

# Problema de Colocação Ótima de Câmeras de Segurança



**Antoniél Magalhães**  
**João Leahy**  
**Luis Felipe**



# Agenda

1. Introdução
2. Descrição do Problema
3. Solução Algorítmica
4. Detalhes de Implementação
5. Experimentos
6. Resultados
7. Conclusão
8. Referências

# Introdução

- **Contextualização e Motivação:**

- Aplicação de Teoria dos Grafos em problemas de segurança.
- Cenário urbano: bairro de Ondina (Salvador).

- **Justificativa:**

- Modelagem de problemas de vigilância como *Cobertura de Vértices*.
- Permite explorar algoritmos gulosos, programação dinâmica e heurísticas.

- **Objetivo Geral:**

- Encontrar o menor conjunto de locais para instalação de câmeras que cubra todas as áreas de interesse.

# Descrição do Problema

- **Formalização:**

- Grafo  $G = (V, E)$ : vértices representam locais para câmeras; arestas representam conexões que devem ser monitoradas.

- **Restrições e Função Objetivo:**

- Minimizar número de câmeras (subconjunto de vértices).
- Garantir que cada aresta possua ao menos um extremo coberto.

- **Aplicação em Ondina:**

- Extração de dados via OpenStreetMap.
- Simplificação das vias em um grafo.

# Solução Algorítmica

- **Cobertura de Vértices:**
  - Problema clássico de otimização em grafos.
  - NP-completo; soluções exatas podem ser inviáveis para grandes instâncias.
- **Abordagem Gulosa (Greedy):**
  - Selecionar iterativamente o vértice que cobre o maior número de arestas não cobertas.
  - Boa eficiência computacional, mas não garante solução ótima em todos os casos.
- **Pseudo-Código (Cobertura Completa):**

---

## Algorithm 1 Cobertura de Vértices Gulosa

---

```
1:  $C \leftarrow \emptyset$ 
2:  $E' \leftarrow E$                                 ▷ Conjunto de arestas não cobertas
3: while  $E' \neq \emptyset$  do
4:   Selecione  $v \in V$  que incide em mais arestas de  $E'$ 
5:    $C \leftarrow C \cup \{v\}$ 
6:   Remova as arestas incidentes a  $v$  de  $E'$ 
7: end while
```

# Detalhes de Implementação

- **Ferramentas e Linguagens:**
  - Python + NetworkX.
  - Scripts para extração de dados (OpenStreetMap) e conversão em .json.
- **Estrutura do Projeto (GitHub):**
  - scripts/: Implementações das heurísticas, criação do grafo, visualizações.
  - instancias/: Armazena dados do bairro de Ondina em formato JSON.
  - resultados/: Saídas dos algoritmos e visualizações.
- **Otimizações:**
  - Uso de set para controle de vértices cobertos.
  - Cálculo incremental das arestas não cobertas.

# Experimentos

- **Metodologia:**

- Geração do grafo de Ondina via dados reais (OSM).
- Aplicação do Algoritmo Guloso e variações (Cobertura Máxima, p-limitada).

- **CrITÉrios de Avaliação:**

- Taxa de cobertura (arestas ou vértices).
- Número de câmeras utilizadas.
- Tempo de execução.

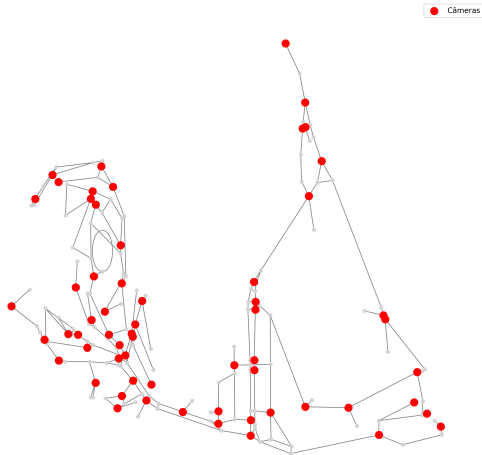
# Resultados

- **Exemplo de Execução em Ondina:**
  - Cobertura Completa: 61 câmeras para cobrir 182 vértices.
  - Cobertura Máxima ( $p=40$ ): 84% de cobertura do grafo.
- **Visualização:**
  - Gráficos mostram os locais de instalação (pontos vermelhos) e vértices cobertos.
- **Análise:**
  - Algoritmos gulosos podem fornecer soluções de boa qualidade em tempo polinomial.
  - Para restrições orçamentárias, a cobertura máxima parcial pode ser suficiente.

