

Trabalho Prático

-

POLICECASEREGISTER

UFMG



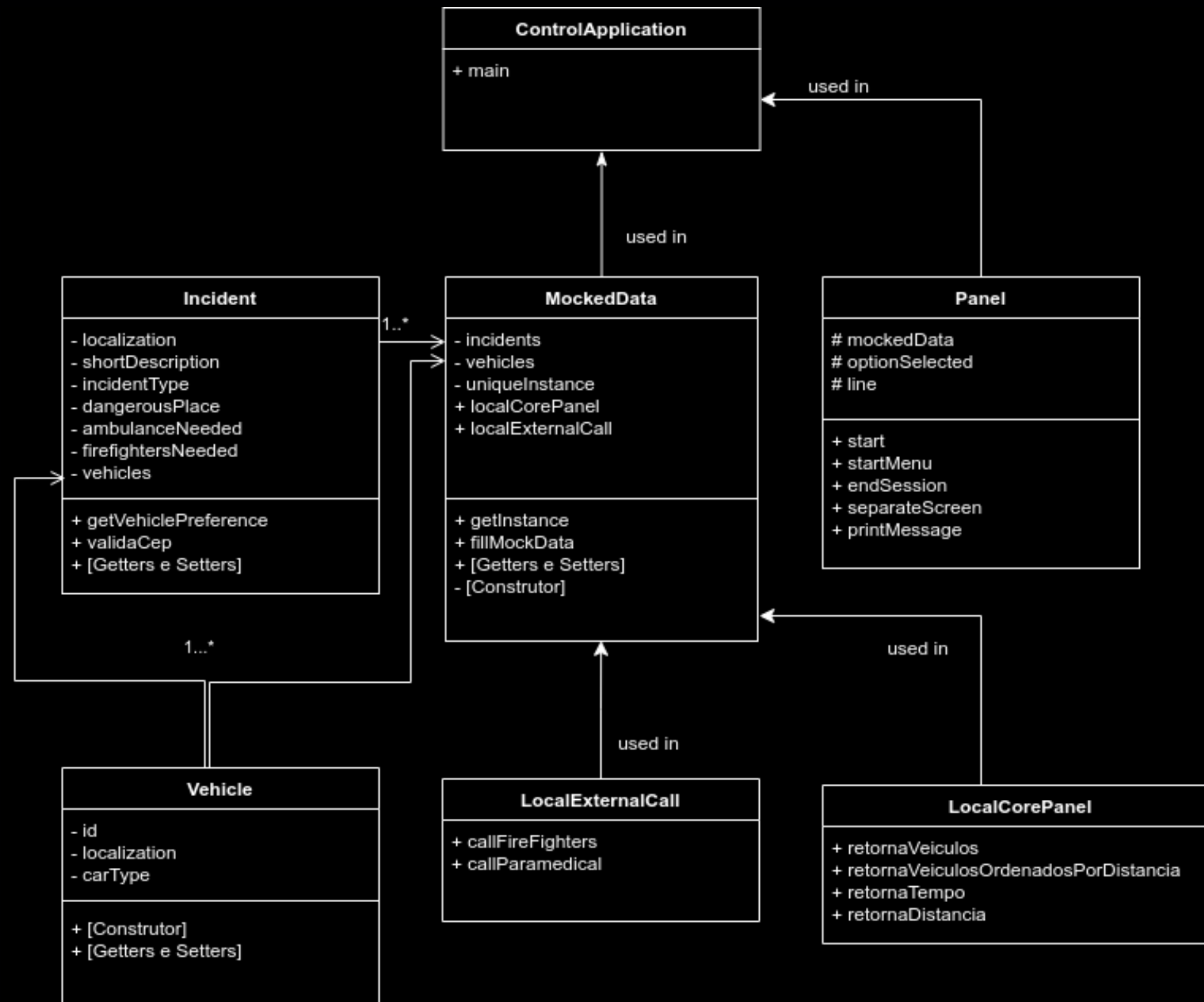
Introdução

Recapitulando

