



Universidade Federal de Santa Catarina
Campus Araranguá - ARA
Centro de Ciências, Tecnologias e Saúde
Departamento de Computação
Plano de Ensino

SEMESTRE 2021.2

I. IDENTIFICAÇÃO DA DISCIPLINA

CÓDIGO	NOME DA DISCIPLINA	Nº DE HORAS-AULA SEMANAIS - TEÓRICAS	Nº DE HORAS-AULA SEMANAIS - PRÁTICAS
DEC7130	ENGENHARIA DE SOFTWARE II	3	1
TOTAL DE HORAS-AULA SEMESTRAIS	HORÁRIO TURMAS TEÓRICAS	HORÁRIO TURMAS PRÁTICAS	MODALIDADE
72	09655 e 04652 - 4.1830-2	09655 e 04652 - 6.1830-2	Remota Assíncrona e Síncrona

II. PROFESSOR(ES) MINISTRANTE(ES)

Profa. Max Pereira
E-mail: max.pereira@ufsc.br

Horário de atendimento: Terça e Quinta das 17:00 às 18:00 por vídeo conferência
(<https://meet.google.com/yjz-qxfi-omc>)

III. PRÉ-REQUISITO(S)

CÓDIGO	NOME DA DISCIPLINA
DEC7138	Análise e Projeto de Software (ENC)
DEC7124	Engenharia de Software I (TIC)

IV. CURSO(S) PARA O(S) QUAL(IS) A DISCIPLINA É OFERECIDA

Engenharia de Computação (ENC)
Tecnologias da Informação e Comunicação (TIC)

V. JUSTIFICATIVA

O profissional responsável por desenvolver sistemas computacionais necessita conhecer e aplicar as principais metodologias adotadas pelo mercado de trabalho para desempenhar sua função com qualidade e ser competitivo no mercado.

VI. EMENTA

Evolução da prática de desenvolvimento de software; Critérios de qualidade de artefatos de software; modelos de ciclo de vida; metodologias de desenvolvimento de software; manutenção de software; engenharia reversa; modelagem formal de sistemas; abordagens voltadas ao reuso de software; teste de software; gerenciamento do processo de produção de software e técnicas de apoio ao gerenciamento do processo de produção de software; apoio automatizado ao desenvolvimento de software.

VII. OBJETIVOS

Objetivo Geral:

Fornecer subsídios ao aluno para que ele possa compreender os processos de desenvolvimento, implementação e manutenção de software.

Objetivos Específicos:

O aluno ao final do curso deve possuir habilidades para:

- Definir engenharia de software explicitando seus conceitos e objetivos;
- Conhecer e aplicar o conceito destinado aos processos de software;
- Conhecer os modelos de ciclo de vida;
- Entender o que é um software de qualidade e conhecer as métricas existentes.

VIII. CONTEÚDO PROGRAMÁTICO

Parte I: Processo de Software

- Visão geral de processo de software
- Modelos de ciclo de vida

Parte II: Evolução de Software

- Processo de manutenção de software
- Engenharia reversa e Reengenharia de software
- Gerenciamento de Sistemas Legados

Parte III: Qualidade de Software

- Verificação e Validação
- Teste de software
- Qualidade de produto e qualidade de processo

IX. COMPETÊNCIAS/HABILIDADES

- Indicar modelos de processos de desenvolvimento de software;
- Especificar processos de software;
- Conhecer aspectos de qualidade de software;
- Identificar diferentes tipos de atividades de manutenção de software, assim como, os principais fenômenos relacionados à evolução de software;
- Aplicar métodos de gerência de projeto de software.

X. METODOLOGIA DE ENSINO / DESENVOLVIMENTO DO PROGRAMA

- A disciplina será ministrada através de aulas expositivas síncronas e atividades assíncronas.
- O material de apoio será postado no Moodle.
- As atividades práticas serão desenvolvidas utilizando ferramentas computacionais.

Requisitos de infraestrutura necessários para ministrar as aulas:

- Acesso à Internet;
- Software livre para modelagem UML;
- Ambiente Virtual de Aprendizagem - Moodle.

