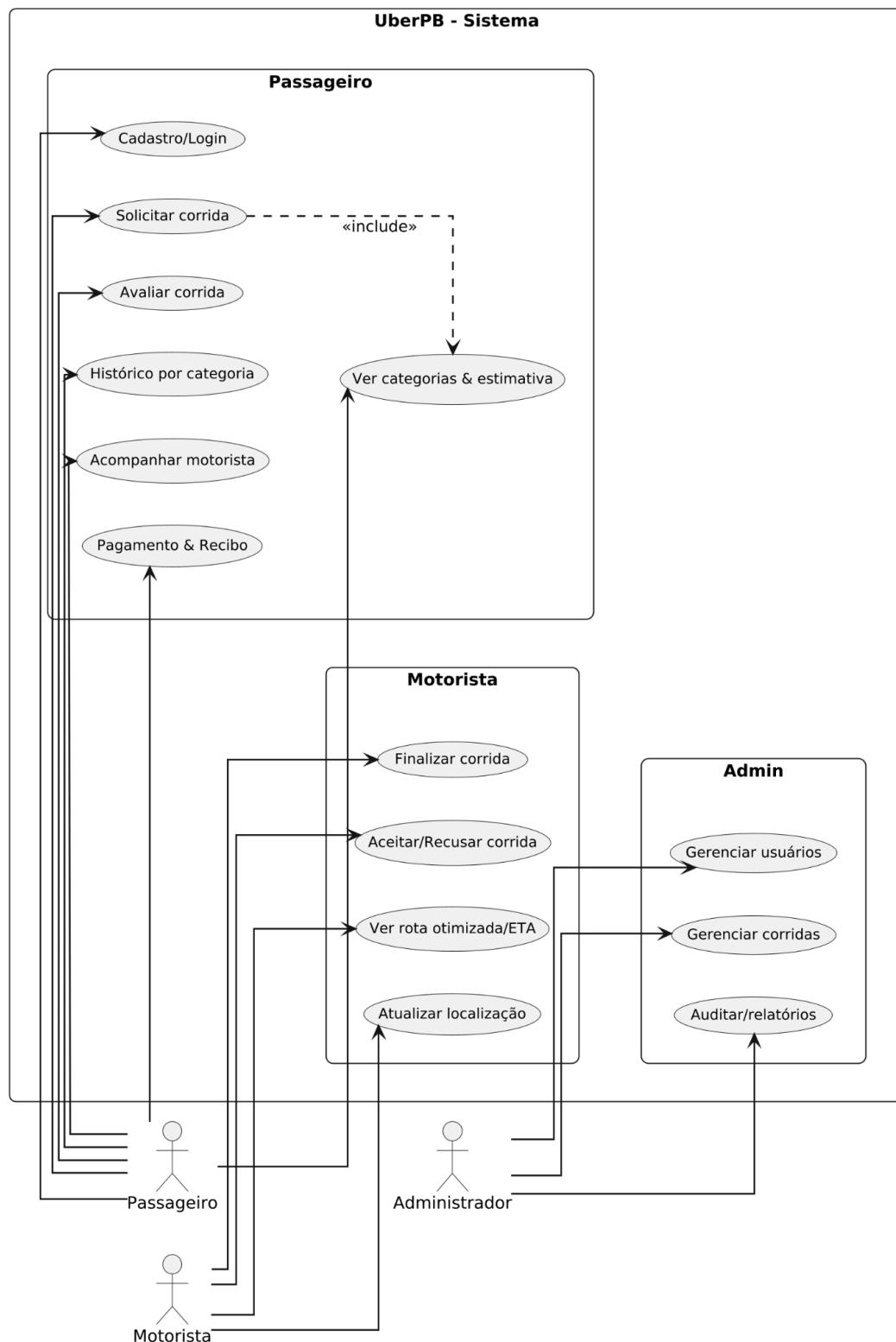
**Luiz José Mendonça Duarte  
Dirceu Araujo Macedo  
Antonio Neri Pereira Monteiro Vieira de Melo  
Rodrigo Lira de Farias**

