



## PLANO DE ENSINO

### 1. IDENTIFICAÇÃO

|                        |                                  |                |            |               |        |                 |             |
|------------------------|----------------------------------|----------------|------------|---------------|--------|-----------------|-------------|
| Componente Curricular: | ENGENHARIA DE SOFTWARE           |                |            |               |        |                 |             |
| Unidade Ofertante:     | FACULDADE DE ENGENHARIA ELÉTRICA |                |            |               |        |                 |             |
| Código:                | FACOM39028                       | Período/Série: | 6º PERÍODO | Turma:        | U      |                 |             |
| Carga Horária:         |                                  |                | Natureza:  |               |        |                 |             |
| Teórica:               | 30                               | Prática:       | 30         | Total:        | 60     | Obrigatório(a): | Optativo(X) |
| Professor(A):          | Eliana Pantaleão                 |                |            | Ano/Semestre: | 2023/2 |                 |             |
| Observações:           |                                  |                |            |               |        |                 |             |

### 2. EMENTA

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

### 3. JUSTIFICATIVA

Engenharia de Software é uma área da computação voltada à especificação, desenvolvimento, manutenção e criação de sistemas de software, com aplicação de tecnologias e práticas de gerência de projetos e outras disciplinas, visando organização, produtividade e qualidade. Os fundamentos científicos para a engenharia de software envolvem o uso de modelos abstratos e precisos que permitem ao engenheiro especificar, projetar, implementar e manter sistemas de software, avaliando e garantindo sua qualidade.

### 4. OBJETIVO

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.

## 5. 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

## 6. METODOLOGIA

- **Conteúdo Programático para Atividades Teóricas Presenciais**

As aulas teóricas serão realizadas às segundas-feiras das 7h10min às 8h50min no Laboratório de Informática Prata, Unipam, bloco G, sala 103.

| <b>Aula</b> | <b>Data</b> | <b>Conteúdo Teórico</b>   |
|-------------|-------------|---|
| 1-2         | 08/01       | Introdução à Engenharia de Software e Engenharia de Sistemas. Processos de desenvolvimento de software.     |
| 3-4         | 15/01       | Engenharia de Requisitos.   |
| 5-6         | 22/01       | Modelagem de Casos de Uso.  |
| 7-8         | 29/01       | Atividade avaliativa.   |
| 9-10        | 05/02       | Modelagem de Classes com UML (Nível Domínio).   |
| 11-12       | 19/02       | Projeto (Design) de Sistemas. Design Orientado a Objetos com UML. Projeto arquitetural – estilos e padrões. |
| 13-14       | 26/02       | Atividade avaliativa.   |
| 15-16       | 04/03       | Modelagem de Classes (Nível Especificação). Modelagem de Interações.  |
| 17-18       | 11/03       | Modelagem de Estados.   |
| 19-20       | 18/03       | Projeto de Interface com o Usuário.   |
| 21-22       | 25/03       | Qualidade de Software (Normas e Padrões) e Verificação e Validação de Software (V&V).                       |

|       |       |  |
|-------|-------|--|
| 23-24 | 01/04 | Atividade avaliativa.                                |
| 25-26 | 08/04 | Atividades relativas ao trabalho final.              |
| 27-28 | 15/04 | Apresentação do Trabalho Final.                      |
| 29-30 | 22/04 | Atividade avaliativa de recuperação de aprendizagem. |

- **Conteúdo Programático para Atividades Práticas Presenciais**

As aulas práticas serão realizadas às segundas-feiras das 8h50min às 10h40min no Laboratório de Informática Prata, Unipam, bloco G, sala 103.

| <b>Aula</b> | <b>Data</b> | <b>Conteúdo Prático</b>  |
|-------------|-------------|--|
| 1-2         | 08/01       | Pesquisa e relatório sobre Ferramentas CASE.   |
| 3-4         | 15/01       | Dinâmica para eliciação de requisitos.   |
| 5-6         | 22/01       | Pesquisa e apresentação de diagramas de casos de uso. Construção de um diagrama de casos de uso. |
| 7-8         | 29/01       | Atividade avaliativa.  |
| 9-10        | 05/02       | Pesquisa e apresentação de diagramas de classes. Construção de um diagrama de classes.           |
| 11-12       | 19/02       | Pesquisa e apresentação de diagramas de arquitetura. Construção de um diagrama de arquitetura.   |
| 13-14       | 26/02       | Pesquisa e apresentação de diagramas de interações. Construção de um diagrama de interações.     |
| 15-16       | 04/03       | Atividade avaliativa.  |
| 17-18       | 11/03       | Pesquisa e apresentação de diagramas de estados. Construção de um diagrama de estados.           |
| 19-20       | 18/03       | Construção de interface com o usuário.   |
| 21-22       | 25/03       | Atividade de avaliação de qualidade de software.   |
| 23-24       | 01/04       | Atividade avaliativa.  |

