



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

### SEMESTRE 2022. I

#### 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	Presencial

#### II. PROFESSOR(ES) MINISTRANTE(ES)

Profa. Andréa Sabedra Bordin

E-mail: [andrea.bordin@ufsc.br](mailto:andrea.bordin@ufsc.br)

Horário de atendimento: quarta-feira das 16hs às 18hs por videoconferência.

Sala: 316

#### 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 e atividades realizadas em sala de aula.
- 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 as avaliações ou não apresentar trabalhos no prazo estabelecido será atribuída nota 0 (zero). (Art. 70, § 4º da Res. nº 17/CUn/1997)

**Avaliações:**

- **Trabalhos práticos (TP):** Três (3) 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 de P.

**Observações:**

- **REC:** Não há avaliação de recuperação nas disciplinas de caráter prático que envolvem 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.
- A complementação da carga horária da disciplina ocorrerá da seguinte forma: (i) a Semana de Integração Acadêmica será contabilizada como dias letivos, conforme calendário acadêmico de 2022; e (ii) serão solicitados trabalhos de carácter prático-teórico para complementação de carga horária da disciplina.

## XII. CRONOGRAMA

SEMANA	DATAS	ASSUNTO
1	18/04/2022 a 22/04/2022	E1 - Apresentação do plano de ensino. E2 - Unidade I: Introdução a processos de software.
2	25/04/2022 a 29/04/2022	E1 - Unidade I: Processos de software prescritivos. E2 - Unidade I: Exercícios
3	02/05/2022 a 06/05/2022	E1: Unidade I: Processos de software ágeis: XP E2: Unidade I: Exercícios
4	09/05/2022 a 13/05/2022	E1: Unidade I: Processos de software ágeis: Scrum. E2: Unidade I: Processos de software ágeis: Scrum.
5	16/05/2022 a 20/05/2022	E1: Unidade I: Processos de software ágeis: Scrum. E1: Unidade I: Exercícios
6	23/05/2022 a 27/05/2022	<b>E1: Unidade I: Apresentação do Trabalho 1.</b> <b>E2: Unidade I: Apresentação do Trabalho 1.</b>
7	30/05/2022 a 03/06/2022	E1: Unidade II: Manutenção de Software. E2: Unidade II: Manutenção de Software.
8	06/06/2022 a 10/06/2022	E1: Unidade II: Manutenção de Software - Sistemas Legados. E2: Unidade III: Manutenção de Software - Reengenharia de Software.
9	13/06/2022 a 17/06/2022	E1: Unidade III: Verificação e Validação e Inspeção de Software. E2: Unidade III: Verificação e Validação - Testes.
10	20/06/2022 a 24/06/2022	<b>E1: Apresentação do Trabalho 2.</b> <b>E2: Apresentação do Trabalho 2.</b>
11	27/06/2022 a 01/07/2022	E1: Unidade III: Qualidade de Software. E2: Unidade III: Modelos de maturidade: CMMI.

12	04/07/2022 a 08/07/2022	E1: Unidade III: Modelos de maturidade: MPS.BR. E2: Unidade III: Exercícios.
13	11/07/2022 a 15/07/2022	<b>E1: Apresentação do Trabalho 3 - Tópicos Especiais.</b> <b>E1: Apresentação do Trabalho 3 - Tópicos Especiais</b>
14	18/07/2022 a 22/07/2022	E1: Exercícios. E2: Exercícios
15	25/07/2022 a 29/07/2022	E1: Exercícios. <b>E2: Prova (P).</b>
16	01/08/2022 a 03/08/2022	E1: Divulgação de notas <b>E2: Prova de recuperação.</b>

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

### **XIII. FERIADOS PREVISTOS PARA O SEMESTRE**

### **XIV. BIBLIOGRAFIA BÁSICA**

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

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

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

### **XV. BIBLIOGRAFIA COMPLEMENTAR**

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

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 ao 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, 2009.

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

Professor(a):

Aprovado pelo Colegiado do Curso em \_\_/\_\_/\_\_ Presidente do Colegiado:

