

# Malha de Trânsito

Uso e controle de  
múltiplas Threads com  
semáforos e monitores.

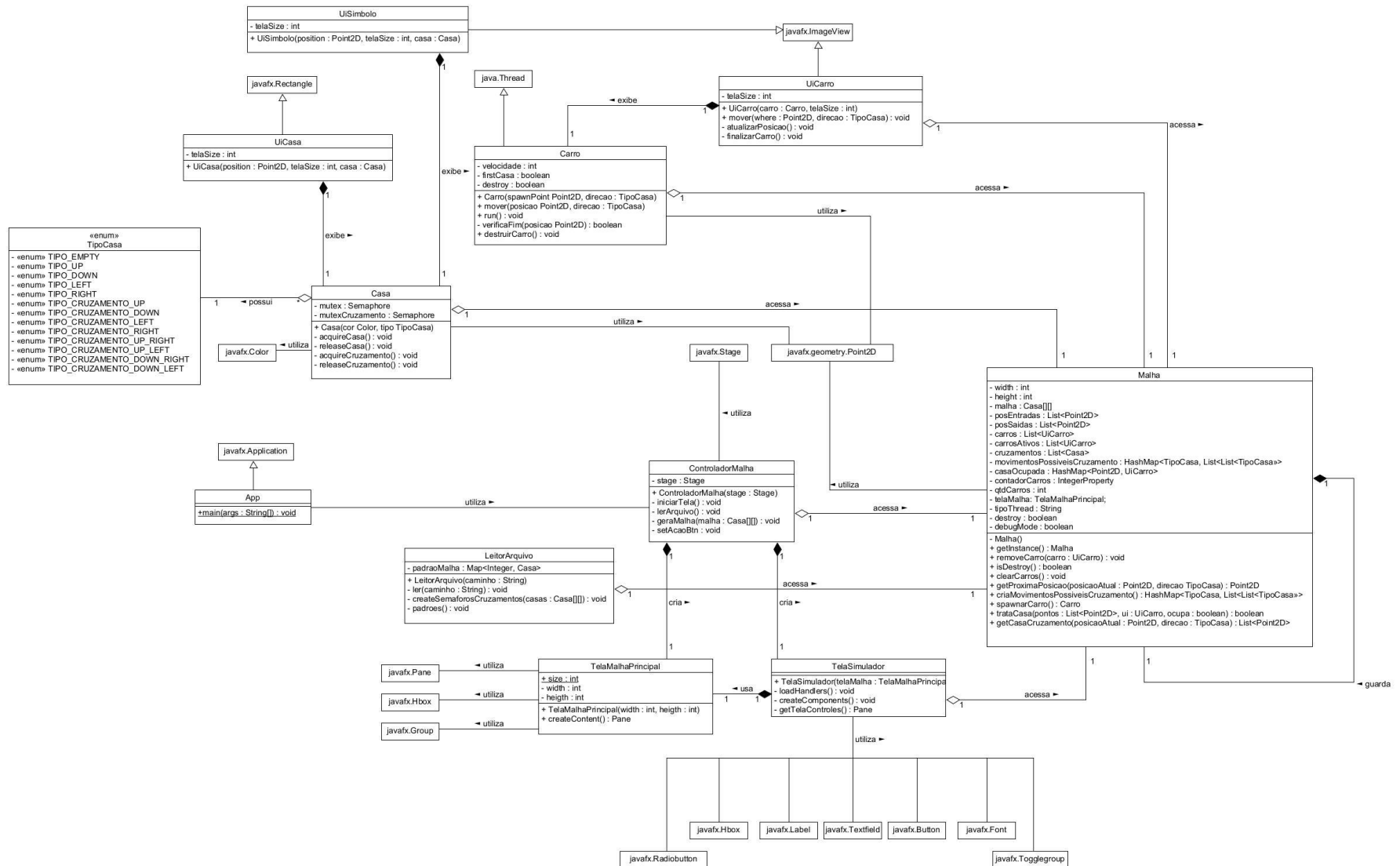
Gabriel Dolzan, Luis Felipe da Silva e  
Murilo Goedert

