

THREADS

65DSD

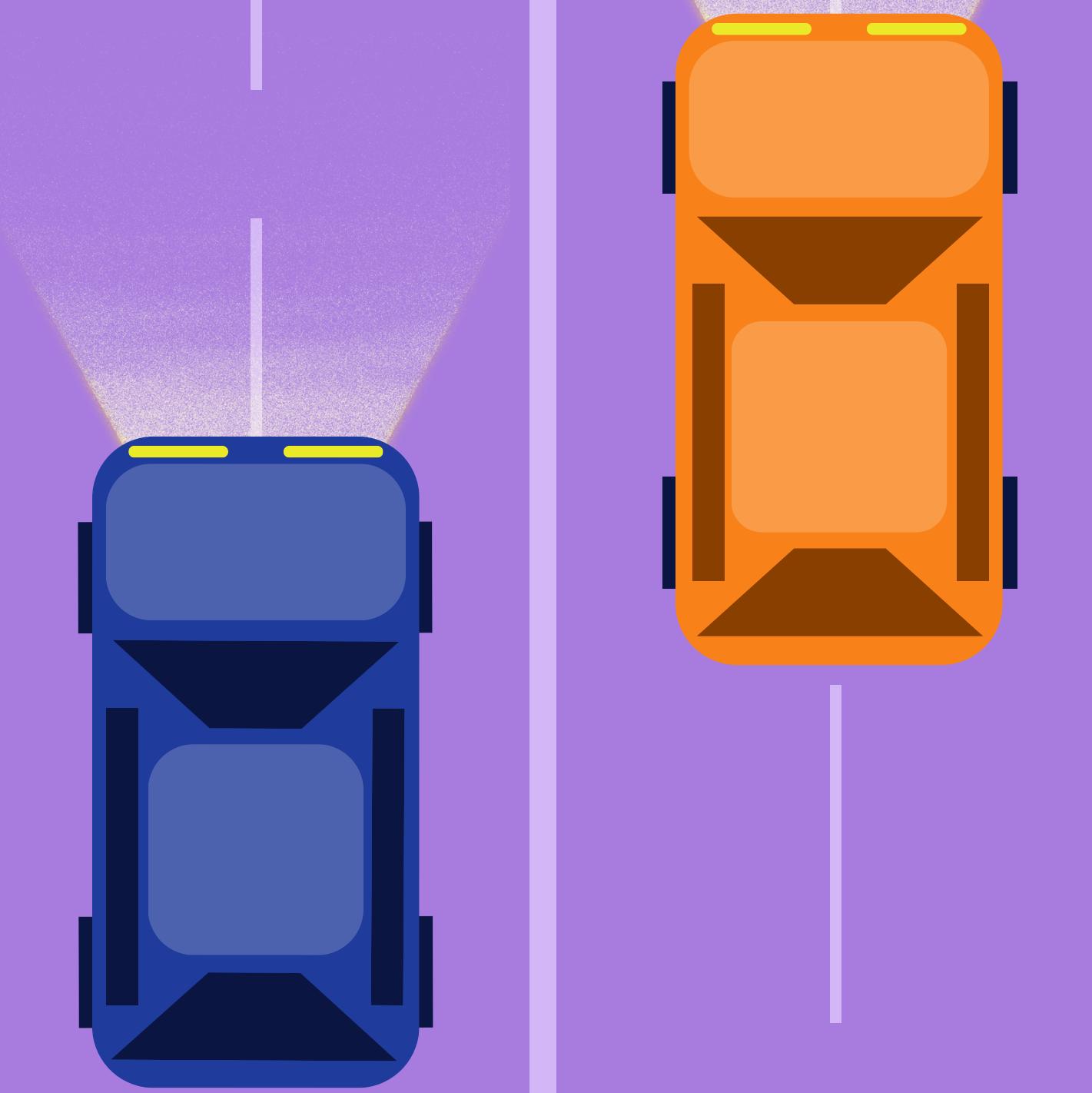
Guilherme Schneidt
Mariana Lino

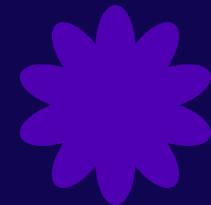




CLASSES DO SISTEMA

- MAP
- NODE
- PATH
- CAR

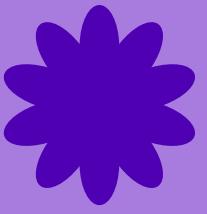




MAP

- A classe Map gerencia o mapa como um todo, funcionando como uma matriz de objetos Node.
- Contém métodos genéricos para localizar posições específicas de nodes e adicionar informações extras durante a inicialização.

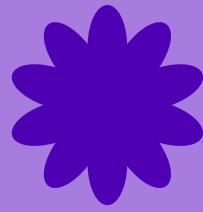




NODE

- A classe Node representa uma posição na matriz e é crucial para a aplicação.
- Cada node possui informações sobre sua direção e regras para movimentação para os nodes adjacentes.
- Fornece informações críticas, como se é entrada ou saída de um cruzamento, bem como a entrada e saída do mapa.
- Cada node possui um mecanismo de exclusão mútua, como um monitor ou semáforo, para garantir a consistência das operações.
- O movimento simples ocorre quando um carro se move para um node adjacente diretamente.

