



## Plano de Ensino

**Universidade Federal do Espírito Santo**

**Campus de São Mateus**

**Curso:** Engenharia da Computação - São Mateus

**Departamento Responsável:** Departamento de Computação e Eletrônica

**Data de Aprovação (Art. nº 91):** 27/08/2025

**DOCENTE PRINCIPAL :** MARIA DAS GRACAS DA SILVA TEIXEIRA

Matrícula: 1653314

**Qualificação / link para o Currículo Lattes:** <http://lattes.cnpq.br/5727201281670356>

**Disciplina:** ENGENHARIA DE SOFTWARE

**Código:** DCE08156

**Período:** 2025 / 2

**Turma:** 33.1

**Pré-requisito:**

**Carga Horária Semestral:** 75

Disciplina: DCE08350 - PROGRAMAÇÃO III

### Distribuição da Carga Horária Semestral

<b>Créditos:</b> 5	<b>Teórica</b>	<b>Exercício</b>	<b>Laboratório</b>	<b>Extensão</b>
	75	0	0	

### Ementa:

Processo de software. Modelos de Processo de Software. Planejamento e Gerência de Projetos de Software. Garantia e Controle da Qualidade. Gerência de Configuração. Análise de Requisitos. Projeto de Sistema. Implementação e Testes. Entrega e Manutenção.

### Objetivos Específicos:

Caracterizar a Engenharia de Software e os principais conceitos relacionados à ela;  
Compreender o processo do desenvolvimento de software, suas fases, atividades, artefatos e a importância das mesmas para a produção de software de qualidade;  
Executar as principais atividades do processo de software.

### Conteúdo Programático:

1. Introdução - Software e Engenharia de Software.

Noções gerais

Ética em Engenharia de Software

2. O Processo de Software

Definições de processo de software

Atividades típicas do processo de software

Modelos de processo de software

3. Levantamento e Análise de Requisitos

Conceitos básicos

Métodos e técnicas de análise de requisitos

Documentação de requisitos

4. Design de Software

Conceitos básicos

Atividades relacionadas ao Design

Documentação de design

Design patterns

5. Implementação e Testes

Implementação

Conceitos básicos e atividades de controle de qualidade (testes)

6. Entrega e Manutenção

Implantação de sistemas

Manutenção e tipos de manutenção

## 7. Gerência de Projetos de Software

Atividades típicas de gerência

Alocação de recursos e gestão de pessoal

Elaboração de cronograma

Gerência de riscos

Plano de projeto

### Metodologia:

- Aulas teórico-expositivas - onde é apresentado e debatido o conteúdo programático previsto;
- Realização de exercícios e outras atividades práticas a fim de melhor compreender o conteúdo teórico;
- Uso de material complementar ao conteúdo programático;
- Atividade Avaliativa: Fichamento de publicação - composto de leitura e fichamento de publicação, bem como de apresentação e debate do tema;
- Trabalho prático - composto de material escrito, apresentação e exercícios complementares realizados ao longo do semestre.

A disciplina será ministrada utilizando o Google Classroom como plataforma de apoio a aprendizagem e ambiente de construção de conhecimento coletivo. Alunos e professora poderão utilizar fóruns, chats, conferências web para trocarem opiniões e dúvidas sobre os conteúdos ministrados. Serão disponibilizados na plataforma: recomendações de leitura, listas de exercício, atividades avaliativas, entre outros. Esses recursos servirão de base e de apoio para atingir o objetivo da disciplina e atendimento da ementa, cujo conteúdo é ministrado nas aulas teórico-expositivas e nas atividades complementares, para fixação de conteúdo.

Além dos recursos do Google Classroom, os alunos precisarão utilizar um editor de UML para realização de exercícios e do trabalho prático da disciplina, sendo recomendada a ferramenta Visual Paradigm (que tem uma versão Community free). Os alunos podem escolher outro editor de UML (desde que a licença seja free, ou a aquisição da licença seja comprovada, e a professora seja comunicada da troca).

