



UNIVERSIDADE PRESBITERIANA MACKENZIE



| | | |
|---|---|---|
| Unidade Universitária: FACULDADE DE COMPUTAÇÃO E INFORMÁTICA | | |
| Curso: Ciência da Computação | | |
| Disciplina: ENGENHARIA DE SOFTWARE | | Código da Disciplina: ENEC50297 |
| Professor(es): ANA GRASIELLE DIONISIO CORREA RENATA MENDES DE ARAÚJO | DRT: 1139178 | Etapa: 05 |
| Carga horária: 04 H/A – 04 SALA DE AULA | | Semestre Letivo: 1ºSEM/2020 |
| <p>Ementa:</p> <p>Estudo dos processos de desenvolvimento de software. Caracterização das atividades de levantamento de requisitos, análise, projeto, implementação, teste, implantação e manutenção de software. Análise comparativa de processos de desenvolvimento de software.</p> | | |
| Objetivos: Ao final da disciplina, o aluno deverá ser capaz de: | | |
| <p>Fatos e Conceitos</p> <ul style="list-style-type: none"> - Conhecer os objetivos, conceitos básicos e desafios da Engenharia de Software. - Conhecer os principais processos, fases, métodos e ferramentas do processo de desenvolvimento de software. - Conhecer os principais modelos de processo de desenvolvimento de software, suas vantagens e desafios. - Conhecer os desafios históricos e atuais da engenharia de software | <p>Procedimentos e Habilidades</p> <ul style="list-style-type: none"> - Participar do desenvolvimento de projetos de software utilizando os processos, etapas, métodos e ferramentas definidos para o processo pela organização. - Analisar um contexto de produção de software avaliando alternativas de processos, atividades, métodos e ferramentas, sugerindo ou adaptando processos para sua condução equilibrando custo, produtividade e qualidade de produção. - Criticar a relevância e adequação de práticas de engenharia de software em diferentes contextos de produção, reconhecendo as relações de custo e benefício das práticas de desenvolvimento de software. | <p>Atitudes, Normas e Valores</p> <ul style="list-style-type: none"> - Identificar o conhecimento da Engenharia de Software como componente importante na sua formação profissional. - Promover o uso e construção de soluções tecnológicas para o benefício da sociedade e sua qualidade de vida, destacando como a computação pode ser um meio eficaz e eficiente para trazer mudanças. - Comunicar-se de forma apropriada utilizando as linguagens e técnicas de exposição e registro de conhecimento, ideias, propostas e resultados, contribuindo para a plena realização do exercício profissional. - Respeitar compromissos assumidos em atividades profissionais colaborando para atingir metas definidas pela equipe em projetos. - Cooperar nas equipes em que trabalha, esforçando-se para atingir alta qualidade em processos e produtos, cumprindo suas atribuições com qualidade e pontualidade, sendo resiliente a alterações de |

| | | |
|--|--|--|
| | | projeto, de planejamento e de realização de tarefas. |
| <p><i>Conteúdo Programático:</i></p> <ol style="list-style-type: none">1. Engenharia de Software:<ol style="list-style-type: none">a. Histórico. Essência. Conceito de Software. Conceito de processo de software.2. Processo de Software<ol style="list-style-type: none">a. Estrutura.b. Modelos de processos de softwarec. Aspectos humanos da engenharia de software3. Engenharia de Requisitos<ol style="list-style-type: none">a. Modelagem de requisitos. Casos de Uso. Cenários. Histórias de usuários.b. Modelagem Formal.4. Projeto de Software.<ol style="list-style-type: none">a. Modelo de Projeto. Arquitetura, Interface. Componentes. Padrões.c. Segurança5. Implementação e Testes6. Gerência de Configuração7. Gerência de Projetos de Software<ol style="list-style-type: none">a. Conceitos.b. Estimativas.c. Métricas.c. Riscos.8. Manutenção de Software<ol style="list-style-type: none">a. Reengenhariab. Engenharia Reversa.c. Reestruturação9. Gestão da Qualidade<ol style="list-style-type: none">a. Conceito de qualidadeb. Técnicas de revisãoc. Garantia de Qualidade de Softwared. Verificação e Validaçãoe. Métricasf. Modelos de qualidade de processo10. Tendências | | |
| <p><i>Metodologia:</i></p> <p>Aulas expositivas e práticas, com base em metodologias baseadas em resolução de problemas e debates. Utilização do ambiente Mackenzie Virtual.</p> | | |

Critério de Avaliação:**--- N1 ---**

A1 – Projeto (em grupos de até 3 alunos) - 70%

PP1 – Prova Parcial 1 - prova individual – 30%

Sendo que: $NI1 = 0,7 * A1 + 0,3 PP1$

--- N2 ---

A2 - Projeto (em grupos de até 3 alunos) - 70%

PP2 – Prova Parcial 2 - prova individual – 30%

Sendo que: $NI2 = 0,7 * A2 + 0,3 * PP2$

-- NP --

Participação – até 1,0 ponto.

--- Média intermediária (MI) ---

$MI = (NI1 + NI2)/2 + NP$

A PROVA SUBSTITUTIVA deve substituir a menor Nota Intermediária (NI1 ou NI2). A prova substitutiva da disciplina será realizada de acordo com o calendário estabelecido pela FCI. A PROVA SUBSTITUTIVA contém todo o conteúdo do semestre.

CRITÉRIOS DE APROVAÇÃO

Se $MI \geq 7.5$ e $FREQUENCIA \geq 75\%$, APROVADO.

Se $MI \geq 8.5$ e $65\% \leq FREQUENCIA < 75\%$, APROVADO.

Se $FREQUENCIA \geq 75\%$ e $(MI+PROVA FINAL)/2 \geq 6.0$, APROVADO

Bibliografia Básica:

PRESSMAN, R.S.; MAXIM, B.R.; **Engenharia de Software, uma abordagem profissional**. 8ª ed. Porto Alegre. AMGH, 2016.

SOMMERVILLE, I. **Engenharia de Software**. 9ª ed. São Paulo. Pearson/Addison Wesley, 2011.

WAZLAWICK, R. **Engenharia de Software: conceitos e práticas**. 2ª edição. Rio de Janeiro: Elsevier. Campus 2019.

Bibliografia Complementar:

BEZERRA, E. **Princípios de análise e projeto de sistemas com UML**. 2ª ed. Rio de Janeiro: Elsevier; Campus. 2007.

COCKBURN, Alistair. **Escrevendo Casos de Uso Eficazes**. Porto Alegre. Bookman. 2005.

HORSTMAN, C.S. **Conceitos de Computação com Java**. 5. Ed. New York: Wiley, 2009.

PFLEEGER, S.L. **Engenharia de Software: teoria e prática**. 2a ed. São Paulo: Pearson Prentice Hall, 2012.

SCHACH, S.R. **Engenharia de Software: os paradigmas clássico e orientado a objetos**. São Paulo: McGraw-Hill, 2008.

STEPHENS, R. **Beginning Software Engineering**. 1a edição. Wrox. 2015.

DUARTE, L. **Scrum e Métodos Ágeis: Um guia prático**. 5a ed. LuizTools, 2016.