|       |       |  |
|-------|-------|--|
| 25-26 | 08/04 | Apresentação do Trabalho Final.                      |
| 27-28 | 15/04 | Atividade avaliativa.                                |
| 29-30 | 22/04 | Atividade avaliativa de recuperação de aprendizagem. |

- **Conteúdo Programático para Atividades Acadêmicas Extras (AAE)**

| <b>Aula</b> | <b>Data</b> | <b>Conteúdo - Atividades Acadêmicas Extras (AAE)</b>   |
|-------------|-------------|--|
| 1-2         | 12/01       | Teórica: Paradigmas da Engenharia de Software, Modelos de Desenvolvimento e Análise Orientada a Objetos (AOO) com UML. |
| 3-4         | 19/01       | Prática: levantamento de requisitos para o trabalho final.   |
| 5-6         | 02/02       | Teórica: SCRUM.  |
| 7-8         | 09/02       | Prática: construção do modelo de casos de uso para o trabalho final.   |
| 9-10        | 15/03       | Teórica: Devops.   |
| 11-12       | 22/03       | Prática: projeto de interfaces com o usuário para o trabalho final.  |

|  | <b>Teórica</b> | <b>Prática</b> |
|--|----------------|----------------|
| <b>C.H Presencial Total</b>                    | 30             | 30             |
| <b>C.H. Atividades Acadêmicas Extras Total</b> | 6              | 6              |
| <b>C.H. Total da disciplina</b>                | 36             | 36             |

- **Atendimento**

O atendimento aos alunos da disciplina será realizado de forma presencial no Unipam, Bloco Alfa, sala 8, de acordo com o seguinte planejamento: quartas-feiras entre 14h50min e 16h30min, ou outro dia desde que agendado com a professora previamente.

## 7. **AVALIAÇÃO**

- **Aproveitamento**

A avaliação de desempenho dos discentes será feita por apresentação e entrega de atividades, três provas e um trabalho final. O cronograma de atividades avaliativas e a distribuição da pontuação é apresentada a seguir.

Os resultados das avaliações serão divulgados no mural do curso, sendo que as notas serão apresentadas pelos números de matrícula dos alunos. A divulgação das notas deve acontecer em até 15 dias úteis após a sua realização e a vista de prova será marcada com os alunos, a partir da data de divulgação das notas, respeitando-se o prazo de no máximo 5 dias úteis, como previsto na Resolução do CONGRAD (Nº46/2022).

| <b>DATA</b> | <b>ATIVIDADE AVALIATIVA</b>                                  | <b>PONTUAÇÃO</b> |
|-------------|--|------------------|
| 29/01       | Prova 1  | 20 pontos        |
| 04/03       | Prova 2  | 20 pontos        |
| 01/04       | Prova 3  | 20 pontos        |
| -           | Apresentação e entrega de atividades no decorrer do período. | 20 pontos        |
| 15/04       | Trabalho Final   | 20 pontos        |
| TOTAL       |  | 100 pontos       |

- **Frequência**

A frequência para aulas presenciais será aferida por chamada oral durante as aulas. Já as horas vinculadas à parte assíncrona serão aferidas com a entrega das atividades.

- **Recuperação**

A prova de recuperação somente será aplicada àquele estudante que não obtiver o rendimento mínimo para aprovação e com frequência mínima de 75% no componente curricular, conforme Resolução do CONGRAD (Nº46/2022). Esta prova substituirá a menor nota obtida entre as provas 1, 2 ou 3, abrangendo todo o conteúdo da disciplina.

## 8. **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.

### **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.

## 9. APROVAÇÃO

Aprovado em reunião do Colegiado conforme Decisão Administrativa do Colegiado anexada ao processo referenciado.

Coordenação do Curso de Graduação: \_\_\_\_\_



Documento assinado eletronicamente por **Eliana Pantaleão, Professor(a) do Magistério Superior**, em 02/02/2024, às 09:23, 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 **Daniel Costa Ramos, Coordenador(a)**, em 15/02/2024, às 08:54, 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](https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **4958401** e o código CRC **B8D8C6BB**.