

PLANO DE ENSINO
ÁREA TI & Computação

UNIDADE CURRICULAR: Modelos, métodos e técnicas da engenharia de software (0021638)

Período letivo	Carga horária
2024/2	160h
Cursos	
Análise e Desenvolvimento de Sistemas, Análise e Desenvolvimento de Sistemas (Semipresencial) 30/70, Ciência da Computação, Ciência da Computação (Semipresencial) 30/70, Ciência Dos Dados (Semipresencial) 30/70, Sistemas de Informação, Sistemas de Informação (Semipresencial) 30/70	

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; Quais ferramentas podem ser utilizadas para criar modelos de softwares?; Como a realização de prototipações pode contribuir com o desenvolvimento e modelagem de um software? 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; Utilizar ferramentas de prototipagem de software e aplicar os tipos de prototipagem conforme o projeto; 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. Define-se três avaliações principais, que marcam o processo avaliativo na medida em que o aluno percorre seu processo formativo, a saber: A1 - avaliação discursiva em que o aluno demonstrará competências por meio de expressão de linguagem, códigos e signos da área, valendo 30 pontos. A2 - avaliação composta por questões objetivas que visam permitir fazer análises e estabelecer relações evidenciando as competências de leitura e interpretação, valendo 30 pontos. A3 - avaliação que acompanhe o processo de ensino-aprendizagem ao longo do semestre e resulte no desenvolvimento de um projeto ou produto, estudo técnico, croqui, solução digital, arte ou outro formato equivalente e condizente com a Unidade Curricular que permita tangibilizar um desempenho de compreensão, valendo 40 pontos. A nota final será composta pela soma das notas da A1, A2 e A3 (A1+A2+A3). Nas unidades curriculares presenciais, estará aprovado o aluno que obtiver, na soma das três avaliações (A1+A2+A3), a nota mínima de 70 pontos e atingir, no mínimo, 75% de frequência nas aulas presenciais. Nas unidades curriculares digitais (UCD), estará aprovado o aluno que obtiver, na soma das três avaliações (A1+A2+A3), a nota mínima de 70 pontos. O aluno que tenha obtido nota final inferior a 70 pontos, possuem no mínimo 40 (quarenta) pontos em uma das seguintes somas: A1+A3 ou A2+A3 e tiver, no mínimo 75% de presença nas aulas da unidade curricular presencial, poderá realizar avaliação integrada (AI) conforme calendário acadêmico. A nota será atribuída numa escala de 0 (zero) a 30 (trinta) pontos e substituirá, entre A1 e A2, a menor nota. 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. Após o lançamento da nota da avaliação integrada (AI), o aluno que obtiver 70 pontos, como resultado da soma das avaliações (A1, A2 e A3), será considerado aprovado. O aluno reprovado na unidade curricular deverá refazê-la, na modalidade presencial ou digital, respeitada a oferta. A reprovação em componente curricular não interrompe a progressão do aluno no curso.

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. Modelo de processo unificado. Engenharia de requisitos. Estudo de viabilidade. Levantamento e técnicas elicitação 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

- DELAMARO, Marcio. Introdução ao teste de software. Rio de Janeiro: LTC, 2016. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788595155732/>.
- PRESSMAN, Roger; MAXIM, Bruce. Engenharia de Software. Uma abordagem profissional. 8a. Ed. Bookman, 2016. <https://integrada.minhabiblioteca.com.br/#/books/978858055349/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

- FOGGETTI, Cristiano (org.). Gestão ágil de projetos. São Paulo: Pearson, 2015. E-book. Disponível em: <https://plataforma.bvirtual.com.br/Leitor/Publicacao/22131/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>
- 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>.
- PFLEEGER, Shari Lawrence. Engenharia de software: teoria e prática. 2. ed. São Paulo: Pearson, 2004. E-book. Disponível em: <https://plataforma.bvirtual.com.br/Leitor/Publicacao/476/pdf/0>
- 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>.

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. Modelo de processo unificado. 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. Modelagem e notação UML (Unified Modeling Language): diagramas estruturais e comportamentais. Modelo de casos de uso. Diagrama de atividades. Diagrama de classes. Diagrama de sequência. Diagrama de estados. Diagrama de componentes. 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.