XI. METODOLOGIA E INSTRUMENTOS DE AVALIAÇÃO

- A verificação do rendimento escolar compreenderá frequência e aproveitamento nos estudos, os quais deverão ser atingidos conjuntamente. Será obrigatória a frequência às atividades correspondentes a cada disciplina, no mínimo a 75% das mesmas (Frequência Suficiente - FS), ficando nela reprovado o aluno que não comparecer a mais de 25% das atividades (Frequência Insuficiente - FI).
- A nota mínima para aprovação na disciplina será $MF \geq 6,0$ (seis) e Frequência Suficiente (FS). (Art. 69 e 72 da Res. nº 17/CUn/1997).
- O aluno com Frequência Suficiente (FS) e média das notas de avaliações do semestre MF entre 3,0 e 5,5 terá direito a uma nova avaliação no final do semestre (REC), exceto as atividades constantes no art.70, § 2º. A Nota Final (NF) será calculada por meio da média aritmética entre a média das notas das avaliações parciais (MF) e a nota obtida na nova avaliação (REC). (Art. 70 e 71 da Res. nº 17/CUn/1997).
- Ao aluno que não comparecer efetuar às avaliações ou não apresentar trabalhos no prazo estabelecido será atribuída nota 0 (zero). (Art. 70, § 4º da Res. nº 17/CUn/1997)
- O registro de frequência será efetuado para aulas assíncronas e síncronas. No primeiro caso serão disponibilizadas atividades com tempo de execução de 48 horas em que, a partir da execução destas, os alunos terão a presença registrada. Para o segundo caso ao final das aulas será realizado o registro.

Avaliações:

- **Trabalhos práticos (TP):** Quatro (4) trabalhos em dupla realizados de forma assíncrona quanto ao seu desenvolvimento e síncrona no que se refere à apresentação do trabalho (quando for solicitado). Os requisitos do trabalho serão divulgados no decorrer da disciplina.
- **Prova (P):** Atividade individual realizada de forma síncrona no horário regular da disciplina.
- A **Média Final (MF)** será calculada da seguinte forma: $MF = P * 0,4 + TP * 0,6$
- A **avaliação de recuperação (REC)** seguirá a mesma regra da P.

Observações:

- **REC:** Não há avaliação de recuperação nas disciplinas de **caráter prático** que envolve atividades de laboratório (Res.17/CUn/97).
- **Nova avaliação:** O aluno, que por motivo de força maior e plenamente justificado, deixar de realizar atividades avaliativas previstas no plano de ensino, deverá formalizar pedido à Chefia do Departamento de Ensino, na Secretaria Integrada de Departamento - SID, ao qual a disciplina pertence, dentro do prazo de 3 (três) dias úteis, apresentando documentação comprobatória.

XII. CRONOGRAMA

SEMANA	DATAS	ASSUNTO
1	25/10/2021 a 30/10/2021	E1: Apresentação do plano de ensino. (síncrono) E2: Unidade I: Introdução a processos de software. (síncrono)
2	01/11/2021 a 06/11/2021	E1: Unidade I: Processos de software prescritivos.(síncrono) E2: Tarefa. (assíncrono)
3	08/11/2021 a 13/11/2021	E1: Unidade I: Processos ágeis. (síncrono). E2: Tarefa. (assíncrono)
4	15/11/2021 a 20/11/2021	E1: Unidade I: Processos ágeis. (síncrono) E2: Apresentação do Trabalho 1. (síncrono)
5	22/11/2021 a 27/11/2021	E1: Unidade II: Manutenção (síncrono). E2: Tarefa. (assíncrono)
6	29/11/2021 a 04/12/2021	E1: Unidade II: Manutenção (síncrono). E2: Tarefa. (assíncrono)
7	06/12/2021 a 11/12/2021	E1: Unidade I: Manutenção (síncrono) E2: Apresentação do Trabalho 2. (síncrono)
8	13/12/2021 a 18/12/2021	E1: Unidade III: Verificação e Validação - Inspeções. (síncrono) E2: Tarefa. (assíncrono)
9	31/01/2022 a 05/02/2022	E1: Unidade III: Verificação e Validação - Testes. (síncrono) E2: Tarefa. (assíncrono)
10	07/02/2022 a 12/02/2022	E1: Unidade III: Qualidade de Software (síncrono). E2: Tarefa. (assíncrono) Apresentação do Trabalho 3 (síncrono).
11	14/02/2022 a 19/02/2022	E1: Unidade III: Modelos de maturidade: CMMI (síncrono). E2: Tarefa. (assíncrono)
12	21/02/2022 a 26/02/2022	E1: Unidade II: Modelos de maturidade: CMMI (síncrono). E2: Tarefa. (assíncrono)

13	28/02/2022 a 05/03/2022	E1: Unidade II: Modelos de maturidade: MPS.BR (síncrono). E2: Tarefa. (assíncrono)
14	07/03/2022 a 12/03/2022	E1: Unidade II: Gerenciamento de projeto de software (síncrono). E2: Apresentação do Trabalho 4 (síncrono).
15	14/03/2022 a 19/03/2022	E1: Exercícios. (assíncrono) E2: Prova (P). (síncrono)
16	21/03/2022 a 26/03/2022	E1: Divulgação de notas. (assíncrono) E2: Prova de recuperação (síncrono).

