



**UDESC**

**IBIRAMA**

**UNIVERSIDADE ESTADUAL DE SANTA CATARINA  
CEAVI - CENTRO DE EDUCAÇÃO SUPERIOR DO ALTO VALE DO ITAJAÍ  
DEPARTAMENTO DE ENGENHARIA DE SOFTWARE**

**Augusto Rustick e Denis Zickuhr**

**Trabalho 2 - Desenvolvimento de sistemas paralelos e distribuídos**

**Ibirama**

# 1. TEMA

O tema deste projeto consiste na implementação de um sistema de controle de movimentação de veículos em um ambiente urbano. O objetivo é criar um software capaz de gerenciar o fluxo de veículos em uma rede de ruas, garantindo a ausência de impasses e sobreposições.

O contexto do projeto se baseia na necessidade de melhorar a eficiência e a segurança do tráfego urbano. A crescente quantidade de veículos em áreas urbanas tem levado a congestionamentos, acidentes e atrasos significativos. Portanto, a proposta é desenvolver um sistema capaz de otimizar a movimentação dos veículos, minimizando o tempo de viagem e reduzindo os riscos de acidentes.

A importância desse tema reside na possibilidade de promover uma melhoria significativa na mobilidade urbana, beneficiando não apenas os motoristas, mas também os pedestres e o meio ambiente. Com um sistema eficiente de controle de tráfego, é possível reduzir o tempo de deslocamento, diminuir a emissão de gases poluentes e aumentar a segurança nas vias públicas. Ao implementar esse sistema, espera-se obter resultados como a diminuição de congestionamentos, a otimização dos tempos de viagem e a redução de acidentes de trânsito.

Além disso, a solução proposta também deve ser flexível e escalável, permitindo a adaptação a diferentes cenários urbanos e a integração com futuras tecnologias de veículos autônomos e sistemas de transporte inteligentes.

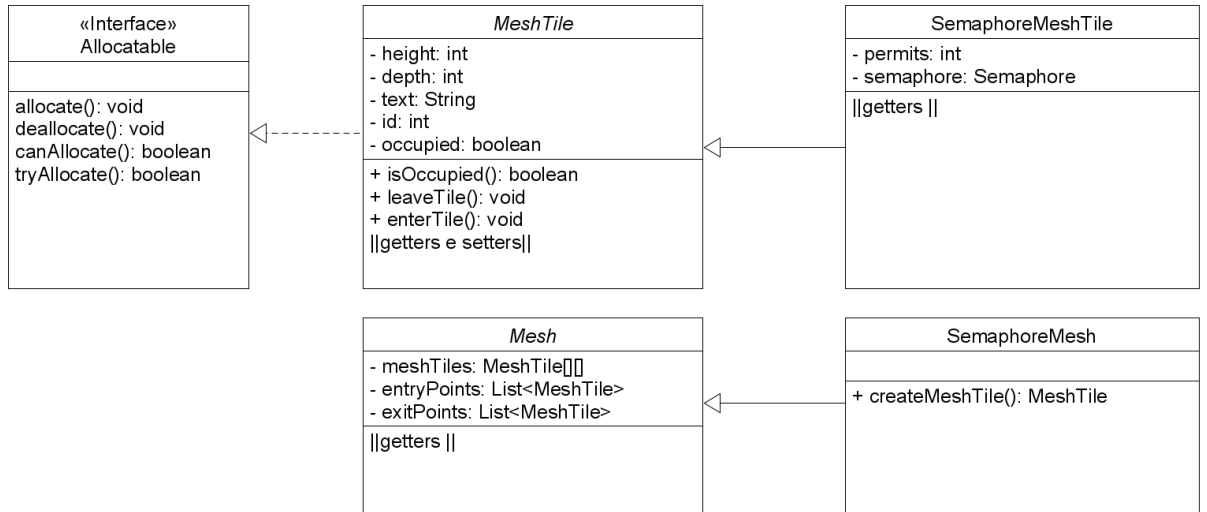
## 2. DIAGRAMAS

Nesta seção, apresentaremos a arquitetura do sistema e os diagramas de classes que descrevem a estrutura e as relações entre as diferentes partes do sistema.

### 2.1. Diagrama de semáforos

O diagrama de semáforos é uma representação visual das classes e suas interações relacionadas à implementação dos semáforos no sistema. Esses semáforos desempenham um papel crucial na coordenação do tráfego e no controle do acesso concorrente aos recursos compartilhados.

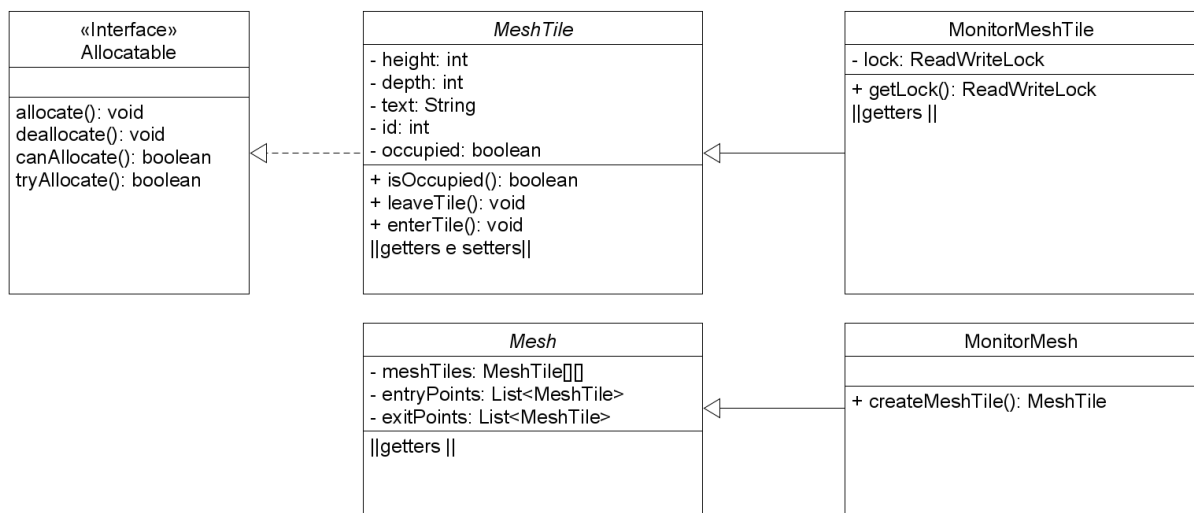
Através do diagrama de semáforos, é possível entender como as classes se relacionam e como a lógica de controle de semáforos é implementada no sistema. Esse diagrama oferece uma visão geral clara da estrutura e funcionalidades relacionadas aos semáforos no contexto do sistema em questão.



## 2.2. Diagrama de monitores

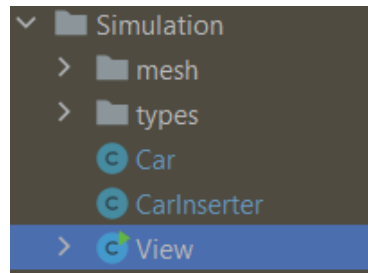
O diagrama de monitores é uma representação visual das classes e suas interações relacionadas à implementação dos monitores no sistema. Os monitores desempenham um papel fundamental na coordenação do acesso concorrente a recursos compartilhados, garantindo a exclusão mútua e a sincronização adequada.

Ao analisar o diagrama de monitores, é possível compreender como as classes se relacionam e como a lógica de controle de acesso concorrente por meio de monitores é implementada no sistema. Esse diagrama oferece uma visão clara da estrutura e funcionalidades relacionadas aos monitores no contexto específico do sistema em questão.

