

**DIRETORIA DE ENSINO/DEPARTAMENTO DE ENSINO
COORDENAÇÃO DO CURSO TÉCNICO INTEGRADO EM INFORMÁTICA
PROGRAMA DE UNIDADE DIDÁTICA – PUD**

DISCIPLINA: ENGENHARIA DE SOFTWARE		
Código: 31.101	Carga horária total: 40h	Créditos: 02
Nível: Médio-Técnico	Ano: 3º	Pré-requisitos: Não
CARGA HORÁRIA	Teórica: 30h	Prática: 10h
	Presencial: Sim	Distância: Não
	Prática Profissional: Não se aplica.	
	Atividades não presenciais: Não se aplica.	
	Extensão: Não se aplica.	
EMENTA		
Fundamentos da Engenharia de Software. Evolução e desafios do desenvolvimento de software. Modelos de processo e metodologias tradicionais e ágeis. Engenharia de requisitos: elicitação, análise, especificação e validação. Introdução à modelagem de sistemas com UML. Noções de projeto de interfaces e experiência do usuário.		
OBJETIVO		
<ul style="list-style-type: none"> -Compreender os fundamentos, processos e metodologias da Engenharia de Software. Identificar, analisar e especificar requisitos de sistemas. -Aplicar técnicas básicas de modelagem utilizando UML. -Reconhecer práticas de gerenciamento e organização do desenvolvimento de software. -Desenvolver protótipos simples de interface com base em princípios de usabilidade e experiência do usuário. 		
PROGRAMA		
UNIDADE 1 – Fundamentos da Engenharia de Software <ul style="list-style-type: none"> 1.1 Conceitos fundamentais 1.2 Natureza e características do software 1.3 Problemas do desenvolvimento de software e crise do software 1.4 Mitos e equívocos no desenvolvimento de software 		
UNIDADE 2 – Processos e Metodologias de Desenvolvimento <ul style="list-style-type: none"> 2.1 Atividades essenciais do processo de software 2.2 Modelos tradicionais: cascata, incremental e iterativo 2.3 Introdução às metodologias ágeis: princípios gerais, Scrum e XP 		
UNIDADE 3 – Engenharia de Requisitos <ul style="list-style-type: none"> 3.1 Tipos de requisitos 3.2 Técnicas básicas de elicitação 3.3 Análise e especificação de requisitos 3.4 Validação e rastreabilidade 		
UNIDADE 4 – Modelagem de Sistemas com UML <ul style="list-style-type: none"> 4.1 Princípios básicos da UML 4.2 Diagrama de Casos de Uso 4.3 Diagrama de Classes 4.4 Diagrama de Sequência 		
UNIDADE 5 – Projeto de Interfaces e UX <ul style="list-style-type: none"> 5.1 Princípios fundamentais de UI e UX 5.2 Prototipação de baixa fidelidade 5.3 Avaliação de interfaces 		

METODOLOGIA DE ENSINO				
Exposição dialogada dos conteúdos teóricos.				
Aulas práticas de análise de requisitos e modelagem com UML.				
Estudos de caso relacionados ao ciclo de vida de software.				
Utilização de metodologias ativas, como aprendizagem baseada em problemas e projetos.				
Elaboração de protótipos simples de interface.				
Uso de projetor multimídia, quadro branco e laboratório de informática.				
RECURSOS				
Laboratório de informática				
Projetor multimídia				
Computadores				
Quadro branco e pincel				
AVALIAÇÃO				
A avaliação será contínua, diagnóstica e cumulativa.				
Distribuição mínima em quatro notas (N1, N2, N3 e N4), incluindo provas escritas, exercícios, relatórios, seminários e atividades práticas.				
Assiduidade e participação serão consideradas.				
Em caso de não atingir os objetivos, o estudante realizará recuperação paralela mediante atividades orientadas ou atendimento individualizado.				
Após recuperação, poderá realizar nova avaliação (lista de exercícios ou prova).				
BIBLIOGRAFIA BÁSICA				
1. GOMES, André Faria. Agile : desenvolvimento de software com entregas frequentes e foco no valor de negócio. São Paulo, SP: Casa do Código, 2014. <i>E-book</i> . Disponível em: https://plataforma.bvirtual.com.br . Acesso em: 10 out 2025.				
2. VAZQUEZ, Carlos Eduardo; SIMÕES, Guilherme Siqueira. Engenharia de requisitos : software orientado ao negócio. 1. ed. Rio de Janeiro: Brasport, 2016. <i>E-book</i> . Disponível em: https://plataforma.bvirtual.com.br . Acesso em: 10 out 2025.				
3. SOMMERVILLE, Ian. Engenharia De Software .10.ed. São Paulo:Pearson Education do Brasil, 2018. Livro. (768 p.). ISBN 9788543024974.				
BIBLIOGRAFIA COMPLEMENTAR				
1. WILDT, Daniel <i>et al.</i> Extreme programming : práticas para o dia a dia no desenvolvimento ágil de software. São Paulo, SP: Casa do Código, 2015. <i>E-book</i> . Disponível em: https://plataforma.bvirtual.com.br . Acesso em: 10 out 2025.				
2. MASSARI, V. L. Agile Scrum Master no gerenciamento avançado de projetos . 1. ed. Rio de Janeiro: Brasport, 2016. <i>E-book</i> . Disponível em: https://plataforma.bvirtual.com.br . Acesso em: 10 out 2025.				
3. SILVA, Leonardo Soares e; FORTES, Gabriel. Aprenda a programar com python : descomplicando o desenvolvimento de software. São Paulo, SP: Casa do Código, 2022. <i>E-book</i> . Disponível em: https://plataforma.bvirtual.com.br . Acesso em: 10 out 2025.				
4. GALLOTTI, Giocondo Marino Antonio (org.). Arquitetura de software . São Paulo: Pearson, 2016. <i>E-book</i> . Disponível em: https://plataforma.bvirtual.com.br . Acesso em: 10 out 2025.				
5. LISBOA, Flávio. Arquitetura de software distribuído : boas práticas para um mundo de microserviços. São Paulo, SP: Casa do Código, 2021. <i>E-book</i> . Disponível em: https://plataforma.bvirtual.com.br . Acesso em: 10 out 2025.				