



UNIVERSIDADE FEDERAL DE UBERLÂNDIA



FICHA DE COMPONENTE CURRICULAR

CÓDIGO:	COMPONENTE CURRICULAR: ENGENHARIA DE SOFTWARE	
UNIDADE ACADÊMICA OFERTANTE: FACULDADE DE COMPUTAÇÃO		SIGLA: FACOM
CH TOTAL TEÓRICA: 30 horas	CH TOTAL PRÁTICA: 30 horas	CH TOTAL: 60 horas

1. **OBJETIVOS**

Ao final da disciplina o estudante será capaz de:

O estudante será capaz de especificar, modelar e desenvolver sistemas computacionais, com base em métodos, técnicas, ferramentas e procedimentos propostos pela Engenharia de Software.

Entre as competências a serem desenvolvidas no estudante destacam-se:

1. Ser capaz de utilizar técnicas adequadas de observação, compreensão, registro e análise das necessidades dos usuários e de seus contextos sociais, culturais, legais, ambientais e econômicos;
2. Formular, de maneira ampla e sistêmica, questões de engenharia, considerando o usuário e seu contexto, concebendo soluções criativas, bem como o uso de técnicas adequadas;
3. Ser capaz de modelar os fenômenos, os sistemas físicos e químicos, utilizando as ferramentas matemáticas, estatísticas, computacionais e de simulação, entre outras;
4. Prever os resultados dos sistemas por meio dos modelos;
5. Conceber experimentos que gerem resultados reais para o comportamento dos fenômenos e sistemas em estudo;
6. Verificar e validar os modelos por meio de técnicas adequadas;
7. Ser capaz de conceber e projetar soluções criativas, desejáveis e viáveis, técnica e economicamente, nos contextos em que serão aplicadas;
8. Projetar e determinar os parâmetros construtivos e operacionais para as soluções de Engenharia;
9. Aplicar conceitos de gestão para planejar, supervisionar, elaborar e coordenar projetos e serviços de Engenharia;
10. Ser capaz de expressar-se adequadamente, seja na língua pátria ou em idioma diferente do Português, inclusive por meio do uso consistente das tecnologias digitais de informação e comunicação (TDICs), mantendo-se sempre atualizado em termos de métodos e tecnologias disponíveis;
11. Ser capaz de interagir com as diferentes culturas, mediante o trabalho em equipes presenciais ou a distância, de modo que facilite a construção coletiva;
12. Gerenciar projetos e liderar, de forma proativa e colaborativa, definindo as estratégias e construindo o consenso nos grupos;

13. Aprender a aprender.

2. **EMENTA**

Especificação, análise e desenvolvimento de sistemas computacionais utilizando paradigmas da Engenharia de Software.

3. **PROGRAMA**

1. Introdução

- 1.1 Conceitos básicos
- 1.2 Engenharia de Software e Engenharia de Sistemas

2. Paradigmas da Engenharia de Software

- 2.1 Modelos Prescritivos de Processo
 - 2.1.1 Cascata, Espiral, Iterativo e Incremental, Prototipação e RAD

3. Processos de Desenvolvimento de Software

- 3.1 Exemplos de processos: RUP, XP, FDD e Praxis
- 3.2 Automatização de processos com ferramentas CASE

4. Engenharia de Requisitos

- 4.1 Especificação de requisitos de um sistema
- 4.2 Priorização e rastreabilidade
- 4.3 Documentação
- 4.4 Modelagem com SysML

5. Análise de Sistemas

- 5.1 Paradigmas de Análise
- 5.2 Análise Orientada a Objetos
 - 5.2.1 Introdução à UML (Unified Modeling Language)
 - 5.2.2 Modelagem de Casos de Uso
 - 5.2.3 Modelagem de Classes (Nível Domínio)
 - 5.2.4 Modelagem de banco de dados (DER)

6. Design Orientado a Objetos

- 6.1 Design Arquitetural
- 6.2 Diagramas da UML voltados ao design de um sistema
 - 6.2.1 Modelagem de Classes (Nível Especificação)
 - 6.2.2 Modelagem de Interações
 - 6.2.3 Modelagem de Estados
- 6.3 Design de Interface com o Usuário

7. Qualidade de Software

- 7.1 Normas e Padrões de Qualidade

8. Verificação e Validação de Software (V&V)

- 8.1 Conceitos básicos de V&V

8.2 Teste de Software

4. **BIBLIOGRAFIA BÁSICA**

1. BEZERRA, E.. **Princípios de análise e projeto de sistemas com UML**. 3. ed. Rio de Janeiro: Elsevier, 2015.
2. PRESSMAN, R. **Engenharia de software: uma abordagem profissional**. Porto Alegre: McGraw-Hill, 2011.
3. SOMMERVILLE, I. **Engenharia de software**. São Paulo: Prentice Hall, 2011.

5. **BIBLIOGRAFIA COMPLEMENTAR**

1. FRIEDENTHAL, S.; STEINER, R.; MOORE, A. **A practical guide to SysML**. 1. ed. Waltham: Elsevier Science, 2011.
2. HEUSER, C. A. **Projeto de banco de dados**. 4. ed. Porto Alegre: Bookman, 2008.
3. JACOBSON, I; BOOCH, G.; RUNBAUGH, J; **UML: guia do usuário**. Rio de Janeiro: Campus, 2012.
4. PAULA FILHO, W. de P. **Engenharia de software: fundamentos, métodos e padrões**. 3. ed. Rio de Janeiro: LTC, 2009.
5. PERRY, S.; HOLT, J. **SysML for systems engineering**. 2ed. INSPEC/IEE, 2013
6. TONSIG, S. L. **Engenharia de software: análise e projeto de sistemas**. 2. ed. rev. e ampl São Paulo: Ciência Moderna, 2008.

6. **APROVAÇÃO**

Pedro Luiz Lima Bertarini
Coordenador(a) do Curso de Engenharia
Eletrônica e de Telecomunicações
Campus Patos de Minas

Mauricio Cunha Escarpinati
Diretor(a) da Faculdade de Computação



Documento assinado eletronicamente por **Mauricio Cunha Escarpinati, Diretor(a)**, em 25/01/2022, às 15:33, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Pedro Luiz Lima Bertarini, Coordenador(a)**, em 22/02/2022, às 15:34, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2955857** e o código CRC **663EF4EC**.