

## Relatório de Documentação do Código

**1. Simulacao:** Classe principal da lógica de simulação. Gerencia o ciclo de vida da simulação, controla o avanço dos veículos, atualiza os semáforos, coleta estatísticas e permite salvar/carregar o estado. Possui métodos para iniciar, pausar, continuar, parar e executar a simulação com diferentes modelos de controle de semáforo.

**2. Semaforo:** Representa um semáforo em uma interseção. Controla o estado atual (VERDE, AMARELO, VERMELHO), tempos de cada fase, e lógica de transição de estados. Permite atualizar o ciclo conforme diferentes modelos (fixo, otimização de espera, otimização de energia).

**3. Veiculo:** Modela um veículo na simulação. Possui um caminho (rota), posição atual, métodos para avançar, checar se chegou ao destino, e obter informações sobre sua posição na rede.

**4. Grafo:** Estrutura de dados que representa o grafo da cidade, com interseções (nós) e ruas (arestas). Permite adicionar/remover vértices e ruas, buscar interseções e ruas, e acessar listas de semáforos.

**5. Intersecao:** Representa uma interseção no grafo. Armazena informações como id, coordenadas, ruas adjacentes, referência ao semáforo e ao grupo de semáforos, além do grafo ao qual pertence.

**6. Rua:** Modela uma rua entre duas interseções. Guarda origem, destino, tempo de travessia e uma fila de veículos presentes na rua.

**7. ControladorSemaforos:** Responsável por atualizar o estado de todos os semáforos do grafo a cada ciclo da simulação.

**8. GeradorDeSemaforos:** Cria e configura semáforos nas interseções do grafo, definindo tempos iniciais de cada fase.

**9. GrupoSemaforo:** Agrupa semáforos que podem ser controlados em conjunto (por exemplo, em cruzamentos complexos).

**10. GeradorVeiculos:** Gera veículos aleatórios com rotas válidas no grafo, podendo criar múltiplos veículos e sortear origem/destino.

**11. VeiculoProgramado:** Associa um veículo a um tempo de entrada na simulação, útil para cenários programados.

**12. Fila, LinkedList, HashSet, HashMap, Node:** Implementações próprias de estruturas de dados para filas, listas, conjuntos e mapas, usadas para manipulação eficiente dos elementos da simulação.

**13. VisualizacaoSimulador:** Painel gráfico (JPanel) que desenha o grafo, veículos e semáforos, permitindo visualizar o estado atual da simulação.

**14. SimulacaoGUI:** Interface gráfica principal da aplicação. Permite ao usuário configurar, iniciar, pausar, parar, salvar e carregar simulações, além de visualizar o painel de simulação.

**15. ImportadorDeGrafo:** Importa o grafo da cidade a partir de arquivos JSON, criando as interseções e ruas conforme os dados do arquivo.

**16. SemaforoJsonUtils:** Utilitário para carregar e interpretar dados de semáforos a partir de arquivos JSON.

**17. TrafficLightJson, IntersecaoJson, RuaJson:** Classes de modelo usadas para deserialização dos dados do JSON da cidade.

**18. Dijkstra:** Algoritmo utilizado para encontrar o menor caminho entre interseções no grafo (usado pelo GeradorVeiculos).

**19. TrabalhoSekeffApplication:** Classe principal (main) que inicializa o sistema, carrega o grafo, configura semáforos e inicia a interface gráfica.

**20. Outros utilitários em CidadeJson:** Como deserializadores personalizados para listas (ex: LinkedListDeserializerIntersecao).