

**PLANO DE ENSINO - 2019 / 2º SEMESTRE**

<b>Curso:</b> Sistemas de Informação (Bacharelado)		<b>Disciplina:</b> Engenharia de Software
<b>3º Semestre</b>	<b>Turno:</b> Diurno / Noturno	<b>C/H Semestral:</b> 80
<b>PROFESSOR RESPONSÁVEL</b>		<b>PROFESSOR EXECUTOR</b>
Esp. Wilson Amaral Martins		

**EMENTA**

Estudo dos conceitos e fundamentos da Engenharia de Software. Compreensão dos principais modelos de processos de software, enfatizando a importância da utilização de métodos e tecnologias para seu desenvolvimento.

**REQUISITOS**

Nenhum identificado

**OBJETIVOS**

<i>Cognitivos</i>	Estudar os conceitos, metodologias e as diversas práticas associadas à engenharia de software com foco no planejamento, desenvolvimento, suporte e manutenção de software.
<i>Habilidades</i>	Desenvolver o pensamento crítico e a habilidade de pensar conceitualmente. Integrar a teoria apreendida à prática do planejamento, desenvolvimento, suporte e manutenção de software. Desenvolver a aprendizagem autônoma e as relações interpessoais.
<i>Atitudes</i>	Valorizar o desenvolvimento de trabalhos em equipe, bem como a interação no processo de aquisição do conhecimento. Respeitar a opinião do outro. Cooperar nos trabalhos em equipe. Participar das aulas com a exposição de sua experiência pessoal em relação aos tópicos da Engenharia de Software, tanto em sala de aula quanto no ambiente virtual de ensino.

UNID.	C/H	CONTEÚDO
I	2	Apresentação da metodologia, plano de ensino e revisão da bibliografia
II	2	Apresentação do Plano de Ensino e Processo de Avaliação do curso que será composto por e notas, a saber: 1. Trabalhos impressos 1.1: Documento de Regras de Negócios - Nota: 0,5 1.2: Especificação de Caso de Uso - Nota: 0,5 1.3: Estratégia de Testes - Nota: 0,52. Trabalho em Grupo (5 pessoas) SWEBOK - Nota: 1,53. Avaliação Bimestral - Nota: 2,0
III	4	Introdução aos conceitos de software, Introdução aos objetivos da engenharia de Software
IV	8	O ciclo de vida de desenvolvimento de software. Atividades do processo de desenvolvimento de software. Modelos de processo de o processo de desenvolvimento de software.
V	4	Gerenciamento de Projetos de Software com o uso do Project Manager Body of Knowledge.
VI	8	Desenvolvimento dos artefatos de Visão, Requisitos, Regras de Negócio, Protótipo do Sistema, Documento de Mensagens do Sistema, Especificação de Casos de Uso e Estratégia de Testes
VII	4	Apresentação das metodologias Ágeis de desenvolvimento de software com ênfase no SCRUM e eXtreme Programming.
VIII	8	Requisitos de Software; Fundamentos de Requisitos de Software; Definição de um Requisito de Software; Requisitos de Produto e Processo; Requisitos funcionais e não funcionais; Propriedades Emergentes; Requisitos Quantificáveis; Requisitos do sistema e requisitos de software; Processo de Requisitos; Modelos de Processo; Suporte e Gerenciamento de Processos; Qualidade e Melhoria do Processo; Elicitação de Requisitos; Requisitos Fontes; Técnicas de Elicitação; Análise de Requisitos; Classificação de requisitos; Modelagem Conceitual; Desenho Arquitetural e Alocação de Requisitos; Negociação de Requisitos; Análise Formal; Especificação de Requisitos; Documento de Definição do Sistema; Especificação de requisitos de sistema; Especificação de Requisitos de Software; Validação de Requisitos;
IX	8	Desenho do Software; Fundamentos de Design de Software; Contexto do Design do Software; Processo de Design de Software; Princípios de Design de Software; Questões-chave no projeto de software; Concorrência; Controle e Manejo de Eventos; Persistência de dados; Distribuição de Componentes; Erro e tratamento de exceções e tolerância a falhas; Interação e Apresentação; Segurança; Estrutura e Arquitetura do Software; Padrões de Design; Design da interface do usuário; Princípios Gerais de Design da Interface do Usuário; Problemas de design da interface do usuário; O Projeto de Modalidades de Interação do Usuário; Processo de Design da Interface do Usuário; Metáforas e modelos conceituais; Análise e Avaliação da Qualidade do Projeto de Software; Atributos de qualidade; Técnicas de Análise e Avaliação da Qualidade; Noções de Design de Software; Descrições Estruturais (Vista Estática); Descrições comportamentais (visualização dinâmica);
X	8	Construção de software; Fundamentos de Construção de Software; Minimização da Complexidade; Antecipar a Mudança; Construindo para Verificação; Reutilização; Padrões em Construção; Gerenciando a Construção; Construção em modelos de ciclo de vida; Planejamento de Construção; Medição de Construção; Considerações práticas; Codificação; Teste de Construção; Construção para Reutilização; Construção com Reutilização; Qualidade de Construção; Integração; Tecnologias de Construção; Projeto e uso da API; Questões de tempo de execução orientadas a objetos; Parametrização e genéricos; Manuseio de Erros, Manipulação de Exceções e Tolerância a Falhas; Técnicas de Construção Baseadas no Estado e na Mesa; Configuração do tempo de execução e internacionalização; Métodos de Construção para Software Distribuído; Construindo Sistemas Heterogêneos; Análise de desempenho e ajuste; Padrões de plataforma; Ferramentas de Construção de Software; Ferramentas de teste de unidade;
XI	8	Manutenção do Software; Fundamentos de Manutenção de Software; Definições e Terminologia; Natureza da Manutenção; Necessidade de manutenção; Maioria dos Custos de Manutenção; Evolução do Software; Categorias de manutenção; Questões-chave na manutenção do software; Problemas técnicos; Problemas de Gerenciamento; Estimativa de custo de manutenção; Medição da Manutenção do Software; Processo de Manutenção; Processos de Manutenção; Atividades de manutenção; Técnicas de Manutenção; Compreensão do programa; Reengenharia; Engenharia Inversa; Migração de dados.
XII	8	Testes de Software; Fundamentos de Testes de Software; Terminologia relacionada ao teste; Questões-chave; Relação de Testes com Outras Atividades; Níveis de Teste; O alvo do teste; Objetivos do Teste; Técnicas de Teste; Baseado na Intuição e Experiência do Software Engineer; Técnicas baseadas em domínio de entrada; Técnicas baseadas em código; Técnicas baseadas em falhas; Técnicas baseadas em uso; Técnicas de Teste Baseadas em Modelos; Técnicas Baseadas na Natureza da Aplicação; Seleção e combinação de técnicas; Medidas Relacionadas ao Teste; Avaliação do Programa em Teste; Avaliação dos Testes Realizados; Processo de Teste; Considerações práticas; Atividades de Teste; Ferramentas de Teste de Software; Suporte à Ferramenta de Teste; Categorias de ferramentas.

