



Plano de Ensino

Universidade Federal do Espírito Santo

Campus de São Mateus

Curso: Ciência da Computação - São Mateus

Departamento Responsável: Departamento de Computação e Eletrônica

Data de Aprovação (Art. nº 91): 19/07/2023

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: DCE16371

Período: 2023 / 2

Turma: 3704.1-PPC2023

Pré-requisito:

Carga Horária Semestral: 60

Disciplina: DCE16361 - PROGRAMAÇÃO ESTRUTURADA

Distribuição da Carga Horária Semestral

Créditos: 4	Teórica	Exercício	Laboratório
	60	0	0

Ementa:

Introdução a engenharia de software. Ética em computação. Modelos de processo de desenvolvimento de software. Gerência de projetos de software. Garantia e controle da qualidade de software. Atividades típicas de um processo de desenvolvimento de software: engenharia de requisitos. Design. Implementação. Testes. Entrega. Manutenção. Método Orientado à Objeto.

Objetivos Específicos:

Caracterizar software como produto e seu processo de desenvolvimento como uma disciplina de engenharia, parte integrante da engenharia de sistemas de informação. [Familiaridade] Identificar e caracterizar os principais modelos de processos de desenvolvimento de software. [Familiaridade / Uso] Compreender as atividades típicas de um processo de desenvolvimento de software. [Familiaridade] Compreender as principais atividades típicas do gerenciamento de um projeto de software. [Familiaridade] Compreender as atividades básicas de garantia e controle de qualidade de software, incluindo as principais normas e padrões de qualidade. [Familiaridade]

Conteúdo Programático:

1. Introdução - Software e Engenharia de Software.
2. O Processo de Software
 - O que é processo de software;
 - Atividades típicas do processo de software;
 - Modelos de processo de software;
 - Automatização do processo de software.
3. Levantamento e Análise de Requisitos
 - Requisitos de software e tipos de requisitos;
 - Atividades relacionadas à requisitos;
 - Métodos e técnicas de análise de requisitos;
 - Documentação de requisitos.
4. Design de Software
 - Conceitos básicos;
 - Modelos;
 - Design patterns.
5. Implementação e Testes
 - Implementação;
 - Conceitos básicos sobre controle de qualidade - Testes
 - Tipos de teste;

Estratégias de teste.

6. Entrega e Manutenção

Implantação de sistemas;
Manutenção e tipos de manutenção;
Problemas da manutenção.

7. Gerência de Projetos de Software

Atividades típicas de gerência;
Escopo;
Estimativas;
Alocação de recursos e gestão de pessoal;
Elaboração de cronograma;
Gerência de riscos;
Plano de projeto.

8. Garantia e Controle da Qualidade

Algumas técnicas e metodologias para garantia e controle de qualidade: Revisões; Documentação; Normas e modelos de qualidade de processo de software.

Metodologia:

- Aulas teórico-expositivas - onde é apresentado o conteúdo programático;
- Realização de exercícios;
- Leituras complementares ao conteúdo programático;
- Fichamento de publicação - composto de leitura e fichamento de publicação (a ser enviado para a professora), 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 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.

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 - Uma Abordagem Profissional. 8a ed. 2016. São Paulo: McGraw-Hill, 2. xxxi, 720 p.

SOMMERVILLE, Ian. Engenharia de software. 8a ed. São Paulo: Pearson, 2007. xiv, 552 p.
PAULA FILHO, Wilson de Pádua. Engenharia de software: fundamentos, métodos e padrões. 3a ed. Rio de Janeiro: LTC, 2009. xiii, 1248 p.

Bibliografia complementar:

COCKBURN, Alistair. Escrevendo casos de uso eficazes: [um guia prático para desenvolvedores de software]. Porto Alegre, RS: Bookman, 2005. viii, 254 p.
WAZLAWICK, Raul Sidnei. Análise e design orientados a objetos para sistemas de informação: modelagem com UML, OCL e IFML. 3a ed. Rio de Janeiro, RJ: Elsevier, 2015. 462 p.
BEZERRA, Eduardo. Princípios de análise e projeto de sistemas com UML. 2a ed. totalmente rev. e atual. Rio de Janeiro: Elsevier: Campus, 2007. xvii, 369 p.
MARTINS, José Carlos Cordeiro. Gerenciando projetos de desenvolvimento de software com PMI, RUP e UML. 4a ed. Rio de Janeiro, RJ: Brasport, 2007. 325 p.
LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento interativo. 3a ed. Porto Alegre: Bookman, 2007. xiv, 695 p.

Cronograma:

Observação:

Apresentação da disciplina (2h)

1. Introdução (6h)
 2. O Processo de Software (12h)
 3. Levantamento e Análise de Requisitos (11h)
 4. Design de Software (10h)
 5. Implementação e Testes (2h)
 6. Entrega e Manutenção (2h)
 7. Gerência de Projetos de Software (11h)
 8. Garantia e Controle da Qualidade (4h)
- Atividades Avaliativas (15 horas)

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.