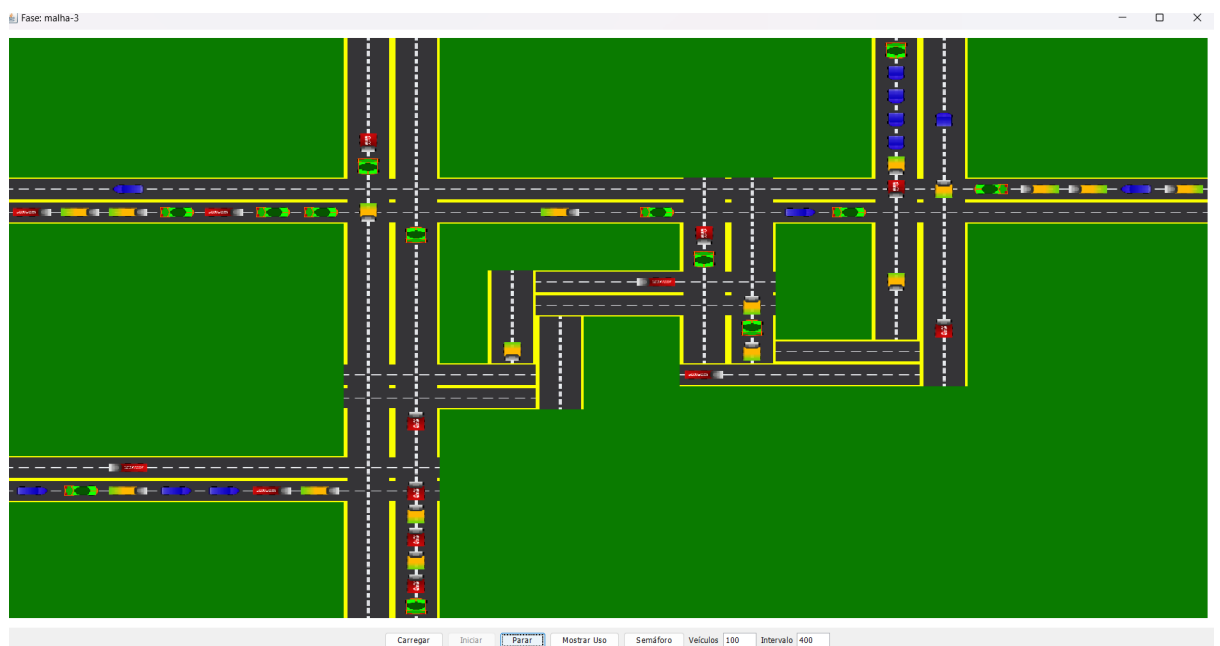
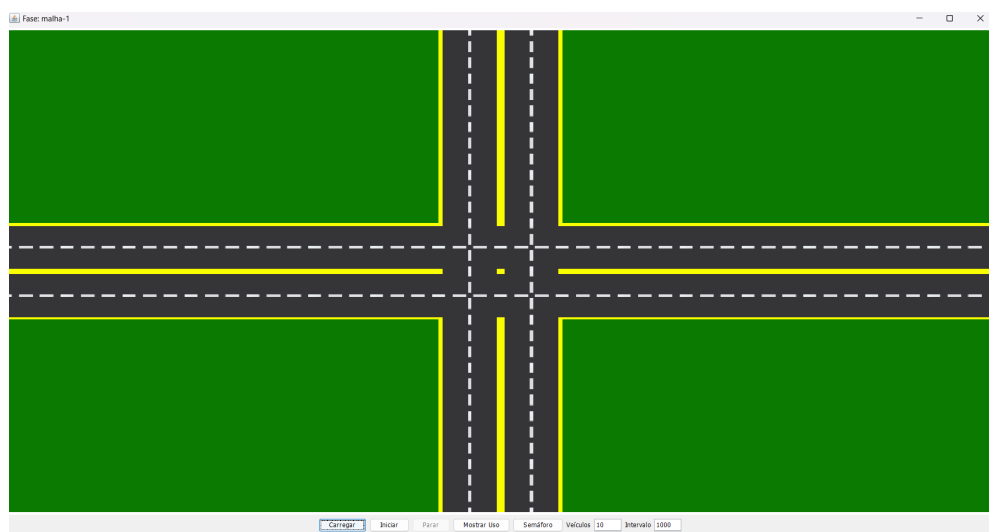


## 3. COMO EXECUTAR O PROJETO

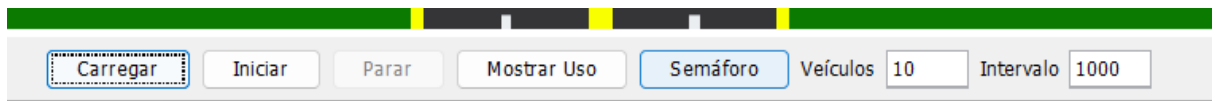
Para executar o projeto, basta executar o arquivo View.java que está contido no *package Simulation*:



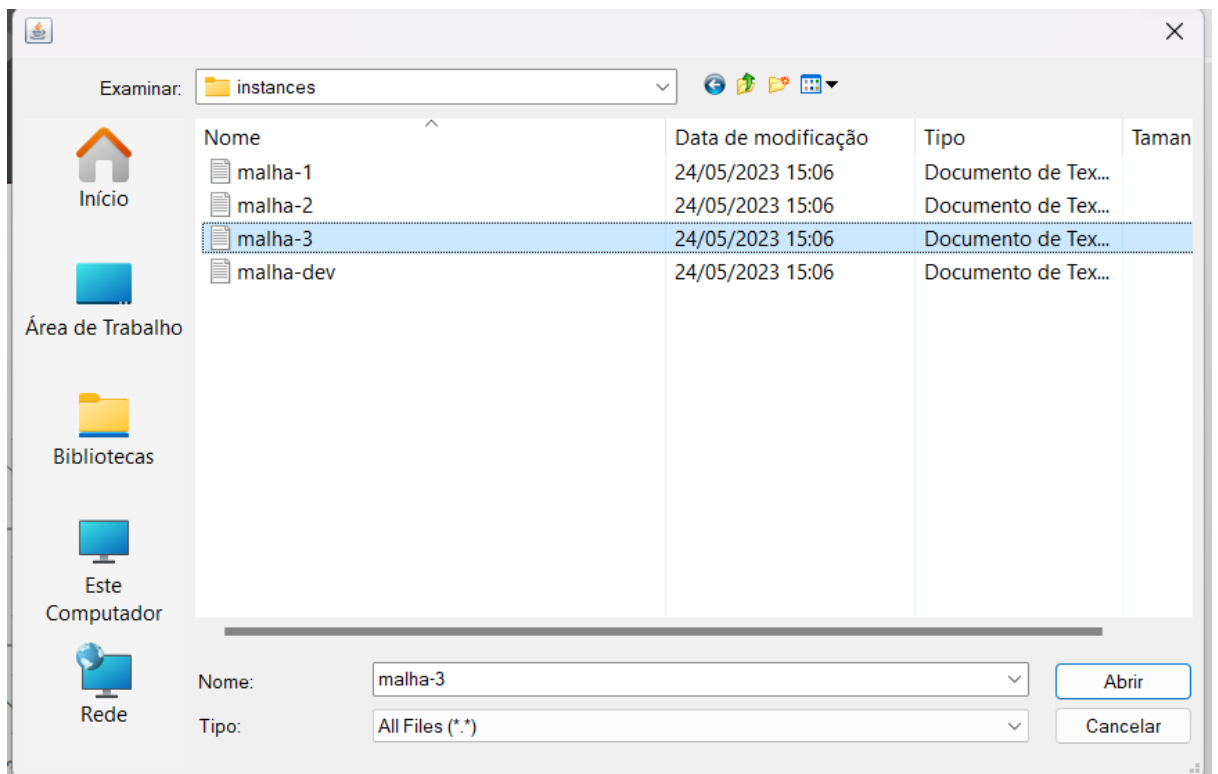
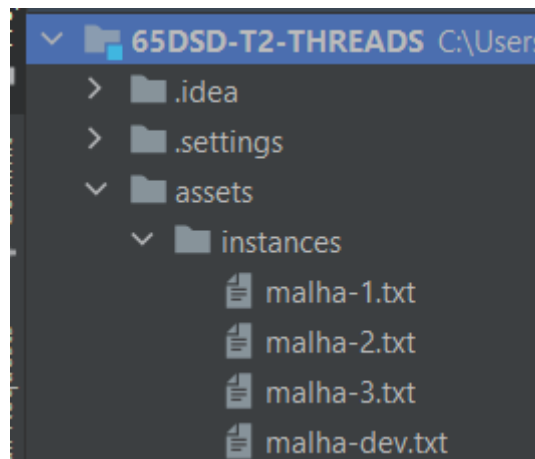
Esse arquivo abrirá a seguinte janela:



Na parte inferior possui os componentes que a aplicação usa para executar



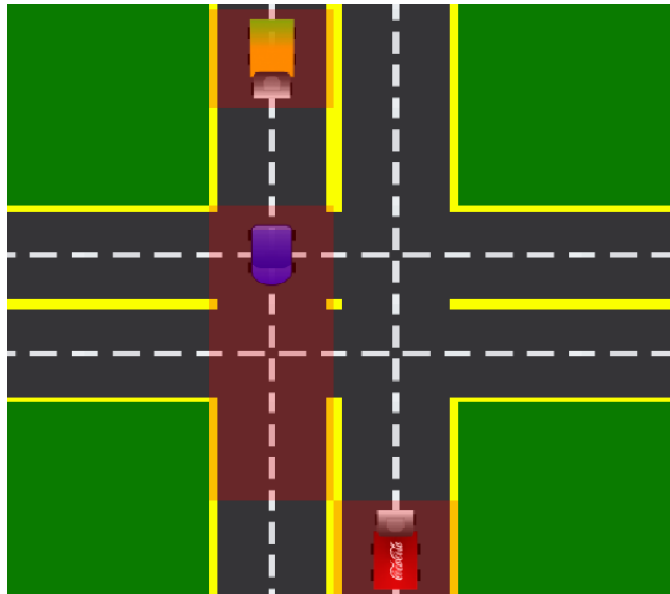
Clicando na opção de carregar, é possível fazer a escolha das malhas, basta escolher o arquivo pelo diretório Assets/instances/..



Além disso, se clicar em semáforo ou monitor, a aplicação troca o tipo de solução para os carros se locomoverem nas ruas.

<input type="button" value="Mostrar Uso"/>	<input type="button" value="Semáforo"/>	<input type="button" value="Mostrar Uso"/>	<input type="button" value="Monitor"/>	<input type="button" value="V"/>
--	---	--	--	----------------------------------

E o botão de “mostrar uso” abre um campo vermelho que mostra quais pedaços da estrada estão em uso



Por fim, a quantidade de veículos que a aplicação vai tentar colocar e o tempo (em milissegundos) que cada carro demora para ser colocados depois do último ter sido posto.

Veículos	<input type="text" value="10"/>	Intervalo	<input type="text" value="1000"/>
----------	---------------------------------	-----------	-----------------------------------