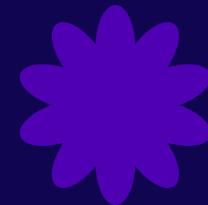




PATH

- O Path é uma lista encadeada de nodes que representa o movimento composto da thread.
- É gerado quando um carro alcança a entrada de um cruzamento e excluído quando o carro completa a travessia.
- Contém os nodes que serão reservados na ordem para atravessar o cruzamento, liberando-os conforme o carro avança.





CAR

- A thread principal da aplicação, representada pela classe Car.
- Alterna entre movimentos simples e compostos.
- O movimento simples ocorre em direções comuns, enquanto o movimento composto segue um Path.
- Aguarda a reserva do caminho completo antes de começar a atravessar o cruzamento.
- Libera os nodes conforme avança pelo caminho e exclui o Path quando completa a travessia.



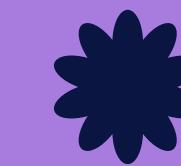
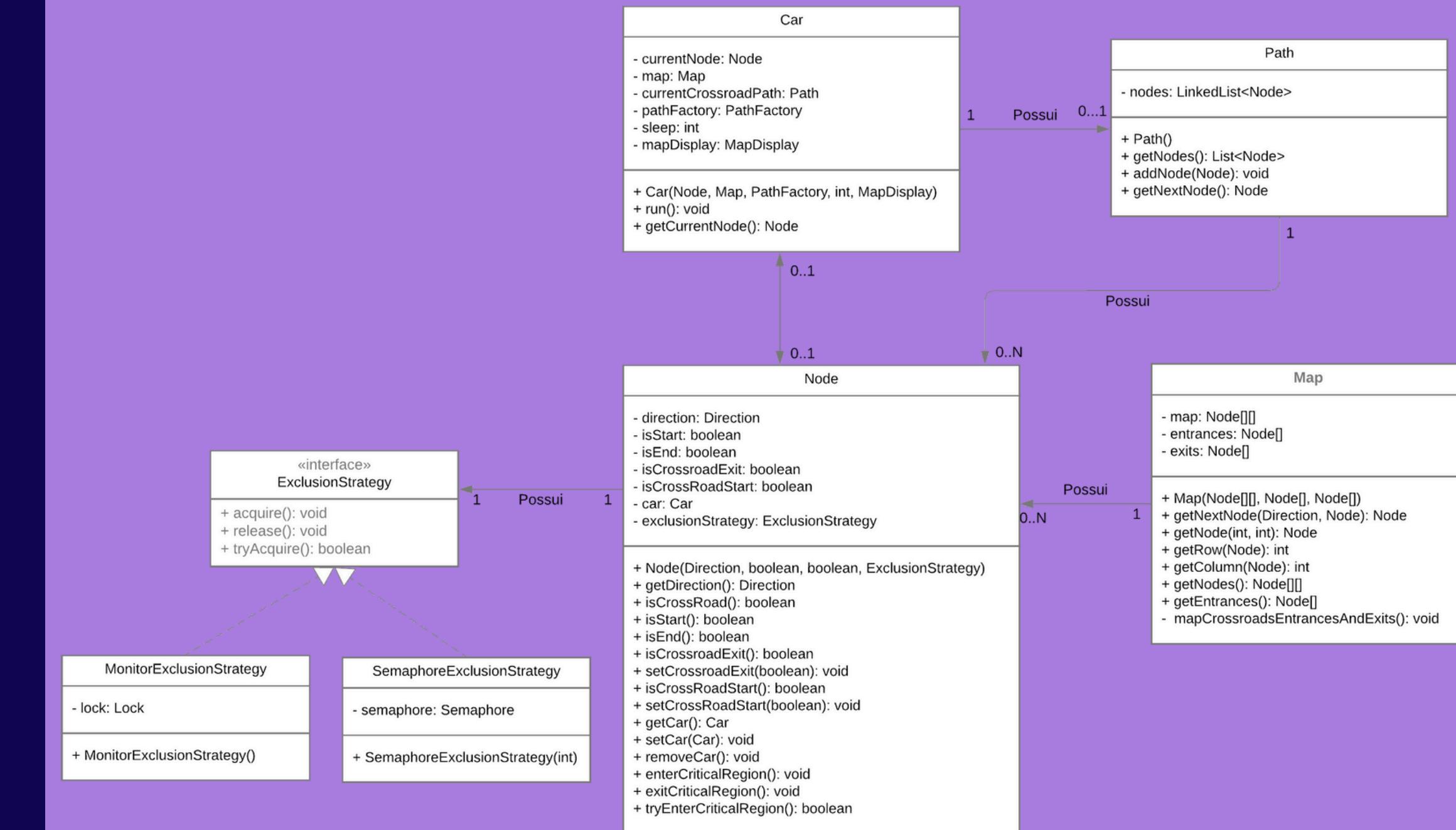


