

Trabalho Prático de Sistemas Distribuídos

(Avaliação Contínua)

Licenciatura em Engenharia Informática - LEI

Licenciatura em Segurança Informática em Redes de Computadores - LSIRC

Ano Letivo 2025/2026

Professor: Ronaldo Salles (rmo@estg.ipp.pt)

1. Introdução

O presente relatório descreve o projeto prático (avaliação contínua) da unidade curricular de Sistemas Distribuídos, cujo objetivo é o desenvolvimento de uma aplicação distribuída em Java que simula um sistema de tráfego urbano. A simulação envolve a interação entre veículos, cruzamentos e semáforos, representando o fluxo de tráfego numa malha urbana 3x3. O sistema deve ser implementado sobre uma arquitetura distribuída composta por processos e threads, com comunicação via sockets (TCP/UDP/Multicast) ou middleware.

2. Objetivo

O objetivo principal do projeto é projetar e implementar um simulador de tráfego urbano distribuído, que permita avaliar políticas de controlo de semáforos, desempenho sob diferentes cargas de tráfego e estratégias de seleção de rotas. O

sistema deve garantir sincronização entre processos, recolha de estatísticas e apresentação de resultados num dashboard interativo.

Os objetos do sistema são os veículos (carros, motos ou camiões) que transitam por uma área pré-definida passando por cruzamentos e semáforos até um ponto de saída do sistema. Cada cruzamento possui semáforos para controle do trânsito, a temporização dos semáforos deve ser configurada com o intuito de reduzir o congestionamento no sistema.

A aplicação (simulação) deve ser implementada sobre um sistema distribuído, utilizando **processos** e **threads** em Java, com comunicação por **sockets** (**TCP/UDP/Multicast**) ou middleware entre processos distribuídos. Os cruzamentos devem ser implementados por meio de **processos** e os semáforos por **threads**.

3. Cenário de Simulação

A simulação baseia-se numa malha 3x3, onde os nós representam pontos de entrada (E1, E2, E3), cruzamentos (Cr1 a Cr5) e um ponto de saída (S). As arestas correspondem a ruas que podem ser de sentido único ou duplo. Cada veículo segue um percurso definido até ao ponto de saída, sendo o tempo de deslocamento entre nós fixo (t).

As arestas da malha representam as ruas que podem ser de sentido único (apenas um sentido no fluxo do tráfego) ou duplo (dois sentidos do fluxo de tráfego). As figuras 1 e 2 a seguir ilustram esse ambiente simulado.

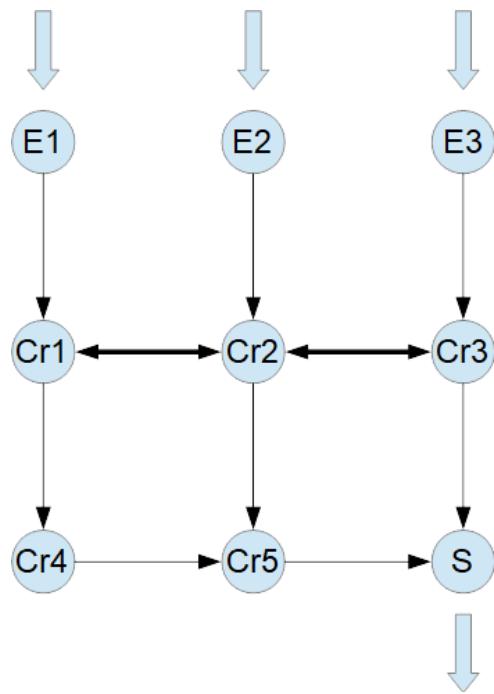


Figura.1: Os veículos entram no sistema pelos pontos E1, E2 ou E3; passam pelos cruzamentos (Cr1...Cr5) até alcançarem o ponto de saída S e abandonarem o sistema. O sentido das vias é indicado pelas setas e o tempo para o veículo sair de um ponto e chegar no seguinte é fixo e igual a t (esse é o tempo de deslocamento na rua, mas não considera o tempo de espera na fila em cada semáforo).

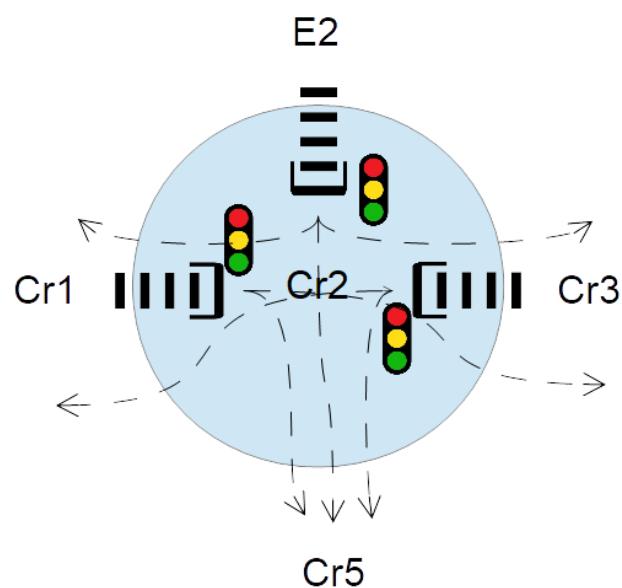


Figura.2: Visão do cruzamento 2 (Cr2) com as filas de veículos, semáforos e fluxo dos veículos que passam pelo cruzamento.