Professor: Fernando dos Santos  
27/10/2022

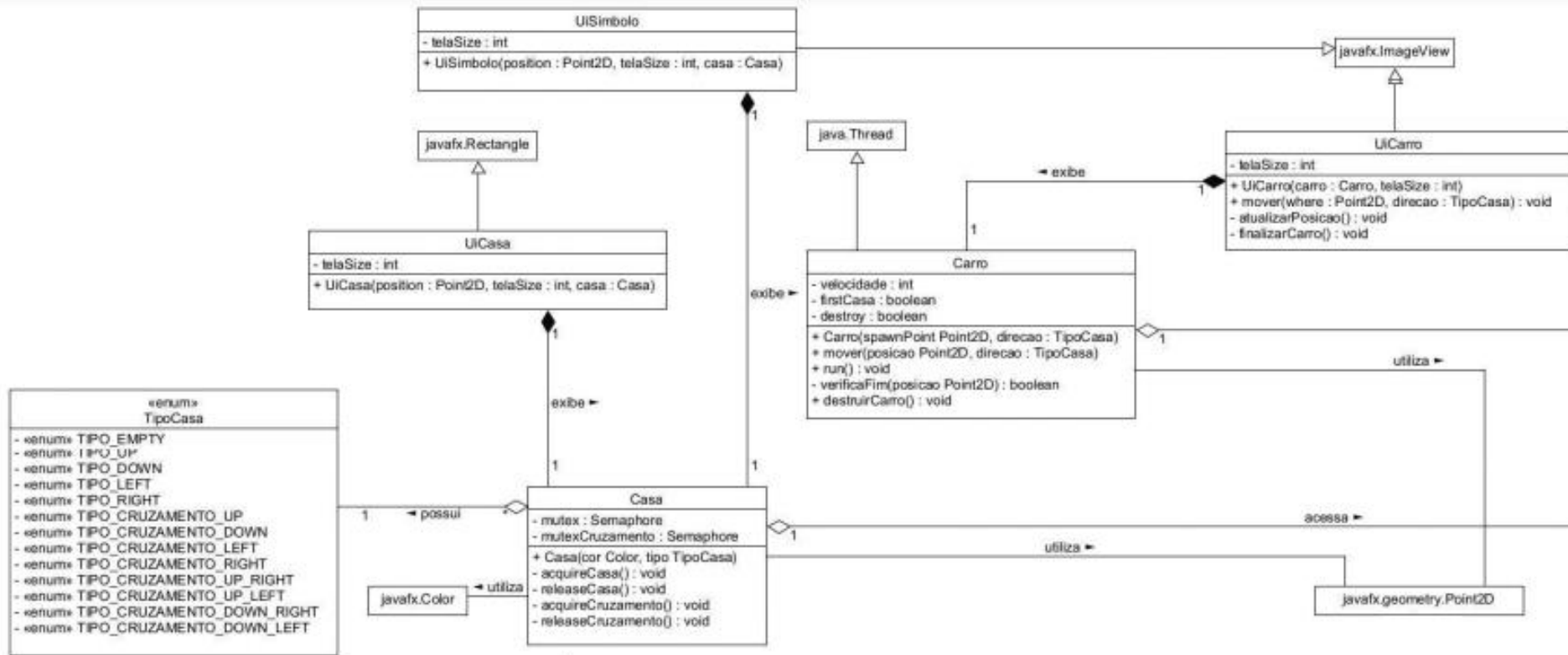
# Tecnologias



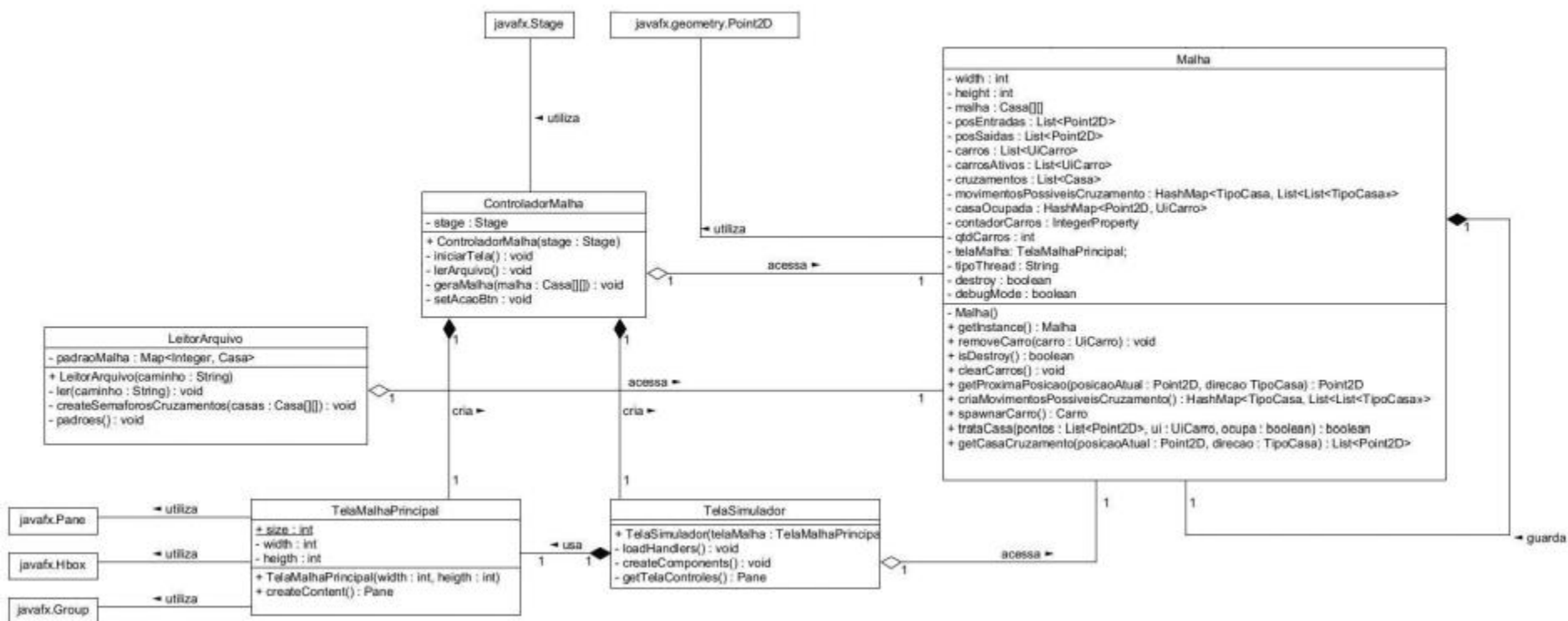
# Diagrama de Classes



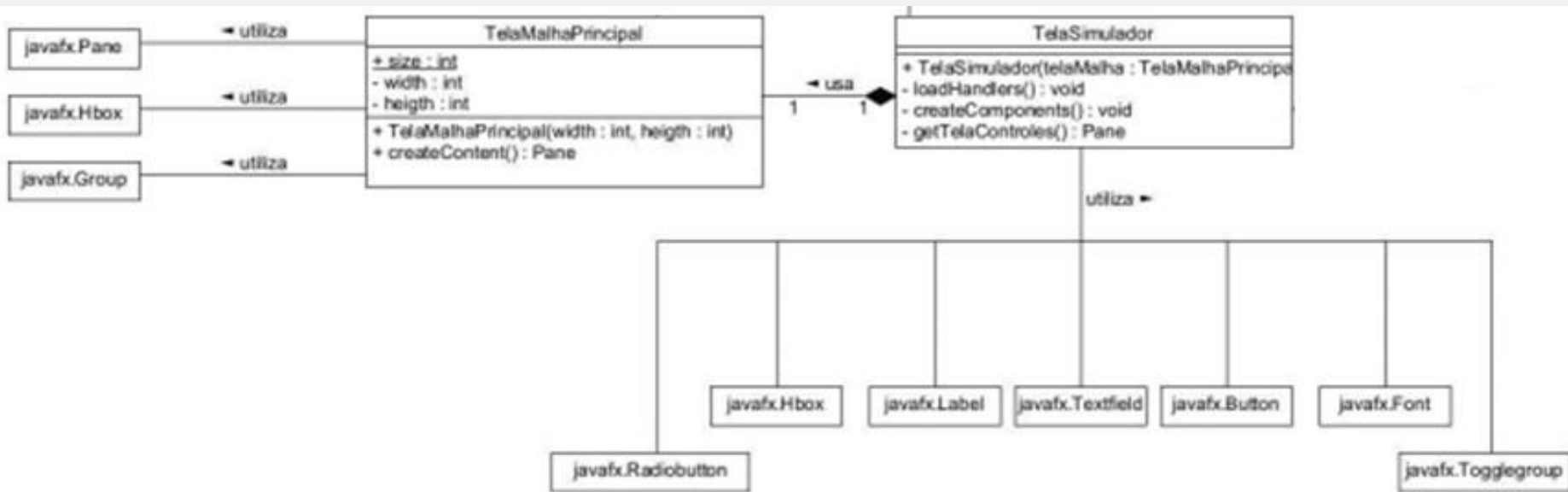
# Classes de Modelo



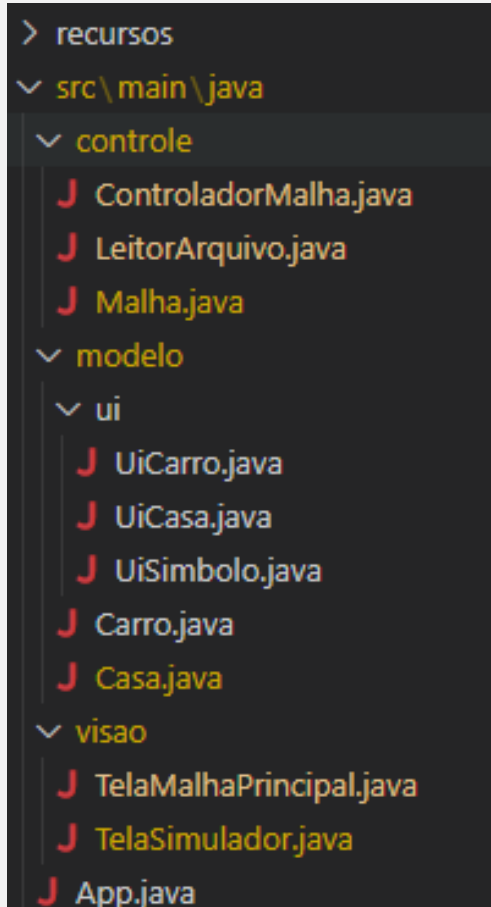
# Classes de Controle



# Classes de Visão



# Estrutura do Projeto



- MVC
- Singleton
- Observer (Listeners)

# Singleton

J Malha.java 9, M X

Trabalho2 > src > main > java > controle > J Malha.java > ...

```
19
20 // Singleton guardando os dados importantes para o projeto
21
22 private static Malha instance; // Instancia do Singleton
23 private TelaMalhaPrincipal telaMalha; // Tela principal
24 private int width; // Largura da malha
