

Documento Técnico

Representação do Grafo

O ambiente urbano é modelado como um **grafo direcionado**, no qual:

- **Vértices (nós)** representam cruzamentos ou pontos de interesse.
- **Arestas** representam as ruas que conectam esses pontos.

A estrutura de dados foi implementada manualmente por meio da classe **Grafo<T>**, parametrizada para aceitar objetos do tipo **Node**.

- Cada **Node** armazena:
 - Identificação única;
 - Geolocalização (latitude e longitude).
- Cada **Edge** representa:
 - Uma conexão entre dois nós;
 - Identificadores de origem e destino;
 - Comprimento da via.

O grafo é carregado dinamicamente a partir de um arquivo **.json**, utilizando a biblioteca **Gson**, que interpreta os dados e instancia os objetos correspondentes. Essa abordagem facilita a alteração do mapa e a experimentação com diferentes configurações urbanas.

Estrutura de Dados Auxiliares

Além da estrutura principal do grafo, o sistema utiliza as seguintes estruturas auxiliares:

- **ListaEncadeada<T>**:
Lista ligada implementada manualmente, utilizada para armazenar a lista de nós e a lista de semáforos.
- **Set<Edge>**:
Conjunto utilizado para garantir que **nenhuma rua seja controlada por mais de um semáforo**.

Modelagem e Controle de Semáforos

Associação e Funcionamento

A lógica de controle de tráfego é implementada na classe `Semaforo`, que:

- Está associada a uma `Edge` (rua controlada) e a um `Node` (posição geográfica);
- Controla o fluxo com base na proximidade geográfica definida a partir do arquivo `.json`.

Os semáforos alternam seus estados de forma cíclica:

VERDE → AMARELO → VERMELHO

Essa alternância é gerenciada pela classe `ControladorSemaforo`, que:

- Implementa a interface `Listener`;
- É acionada periodicamente pelo simulador via o padrão **Observer**.

Lógica Detalhada dos Semáforos

Identificação das Interseções

- Os semáforos são posicionados com base nas **coordenadas dos objetos** `TrafficLight` importados do `.json`.
- Cada `TrafficLight` é associado ao **Node mais próximo**.
- Para evitar múltiplos semáforos desnecessários em cruzamentos complexos, os nós são **agrupados por proximidade geográfica**, utilizando a função `agruparNosProximos(...)`.

Filtragem das Ruas Válidas

- Para cada grupo de nós (interseção real), são selecionadas as **arestas (Edge)** que chegam ao grupo.
- Ruas com comprimento **abaixo de 15 metros** são descartadas.
- Um semáforo é criado **para cada Edge válida** que entra no grupo.

Agrupamento e Sincronização

- Cada semáforo é associado a uma **Edge**.
 - Com a função **eVertical(Edge)**, determina-se se a rua é vertical (grupo A) ou horizontal (grupo B).
 - O controle alterna entre os grupos A e B em **ciclos de 8 ticks**, assegurando que:

Ruas opostas (ida e volta) compartilhem o mesmo estado.

Ruas cruzadas **nunca tenham verde ao mesmo tempo**, evitando colisões lógicas.
-

Geração e Movimentação de Veículos

- Os veículos são instanciados como objetos **Carro**.
 - Cada carro:
 - Escolhe um caminho aleatório no grafo;
 - utiliza-se do algoritmo de Dijkstra para conseguir o caminho a ser percorrido;
 - Movimenta-se entre os nós usando interpolação linear;
 - Consulta os estados dos semáforos antes de avançar;
 - Respeita os sinais (parando no vermelho, avançando no verde).
 - A geração automática de veículos é feita por um **temporizador** (**javax.swing.Timer**), que cria um novo carro em intervalos regulares.
-

5. Visualização: Classe **GrafoViewer**

A interface gráfica da simulação é implementada na classe **GrafoViewer**, que estende **JPanel**. Essa classe é responsável por renderizar:

- **Ruas** (arestas): linhas cinzas;
- **Semáforos**: setas coloridas conforme o estado (verde, amarelo, vermelho);
- **Veículos**: pequenos quadrados laranja em movimento;
- **Identificação dos nós**: com exibição opcional de coordenadas e nomes.

A renderização converte latitude e longitude para coordenadas relativas à tela, garantindo **representação proporcional e coerente** com a geografia.

A interface é atualizada pelo método `repaint()`, acionado automaticamente pelo simulador ou por alterações no estado dos semáforos.

Resultados e Benefícios

- Simulação fiel de **cruzamentos urbanos com vias de mão dupla**.
- Transições de semáforos suaves e seguras.
- Lógica baseada em geometria de vias, adaptável a diferentes mapas e tipos de cruzamento.
- Visualização clara e informativa dos elementos da simulação.