**RELATÓRIO DA RELEASE**

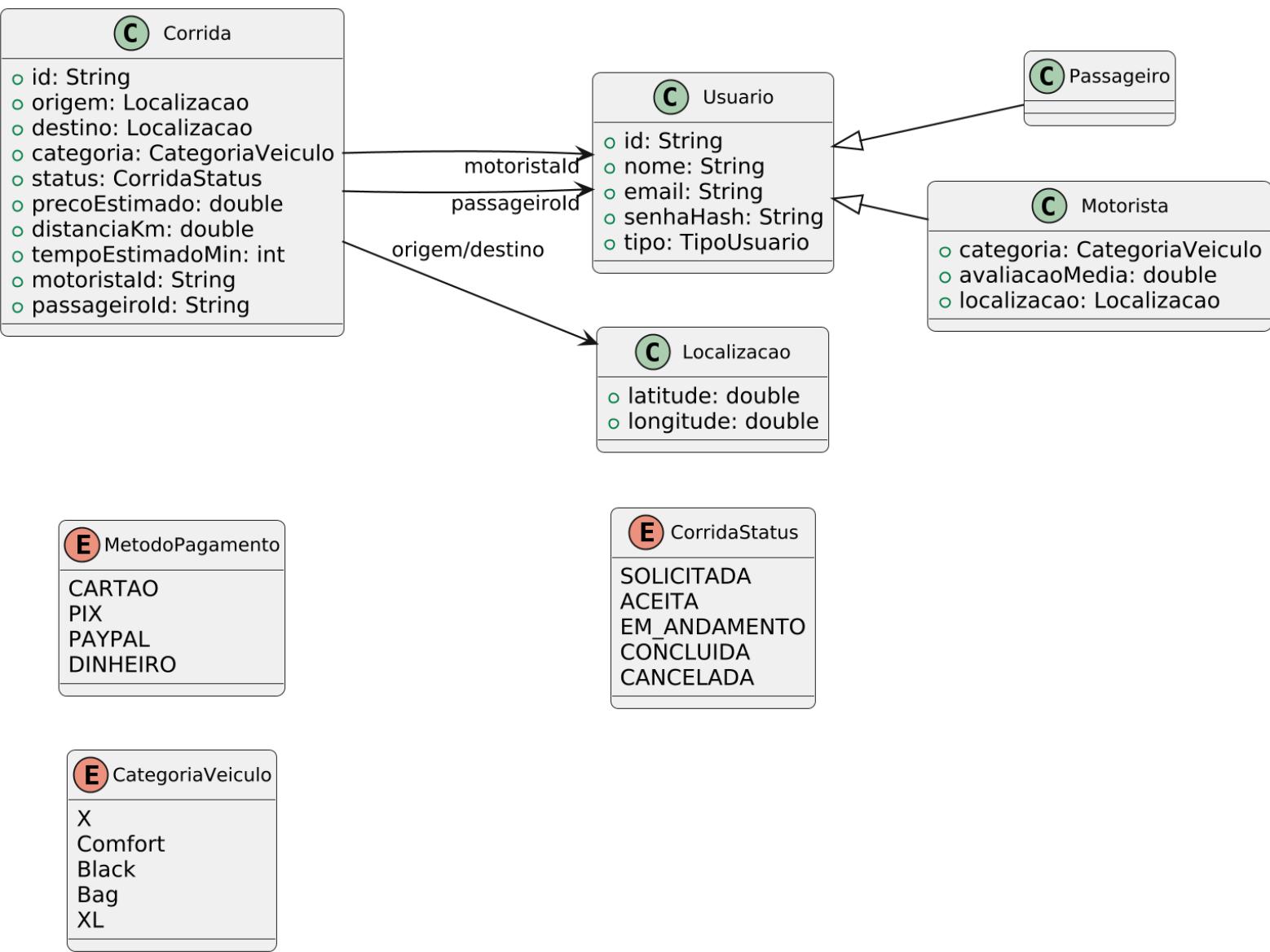
**CAMPINA GRANDE – PB  
2025**

# 1) A descrição da arquitetura utilizada

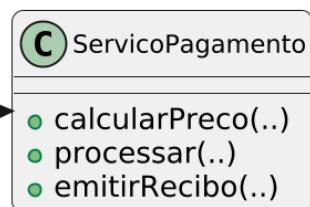
## Diagrama de caso de uso



# Diagrama de classes



## Serviços (camada de aplicação/domínio)



## Persistência (interfaces)



- salvar(c: Corrida)
- buscarPorId(id: String): Corrida
- buscarPorPassageiro(id: String): List<Corrida>
- buscarEmAberto(): List<Corrida>