4. Veículos

Os veículos são representados como objetos com atributos próprios, incluindo identificador único, tipo (moto, carro, camião), ponto de entrada, tempos de chegada e saída, e o caminho percorrido até à saída. Os tempos de deslocamento variam conforme o tipo de veículo: $t_{moto} = 0.5 \times t_{carro}$ e $t_{caminhão} = 4 \times t_{moto}$.

- Identificador único
- Tipo (moto, carro ou camião)
- Ponto de entrada: E1/E2/E3
- Tempo de chegada no sistema
- Tempo de saída do sistema
- Caminho até S

Veículos que chegam por **E1** têm os seguintes caminhos possíveis:

- E1 → Cr1 → Cr4 → Cr5 → S (34%)
- E1 → Cr1 → Cr2 → Cr5 → S (33%)
- E1 → Cr1 → Cr2 → Cr3 → S (33%)

Veículos que chegam por **E2** têm os seguintes caminhos possíveis:

- E2 → Cr2 → Cr5 → S (34%)
- E2 → Cr2 → Cr3 → S (33%)
- E2 → Cr2 → Cr1 → Cr4 → Cr5 → S (33%)

Veículos que chegam por **E3** têm os seguintes caminhos possíveis:

- E3 → Cr3 → S (34%)
- E3 → Cr3 → Cr2 → Cr5 → S (33%)
- E3 → Cr3 → Cr2 → Cr1 → Cr4 → Cr5 → S (33%)

Dois modelos de chegada de veículos no sistema:

- **Não realista:** veículos chegam em cada ponto de entrada E_i em intervalos fixos e regulares Δt_i
- **Mais realista:** veículos chegam no sistema de modo aleatório segundo um *Processo Estocástico de Poisson* de taxa λ_i (quantidade de veículos por segundo). Neste caso, o intervalo entre chegadas consecutivas de veículos é independente e modelado por uma *distribuição exponencial negativa* de taxa λ_i .

A escolha do caminho que cada veículo seguirá no sistema é dada inicialmente por uma distribuição uniforme conforme indicado acima.

O tempo de cruzamento de um nó ao seguinte da malha (deslocamento nas ruas) varia com o tipo do veículo: $t_{moto} = 0.5 \times t_{carro} = 0.25 \times t_{caminhão}$

5. Semáforos (Threads)

Os semáforos são implementados como threads dentro de cada processo de cruzamento. Cada semáforo alterna entre dois estados: aberto (verde) e fechado (vermelho). Devem gerir uma fila de veículos e calcular o tamanho máximo e médio da fila durante a simulação. A temporização (ciclo verde/vermelho) deve ser configurável e ajustável dinamicamente para otimizar o fluxo de tráfego. Políticas adaptativas, como o controlo por densidade de veículos, podem ser integradas para maior realismo. **O tempo de passagem de cada veículo no semáforo é dado por t_{sem} (quando o semáforo abre, o veículo que se encontra na primeira posição da fila demora um tempo t_{sem} a atravessar o semáforo e assim sucessivamente).**

6. Cruzamentos (Processos)

Os cruzamentos são implementados como processos distintos, responsáveis por coordenar os semáforos e assegurar que **apenas um sentido de tráfego se encontra aberto em cada instante**. Cada cruzamento gera a lógica de sincronização dos semáforos e pode aplicar políticas de temporização baseadas em ciclos fixos ou adaptativos. O cruzamento Cr4 possui um semáforo exclusivo para peões, simplificando a sua lógica.

7. Execução da Simulação

A simulação deve ser baseada no paradigma de eventos discretos. Cada evento representa uma alteração de estado no sistema, como a chegada de um veículo, a mudança de um semáforo ou a saída de um veículo do sistema. Os elementos fundamentais da simulação incluem:

- Lista de Eventos: Estrutura de dados que armazena todos os eventos futuros, ordenados pelo tempo de ocorrência.
- Estados do Sistema: Número de veículos nas filas, estados dos semáforos, tempos médios de espera e deslocamento.
- Linha do Tempo: A simulação avança segundo o tempo (discreto ou contínuo), onde o relógio é atualizado para o instante do próximo evento.
- Critério de Paragem: A simulação termina quando o tempo total atinge um valor máximo predefinido ou quando todos os veículos saem do

sistema.

- Medidas de Desempenho: Tempos mínimo, médio e máximo de travessia por tipo de veículo, comprimentos médio e máximo das filas, quantidade de veículos,
- Tratamento Estatístico: Cálculo de médias, desvios padrão e intervalos de confiança com base em múltiplas execuções sob os mesmos parâmetros.

