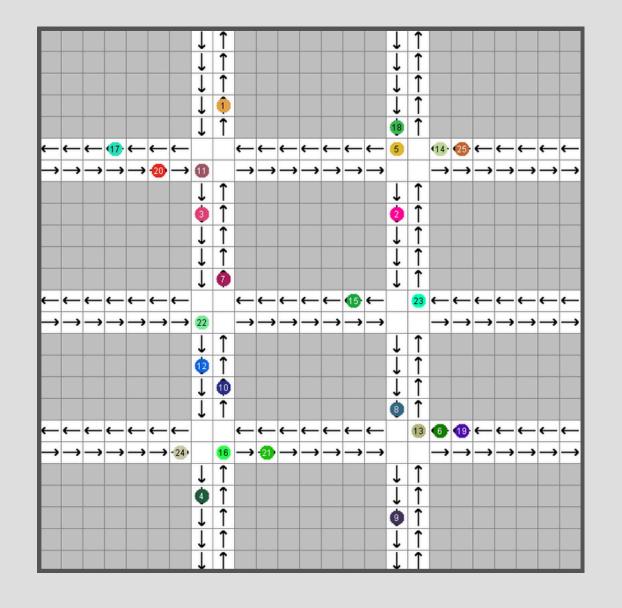
TRABALHO 2 - THREADS

Fernando Prim e Lucas Gitirana



SOBRE O PROJETO

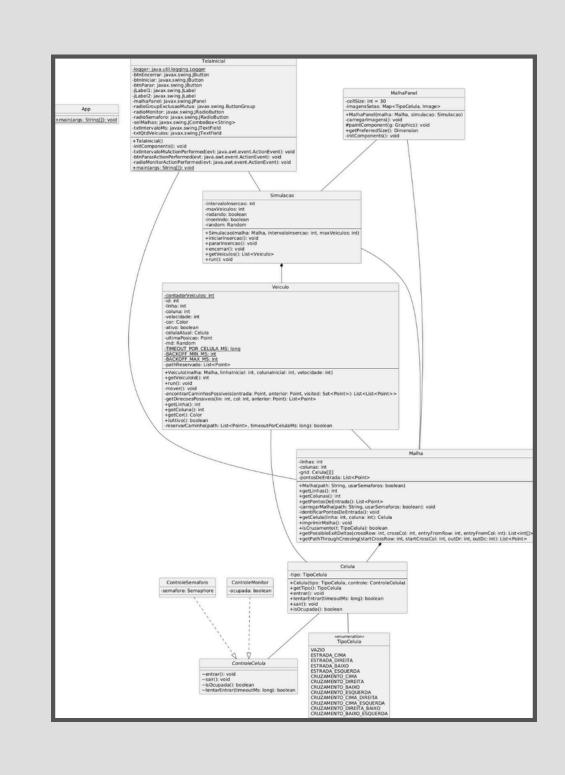
Esse projeto serve para simular uma malha viária, onde os veículos podem percorrer as ruas e cruzamentos de forma a evitar acidentes e sair da malha com segurança!

