



UNEB

UNIVERSIDADE DO
ESTADO DA BAHIA

INFRAESTRUTURA URBANA ENGENHARIA DE PRODUÇÃO CIVIL

Jonas G Santos

Universidade Estadual da Bahia - UNEB

6 de diciembre de 2023

SUMÁRIO

- 1 Introdução
- 2 Problema de Pesquisa
- 3 Hipóteses
- 4 Objetivos
- 5 Justificativa
- 6 Interesse
- 7 Trabalhos Futuros
- 8 Desenvolvimento do Projeto
- 9 Metodologia
- 10 Cronograma

INTRODUÇÃO

Infraestrutura Urbana

Em meio aos desafios da vida moderna, um aspecto que todos compartilhamos é a experiência, muitas vezes exasperante, do trânsito urbano. Engarrafamentos, vias congestionadas e sistemas de transporte sobrecarregados tornaram-se componentes cotidianos das cidades contemporâneas. Esta pesquisa explora as complexidades do trânsito urbano, mergulhando nos problemas de infraestrutura viária e buscando soluções para criar ambientes urbanos mais fluídos e eficientes.

PROBLEMA DE PESQUISA

Como pode-se aprimorar a infraestrutura urbana de maneira eficaz para mitigar e aliviar os persistentes problemas de engarrafamentos, contribuindo assim para uma mobilidade urbana mais eficiente e uma qualidade de vida melhor para os cidadãos?

HIPOTÉSES

Um Software para a simulações de trânsitos seria de grande ajuda para controle de tráfego, ajudando a fazer as melhorias corretas para o local desejado

O objetivo geral deste projeto de pesquisa é realizar uma análise da infraestrutura urbana

OBJETIVOS ESPECÍFICOS

- Identificar Causas e Padrões de Engarrafamentos
- Avaliar Infraestrutura Viária Existente
- Avaliar Impactos Econômicos e Ambientais

JUSTIFICATIVA

A justificativa para a realização desta pesquisa está fundamentada na necessidade premente de encontrar soluções inovadoras e eficazes para aliviar os engarrafamentos urbanos, considerando os seguintes pontos.

INTERESSE

O interesse por esta pesquisa é motivado pela urgência de encontrar soluções inovadoras e práticas para um dos problemas mais evidentes e impactantes nas cidades contemporâneas: o trânsito urbano congestionado.

TRABALHOS FUTUROS

Com a conclusão dessa pesquisa científica, espera-se a atualização do software, com adição de diversas novas funcionalidades.

DESENVOLVIMENTO DO PROJETO

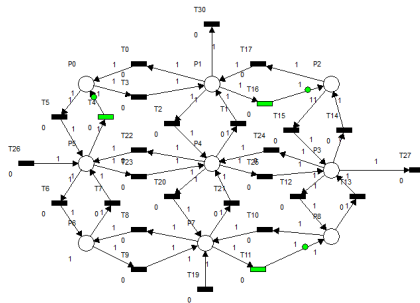
Rede de Petri

HPSim - trafego de carros.hps

File Edit View Window Tools Zoom Simulation Extra ?



trafego de carros.hps



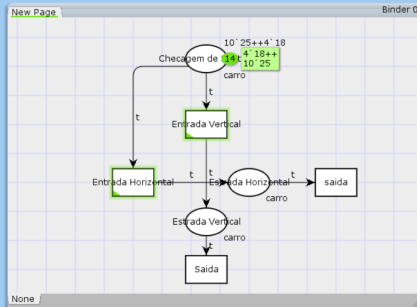
Universidade Estadual da Bahia

DESENVOLVIMENTO DO PROJETO

Rede de Petri Colorida

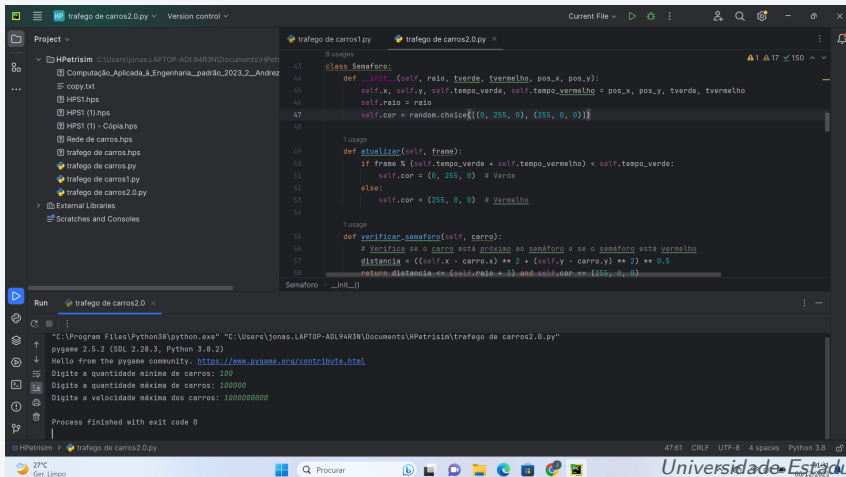
CPN Tools (Version 4.0.1, February 2015)