Observações importantes:

- Toda a **temporização** da simulação fica a critério do aluno.
- Deve ser possível verificar a lista completa de todos os eventos no sistema (implementação da lista ou log de todos os eventos).
- Deve-se poder acompanhar o percurso completo de um veículo selecionado desde a entrada até a saída do sistema.
- Devem ser testadas **várias configurações diferentes** a fim de que haja filas de espera em praticamente todos os semáforos (sistema em baixa carga, média carga e alta carga).
- Na situação de alta carga deve-se buscar **a melhor política de temporização dos semáforos** a fim de reduzir os congestionamentos.
- Deve-se avaliar **diferentes políticas de seleção de caminhos** e o respetivo impacto no desempenho do sistema (seleção aleatória, menor caminho, caminhos fixos, caminhos de menores filas nos semáforos, etc...)
- As medidas de simulação devem ser apresentadas em um dashboard que se atualiza durante a execução da simulação e ao final retorna as medidas de desempenho gerais do sistema para cada cenário simulado.
- A simulação deve **avaliar o desempenho do sistema** em diferentes situações de carga (diferentes taxas de chegadas de veículos no sistema). A simulação deve permitir avaliar:
 - A carga máxima de veículos a entrar no sistema para que as filas não ultrapassem um valor determinado máximo.
 - O efeito de diferentes temporizações dos semáforos no congestionamento do sistema.
 - O efeito de diferentes algoritmos de seleção de caminhos no congestionamento do sistema.

8. Dashboard

O dashboard apresenta em tempo real as principais métricas de desempenho do sistema, incluindo o tamanho atual e máximo das filas, a taxa de ocupação dos cruzamentos e o tempo médio de permanência dos veículos. Cada processo deve enviar periodicamente os dados para um servidor central responsável pela agregação e visualização das informações. O dashboard deve também apresentar gráficos e indicadores que permitam comparar diferentes políticas de controlo e níveis de carga.

Cada processo deve coletar estatísticas para serem apresentadas em um *dashboard* de status da simulação:

- Tamanho atual das filas nos semáforos
- Tamanho máximo e médio das filas
- Quantos veículos de cada tipo passaram pelo cruzamento
- Outras medidas que julgar necessárias

O processo do nó S deve apresentar também as estatísticas globais da simulação:

- Tempo mínimo, médio e máximo que cada tipo de veículo levou para cruzar o sistema (*dwelling time*)
- Quantidade de veículos de cada tipo que cruzou o sistema
- Outras medidas que julgar necessárias

9. Entrega

O trabalho deve ser entregue em formato .zip, contendo **o relatório, o código fonte integralmente comentado** e os **diagramas da arquitetura**. A avaliação considerará a completude da arquitetura, a correta utilização de processos e threads, a implementação do dashboard, a análise de desempenho e a documentação do código.

O **relatório completo** deverá conter:

- O modelo de simulação, suas características e simplificações
- A arquitetura projetada para o sistema, todos os componentes, suas funções e objetivos, comunicação entre os componentes, diagrama explicativo da arquitetura, etc

- Explicação sobre todas as classes e métodos usados, seus objetivos e funções
- Discussão completa sobre a avaliação do desempenho do sistema, comparação entre os cenários simulados e as melhores configurações
- Capturas de tela, especialmente do dashboard e explicação sobre a implementação e medidas presentes no dashborad
- Conclusão, considerações finais, dificuldades encontradas e soluções
- Referências que foram consultados para a realização do trabalho

Código fonte:

- Código de todas as classes implementadas
- O código deve ser integralmente comentado
- Deve seguir as melhores práticas

Submeter todo o material em **um ficheiro .zip** pelo **moodle** até o prazo estipulado. O ficheiro deve ser nomeado com o número dos alunos do grupo, exemplo: **8321001_8321002_8321003.zip**

10. Datas Importantes

- Apresentação do Enunciado: 13 de outubro de 2025
- Escolha dos Grupos (2 ou 3 alunos): **20 de outubro de 2025**
- **Submissão do Trabalho** pelo Moodle: **08 de dezembro de 2025**
- **Defesa Oral: 10, 15 e 17 de dezembro de 2025 (turmas)**
- **Defesa Escrita: 15 de dezembro 2025 (todos)**

11. Critérios de Avaliação

Item	Pontuação
Modelo completo de simulação (Relatório)	4
Arquitetura Completa do SD (Relatório)	5
Implementação em JAVA: classes, processos, threads, sockets... (Código)	5
<i>Dashborad</i> e medidas de desempenho do sistema (Relatório + Código)	3
Avaliação do desempenho do sistema frente a diferentes cenários e configurações (Relatório)	3

Total Pontos	20

. **Defesa:** Oral (PL) + Escrita (T) = valor de 0% a 100% [0,1]

. **Nota AC** = Pontos recebidos x Defesa

12. Conclusão

O projeto visa desenvolver competências em programação distribuída, simulação de sistemas complexos e análise de desempenho. A correta modelação de cruzamentos e semáforos, aliada à recolha e tratamento estatístico dos resultados, permitirá avaliar o impacto de diferentes políticas de controlo sobre o fluxo de tráfego urbano.

13. Plágio

Qualquer trabalho que se detete que tenha sido copiado, mesmo que parcialmente, receberá o valor zero. Todas as fontes utilizadas devem ser devidamente citadas.