25 private int height; // Altura da malha
26 private Casa[][] malha; // A malha
27 private List<Point2D> posEntradas = new ArrayList<>(); // Lista das posições de entrada
28 private List<Point2D> posSaidas = new ArrayList<>(); // Lista das posições de saída
29 private List<UiCarro> carros = new ArrayList<>(); // Lista dos carros inseridos
30 private List<UiCarro> carrosAtivos = new ArrayList<>(); // Lista dos carros inseridos
31 private List<Casa> cruzamentos = new ArrayList<>(); // Lista dos carros inseridos
32 private HashMap<TipoCasa, List<List<TipoCasa>>> movimentosPossiveisCruzamento = this
33     .criaMovimentosPossiveisCruzamento(); // Movimentos possiveis quando se esta em um cruzamento
34 private HashMap<Point2D, UiCarro> casaOcupada = new HashMap<>(); // Controle de ocupacao de casa
35 private IntegerProperty contadorCarros = new SimpleIntegerProperty(); // Contador de carros (listener)
36 private int qtdCarros = 0; // Quantidade de carros
37 private String tipoThread = "Semáforo"; // Tipo dos controladoes (Semáforo ou Monitor)
38 private boolean destroy = false; // Controle para destruir threads
39 private boolean debugMode = false; // Controle para visualizar acquire e release
40
41 private Malha() {
42 }
43
44 public synchronized static Malha getInstance() {
45     if (instance == null) {
46         instance = new Malha();
47     }
48
49     return instance;
50 }
51
```