XIII	8	Apresentação das principais métricas de software e detalhamento da contagem utilizando Pontos de Função
------	---	---

### ESTRATÉGIA DE ENSINO

Apresentação de conteúdo em sala de aula, disponibilização de conteúdo disponível no Blackboard. Exercícios de fixação de aprendizagem desenvolvidos individualmente e em grupo. Elaboração de um projeto de software como trabalho prático em grupo. Inserção de material produzido (trabalhos e produção textual) no Ambiente Virtual de Aprendizagem (Blackboard).

### RECURSOS DISPONÍVEIS

Blackboard e projetor.

### AVALIAÇÃO

O semestre letivo é composto por 02 (duas) avaliações de aprendizagem, com conteúdos cumulativos: - Avaliação Regimental (A1): 5,0 (cinco) - Avaliação Docente (A2): 5,0 (cinco) Para as disciplinas que não possuem PRI as avaliações A1 e A2 são de responsabilidade de cada docente. A Nota Final (NF) é obtida pelo somatório de A1 e A2. Assim: A1 A2 = NF Para aprovação o estudante deverá obter NF igual ou superior a 6,0 (seis) e, no mínimo, 75% (setenta e cinco por cento) de presenças. Se a NF for inferior a 6,0 (seis) e o estudante tiver obtido ao menos 1,0 (um) na A1 ou na A2, poderá realizar uma Avaliação Final (AF), correspondente a 5,0 (cinco). Neste caso, a AF substituirá a menor nota lançada no sistema.

BIBLIOGRAFIA BÁSICA	BIBLIOGRAFIA COMPLEMENTAR
<p>BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML: guia do usuário. 2 ed., rev. e atual. Rio de Janeiro: Campus, 2012.</p> <p>PRESSMAN, Roger S. Engenharia de software: uma abordagem profissional. 7. ed. Porto Alegre: McGraw-Hill, 2011.</p> <p>SOMMERVILLE, Ian. Engenharia de software. 9. ed. São Paulo: Pearson Prentice Hall, 2011</p>	<p>COHN, Mike. Desenvolvimento de Software com Scrum - Aplicando Métodos Ágeis com Sucesso. Porto Alegre: Bookman, 2011.</p> <p>LIMA, Adilson da Silva. UML 2.3: do requisito à solução. 1. ed. rev. e atual. São Paulo: Érica, 2011. 368 p.</p> <p>MARTINS, José Carlos Cordeiro. Gerenciando projetos de desenvolvimento de software com PMI, RUP e UML. 4. ed. Rio de Janeiro: Brasport, 2007. xxiv, 325 p.</p> <p>OKUYAMA, Fabio Yoshimitsu; MILETTO, Evandro Manara; NICOLAO, Mariano. Desenvolvimento de Software I: Conceitos Básicos - Série Tekne. Porto Alegre : Bookman, 2014</p> <p>PFLEEGER, Shari Lawrence. Engenharia de Software: teoria e prática - material na web. Disponível em <a href="http://wps.prenhall.com/br_pfleeger_engsoftware_2/">http://wps.prenhall.com/br_pfleeger_engsoftware_2/</a>. Acessado em 20 de outubro de 2013.</p>