**Fluxos de Eventos**

**1. Captura e Identificação de Veículos**

**Fluxo Principal:**

1. O sistema inicializa o hardware, conectando a câmera ao Arduino e confirmando sua funcionalidade.
2. A câmera começa a capturar imagens em tempo real de uma via pública.
3. O script em Python com OpenCV processa as imagens capturadas.
4. Veículos são identificados e contados automaticamente.
5. O número total de veículos é exibido no painel do sistema.
6. O sistema armazena apenas a contagem total no banco de dados, sem informações pessoais ou identificadores específicos.

**Fluxo Alternativo 1 (Falha na Conexão do Hardware):**

1. Caso o hardware (câmera ou Arduino) não esteja funcional, o sistema exibe uma mensagem de erro ao operador.
2. O operador é instruído a verificar conexões e reiniciar o hardware.

**Fluxo Alternativo 2 (Erro no Processamento de Imagens):**

1. Se o script de processamento encontrar erros, o sistema registra a falha e interrompe a contagem.
2. O operador é notificado para revisar o script ou a qualidade da câmera.

**2. Registro de Fluxo em Diferentes Horários**

**Fluxo Principal:**

1. O sistema inicia o armazenamento contínuo de dados de contagem de veículos.
2. Para cada contagem, o sistema registra:
   * Data e hora da captura.
   * Número total de veículos detectados.
3. Os dados são enviados ao banco de dados MongoDB.
4. O sistema verifica duplicações antes de salvar os registros.
5. Dados registrados são consolidados em relatórios disponíveis para consulta.

**Fluxo Alternativo 1 (Falha no Armazenamento):**

1. Caso o MongoDB não esteja acessível, o sistema armazena os dados temporariamente em cache.
2. Quando a conexão com o banco de dados é restabelecida, os dados em cache são sincronizados.

**Fluxo Alternativo 2 (Duplicação de Dados):**

1. Se registros duplicados forem detectados, o sistema elimina duplicatas antes de armazenar os dados.

**3. Exibição Intuitiva de Dados**

**Fluxo Principal:**

1. O usuário acessa a página de gráficos no sistema.
2. O sistema exibe os gráficos padrão com dados gerais.
3. O usuário seleciona filtros, como:
   * Intervalos de datas.
   * Horários específicos.
4. O sistema aplica os filtros, atualiza os gráficos e apresenta os dados ajustados.
5. O usuário interage com os gráficos para explorar informações detalhadas.
6. Se necessário, o usuário exporta os gráficos e dados filtrados.

**Fluxo Alternativo 1 (Filtros Inválidos):**

1. Caso o usuário insira filtros incompatíveis (ex.: datas futuras ou intervalos inválidos), o sistema exibe uma mensagem de erro.
2. O sistema solicita que o usuário insira parâmetros válidos.

**Fluxo Alternativo 2 (Erro no Gráfico):**

1. Se o gráfico não for gerado corretamente, o sistema apresenta uma mensagem de erro.
2. O sistema registra a falha para análise posterior e oferece a opção de recarregar a página.

**4. Análise de Dados e Histórico**

**Fluxo Principal:**

1. O usuário acessa a seção de análise de dados no sistema.
2. O sistema exibe a interface padrão com gráficos históricos gerais.
3. O usuário aplica filtros para selecionar períodos específicos, como datas e horários.
4. O sistema recupera os dados históricos do banco de dados MongoDB.
5. Os gráficos são gerados com base nos filtros aplicados.
6. O usuário analisa os padrões de tráfego exibidos nos gráficos.
7. O usuário utiliza as informações para planejar ações ou exportar os resultados.

**Fluxo Alternativo 1 (Filtros Inválidos):**

1. Caso o usuário aplique filtros inválidos, o sistema exibe uma mensagem de erro.
2. O sistema solicita parâmetros corretos antes de continuar.

**Fluxo Alternativo 2 (Dados Incompletos):**

1. Se os dados históricos estiverem incompletos ou ausentes para o período selecionado, o sistema notifica o usuário.
2. O sistema oferece a opção de exibir os dados disponíveis.