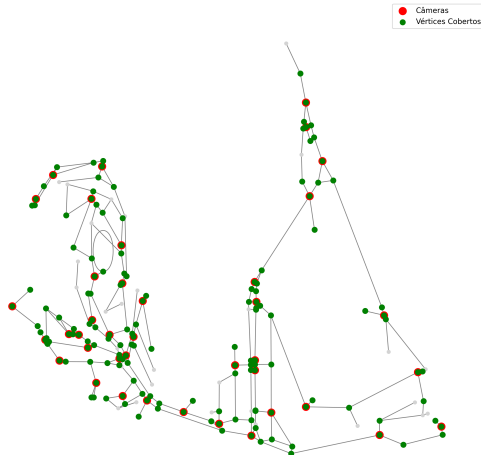


# Comparação: Cobertura Completa vs. Máxima

Cobertura Completa  
61 câmeras cobrindo 182 vértices



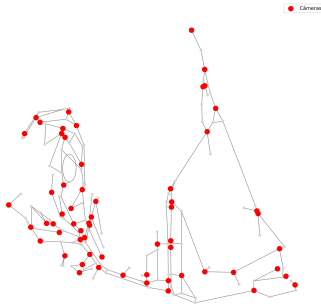
Cobertura Máxima  
40 câmeras cobrindo 153 vértices



# Comparação com Algoritmo Genético

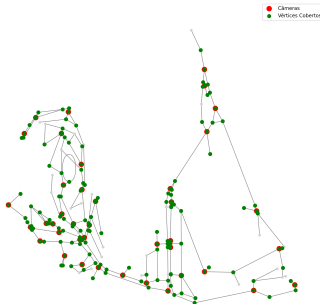
Cobertura Completa (Guloso)

61 câmeras cobrindo 182 vértices



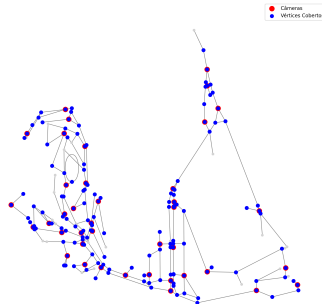
Cobertura Máxima (Guloso)

40 câmeras cobrindo 153 vértices



Cobertura Máxima (Genético)

40 câmeras cobrindo 156 vértices



# Conclusão

- **Aplicabilidade:**
  - Teoria dos Grafos oferece base sólida para problemas de vigilância urbana.
  - Solução viável para cenários reais de segurança pública.
- **Contribuições:**
  - Integração de conceitos de Cobertura de Vértices com dados abertos do OSM.
  - Comparação entre abordagens gulosas e outras heurísticas (por exemplo, genéticas).

# Referências

- Lopes Filho, J. G. (2019). *Problema do Caixeiro Viajante com Coleta Opcional de Bônus, Tempo de Coleta e Passageiros*. Tese de Doutorado, UFRN.
- Carvalho, M. R. (2022). Métodos Heurísticos para o TSP-OBP. *Journal of Combinatorial Optimization*.
- Carnielli, W. e Epstein, R. (2017). *Computabilidade e Funções Computáveis*. UNESP.
- Goldbarg, M. e Goldbarg, E. (2012). *Grafos: Conceitos, algoritmos e aplicações*. Elsevier.
- Repositório do Projeto:  
<https://github.com/antoniell/mata53-projeto-final>