

PLANO DE ENSINO

ÁREA Engenharias

UNIDADE CURRICULAR: Modelos, métodos e técnicas da engenharia de software (0005317)

Período letivo	Carga horária
2024/1	160h
Curso	
Engenharia da Computação	

Tópicos geradores

Como implementar software através de conceitos, métodos e práticas da engenharia de software?; Quais os principais modelos de engenharia de software e como podem ser aplicados?; Modelos de processos de software baseados em cascata, prototipação, incremental e espiral; Estudo da viabilidade de software através da engenharia de requisitos; Como garantir o desenvolvimento de software por meio de uma arquitetura adequada e utilizando metodologias de projetos?; Qual o diferencial de uma solução que utiliza padrões de projetos?; Projetar e desenvolver softwares utilizando metodologias ágeis de desenvolvimento; Conceito de DevOps para integração e entrega contínua de software;

Metas de compreensão

Reconhecer as fases da engenharia de requisitos, bem como sua aplicação e importância desde o levantamento de requisitos até a gestão de requisitos; Aplicar princípios e práticas do desenvolvimento ágil em desenvolvimento de software; Identificar e avaliar a aplicabilidade dos modelos de processo de software e as boas práticas da engenharia de software; Diferenciar tipos de arquitetura de software, compreendendo seus benefícios e quais aplicá-los em determinadas situações; Aplicar integração e entrega contínua envolvendo práticas de DevOps; Analisar as diferentes técnicas para coleta de requisitos, e diferenciar como cada uma delas pode ser utilizada; Criar soluções com os principais padrões de projeto e boas práticas de arquitetura de software; Meta máxima: aplicar o desenvolvimento ágil com frameworks atuais, de acordo com seus artefatos;

Desempenho de compreensão

Os desempenhos de compreensão são constituídos por experiências, estudos, pesquisas e práticas desenvolvidos no decorrer do semestre letivo que possibilitam ao aluno expressar, de diferentes formas, evidências de que atingiu a compreensão. Tais desempenhos demonstram com clareza que os alunos dominam as metas de compreensão, por meio de sua atuação em projetos desafiadores e acessíveis que promovem o seu envolvimento reflexivo sobre a situação a ele exposta. Nesta Unidade Curricular, os alunos demonstrarão a sua compreensão por meio do(s) seguinte(s) desempenho(s): (descrição dos professores da UC do desempenho ou desempenhos definidos no planejamento).

Avaliação continuada

A avaliação contínua está diretamente ligada à realização do trabalho pedagógico e se concretiza no acompanhamento do processo de ensino e aprendizagem por várias formas. Constitui-se de momentos e instrumentos imprescindíveis utilizados pelo professor para que as metas de compreensão sejam atingidas. Os estudantes terão dois momentos (A1 e A2), previstos no calendário, com o objetivo de refletir sobre o seu nível de compreensão das habilidades de sua área de estudos e competências desenvolvidas na Unidade Curricular, e a avaliação dos desempenhos ao longo do semestre (A3). A nota final do semestre será composta por: A1 - Avaliação dissertativa em que o aluno demonstrará habilidades na expressão da linguagem, códigos e signos da área, valendo 30 pontos. A2 - Avaliação com questões de múltipla escolha em que o aluno demonstrará habilidades de leitura, interpretação, análise de informações e estabelecimento de relações, valendo 30 pontos. A3 - Avaliação de desempenho como resultado do processo composto por uma variedade de feedbacks para o atingimento das metas de compreensão e a concretização dos desempenhos parciais e consequente construção do desempenho final, valendo 40 pontos. Critérios de atribuição da nota final e de aprovação: A nota final será a soma das notas da A1, A2 e A3 (A1+A2+A3) Para aprovação, a nota final deverá ser igual ou maior que 70 e a frequência igual ou superior a 75% da Carga Horária da Unidade Curricular. Obs: Caso a frequência do aluno seja menor que 75%, ele estará reprovado, independentemente da nota. Avaliação Integrada: O aluno que tiver frequência de 75% ou mais e nota inferior a 70 pontos poderá realizar a Avaliação Integrada (AI). A AI valerá 30 pontos e a nota, se maior, substituirá a menor das notas entre a A1 e a A2. Caso a nota da AI, por ser maior, substitua a nota da A1 ou da A2, a soma das três notas (A1+A2+A3) deverá ser 70 pontos ou mais, para a aprovação. Se a nota da AI for inferior à nota da A1 e, também, da A2, não haverá substituição e o aluno estará reprovado na Unidade Curricular.

