**ROTEIRO DE AULA PRÁTICA – CAPÍTULO 8**

**DISCIPLINA: ANÁLISE E PROJETO DE SISTEMAS**

**TÍTULO DA AULA: MANUTENÇÃO E EVOLUÇÃO DE SISTEMAS**

**1. Objetivos da Aula**

• Compreender os tipos de manutenção de software (corretiva, adaptativa, perfectiva e preventiva).  
• Aplicar técnicas de refatoração para melhorar a qualidade do código.  
• Analisar um sistema simples e propor melhorias evolutivas.  
• Criar um pequeno exemplo prático de reengenharia.

**2. Recursos Necessários**

• Computadores com ambiente configurado para Python e/ou C.  
• Editor de código (VS Code, Visual Studio, Code::Blocks ou similar).  
• Material de apoio: Capítulo 8 do livro-texto.  
• Ferramentas de versionamento (GitHub, GitLab ou similar).

**3. Estrutura da Aula**

1. Abertura (10 minutos): Introdução aos conceitos de manutenção e evolução.  
2. Revisão Conceitual (20 minutos): Exemplos reais de manutenção de sistemas.  
3. Demonstração (20 minutos): Refatorar um código existente para melhorar legibilidade e desempenho.  
4. Atividade Prática (40 minutos): Refatorar e documentar um código legado, propondo melhorias.  
5. Encerramento (20 minutos): Discussão das melhorias aplicadas e lições aprendidas.

**4. Relatório Final**

O relatório deve conter:  
• Resumo teórico sobre manutenção e evolução de software.  
• Código original (antes da refatoração) e código atualizado.  
• Descrição das melhorias aplicadas.  
• Reflexões sobre o impacto da manutenção no ciclo de vida do software.

**5. Critérios de Avaliação**

• Clareza do resumo teórico (2,0).  
• Qualidade da refatoração e código atualizado (4,0).  
• Justificativa das melhorias aplicadas (2,0).  
• Organização e apresentação do relatório (2,0).

**6. Conclusão**

Ao final desta prática, o estudante será capaz de identificar problemas em códigos existentes e aplicar técnicas de manutenção e evolução para melhorar a qualidade do sistema.