**1. Especificação Detalhada de Classes e Métodos**

Cada classe e método pode ser descrito assim:

**Classe CrimeData**

* **Descrição:** Representa os dados de crimes importados do CSV.
* **Atributos:**
  + data: str (formato YYYY-MM-DD)
  + hora: str (formato HH:MM)
  + dia\_semana: str
  + tipo\_crime: str
  + bairro: str
  + logradouro: str
  + coordenadas: Tuple[float, float]
  + setor: str
* **Métodos:**
  + processar\_dados(self) -> pd.DataFrame
    - **Descrição:** Carrega e valida os dados do arquivo CSV. Retorna um DataFrame.
    - **Exemplo de Uso:** dados = CrimeData().processar\_dados()

**Classe PrevisorCrime**

* **Descrição:** Responsável por treinar e fazer previsões de locais e horários de crimes.
* **Atributos:**
  + algoritmo: Modelo preditivo (e.g., RandomForest, Keras Sequential)
  + parametros: Configurações específicas do modelo.
* **Métodos:**
  + treinar\_modelo(self, dados: pd.DataFrame) -> None
    - **Descrição:** Treina o modelo de previsão com dados históricos.
  + prever\_local\_horario(self, dados: pd.DataFrame) -> pd.DataFrame
    - **Descrição:** Previsão de locais e horários de ocorrência.
    - **Saída:** DataFrame com bairro, logradouro, coordenadas, objetivo.

**Classe CartaoPrograma**

* **Descrição:** Gera o cartão programa com objetivos, missões e locais de patrulhamento.
* **Atributos:**
  + local\_patrulhamento: str
  + horario\_inicio: str
  + horario\_termino: str
  + setor: str
  + num\_indicacao\_ponto: int
  + objetivo: str
  + missao: str
  + observacao: str
* **Métodos:**
  + gerar\_pdf(self, caminho: str) -> None
    - **Descrição:** Gera o PDF do cartão programa.
    - **Parâmetros:** caminho: Caminho de destino para salvar o PDF.
    - **Exemplo de Uso:** CartaoPrograma().gerar\_pdf('cartao\_programa.pdf')

**2. Diagrama de Fluxo e Arquitetura do Sistema**

O diagrama de fluxo ajuda a visualizar a interação entre classes e a sequência de operações, como na criação do cartão programa:

1. **Carregar Dados:** CrimeData.processar\_dados()
2. **Treinar e Prever com Modelo:** PrevisorCrime.treinar\_modelo(), PrevisorCrime.prever\_local\_horario()
3. **Gerar Cartão Programa:** CartaoPrograma.gerar\_pdf()

**3. Estrutura de Dados e Regras de Negócio**

**Estrutura do CSV**

* **Campos Obrigatórios:** bairro, logradouro, coordenadas, tipo\_crime, setor
* **Regras:**
  + **Objetivo:** Definir o objetivo de cada ponto (e.g., "aumento de furtos") com base na análise do tipo\_crime.
  + **Missão:** Associada ao objetivo, com opções de ações ("ponto base", "batida policial") que podem ser editadas pelo usuário.

**Transformações e Regras:**

* **Tempo de Permanência Máximo:** 30 minutos.
* **Sequência de Pontos:** Ordenação por horario\_inicio.

**4. Testes Unitários e Cenários de Validação**

Para cada função crítica, incluir cenários de teste e saídas esperadas. Por exemplo:

**Teste para processar\_dados**

* **Entrada:** Arquivo CSV com registros incompletos (sem bairro)
* **Saída Esperada:** Erro de validação ou relatório indicando campos ausentes.

**Teste para gerar\_pdf**

* **Entrada:** Ponto de patrulhamento com objetivo, missao, observacao.
* **Saída Esperada:** Arquivo PDF gerado com os campos preenchidos.

**5. Protocolo de Interface de Usuário e Layout Especificado**

**Mockups e Layouts**

Criar mockups para cada página ajuda a IA a entender como o código precisa construir a interface. Descrever cada elemento visual com detalhamento, incluindo:

* **Página Inicial:** Gráficos com tendências, botões de navegação para os módulos de Importar Dados, Configuração do Cartão.
* **Página de Configuração do Cartão Programa:** Campos de edição para missão e observação, uma lista visual dos pontos com objetivo e horários.

**Especificações de Interação**

* **Exemplo de Botões:** Botão de “Salvar” deve acionar a função CartaoPrograma.gerar\_pdf().
* **Filtros de Visualização:** Por bairro, setor, dia da semana.