- Tool box
 - Auxiliary
 - Create
 - Declare
 - Hierarchy
 - Monitoring
 - Net
 - Simulation
 - State space
 - Style
 - View
- Help
- Options
- atividade petri_colorida.cpn
 - Step: 0
 - Time: 0
 - Options
 - History
 - Declarations
 - Standard priorities
 - Standard declarations
 - colset UNIT
 - colset BOOL
 - colset INT
 - colset INTINF
 - colset TIME
 - colset REAL
 - colset STRING = string;
 - colset carro=int;
 - var t:carro;
 - Monitors
 - New Page



DESENVOLVIMENTO DO PROJETO

Codigo Python



```
Project: HPetrism C:\Users\jonas.LAPTOP-ADL94R3N\Documents\HPetrism\
  Computação_Aplicada_A_Engenharia_padrao_2023_2_Andrez
  copy.txt
  HPS1.hps
  HPS1 (1).hps
  HPS1 (1) - Cópia.hps
  Rede de carros.hps
  trafego de carros.hps
  trafego de carros.py
  trafego de carros1.py
  trafego de carros2.0.py
  External Libraries
  Scratches and Consoles

trafego de carros2.0.py
trafego de carros1.py
trafego de carros2.0.py

class Semaphore:
    def __init__(self, raio, tverde, tvermelho, pos_x, pos_y):
        self.x, self.y, self.tempo_verde, self.tempo_vermelho = pos_x, pos_y, tverde, tvermelho
        self.raio = raio
        self.cor = random.choice([(0, 255, 0), (255, 0, 0)])

    def atualizar(self, frame):
        if frame % (self.tempo_verde + self.tempo_vermelho) < self.tempo_verde:
            self.cor = (0, 255, 0) # Verde
        else:
            self.cor = (255, 0, 0) # Vermelho

    def verificar_semaforo(self, carro):
        # Verifica se o carro está próximo ao semáforo e se o semáforo está vermelho
        distancia = ((self.x - carro.x) ** 2 + (self.y - carro.y) ** 2) ** 0.5
        return distancia <= (self.raio + 3) and self.cor == (255, 0, 0)

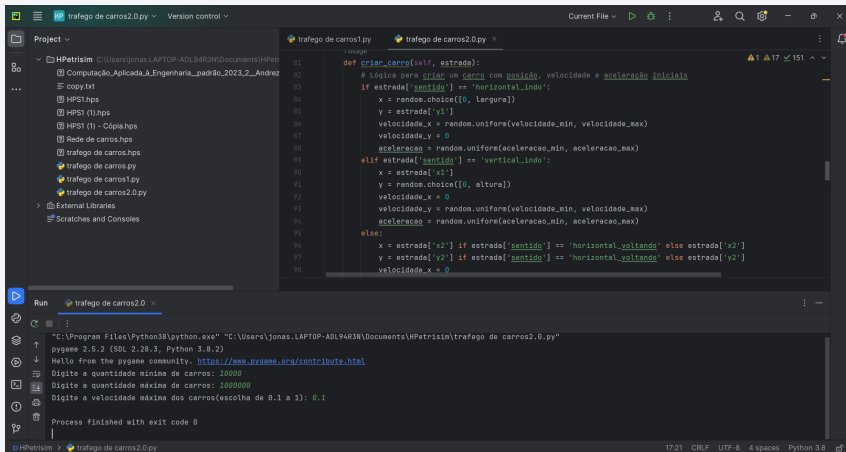
    def __init__()
```

Run: trafego de carros2.0

```
"C:\Program Files\Python38\python.exe" "C:\Users\jonas.LAPTOP-ADL94R3N\Documents\HPetrism\trafego de carros2.0.py"
pygame 2.5.2 (SDL 2.28.3, Python 3.8.2)
Hello from the pygame community. https://www.pygame.org/contribute.html
Digite a quantidade minima de carros: 100
Digite a quantidade máxima de carros: 100000
Digite a velocidade máxima dos carros: 1000000000
Process finished with exit code 0
```