Ementa

Conceitos, métodos e práticas da engenharia de software. Modelos de processos de software. Modelo de processo cascata. Modelo de processo de prototipação. Modelo de processo incremental. Modelo de processo espiral. Engenharia de requisitos. Estudo de viabilidade. Levantamento de requisitos. Análise e negociação de requisitos. Especificação de requisitos. Validação de requisitos. Gestão de requisitos. Visão de análise e projeto. Projeto e tipos Arquitetura de software. Padrões de projeto. Integração e entrega contínua. Paradigma de desenvolvimento ágil. Manifesto ágil. Frameworks para desenvolvimento ágil de software. Visão geral DevOps.

Certificação

Qualificação Profissional em Modelos, Métodos e Técnicas da Engenharia de Software

Competências

Compreender e aplicar os diferentes modelos de processos de desenvolvimento de software para diferentes problemas computacionais; Identificar e analisar requisitos e especificações para problemas específicos e planejar estratégias para suas soluções, bem como sua gestão; Projetar e criar soluções computacionais com qualidade em harmonia com o ambiente social e físico no seu entorno de aplicação considerando os princípios e boas práticas de engenharia de software; Avaliar e tomar decisões com base no conhecimento de arquitetura de software e considerando aspectos de infraestrutura, consciente dos aspectos éticos, legais e dos impactos ambientais decorrentes; Aplicar os princípios de interação humano computador para avaliar e construir sistemas de software com enfoque na experiência do usuário; Compreender e aplicar práticas e princípios de desenvolvimento ágil para sistemas de software que se adequam a esse novo paradigma;

Bibliografia básica

MUNIZ, Antonio; SANTOS, Rodrigo; IRIGOYEN, Analia; MOUTINHO, Rodrigo. Jornada DevOps Unindo cultura ágil, Lean e tecnologia para entrega de software com qualidade. 1a. Edição. São Paulo: Editora Brasport, 2019.

PRESSMAN, Roger; MAXIM, Bruce. Engenharia de Software. Uma abordagem profissional. 8a. Ed. Bookman, 2016. <https://integrada.minhabiblioteca.com.br/#/books/9788580555349/cfi/3!/4/2@100:0.00>

SOMMERVILLE, Ian. Engenharia de Software. 9. ed. São Paulo: Pearson Prentice Hall, 2011. https://bv4.digitalpages.com.br/?term=engenharia%2520de%2520software&searchpage=1&filtro=todos&from=busca&page=_14&ion=0#/legacy/276

Bibliografia complementar

PFLEEGER, Shari Lawrence. Engenharia de software: teoria e prática. 2. ed. São Paulo: Prentice Hall, 2004.

<https://bv4.digitalpages.com.br/?term=engenharia%2520de%2520software&searchpage=1&filtro=todos&from=busca#/legacy/476>

FOGGETTI, Cristiano. Gestão Ágil de Projetos. Pearson, 2015. <https://bv4.digitalpages.com.br/?term=desenvolvimento%2520%25C3%25A1gil&searchpage=1&filtro=todos&from=busca&page=-7§ion=0#/legacy/22131>

RANGEL, Pablo. Sistemas orientados a objetos. Rio de Janeiro: Brasport, 2021. E-book. Disponível em: <https://plataforma.bvirtual.com.br/Leitor/Publicacao/197367/epub/0>.

MORAIS, Izabelly Soares de. Engenharia de software. São Paulo: Pearson, 2017. E-book. Disponível em: <https://plataforma.bvirtual.com.br/Leitor/Publicacao/184098/pdf/0>.

FOWLER, Martin; SCOTT, Kendall. UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos. 3ª. ed. Porto Alegre: Bookman, 2004. <https://integrada.minhabiblioteca.com.br/#/books/9788560031382/cfi/6/2!/4/2@0:0.131>

Conteúdo programático

Conceitos, métodos e práticas da engenharia de software. Modelos de processos de software. Modelo de processo cascata. Modelo de processo de prototipação. Modelo de processo incremental. Modelo de processo espiral. Engenharia de requisitos. Estudo de viabilidade. Levantamento de requisitos. Análise e negociação de requisitos. Especificação de requisitos. Validação de requisitos. Gestão de requisitos. Visão de análise e projeto. Projeto e tipos Arquitetura de software. Padrões de projeto. Integração e entrega contínua. Paradigma de desenvolvimento ágil. Manifesto ágil. Frameworks para desenvolvimento ágil de software. Visão geral DevOps.