



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE - CTS  
DEPARTAMENTO DE COMPUTAÇÃO – DEC

PLANO DE ENSINO

SEMESTRE 2018.2

I. IDENTIFICAÇÃO DA DISCIPLINA:

CÓDIGO	NOME DA DISCIPLINA	N.º DE HORAS-AULA SEMANAIS		TOTAL DE HORAS-AULA SEMESTRAIS
		TEÓRICAS	PRÁTICAS	
DEC7130	ENGENHARIA DE SOFTWARE II	3	1	72

HORÁRIO		MÓDULO
TURMAS TEÓRICAS	TURMAS PRÁTICAS	Presencial
09655 – 2-1420-2 e 4-1420-2		
04652 – 4-1830-2 e 6-2020-2		

II. PROFESSOR(ES) MINISTRANTE(S)	HORÁRIO DE ATENDIMENTO AO ESTUDANTE
Tatiana Nilson dos Santos Email: <a href="mailto:tatiana.santos@ufsc.br">tatiana.santos@ufsc.br</a>	Segunda-feira: 18:00-19:00 – Sala A307 Sexta-feira: 17:00-18:00 – Sala A307

III. PRÉ-REQUISITO(S) SUGERIDO ENC

CÓDIGO	NOME DA DISCIPLINA
DEC7138	Análise e Projeto de Software

III. PRÉ-REQUISITO(S) SUGERIDO TIC

CÓDIGO	NOME DA DISCIPLINA
DEC7124	Engenharia de Software I

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, gerenciamento 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: Fundamentos em Engenharia de Software

- Visão geral da engenharia de *software*
- Desenvolvimento de processo de *software*
- Verificação e Validação de *software*

### Parte II: Avaliação de Software

- Manutenção de software e suas variações
- Evolução de software e sistemas legados
- Qualidade e certificações

### Parte III: Gerenciamento

- Gerenciamento de projeto de *software*
- Reuso de Software e suas variações

## IX. METODOLOGIA DE ENSINO / DESENVOLVIMENTO DO PROGRAMA

Exposição dialogada, utilizando projetores de slides, trabalhos dirigidos com levantamento bibliográfico e atualização de assuntos, bem como todos os equipamentos necessários para o desenvolvimento das aulas laboratoriais de acordo com cada assunto ministrado; Material de apoio postado no Moodle. Desenvolvimento de trabalhos e exercícios semanais

## X. 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, ficando nela reprovado o aluno que não comparecer, no mínimo a 75% das mesmas.
- A nota mínima para aprovação na disciplina será 6,0 (seis). (Art. 69 e 72 da Res. nº 17/CUn/1997).
- A média final (MF) será composta pelo cálculo aritmético das seguintes notas:

$$MF = \frac{P1 + P2 + P3 + TF}{4}$$

- Onde:  
Primeira (P1), segunda (P2) e terceira (P3) prova, respectivamente = peso 10,0  
Trabalho Final (TF) = peso 10,0
- **Obs: Atividades extras, que por ventura forem realizadas em sala, poderão ser contabilizadas nas notas das provas (modificando assim seu peso)**
- Critério para aprovação: Média Final (MF)  $\geq$  6 e frequência suficiente (FS).
- O aluno com frequência suficiente (FS) e média das notas de avaliações do semestre 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 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 às avaliações ou não apresentar trabalhos no prazo estabelecido será atribuída nota 0 (zero). (Art. 70, § 4º da Res. nº 17/CUn/1997)

\*As provas poderão conter questões objetivas, objetivas mistas e dissertativas.

#### Observações:

#### Avaliação de recuperação

- Não há avaliação de recuperação nas disciplinas de **caráter prático** que envolve atividades de laboratório (Res.17/CUn/97).

#### Nova avaliação

- Para pedido de segunda avaliação somente em casos em que o aluno, por motivo de força maior e plenamente justificado, deixar de realizar avaliações previstas no plano de ensino, deverá formalizar pedido de avaliação à Secretaria Integrada de Departamento dentro do prazo de **3 dias úteis** apresentando comprovação.

