

Este sistema utiliza sensores para monitorar variáveis ambientais críticas, como qualidade do ar e temperatura. Quando os níveis ultrapassam um limite seguro, o sistema prioriza a análise e gera alertas em tempo real.

Monitoramento Ambiental com Priorização de Eventos

Neste exemplo:

1. Sensores monitoram a qualidade do ar e a temperatura.
2. Um evento crítico ocorre quando os níveis ambientais ultrapassam um limite.
3. Um sistema de priorização garante que eventos críticos sejam tratados antes de processos normais.

Código Python

```
```python
```

```
import simpy
```

```
import random
```

```
class Sensor:
```

```
 def __init__(self, env, name, data_type, critical_threshold, interval):
```

```
 self.env = env
```

```
 self.name = name
```

```
 self.data_type = data_type
```

```
 self.critical_threshold = critical_threshold
```

```
 self.interval = interval
```

```
 self.value = 0
```

```
 self.process = env.process(self.monitor())
```

```
 def monitor(self):
```

```
 while True:
```

```
 Simula a leitura do sensor
```

```
self.value = random.uniform(0, 200) # Valores de 0 a 200

print(f'{self.env.now}: {self.name} ({self.data_type}) = {self.value:.2f}')
```

Verifica se ultrapassa o limite crítico

```
if self.value > self.critical_threshold:

 print(f'{self.env.now}: ALERTA CRÍTICO em {self.name}! Valor =
{self.value:.2f}')

 yield self.env.process(self.handle_critical_event())

else:

 yield self.env.timeout(self.interval) # Tempo entre leituras
```

```
def handle_critical_event(self):

 print(f'{self.env.now}: Tratando evento crítico de {self.name}...')

 yield self.env.timeout(5) # Tempo para resolver o problema

 print(f'{self.env.now}: Evento crítico de {self.name} resolvido.')
```

class EnvironmentalMonitoringSystem:

```
def __init__(self, env):

 self.env = env

 self.sensors = []

def add_sensor(self, sensor):

 self.sensors.append(sensor)

def run(self):

 print(f'Sistema de monitoramento ambiental iniciado no tempo
{self.env.now}.')

 yield self.env.timeout(60) # Executa por 60 segundos

 print(f'Sistema de monitoramento finalizado no tempo {self.env.now}.')
```

Configuração da simulação

```
def simulation():
```

```
 env = simpy.Environment()
```

Criação do sistema de monitoramento ambiental

```
 system = EnvironmentalMonitoringSystem(env)
```

Adiciona sensores ao sistema

```
 sensor_air_quality = Sensor(env, 'Sensor de Qualidade do Ar', 'Qualidade do Ar',
150, 10)
```

```
 sensor_temperature = Sensor(env, 'Sensor de Temperatura', 'Temperatura', 120,
15)
```

```
 system.add_sensor(sensor_air_quality)
```

```
 system.add_sensor(sensor_temperature)
```

Inicia a execução

```
 env.process(system.run())
```

```
 env.run()
```

Inicia a simulação

```
simulation()
```

```
```\n
```

Explicação do Código

1. Sensores:

- Cada sensor monitora uma variável ambiental com base em um intervalo fixo (por exemplo, 10 segundos para qualidade do ar e 15 segundos para temperatura).

- Os valores do sensor são simulados com `random.uniform``, variando entre 0 e 200.

2. Detecção de eventos críticos:

- Quando o valor de um sensor excede o limite crítico (`critical_threshold``), um processo especial é disparado (`handle_critical_event``).
- O evento crítico é tratado antes que o sensor volte à operação normal.

3. Sistema de monitoramento ambiental:

- O sistema centraliza e organiza os sensores, mas cada sensor opera como um processo independente.

4. SimPy:

- Usamos SimPy para simular processos simultâneos de monitoramento e tratamento de eventos críticos.

Saída Esperada

````plaintext`

0: Sistema de monitoramento ambiental iniciado no tempo 0.

0: Sensor de Qualidade do Ar (Qualidade do Ar) = 120.45

0: Sensor de Temperatura (Temperatura) = 105.32

10: Sensor de Qualidade do Ar (Qualidade do Ar) = 155.67

10: ALERTA CRÍTICO em Sensor de Qualidade do Ar! Valor = 155.67

10: Tratando evento crítico de Sensor de Qualidade do Ar...

15: Sensor de Temperatura (Temperatura) = 130.56

15: ALERTA CRÍTICO em Sensor de Temperatura! Valor = 130.56

15: Tratando evento crítico de Sensor de Temperatura...

20: Evento crítico de Sensor de Qualidade do Ar resolvido.

25: Evento crítico de Sensor de Temperatura resolvido.

...

60: Sistema de monitoramento finalizado no tempo 60.

...

#### Expansões Possíveis

1. **Interconexão entre sensores:** Simular como diferentes sensores colaboram em um evento crítico, como um incêndio (temperatura e qualidade do ar simultaneamente).
2. **Ações automatizadas:** Enviar comandos a sistemas externos, como ativar sprinklers ou alertar autoridades.
3. **Visualização de dados:** Plotar os valores do sensor ao longo do tempo usando bibliotecas como Matplotlib.