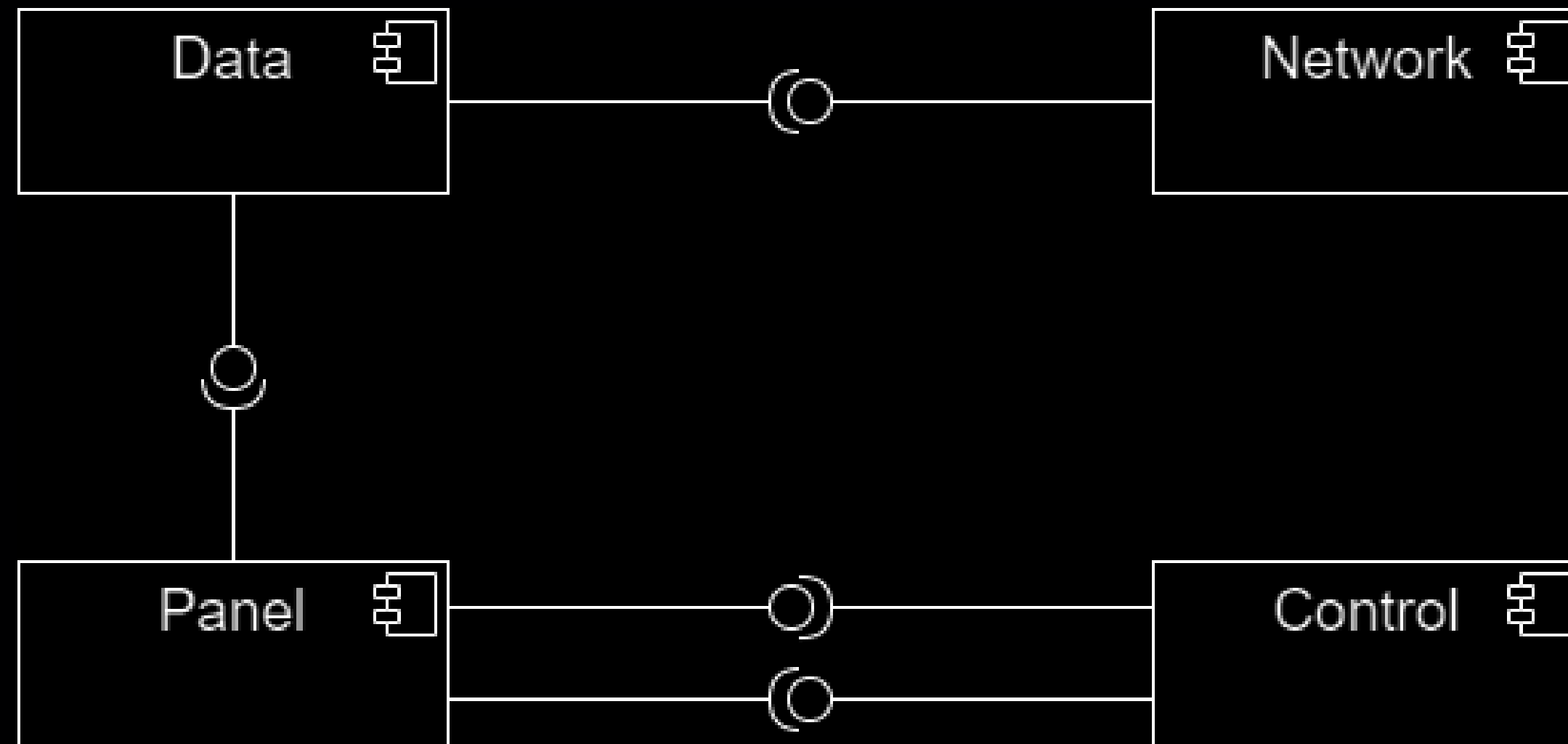
O Police Case Register é um sistema que permite ao usuário **registrar** e **acompanhar** chamadas policiais.