### XI. CRONOGRAMA TEÓRICO / PRÁTICO\*

AULA (semana)	DATA	ASSUNTO
1	11/03 a 16/03	Plano de Ensino
2		Introdução à Engenharia de Software: Visão Geral
3		Unidade I: Processos de software clássicos
4	18/03 a 23/03	Unidade I: Processos de software clássicos
5		Unidade I: Processos Ágeis
6	25/03 a 30/03	Unidade I: Processos Ágeis
7		Unidade I: Verificação e Validação
8	01/04 a 06/04	Unidade I: Verificação e Validação
9		<b>PROVA TEÓRICA UNIDADE I</b>
10	08/04 a 13/04	Unidade II: Manutenção
11		Unidade II: Manutenção
12	15/04 a 20/04	Unidade II: Tipos de Manutenção
13		Unidade II: Reengenharia e Engenharia Reversa
14	22/04 a 27/04	Unidade II: Reengenharia e Engenharia Reversa
15		Unidade II: Evolução de software/sistema legados
16	29/04 a 04/05	Unidade II: Qualidade de Software
17		Unidade II: Modelos de maturidade: CMMI
18	06/05 a 11/05	Unidade II: Modelos de maturidade: CMMI
19		Unidade II: Modelos de maturidade: MPS.BR
20	13/05 a 18/05	Unidade II: Modelos de maturidade: MPS.BR
21		<b>PROVA TEÓRICA UNIDADE II</b>
22	20/05 a 25/05	Unidade III: Abordagens voltadas ao reuso de software
23		Unidade III: Abordagens voltadas ao reuso de software
24	27/05 a 01/06	Unidade III: Abordagens voltadas ao reuso de software
25		Unidade III: Gerenciamento de Software: projetos e requisitos
26	03/06 a 08/06	Unidade III: Gerenciamento de Software: projetos e requisitos
27		Unidade III: Gerenciamento de Software: projetos e requisitos
28	10/06 a 15/06	<b>PROVA TEÓRICA UNIDADE III</b>
29		<b>APRESENTAÇÃO DO TRABALHO FINAL</b>
30	17/06 a 22/06	<b>APRESENTAÇÃO DO TRABALHO FINAL</b>
31		<b>APRESENTAÇÃO DO TRABALHO FINAL</b>
32	24/06 a 29/06	<b>APRESENTAÇÃO DO TRABALHO FINAL</b>
33		Segunda Avaliação (P1, P2 ou P3)
34	01/07 a 06/07	Prova de Recuperação (REC)
35		Divulgação de Notas

\*O calendário está sujeito a pequenos ajustes de acordo com as necessidades das atividades a serem desenvolvidas

XII. FERIADOS PREVISTOS PARA O SEMESTRE 2019.1	
DATA	DESCRIÇÃO
03/04	Aniversário de Araranguá
19/04	Paixão de Cristo
20/04	Dia não letivo
21/04	Tiradentes/Páscoa
01/05	Dia de Trabalho
04/05	Dia da Padroeira de Araranguá
20/06	Corpus Christi
21/06	Dia não letivo
22/06	Dia não letivo

XIII. BIBLIOGRAFIA BÁSICA
SOMMERVILLE, Ian. <b>Engenharia de Software</b> . 9 ed. São Paulo: Pearson Prentice Hall, 2011.
BECK, K. <b>Programação extrema (xp) explicada</b> : acolha as mudanças. Porto Alegre: Bookman, 2004. 182p.
GAMMA, E. ET AL. <b>Padrões de projeto</b> : soluções reutilizáveis de software orientado a objetos. Porto Alegre: Bookman, 2000. 364
XIV. BIBLIOGRAFIA COMPLEMENTAR
MENDES, E.; MOSLEY, N. <b>Web Engineering</b> . New York: Springer, 2007.
LARMAN, C. <b>Utilizando UML e padrões</b> : uma introdução à análise e ao projetos orientados a objetos e ao desenvolvimento interativo. 3. ed. Porto Alegre: Bookman, 2007.
BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. <b>UML - Guia do Usuário</b> . 2. ed. Rio de Janeiro: Campus, 2006.
JACOBSON, I; BOOCH, G.; RUMBAUGH, J. <b>The unified software development process</b> . Boston: Addison-Wesley, 1999.
PRESSMAN, Roger S. <b>Engenharia de software: uma abordagem profissional</b> . 7 ed. Porto Alegre: AMGH, 2011. 780 p.
PAULA FILHO, W. <b>Engenharia de software: fundamentos, métodos e padrões</b> . 3 ed. Rio de Janeiro: LTC, c2009.

Os livros acima citados constam na Biblioteca Universitária e Setorial de Araranguá.



Tatiana Nilson dos Santos  
SIAPE 3017565

Professor responsável

Aprovado na Reunião do Colegiado do Curso 27/03/19



Coordenador

Prof. Fábio Guilherme  
Coordenador do Curso de  
Eng. de Computação - UFSC  
Portaria 2703/2018/GR