DESENVOLVIMENTO DO PROJETO

Código Python



```
Project: HPetrism
  Computação Aplicada à Engenharia - padrão 2023.2 - Andrez
  copy.txt
  HPS1.hps
  HPS1 (1).hps
  HPS1 (1) - Cópia.hps
  Rede de carros.hps
  trafego de carros.hps
  trafego de carros.py
  trafego de carros1.py
  trafego de carros2.0.py
  External Libraries
  Scratches and Consoles

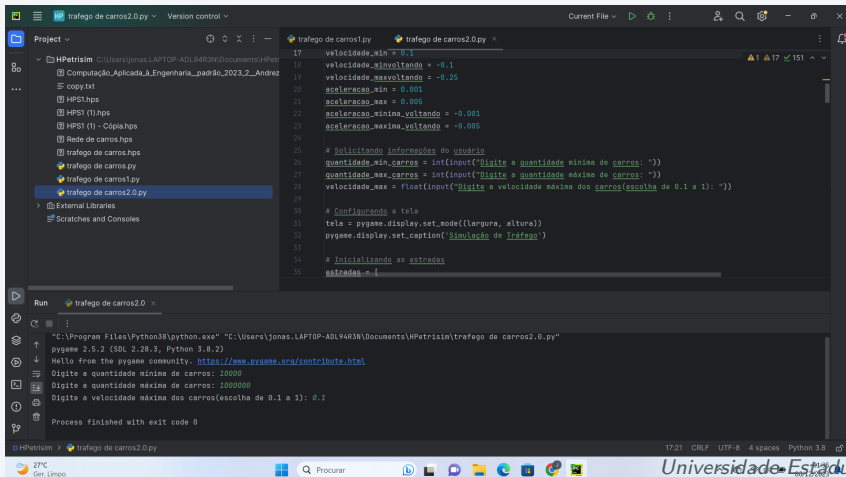
trafego de carros.py
trafego de carros2.0.py

Run: trafego de carros2.0 <

"C:\Program Files\Python38\python.exe" "C:\Users\jonas.LAPTOP-ADL94R3N\Documents\HPetrism\trafego de carros2.0.py"
pygame 2.5.2 (SDL 2.28.3, Python 3.8.2)
Hello from the pygame community. https://www.pygame.org/contribute.html
Digite a quantidade mínima de carros: 10000
Digite a quantidade máxima de carros: 1000000
Digite a velocidade máxima dos carros(escolha de 0.1 a 1): 0.1
Process finished with exit code 0
```

DESENVOLVIMENTO DO PROJETO

Codigo Python



The screenshot shows a Python IDE with a project named 'tráfego de carros2.0.py'. The code is written in Portuguese and defines variables for car speed, acceleration, and quantity. It also includes comments for user input and screen configuration.

```
17 velocidade_min = 0.1
18 velocidade_sinuando = -0.1
19 velocidade_maxvoltando = -0.25
20 aceleracao_min = 0.001
21 aceleracao_max = 0.005
22 aceleracao_minima_voltando = -0.001
23 aceleracao_maxima_voltando = -0.005
24
25 # Solicitando informações do usuário
26 quantidade_min_carros = int(input("Digite a quantidade mínima de carros: "))
27 quantidade_max_carros = int(input("Digite a quantidade máxima de carros: "))
28 velocidade_max = float(input("Digite a velocidade máxima dos carros(escolha de 0.1 a 1): "))
29
30 # Configurando a tela
31 tela = pygame.display.set_mode((largura, altura))
32 pygame.display.set_caption('Simulação de Tráfego')
33
34 # Inicializando as estradas
35 estradas = []
```

The Run console shows the execution of the script, displaying the pygame version and the user input for the number of cars (1000000) and the maximum speed (0.1).

```
Run: tráfego de carros2.0
"C:\Program Files\Python38\python.exe" "C:\Users\jonas.LAPTOP-ADL94R3N\Documents\HPetrism\tráfego de carros2.0.py"
pygame 2.5.2 (SDL 2.28.3, Python 3.8.2)
Hello from the pygame community. https://www.pygame.org/contribute.html
Digite a quantidade mínima de carros: 10000
Digite a quantidade máxima de carros: 1000000
Digite a velocidade máxima dos carros(escolha de 0.1 a 1): 0.1
Process finished with exit code 0
```

METODOLOGIA

A metodologia desse projeto de pesquisa foi realizado da seguinte forma:

- Aprofundamento Teórico do Objeto de Estudo
- Coleta de Dados
- Tratamento de Dados
- Análise de Dados (Gráficos)
- Desenvolvimento do Trabalho

CRONOGRAMA

Discriminação	1 mês	2 meses	3 meses	4 meses	5 meses
Aperfeiçoamento Teórico	x	x	x	x	x
Coleta de Dados		x	x	x	
Tratamento de Dados			x	x	
Análise de Dados				x	x
Discussão do Trabalho					x
Considerações Finais					x
Referências	x	x	x	x	x
Trabalho Acadêmico					x

Cuadro: Cronograma de Atividades

REPERTÓRIO

Github

- <https://github.com/Sanojgds/CodeList>