Obs: O calendário está sujeito a pequenos ajustes de acordo com as necessidades das atividades

XIII. FERIADOS PREVISTOS PARA O SEMESTRE

02/11/2021	Finados
15/11/2021	Proclamação da República
28/02/2022	Carnaval - Ponto Facultativo
01/03/2022	Carnaval
02/03/2022	Quarta-feira de cinzas (Ponto Facultativo até 14 horas)

XIV. BIBLIOGRAFIA BÁSICA

BOURQUE, P.; FAIRLEY, R.E. *Guide to the Software Engineering Body of Knowledge, Version 3.0*, IEEE Computer Society, 2014. Disponível em: <https://www.computer.org/education/bodies-of-knowledge/software-engineering>

GOERICKE, Stephan. *The Future of Software Quality Assurance*. Berlin: Springer, 2020. Disponível em: <https://link.springer.com/book/10.1007/978-3-030-29509-7>

GRUHN, Volker; STRIEMER, Rüdiger. *The Essence of Software Engineering (2018)*. Disponível em: <https://link.springer.com/book/10.1007/978-3-319-73897-0>

HAZZAN, Orit; DUBINSKY, Yael. *Agile Software Engineering*. Berlin: Springer, 2008. Disponível em: <https://link.springer.com/book/10.1007/978-1-84800-198-5>

JALOTE, Pankaj. *Concise Introduction to Software Engineering (2008)*. Disponível em: <https://link.springer.com/book/10.1007/978-1-84800-302-6>

MENS, T.; DEMEYERS, S. *Software Evolution*. Berlin: Springer, 2008. Disponível em: <https://link.springer.com/book/10.1007/978-3-540-76440-3>

O'REGAN, Gerard. *A Practical Approach to Software Quality*. Berlin: Springer, 2002. Disponível em: <https://link.springer.com/book/10.1007/978-0-387-22454-1>

PRESSMAN, Roger S. e MAXIM, Bruce R. *Engenharia de software: uma abordagem profissional*. 8. ed. Porto Alegre: AMGH, 2016. Alguns capítulos disponíveis em: < <https://books.google.com.br/books?id=wexzCwAAQBAJ&lpg=PA15&dq=engenharia%20de%20software&hl=pt-BR&pg=PR3#v=onepage&q=engenharia%20de%20software&f=false> >

WASLAWICK, Raul Sidnei. *Engenharia de software: conceitos e práticas*. 2ª edição. Rio de Janeiro: Elsevier, 2019. . Alguns capítulos disponíveis em: < <https://books.google.com.br/books?id=d1qnDwAAQBAJ&lpg=PP1&dq=engenharia%20de%20software&hl=pt-BR&pg=PT9#v=onepage&q=engenharia%20de%20software&f=false> >

XV. BIBLIOGRAFIA COMPLEMENTAR

BECK, K. Programação extrema (xp) explicada: acolha as mudanças. Porto Alegre: Bookman, 2004. 182p.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML - Guia do Usuário. 2. ed. Rio de Janeiro: Campus, 2006.

GAMMA, E. ET AL. Padrões de projeto: soluções reutilizáveis de software orientado a objetos. Porto Alegre: Bookman, 2000. 364p

JACOBSON, I; BOOCH, G.; RUMBAUGH, J. The unified software development process. Boston: Addison-Wesley, 1999.

LARMAN, C. Utilizando UML e padrões: uma introdução à análise e aos projetos orientados a objetos e ao desenvolvimento interativo. 3. ed. Porto Alegre: Bookman, 2007.

MENDES, E.; MOSLEY, N. Web Engineering. New York: Springer, 2007.

PAULA FILHO, W. Engenharia de software: fundamentos, métodos e padrões. 3.ed. Rio de Janeiro: LTC, c2009.

PRESSMAN, Roger S. Engenharia de software: uma abordagem profissional.7. ed. Porto Alegre: AMGH, 2011. 780 p.

REUSSNER, Ralf. Managed Software Evolution. Springer Open, 2019. Disponível em: <https://link.springer.com/book/10.1007/978-3-030-13499-0>

SOMMERVILLE, I. Engenharia de software. 8. ed. São Paulo: Pearson Addison-Wesley, 2007. xiv, 552 p.

Professor(a):

Aprovado pelo Colegiado do Curso em __/__/__ Presidente do Colegiado: