

Caracterização da disciplina

Código da disciplina:	MCTA010-13	Nome da disciplina:	Engenharia de Software						
Créditos (T-P-I):	(4-0-4)	Carga horária:	48 horas	Aula prática:	N	Câmpus:	SA		
Código da turma:	NA2MCTA033-15SA	Turma:	A2	Turno:	Noturno	Quadrimestre:	1	Ano:	2019
Docente(s) responsável(is):	FRANCISCO DE ASSIS ZAMPIROLI								

Alocação da turma

	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
8:00 - 9:00						
9:00 - 10:00						
10:00 - 11:00						
11:00 - 12:00						
12:00 - 13:00						
13:00 - 14:00						
14:00 - 15:00						
15:00 - 16:00						
16:00 - 17:00						
17:00 - 18:00						
18:00 - 19:00						
19:00 - 20:00		X (sem. I/II)				
20:00 - 21:00		X (sem. I/II)				
21:00 - 22:00					X (sem. I/II)	
22:00 - 23:00					X (sem. I/II)	

Planejamento da disciplina
Objetivos gerais

Apresentar os fundamentos sobre Engenharia de Software, principalmente por meio da explicação e experimentação dos conceitos e do uso prático da modelagem de sistemas simples. Propiciar ao aluno mecanismos para a construção de produtos de software de forma eficiente e que atendem aos padrões de qualidade, confiabilidade e economia de recursos.

Objetivos específicos

Que o aluno seja capaz de compreender os conceitos fundamentais a respeito da Engenharia de Software. Que o aluno entenda todo o ciclo de vida do desenvolvimento do software e adquira a habilidade prática de desenvolver projeto básicos para modelar e solucionar problemas de natureza técnico-científica, independentemente de uma linguagem ou de um paradigma de programação específicos.

Ementa

Introdução a Engenharia de Software. Modelos de processos de desenvolvimento de software. Gerência de projeto. Modelagem e especificação de requisitos de software. Análise de requisitos de software. Verificação e validação de requisitos de software. Noções de métodos formais para especificação e verificação de requisitos. Ferramentas para engenharia de requisitos. Métricas de requisitos de software. Requisitos e prototipagem de interfaces. Aspectos éticos relacionados ao desenvolvimento de software

Conteúdo programático

Aula	Conteúdo	Estratégias didáticas	Avaliação
1	Apresentação	Aula expositiva e Exercícios	Por meio de exercícios
2	Visão Geral de ES	Aula expositiva e Exercícios	Por meio de exercícios
3	Processos de Software	Aula expositiva e Exercícios	Por meio de exercícios
4	Gerência de Projetos	Aula expositiva e Exercícios	Por meio de exercícios
5	Requisitos	Aula expositiva e Exercícios	Por meio de exercícios
6	Análise	Aula expositiva e Exercícios	Por meio de exercícios
7	Projeto	Aula expositiva e Exercícios	Por meio de exercícios
8	Revisão	Aula expositiva e Exercícios	Por meio de exercícios
9	Exercícios	Aula expositiva e Exercícios	Por meio de exercícios
10	Primeira Prova - 19/3	Prova individual	Prova escrita
11	Estratégias de Teste	Aula expositiva e Exercícios	Por meio de exercícios
12	Técnicas de Teste	Aula expositiva e Exercícios	Por meio de exercícios
13	Qualidade	Aula expositiva e Exercícios	Por meio de exercícios
14	Métricas	Aula expositiva e Exercícios	Por meio de exercícios
15	Tópicos	Aula expositiva e Exercícios	Por meio de exercícios
16	Revisão	Aula expositiva e Exercícios	Por meio de exercícios
17	Exercícios	Aula expositiva e Exercícios	Por meio de exercícios
18	Segunda Prova - 16/4	Prova individual	Prova escrita
19	Prova Substitutiva - 23/4	Prova individual	Prova escrita
20	Projeto - 26/4	Apresentação de Projetos	Avaliação do Projetos
21	Projeto - 30/4	Apresentação de Projetos	Avaliação do Projetos
22	Vista	Vista de prova	vista
23	Prova de Recuperação - 8/5	Prova individual	Prova escrita
24	Vista	Vista de prova	vista

Descrição dos instrumentos e critérios de avaliação qualitativa

Ferramentas: CASE

Critérios de Avaliação: Os alunos são avaliados semanalmente através de exercícios práticos realizados em projetos em grupo. Isso permite identificar as dificuldades enfrentadas pelos alunos desde o início do quadrimestre. Os conceitos de teoria são avaliados principalmente em duas avaliações escritas onde os alunos devem demonstrar a capacidade de interpretar código e também de elaborar soluções de problemas e traduzir a solução em um algoritmo formal. As atividades semanais realizadas na disciplina valem 20%. Plágios serão punidos com o conceito F na prova. A primeira prova vale 40%, a segunda prova vale 40%. A prova de recuperação terá toda a teoria. A média final da disciplina será calculada considerando 50% do conceito final (antes do mecanismo de recuperação) e 50% do conceito obtido no

mecanismo de recuperação. Os pesos não determinísticos nos cálculos dos conceitos.

Horário de atendimento: sexta-feira, das 16:00 às 18:00 horas, na sala 819, BLOCO B, SA.

Referências bibliográficas básicas

1. Pressman, Roger S. Engenharia de Software. 6.ed. - Rio de Janeiro: McGraw-Hill, 2006.
2. Sommerville, I. Engenharia de Software. 10.ed. - São Paulo : Addison-Wesley, 2007.
3. Booch, G.; Rumbaugh, J.E.; Jacobson, I. UML, guia do usuário. Rio de Janeiro: Campus, 2000.

Referências bibliográficas complementares

1. BEZERRA, Eduardo. Princípios de análise e projeto de sistemas com UML. Rio de Janeiro: Elsevier, 2007. 369 p. ISBN 9788535216967.
2. BRUEGGE, Bernd.; DUTOIT, Allen H.. Object-oriented software engineering: using UML, Patterns, and java. 2nd ed.. Upper Saddle River, NJ: Prentice Hall, c2004. xxxiv, 762 p. Includes bibliographical references (p. 739-750). ISBN 978130471109.
3. GUEDES, Gilleanes T. A.. UML 2: uma abordagem prática. São Paulo: Novatec editora, 2009. 485 p. ISBN 9788575221938.
4. JACOBSON, Ivar. Object-oriented software engineering: a use case driven approach. [NewYork]: ACM Press, c1992. xx, 524 p. Includes bibliographical references (p. 513-520) and index. ISBN 0201544350.
5. LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3.ed. Porto Alegre: Bookman, 2007. 695 p. ISBN 9788560031528.