|  |  |  |  |
| --- | --- | --- | --- |
| **Curso** | Qualificação em IA Industrial | | |
| **Unidade Curricular** | Fundamentos de Python para IA | | |
| **Turma** |  | **Data** |  |
| **Aluno(s)** |  | | |
| **Professor** |  | **Nota** |  |

**PRÁTICA 2 – MONITORAMENTO DE TEMPERATURAS COM ANÁLISE ESTATÍSTICA E SEGURANÇA**

Construir um sistema orientado a objetos para monitoramento de leituras de temperatura, capaz de armazenar, processar e analisar dados, além de verificar se as leituras estão dentro de limites de segurança predefinidos. O projeto deverá utilizar a biblioteca numpy para cálculos estatísticos de máximo, mínimo, média e aplicar conceitos de programação modular e limpa, com métodos específicos para cada etapa do processo. Ao final, o programa deve apresentar estatísticas claras e emitir alertas para valores fora do intervalo seguro.

**ORIENTAÇÕES TÉCNICAS**

Na construção do seu sistema, é obrigatório aplicar:

* Classes em Python para encapsular dados e métodos, estruturando o programa de forma orientada a objetos.
* Uso do numpy para cálculos estatísticos (média, mínimo e máximo) de forma vetorizada.
* Métodos específicos para adicionar leituras, calcular estatísticas e verificar se as temperaturas estão dentro de um intervalo seguro, que por padrão deve ser entre 20°C e 80°C. O programa deve alertar sempre que encontrar valores fora desse intervalo.
* Mensagens claras ao usuário, incluindo a identificação imediata de temperaturas fora do limite seguro.
* Execução de exemplos práticos ao final, demonstrando o funcionamento do monitor, preferencialmente com a inserção de diferentes leituras que permitam observar tanto o caso seguro quanto o acionamento de alertas. A atividade pode ser desenvolvida no ambiente Google Colab, preferencialmente por sua praticidade e integração com bibliotecas, mas também pode ser executada localmente no VS Code. Para isso, é necessário garantir a instalação prévia das bibliotecas scikit-learn, pandas, numpy, matplotlib e seaborn, que serão utilizadas ao longo do desenvolvimento do pipeline.

Algumas práticas são recomendadas para tornar o projeto mais robusto e didático:

* Definir nomes claros para métodos e variáveis, garantindo semântica e legibilidade.
* Comentar o código explicando o objetivo de cada método ou bloco importante.
* Adicionar prints formatados, com unidades (°C) e uso de emojis ou outros símbolos para melhorar a comunicação dos resultados.
* Organizar o código em blocos bem definidos, facilitando testes futuros e manutenção.

Boa Aula!