



UNIVERSIDADE FEDERAL
DE SANTA CATARINA

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

PLANO DE
ENSINO

SEMESTRE 2020.2

I. IDENTIFICAÇÃO DA DISCIPLINA:

| CÓDIGO | NOME DA DISCIPLINA | Nº 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 | Remota Assíncrona e Síncrona |
| 09655 e 04652 – 4-1830-2 | 09655 e 04652 – 6-1830-2 | |

II. PROFESSOR(ES) MINISTRANTE(S)

Profa. Andréa Sabedra Bordin
E-mail: andreabord@gmail.com
Horário de atendimento: Segunda-feira das 16:00 às 18:00 por vídeo conferência.
Sala: meet.google.com/tjf-awha-czd

III. PRÉ-REQUISITO(S) SUGERIDO ENC

| CÓDIGO | NOME DA DISCIPLINA |
|---------|-------------------------------|
| DEC7138 | Análise e Projeto de Software |

III. PRÉ-REQUISITO(S) 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 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: Gerenciamento

- Gerenciamento da qualidade
- Gerenciamento de projeto de *software*
- Aprimoramento dos processos

IX. 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.

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, 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. Na eventual impossibilidade do aluno estar presente será aplicada a regra da aula assíncrona.

Avaliações:

- Trabalhos práticos (TP): Quatro 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:**Avaliação de recuperação**

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.

XI. CRONOGRAMA TEÓRICO/PRÁTICO

| AULA (semanas) | DATA | ASSUNTO |
|----------------|-------------------------|--|
| 1 | 01/02/2021 a 06/02/2021 | Apresentação da disciplina. Unidade I: Introdução a processos de software (aula síncrona) |
| 2 | 08/02/2021 a 13/02/2021 | Unidade I: Processos de software prescritivos (aula síncrona) |
| 3 | 15/02/2021 a 20/02/2021 | Unidade I: Processos ágeis (aula síncrona). |
| 4 | 22/02/2021 a 27/02/2021 | Unidade I: Processos ágeis (aula síncrona). |
| 5 | 01/03/2021 a 06/03/2021 | Unidade I: Verificação e Validação (aula síncrona). |
| 6 | 08/03/2021 a 13/03/2021 | Unidade I: Manutenção (aula síncrona). |
| 7 | 15/03/2021 a 20/03/2021 | Unidade I: Manutenção (aula síncrona). |
| 8 | 22/03/2021 a 27/03/2021 | Unidade II: Qualidade (aula síncrona). |
| 9 | 29/03/2021 a 03/04/2021 | Unidade II: Qualidade (aula síncrona). |
| 10 | 05/04/2021 a 10/04/2021 | Unidade II: Modelos de maturidade: CMMI (aula síncrona). |
| 11 | 12/04/2021 a 17/04/2021 | Unidade II: Modelos de maturidade: CMMI (aula síncrona). |
| 12 | 19/04/2021 a 24/04/2021 | Unidade II: Modelos de maturidade: MPS.BR (aula síncrona). |
| 13 | 26/04/2021 a 01/05/2021 | Unidade II: Modelos de maturidade: MPS.BR (aula síncrona). |
| 14 | 03/05/2021 a 08/05/2021 | Unidade II: Gerenciamento de projeto de software (aula síncrona). |
| 15 | 10/05/2021 a 15/05/2021 | Prova (P) (aula síncrona). |
| 16 | 17/05/2021 a 22/05/2021 | Divulgação de notas. Prova de recuperação (aula síncrona). |

XII. Feriados previstos para o semestre 2020.2:

| DATA | |
|------------|----------------------------|
| 15/02/2021 | Ponto facultativo Carnaval |
| 16/02/2021 | Carnaval |
| 02/04/2021 | Sexta-feira Santa |
| 03/04/2021 | Aniversário de Araranguá |
| 21/04/2021 | Tiradentes |
| 01/05/2021 | Dia do Trabalho |

| | |
|------------|-------------------------------|
| 04/05/2021 | Dia da Padroeira de Araranguá |
| 03/06/2021 | Corpus Christi |

XIII. BIBLIOGRAFIA BÁSICA

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

RUBIN, Kenneth S. Scrum Essencial [recurso eletrônico]: Um guia prático para o mais popular processo ágil. Traduzido por Roberto Rezende. Rio de Janeiro: Alta Books Editora, 2017. Disponível em: < <https://books.google.com.br/books?id=pAtzDwAAQBAJ&lpg=PT36&dq=scrum&hl=pt-BR&pg=PT9#v=onepage&q=scrum&f=false> >

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> >

XIV. BIBLIOGRAFIA COMPLEMENTAR

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.

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

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

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

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.

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.

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

GOERICKE, S. The Future of Software Quality Assurance. Springer Open, 2019. Disponível em:
<https://link.springer.com/book/10.1007/978-3-030-29509-7>

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

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

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

Andriá S. Bordin

Professor(a) responsável

Aprovado na Reunião do Colegiado do Curso ____ / ____ / ____

Coordenador do Curso