DIAGRAMA DE CLASSES





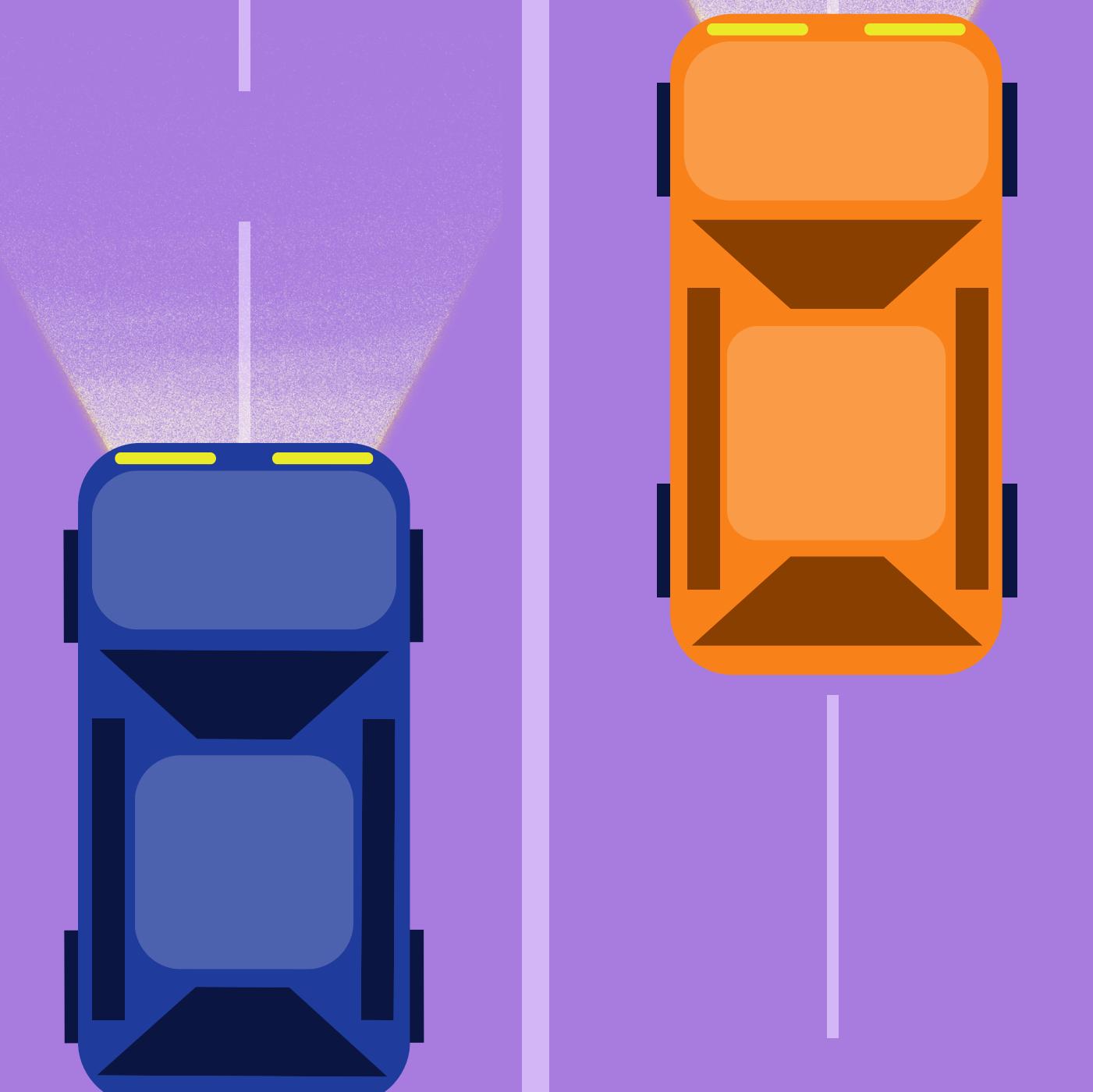
PADRÕES UTILIZADOS

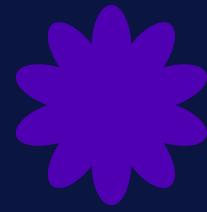
- **STRATEGY**

- Define uma família de algoritmos, encapsula e torna eles intercambiáveis. Essa funcionalidade permite que o cliente escolha o algoritmo a ser utilizado em tempo de execução sem alterar a estrutura do código.

- **FACTORY**

- Geralmente utilizado quando um sistema precisa criar objetos, mas não necessariamente precisa saber qual classe concreta será utilizada, já que permite que as subclasses alterem o tipo de objetos que serão criados.





PROBLEMAS ENCONTRADOS

- Lidar com as exclusões mútuas sem travar todo o sistema enquanto um carro passa;
- Conseguir encerrar a simulação de forma que os carros parassem imediatamente.

E como foram resolvidos?