- Modelo Cascata - Software **crítico**, documentos (diagramas) bem encaixados na estrutura
- Fases: modelagem, implementação, validação e manutenção

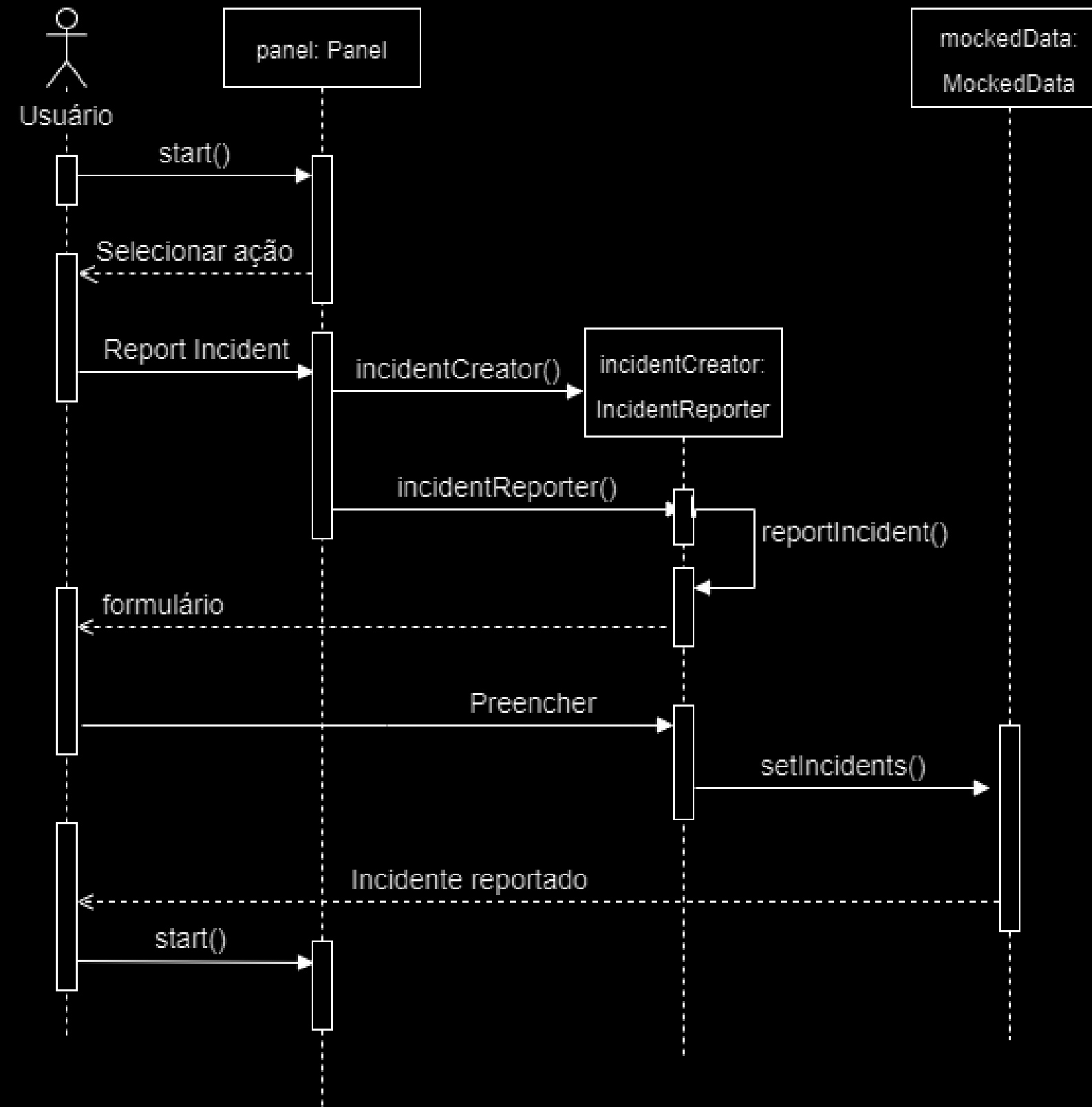
Modelagem



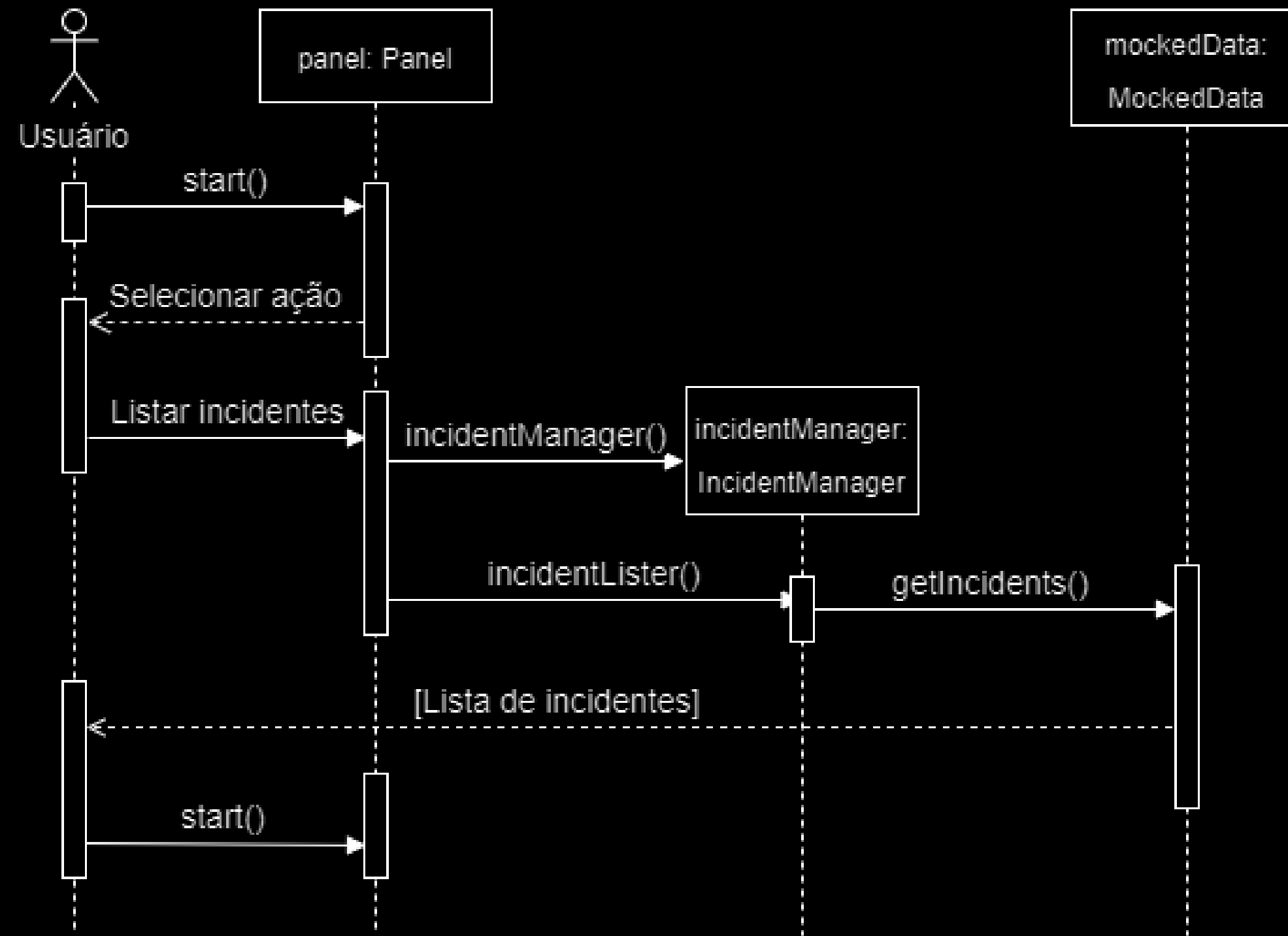
Modelagem



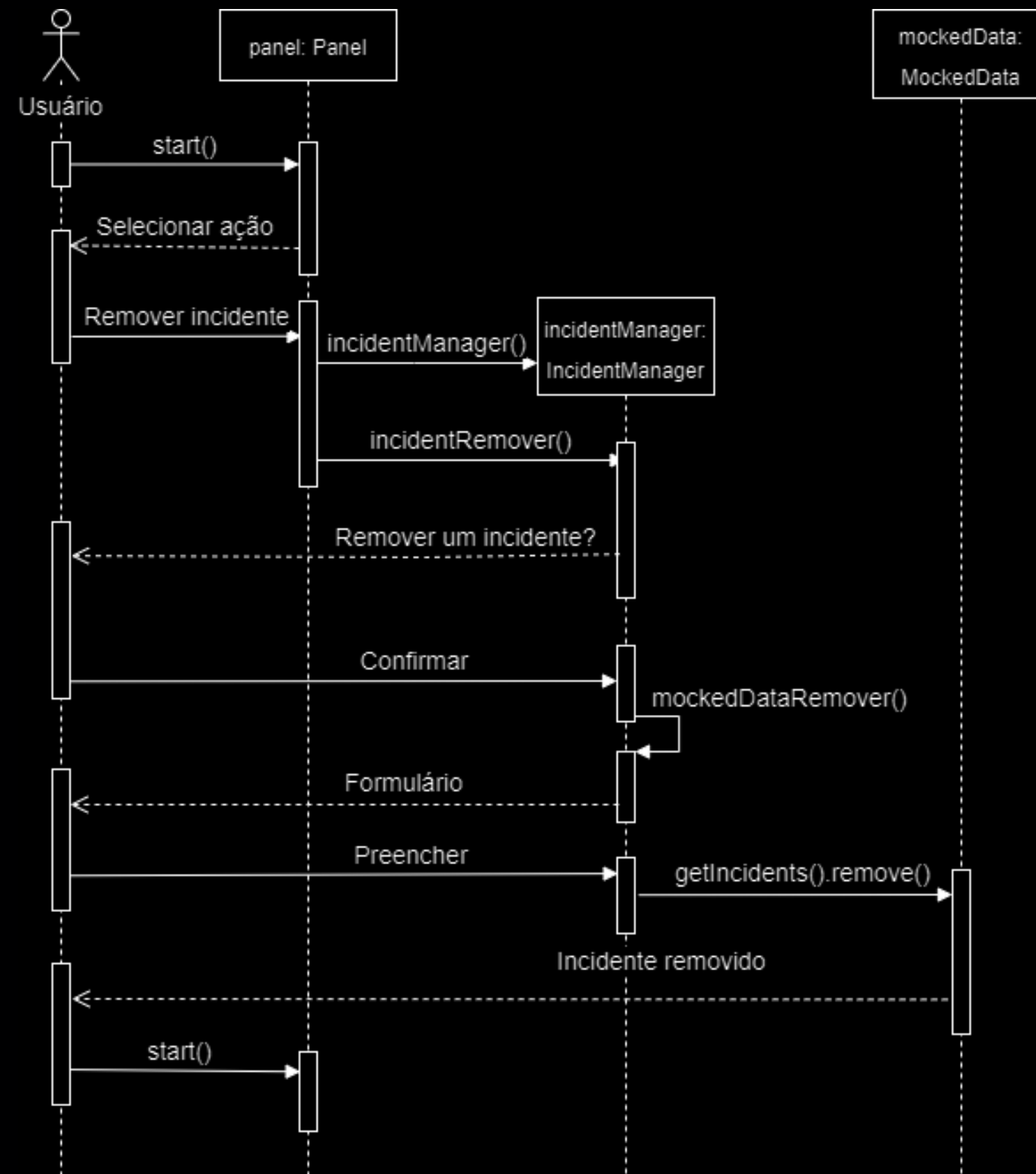
Modelagem



Modelagem

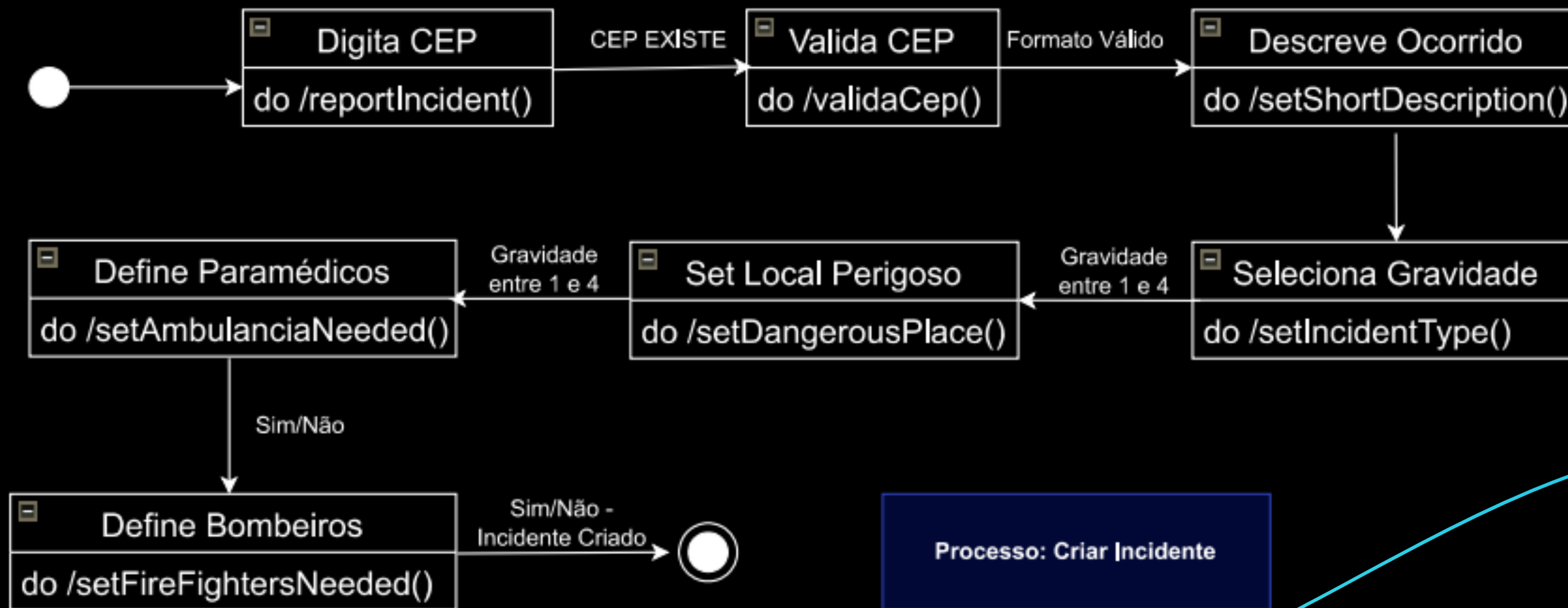


Modelagem



Modelagem

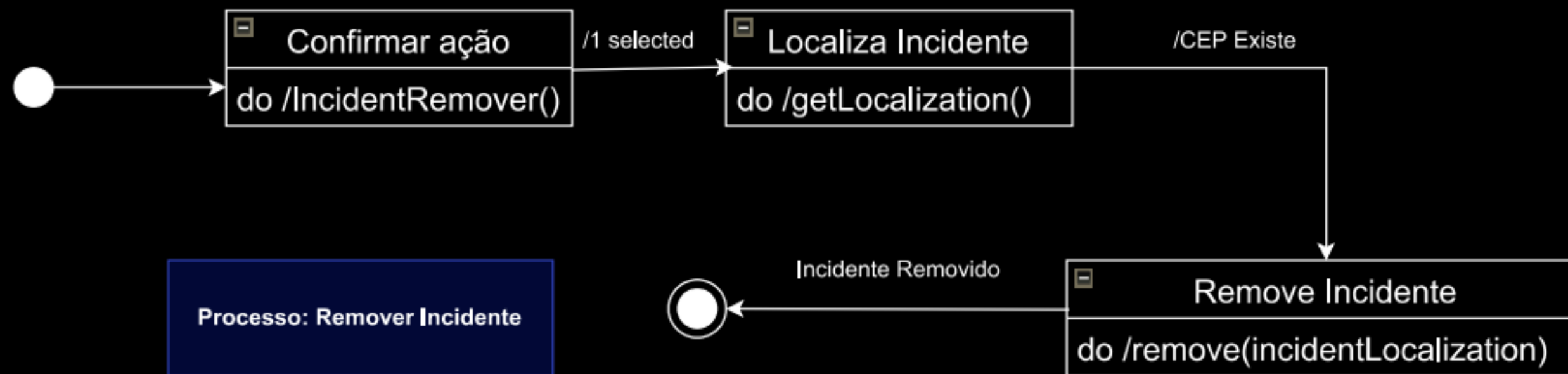