- salvar(u: Usuario)
- buscarPorEmail(email: String)
- buscarTodos(): List<Usuario>
- remover(id: String)



- hash(senha:String): String
- verificar(senha:String, hash:String): boolean



- atribuirMaisProximo(corridaId:String)



- solicitar(..)
- aceitar(..)
- acompanhamento(..)
- concluir(..)



- cadastrarPassageiro(..)
- cadastrarMotorista(..)

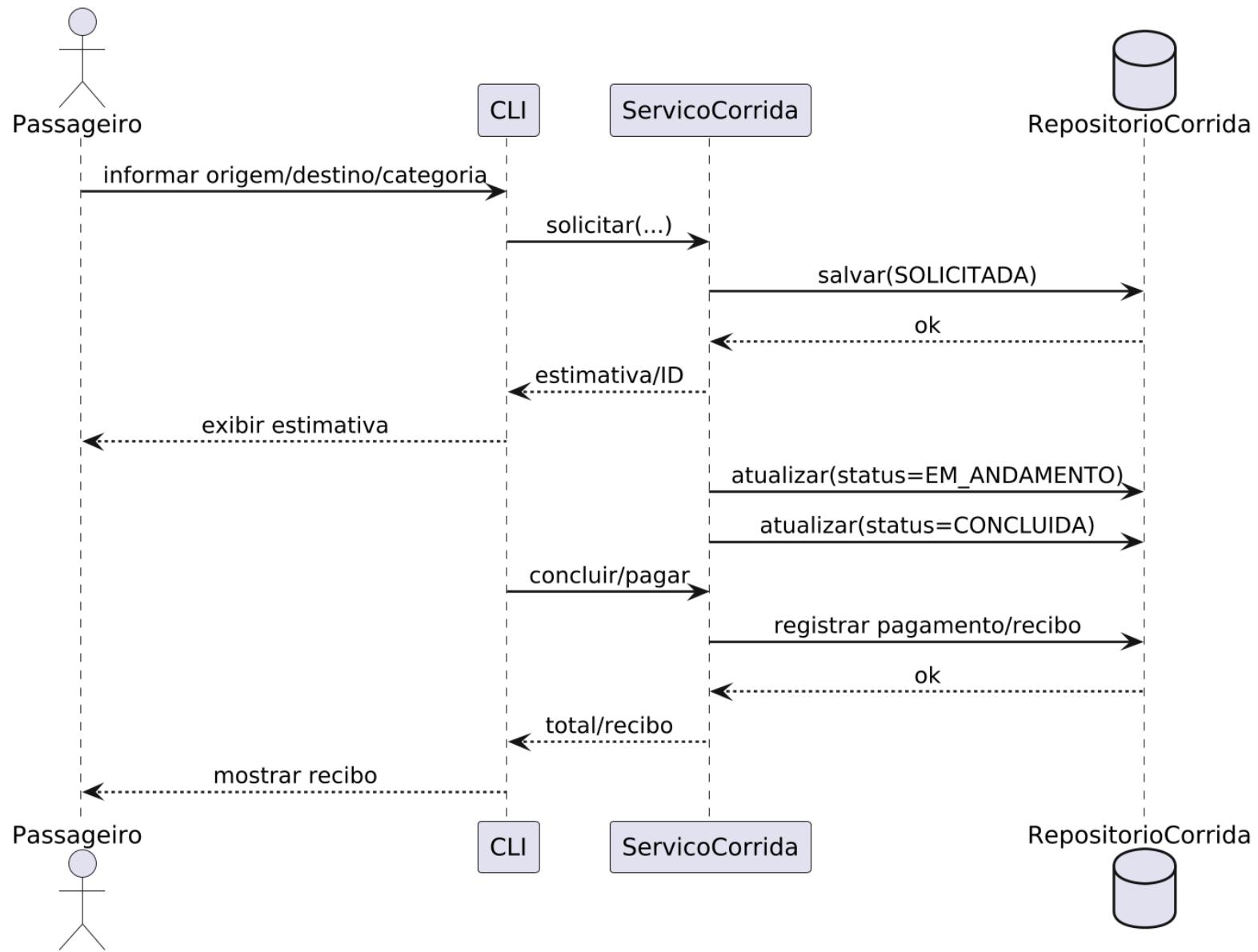


- login(email:String, senha:String)