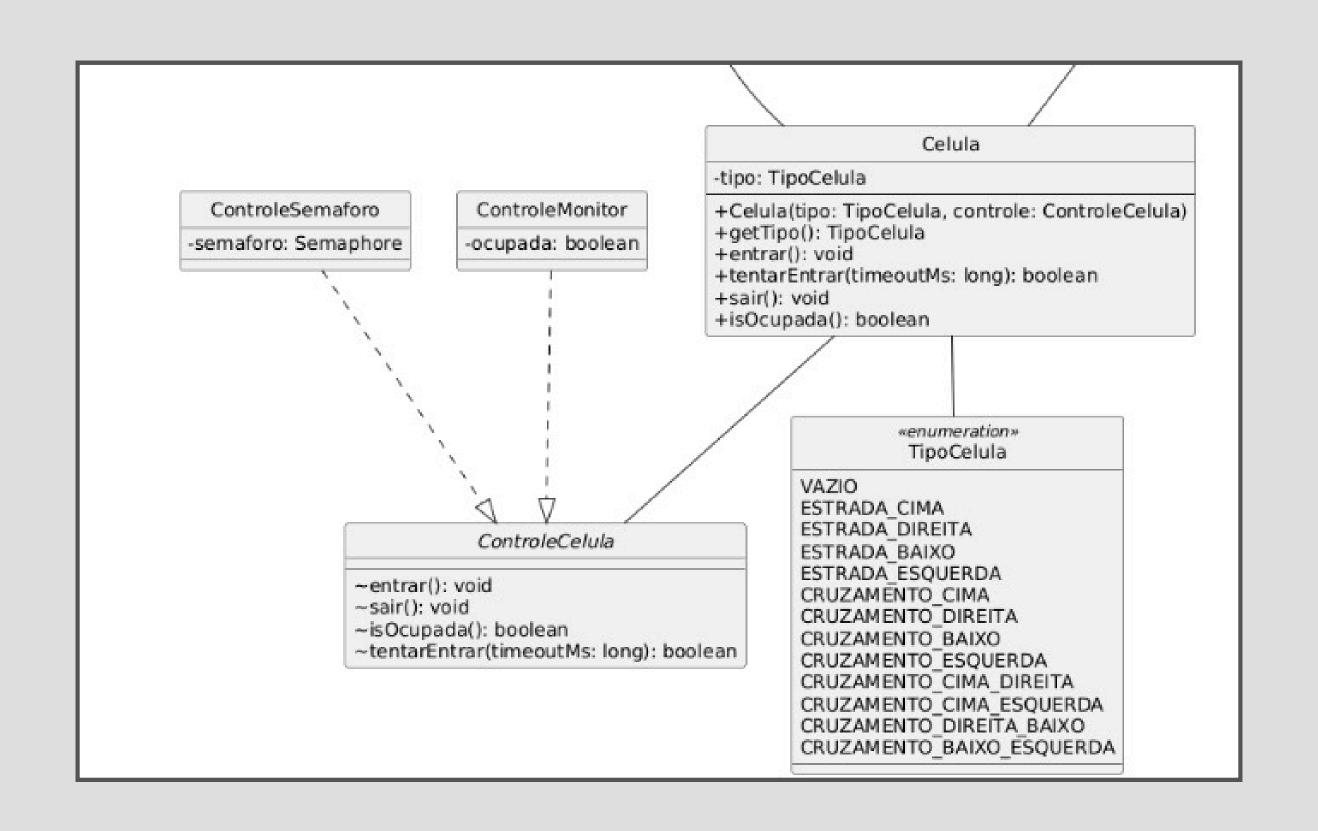


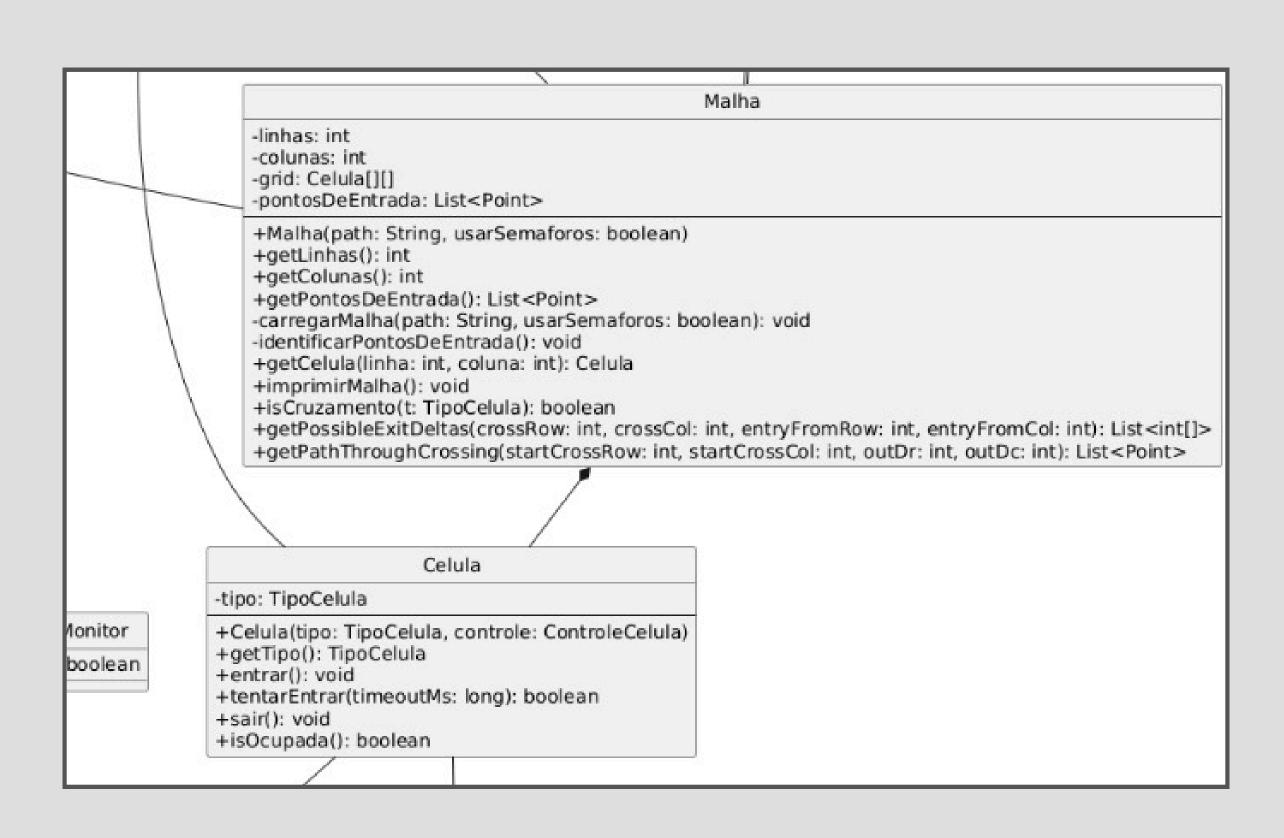
COMO?