Recursos como e-mail e avisos no Google Classroom servirão como forma de contato quando necessário, bem como para o estabelecimento de datas das atividades previstas.

### Critérios / Processo de avaliação da Aprendizagem :

- 1) Nota 1 (N1) - Fichamento e apresentação de publicação, seguido por debate. Pontuação: 0 a 4 pontos;
- 2) Nota 2 (N2) - A N2 dar-se-á a partir de um trabalho prático, para a resolução de um problema proposto. Pontuação: 0 a 6 pontos.

Cálculo da média do semestre: Média Semestral (MS) =  $N1 + N2$

Critério para aprovação no semestre, sem Avaliação Final:  
Se  $MS \geq 7.00$  Então aluno está APROVADO na disciplina  
Caso Contrário o aluno deverá realizar AVALIAÇÃO FINAL

- 3) Nota 3 (Avaliação Final - AF) - Aplicação de uma avaliação teórico-prática  
Conteúdo referente ao semestre. Pontuação: 0 a 10 pontos.

Cálculo da média final: Média Final (MF) =  $(MS + AF) / 2$

Critério para aprovação no semestre, com Avaliação Final:  
Se  $MF \geq 5.00$  Então aluno está APROVADO na disciplina  
Caso Contrário o aluno está REPROVADO na disciplina

A frequência regimental mínima será considerada.

### Bibliografia básica:

PRESSMAN, Roger S. Engenharia de software. 6. ed. São Paulo: McGraw-Hill, 2006. xxxi, 720 p. ISBN 9788586804571 (broch.)

SOMMERVILLE, Ian. Engenharia de software. 8. ed. São Paulo: Pearson, 2007. xiv, 552 p.

ISBN 9788588639287 (broch.)

PAULA FILHO, Wilson de Pádua. Engenharia de software: fundamentos, métodos e padrões. 3. ed. Rio de Janeiro: LTC, 2009. xiii, 1248 p. ISBN 9788521616504 (broch.)

### Bibliografia complementar:

COCKBURN, Alistair. Escrevendo casos de uso eficazes: [um guia prático para desenvolvedores de software]. Porto Alegre, RS: Bookman, 2005. viii, 254 p. ISBN 9788536304571 (broch.)

BRAUDE, Eric J. Projeto de software: da programação à arquitetura : uma abordagem baseada em Java. Porto Alegre: Bookman, 2005. xii, 619 p. ISBN 9788536304939 (broch.)

BEZERRA, Eduardo. Princípios de análise e projeto de sistemas com UML. 2. ed. totalmente rev. e atual. Rio de Janeiro: Elsevier: Campus, 2007. xvii, 369 p. ISBN 9788535216967 (broch.)

MISTRİK, I; BROWN, AW; ALI BABAR, M. Agile Software Architecture : Aligning Agile Processes and Software Architectures. Amsterdam : Morgan Kaufmann, 2014. ISBN: 9780124077720. (eBook)

LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007. xiv, 695 p. ISBN 9788560031528 (broch.).

**Cronograma:**

**Observação:**

Apresentação da disciplina (2h)

1. Introdução (7h)

2. Processo de Software (12h)

3. Levantamento e Análise de Requisitos (12h)

4. Design de Software (10h)

5. Implementação e Testes (2h)

6. Entrega e Manutenção (2h)

7. Gerência de Projetos de Software (10h)

Atividades Avaliativas (18 horas) -> Essa carga horário contempla tanto a apresentação das duas atividades avaliativas quanto o acompanhamento das atividades realizado ao longo do semestre, com o objetivo de auxiliar no aprendizado do aluno

Ao longo do semestre alterações na metodologia/cronograma poderão ocorrer. Desde que mantendo em foco o conteúdo programático. O objetivo é priorizar o aprendizado do aluno, respeitando as regras definidas, e as situações que ocorrerem ao longo do semestre.