

1. Captura e Contagem de Veículos

- **História de Usuário:** Como administrador da cidade, quero que o sistema seja capaz de capturar e contar o número de veículos que passam por um ponto específico para que eu possa monitorar o fluxo de tráfego em tempo real.
 - **Tarefas:**
 - Configurar a câmera com o Arduino para capturar imagens de uma via pública.
 - Desenvolver um script em Python usando OpenCV para processar as imagens e identificar os veículos.
 - Integrar o contador de veículos ao servidor para o armazenamento dos dados.
 - **Critérios de Aceitação:**
 - A contagem de veículos deve ser precisa, com uma margem de erro mínima.
 - O sistema não deve armazenar dados pessoais dos veículos, apenas a contagem total.
-

2. Registro de Fluxo em Diferentes Horários

- **História de Usuário:** Como gestor público, preciso que o sistema registre o fluxo de veículos em horários variados do dia, gerando dados históricos para futuras análises.
 - **Tarefas:**
 - Implementar o armazenamento dos dados de contagem (número de veículos e horário) no MongoDB.
 - Assegurar que o sistema registre os dados continuamente sem interrupções.
 - **Critérios de Aceitação:**
 - Os dados de contagem devem ser armazenados com o horário exato.
 - Os dados agregados devem ser facilmente recuperáveis para futuras análises.
-

3. Integração IoT e Comunicação Arduino

- **História de Usuário:** Como desenvolvedor de infraestrutura, quero que o sistema esteja integrado a uma rede IoT para monitorar simultaneamente múltiplos pontos de tráfego.
 - **Tarefas:**
 - Configurar a comunicação entre o Arduino e o servidor usando protocolos IoT (MQTT ou HTTP).
 - Desenvolver uma central de dados para receber e consolidar informações de múltiplos sensores.
 - **Critérios de Aceitação:**
 - O sistema deve ser capaz de monitorar múltiplos pontos de tráfego simultaneamente.
 - A comunicação deve ser confiável, com baixa latência e sem perda de dados.
-

4. Visualização de Dados em Tempo Real

- **História de Usuário:** Como gestor público, quero visualizar a contagem de veículos em tempo real através de uma interface web para poder tomar decisões rápidas sobre a gestão do tráfego. • **Tarefas:**
 - Desenvolver uma interface web usando HTML, CSS e Bootstrap para exibir a contagem em tempo real.
 - Criar gráficos dinâmicos em JavaScript para ilustrar a evolução do fluxo de veículos.
 - Implementar um painel central que mostre múltiplos pontos de tráfego em um mapa interativo da cidade.
 - **CrITÉrios de Aceitação:**
 - A interface web deve atualizar os dados em tempo real com poucos segundos de intervalo.
 - Os gráficos devem ser responsivos, intuitivos e de fácil interpretação. ○ O painel deve incluir um mapa com os pontos de monitoramento destacados.
-

5. Análise de Dados e Relatórios

- **História de Usuário:** Como gestor público, quero acessar relatórios que mostram o fluxo de veículos em períodos específicos para entender os padrões de tráfego e tomar decisões de planejamento urbano. • **Tarefas:**
 - Integrar o sistema com o plotly para gerar gráficos.
 - Desenvolver relatórios históricos diários, semanais e mensais, baseados nos dados de contagem.
 - Implementar filtros para permitir a análise de diferentes períodos e locais da cidade.
 - **CrITÉrios de Aceitação:**
 - Os relatórios devem ser gerados automaticamente e permitir comparações entre diferentes períodos.
-

6. Segurança e Privacidade

- **História de Usuário:** Como gestor público, preciso garantir que o sistema respeite as diretrizes de privacidade, não capturando dados pessoais dos veículos para evitar problemas legais.
- **Tarefas:**
 - Implementar medidas para garantir que apenas o número total de veículos seja contado, sem capturar informações pessoais ou imagens de motoristas.
 - Auditar regularmente o sistema para garantir que nenhuma informação indevida esteja sendo armazenada.
- **CrITÉrios de Aceitação:**
 - O sistema deve ser capaz de contar os veículos sem armazenar imagens ou dados pessoais.

- Auditorias frequentes devem garantir que as normas de privacidade sejam cumpridas.
-

7. Escalabilidade e Manutenção

- **História de Usuário:** Como administrador do sistema, preciso que ele seja escalável e fácil de manter, para que possamos expandir o monitoramento para outras áreas da cidade no futuro. • **Tarefas:**
 - Desenvolver o sistema de forma modular, permitindo fácil expansão e inclusão de novos pontos de monitoramento.
 - Documentar todo o código e os processos para facilitar a manutenção e futuras integrações.
 - Garantir que o sistema funcione eficientemente, mesmo em áreas com maior volume de tráfego.
 - **Critérios de Aceitação:**
 - O sistema deve ser modular, permitindo que novos pontos sejam adicionados com mínima intervenção no código.
 - A documentação deve ser clara, abrangente e atualizada, facilitando a manutenção.
-

8. Testes e Implementação

- **História de Usuário:** Como desenvolvedor, quero garantir que o sistema seja amplamente testado em cenários reais antes de sua implementação em grande escala, para assegurar sua confiabilidade.
- **Tarefas:**
 - Realizar testes de contagem de veículos em ambientes simulados e reais.
 - Verificar a integração com o banco de dados MongoDB e o Power BI, testando a geração correta de relatórios.
 - Testar o desempenho da interface de visualização em tempo real e corrigir eventuais falhas.
- **Critérios de Aceitação:**
 - O sistema deve funcionar conforme o esperado com diferentes volumes de tráfego.
 - Todos os componentes devem operar de maneira integrada, sem erros ou perdas de dados.