



UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE COMPUTAÇÃO
PLANO DE ENSINO

SEMESTRE 2016.2

I. IDENTIFICAÇÃO DA DISCIPLINA:

CÓDIGO	NOME DA DISCIPLINA	Nº DE HORAS-AULA SEMANAIS		TOTAL DE HORAS-AULA SEMESTRAIS
		TEÓRICAS	PRÁTICAS	
ARA 7130	ENGENHARIA DE SOFTWARE II	3	1	72

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

II. PROFESSOR(ES) MINISTRANTE(S)

Luciana Bolan Frigo

III. PRÉ-REQUISITO(S) SOMENTE PARA CURSO TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO

CÓDIGO	NOME DA DISCIPLINA
ARA 7124	Engenharia de Software I

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

Graduação em Tecnologias da Informação e Comunicação (TIC)
Engenharia de Computação

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

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).
- 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).

$$NF = \frac{MF + REC}{2}$$

- 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)

- **Avaliações**

Média das avaliações: peso 7,0

Média das atividades: peso 3,0

* 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 Direção do Campus Araranguá dentro do prazo de 3 dias úteis apresentando comprovação. ([Ver formulário](#))

XI. CRONOGRAMA TEÓRICO/PRÁTICO		
AULA (semana)	DATA	ASSUNTO
1	19/09/16 a 23/09/16	Unidade I: Plano de Ensino e Introdução à Engenharia de Software: Visão Geral
2	Atividade extra	Unidade I: Processos de software clássicos
3	26/09/16 a 30/09/16	Unidade I: Processos Ágeis
4	03/10/16 a 07/10/16	Unidade I: Processos Ágeis
5	10/10/16 a 14/10/16	Unidade I: Verificação e Validação
6	17/10/16 a 21/10/16	Unidade I: Manutenção
7	24/10/16 a 28/10/16	Unidade II: Qualidade
8	Atividade extra	Unidade II: Abordagens voltada ao reuso de software
9	31/10/16 a 04/11/16	Unidade II: Gerenciamento de Software: projetos e requisitos
10	Sábado 05/11	Prova Teórica Unidade I
11	07/11/16 a 11/11/16	Unidade II: Modelos de maturidade: CMMI
12	14/11/16 a 18/11/16	Unidade II: Modelos de maturidade: CMMI
13	21/11/16 a 25/11/16	Unidade II: Modelos de maturidade: MPS.BR
14	28/11/16 a 02/12/16	Unidade II: Modelos de maturidade: MPS.BR
15	Atividade extra	Unidade II: Reengenharia, evolução de sistemas legados, Apoio automatizado ao gerenciamento de <i>software</i>
16	05/12/16 a 09/12/16	Prova Teórica Unidade II e Segunda avaliação (Prova substitutiva)
17	Sábado 10/12	Nova Avaliação (Prova de recuperação)
18	15/12	Divulgação de Notas

XII. Feriados previstos para o semestre 2016.2:

DATA	
11/08/2016	Feriado Estadual
12/08/2016	Dia não letivo
13/08/2016	Dia não letivo
16 a 18/08/2016	II Semana Acadêmica de Computação
07/09/2016	Independência do Brasil
12/10/2016	Nossa Senhora Aparecida
13 a 15/10/2016	I Simpósio Latino-Americano de Jogos (SLAT Jogos)
28/10/2016	Dia do Servidor Público (Lei 8112 art.236)
29/10/2016	Dia não letivo
02/11/2016	Finados
14/11/2016	Dia não letivo
15/11/2016	Proclamação da República

XIII. BIBLIOGRAFIA BÁSICA

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

XIV. BIBLIOGRAFIA COMPLEMENTAR

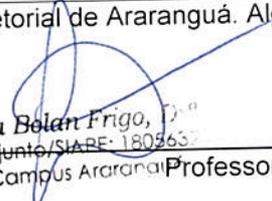
- PRESSMAN, Roger S. **Engenharia de software: uma abordagem profissional**. 7. ed. Porto Alegre: AMGH, 2011. 780 p.
- PAULA FILHO, W. **Engenharia de software: fundamentos, métodos e padrões**. 3.ed. Rio de Janeiro: LTC, c2009.
- MENDES, E.; MOSLEY, N. **Web Engineering**. New York: Springer, 2007.

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

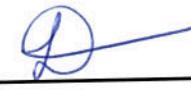
BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML - Guia do Usuário**. 2. ed. Rio de Janeiro: Campus, 2006.

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

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

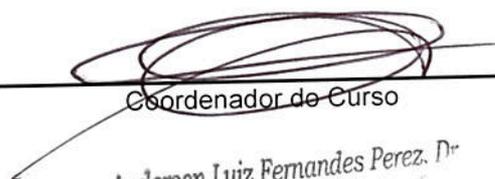

Luciana Belan Frigo, D^o
Prof. Adjunto/SIAPE: 1805632
UFSC/Campus Araranguá Professor

Aprovado na Reunião do Departamento 07/10/16



Chefe de Departamento

Aprovado na Reunião do Colegiado do Curso 11/10/2016



Coordenador de Curso

Anderson Luiz Fernandes Perez, D^o
Prof. Adjunto/SIAPE: 1135450
UFSC/Campus Araranguá