Diagrama de Estados



Processo: Criar Incidente

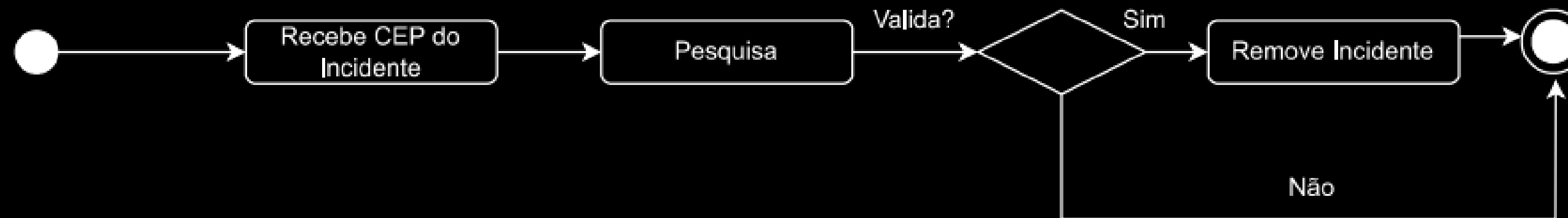
Modelagem

Diagrama de Estados



Modelagem

Diagrama de
Atividades



Processo: Remover Incidente

Desenvolvimento

Durante o processo de desenvolvimento, utilizamos vários dos conteúdos aprendidos na disciplina, em especial:

- Idiomas de programação
- Princípios de Orientação a Objetos
- Interpretação de diagramas

Desenvolvimento