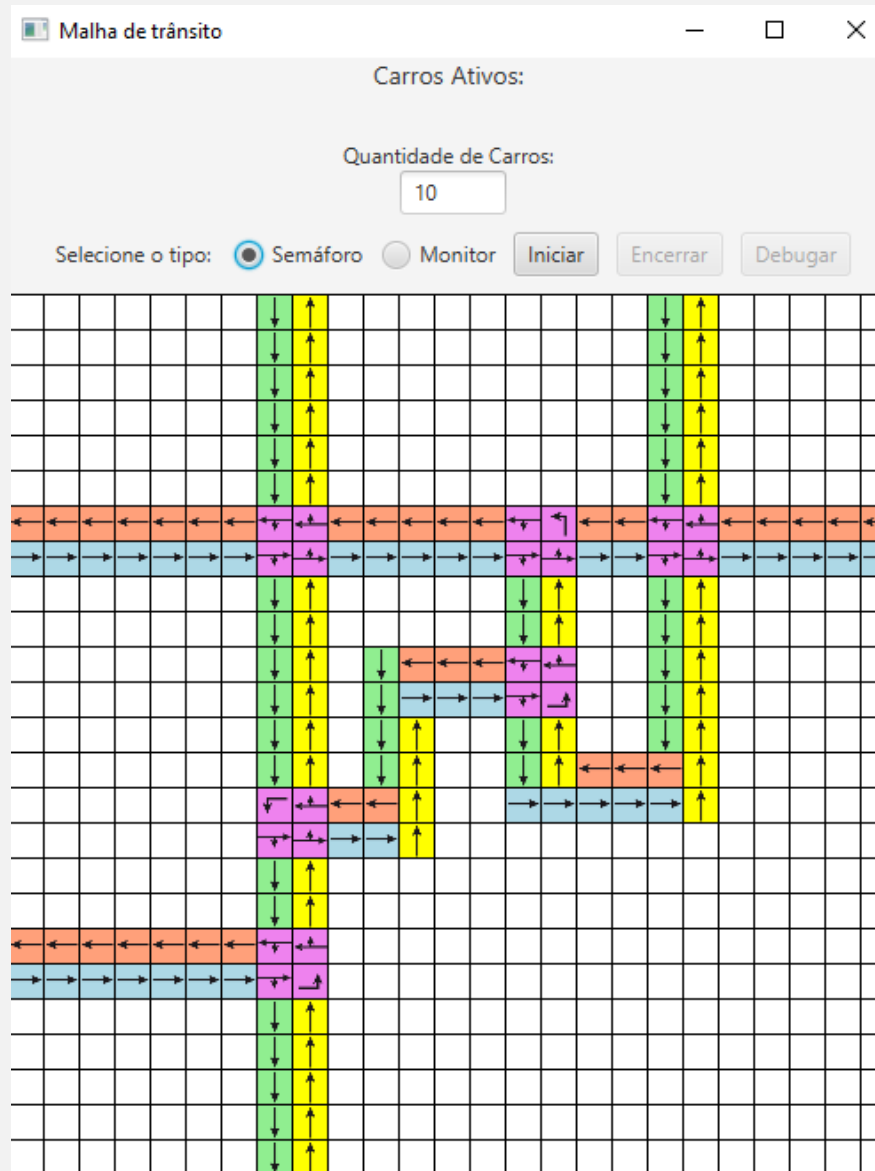
# Observer (Listener)

```
private void loadHandlers(){
    this.malha.getContadorCarros().addListener(new ChangeListener<Number>() {

        @Override
        public void changed(ObservableValue<? extends Number> observable, Number oldValue, Number newValue) {
            contadorCarros.setText(newValue.toString());
        }

    });
}
```

# Tela Desenvolvida





# Obrigado

**UDESC – Universidade do Estado de  
Santa Catarina**