O nosso sistema simula o tráfego viário em uma malha composta por células que podem representar estradas ou cruzamentos, com uso de threads e mecanismos de exclusão mútua (monitores ou semáforos) para evitar colisões.

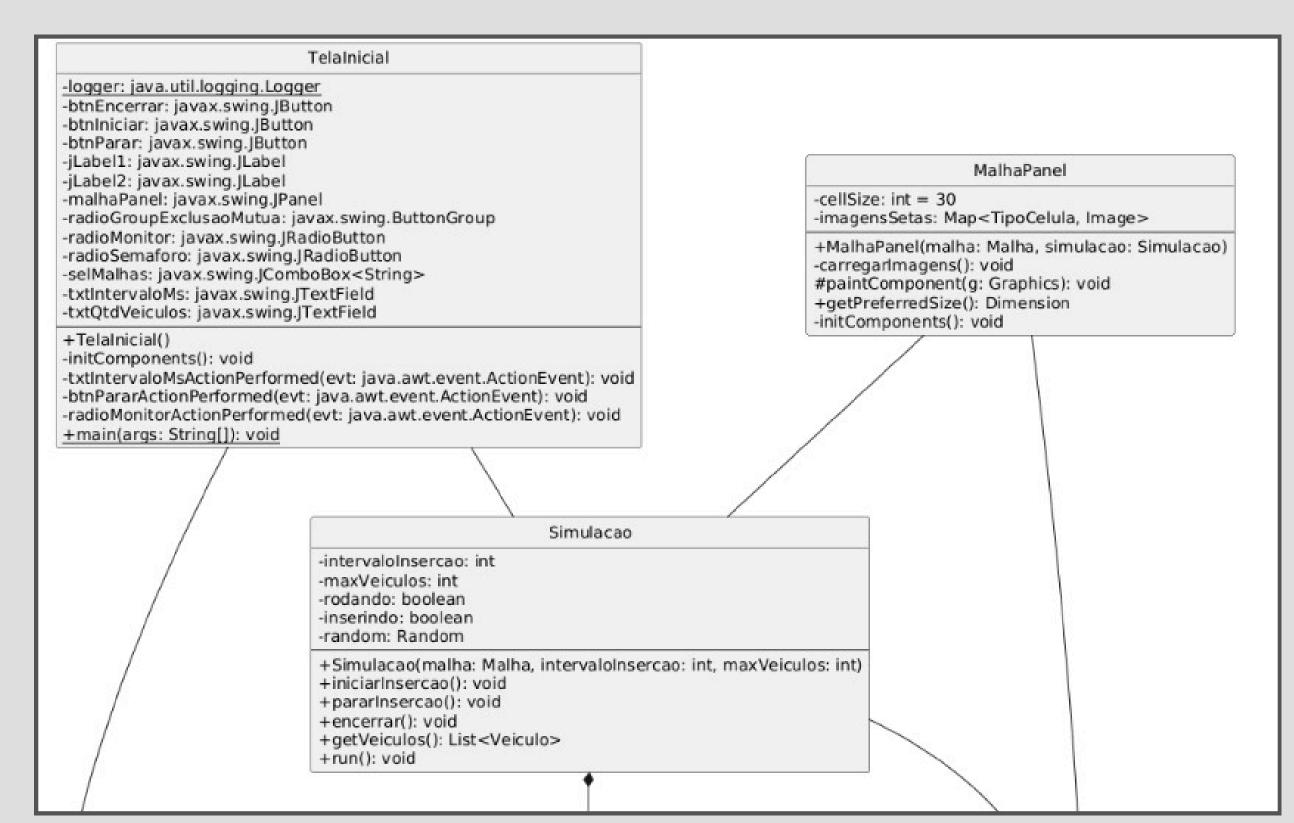
Nele, cada veículo atua como uma thread que é responsável por executar seus movimentos, enquanto a malha gerencia o acesso às células, garantindo reservas em ruas e cruzamentos para prevenir deadlocks e invasões.







Veiculo -contadorVeiculos: int -id: int -linha: int -coluna: int -velocidade: int -cor: Color -ativo: boolean -celulaAtual: Celula -ultimaPosicao: Point -md: Random -TIMEOUT POR CELULA MS: long -BACKOFF MIN MS: int -BACKOFF MAX MS: int -pathReservado: List<Point> +Veiculo(malha: Malha, linhalnicial: int, colunalnicial: int, velocidade: int) +getVeiculoId(): int +run(): void -mover(): void -encontrarCaminhosPossiveis(entrada: Point, anterior: Point, visited: Set<Point>): List<List<Point>> -getDirecoesPossiveis(lin: int, col: int, anterior: Point): List<Point> +getLinha(): int +getColuna(): int +getCor(): Color +isAtivo(): boolean -reservarCaminho(path: List<Point>, timeoutPorCelulaMs: long): boolean



PRINCIPAIS DIFICULDADES ENCONTRADAS

- Aparecimento e direcionamento correto dos veículos
- Lógica dos cruzamentos para evitar deadlocks e carros invadindo a contramão

DEMONSTRAÇÃO PRÁTICA

Hora de andar na BR!