Utilizamos também outros conhecimentos da área de Engenharia de Software:

- Design Pattern: Singleton
- Injeção de dependências com interfaces

3 usages  Espezzialy Souza

```
public class LocalCorePanel implements InterfaceCorePanel {
```

```
private MockedData() {}
```

```
public static synchronized MockedData getInstance() {  
    if (uniqueInstance == null)  
        uniqueInstance = new MockedData();  
  
    return uniqueInstance;  
}
```

Verificação & Validação

Com base nos critérios de V&V estabelecidos na primeira etapa do TP, implementamos testes unitários correspondentes.

```
@Test
public void shouldNotSendNormalCarsToDangerousPlace() {
    Incident incident = new Incident(
        localization: "30240130",
        shortDescription: "Assalto a mão armada",
        IncidentType.LEVE,
        dangerousPlace: true,
        ambulanceNeeded: true,
        firefightersNeeded: false
    );
    assertFalse(incident.getVehiclePreference().keySet().contains(CarType.CARRO));
}
```

Considerações finais

- O sistema pode, evidentemente, ser expandido. Com isso, podem surgir novas observações sobre *arquitetura e refatoração* para melhor adequar o código.
- Os dados implementados atualmente estão em *mock*, i.e., fazem mímica de dados reais. Uma expansão interessante seria a implementação de um banco de dados.

Demonstração