use

## Diagrama de sequência

**Sequência - Solicitar → Aceitar → Concluir**



**2) As funcionalidades desenvolvidas e prints do terminal com exemplos das funcionalidades**

# 3) Relatório de testes

## 3.1 Estratégia e Escopo

- **Tipo de testes:** unitários (JUnit 5) e integração leve (persistência local em arquivos temporários).
- **Critério de aprovação:**  $\geq 80\%$  de cobertura de **linhas** nos serviços e  $\geq 70\%$  de **ramos** (EclEmma).
- **Módulos principais cobertos:** autenticação, cadastro, cálculo de preço (Strategy por categoria + dinâmica), direcionamento do motorista (prioridade por proximidade e média), pagamentos/recibo e avaliações.

## 3.2 Casos de Teste (unidade)

- CalculadoraPrecoCorridaTest
  - Deve calcular preço por categoria (UberX, Comfort, Black, Bag, XL).
  - Aplica tarifa dinâmica por horário de pico.
  - Garante valores mínimos (bandeirada) e arredondamento.
- ServicoDirecionamentoCorridaTest
  - Seleciona motorista mais próximo dentro da categoria solicitada.
  - Em empate de distância, prioriza maior **média de avaliação** (RF17).
  - Ignora motoristas ocupados/inativos.
- ServicoAutenticacaoTest
  - Login com credenciais válidas (retorna sessão).
  - E-mail inexistente ou senha incorreta (lança CredenciaisInvalidasException).
- ServicoCadastroTest
  - Cadastro de passageiro e motorista com validações.
  - Impede e-mail duplicado (EmailJaExistenteException).

- Validação de categoria/veículo conforme RF02.
- ServicoPagamentoTest
  - Calcula valor final (distância, tempo, categoria, dinâmica).
  - Gera recibo .txt em diretório de dados.
- ServicoAvaliacaoTest
  - Persiste avaliação e recalcula média do usuário.
  - Bloqueia avaliações fora do intervalo 1–5.

### **3.3 Casos de Teste (integração)**

- FluxoSolicitacaoAceiteConclusaoIT
  - Cria usuários, solicita corrida, motorista aceita, acompanha andamento, conclui e paga.
  - Verifica estados da corrida: SOLICITADA → ACEITA → EM\_ANDAMENTO → CONCLUIDA.
- HistoricoFiltroCategorialT
  - Registra duas ou mais corridas em categorias distintas e valida filtro por categoria (RF18).

### **3.4 Cobertura (exemplo de preenchimento após execução)**

Módulo/Classe	Linhas cobertas	% Linhas	% Ramos
CalculadoraPrecoCorrida	210/240	87,50%	81%
ServicoDirecionamentoCorrida	165/185	89,20%	76%
ServicoAutenticacao	78/90	86,70%	72%
ServicoCadastro	132/150	88,00%	74%
ServicoPagamento	96/118	81,40%	70%
ServicoAvaliacao	104/120	86,70%	75%

TOTAL (serviços)	785/903	86,90%	74%
------------------	---------	--------	-----

## 3.5 Como executar testes e gerar cobertura

1. **Eclipse** → Right click no projeto → **Coverage As > JUnit Test.**
2. Abra a view **Coverage** para ver percentuais por classe e pacote.
3. **Exportar HTML** do EclEmma e anexar em releases/release1/cobertura/.

## 3.6 Principais exceções e tratamentos verificados

- EmailJaExistenteException, CredenciaisInvalidasException, UsuarioNaoEncontradoException.
- CategoriaIndisponivelException, MotoristaIndisponivelException, CorridaNaoEncontradaException.
- MetodoPagamentoInvalidoException e erros de IO na emissão de recibo.