

Campus: São José dos Campos		
Curso (s): Engenharia de Computação e Ciência da Computação		
Unidade Curricular (UC): Projeto Orientado a Objetos		
Unidade Curricular (UC): <i>[nome da UC em inglês] Object-Oriented Design</i>		
Código da UC: 5168		
Docente Responsável: Fábio Fagundes Silveira / DCT		Contato (e-mail): <i>[opcional]</i> fsilveira@unifesp.br
Docente (s) Colaborador/a (es/as):		Contato (e-mail): <i>[opcional]</i>
Ano letivo: 2023	Termo: 5	Turma (s): I/N
Nome do Grupo/Módulo/Eixo da UC (se houver):		Idioma predominante em que a UC será oferecida: ( X ) Português ( ) English ( ) Español ( ) Français ( ) Libras ( ) Outro:
UC: ( X ) Fixa ( ) Eletiva ( ) Optativa	Oferecida como: ( X ) Disciplina ( ) Módulo ( ) Estágio ( ) Outro:	Oferta da UC: ( X ) Semestral ( ) Anual
Ambiente Virtual de Aprendizagem: ( ) Moodle ( X ) Classroom ( ) Outro: ( ) Não se aplica		
Pré-Requisito (s) - Indicar Código e Nome (s) da (s) UC:		
Carga horária total (em horas): 76		
Carga horária teórica (em horas): 36	Carga horária prática (em horas): 36	Carga horária de extensão (em horas, se houver):
Se houver atividades de extensão, indicar código e nome do projeto ou programa vinculado na Pró-Reitoria de Extensão e Cultura (ProEC):		
Ementa: Fundamentos de Projeto de Software; Modelagem Orientada a Objetos Utilizando UML e SysML; Modelos Arquiteturais; Componentes de Software; Reuso de Software; Padrões de Projeto.		
Conteúdo Programático:  1. Linguagem de Modelagem Unificada (UML) <ul style="list-style-type: none"> <li>a. Histórico e Introdução</li> <li>b. Casos de Uso</li> <li>c. Requisitos com casos de uso</li> <li>d. Diagramas da UML: casos de uso, atividades, classes, seqüência, colaboração ou comunicação, objetos, estados, componentes, implantação, tempo</li> <li>e. Técnicas de análise orientada a objetos: identificação de classes, relacionamentos, operações e atributos.</li> <li>f. Estudo de Caso e Exercícios</li> </ul> 2. Linguagem de Modelagem de Sistemas (SysML) <ul style="list-style-type: none"> <li>a. Histórico e Introdução</li> <li>b. Descrição e Arquitetura da Linguagem</li> <li>c. Especificação da SysML</li> <li>d. Diagramas da SysML</li> <li>e. Estudo de Caso e Exercícios</li> </ul> 3. Padrões de Projeto <ul style="list-style-type: none"> <li>a. Padrões de Criação</li> <li>b. Padrões Estruturais</li> </ul>		

- c. Padrões Comportamentais
- 4. Arquitetura n camadas
  - a. Introdução
  - b. Camada de interface
  - c. Camada de negócios
  - d. Camada de dados
  - e. Estudo de Caso e Exercícios
- 5. Métodos Ágeis
  - a. Introdução
  - b. Principais Métodos Ágeis: XP, SCRUM, outros.
  - c. Elaboração progressiva de requisitos
  - d. Planejamento Iterativo e adaptação.
- 6. Refatoramento
  - a. Introdução
  - b. Princípios
  - c. “Bad Smells”
  - d. Ferramentas de refatoração
  - e. Estudos de Caso e Exercícios

Objetivos:

Gerais:

O objetivo desta disciplina é capacitar os alunos a conceber e especificar projetos de software, fazendo uso de técnicas e métodos avançados da tecnologia orientada a objetos.

Específicos:

Embasar os alunos com os principais conceitos sobre projeto de software;  
Capacitar os alunos no uso de técnicas de modelagem de software orientada a objetos;  
Apresentar aos alunos os principais modelos arquiteturais que embasam o projeto de software;  
Capacitar os alunos no desenvolvimento de software baseado em componentes;  
Capacitar os alunos para o uso dos principais padrões de projeto de software orientado a objetos;  
Despertar no aluno a motivação para o desenvolvimento de software baseado em reuso.

Metodologia de ensino:

A disciplina será intercalada por aulas teóricas e aulas práticas em laboratório. Nas aulas teóricas serão apresentados os principais conceitos e seus relacionamentos; nas aulas de laboratório, os conceitos serão implementados em ferramentas CASE e em linguagem Java. Ademais, desenvolver-se-á atividades à distância, com o apoio da ferramenta *ClassRoom*.

A metodologia de ensino baseada na resolução de problemas (*Problem Based Learning*) será amplamente utilizada. O professor, após apresentar a teoria necessária, irá propor problemas e atuará apenas como facilitador junto aos alunos na resolução do problema.

Avaliação:

A avaliação do aprendizado dar-se-á por meio de uma ou duas provas semestrais (i), trabalhos individuais (ii), trabalhos em grupo (iii) e lista de exercícios (iv). A nota final será uma média ponderada entre as notas dos itens i (peso 3), ii (peso 1), iii (peso 1) e iv (peso 1).

Bibliografia:

Básica:

- 1.GOMAA, H. Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures. Cambridge University Press, 2011. ISBN-13: 978-0521764148.
- 2.VLISSIDES, J., Helm, H., GAMMA, E, JOHNSON, R., Padrões de Projeto, Editora Bookman, 2005.
- 3.GRADY BOOCH; JAMES RUMBAUGH; IVAR JACOBSON. UML: Guia do Usuário. 2ª ed. Rio de Janeiro: Elsevier, 2005.

Complementar:

- 1.BEZERRA, E. Princípios de Análise e Projeto de Sistemas com UML. 2ª ed.Rio de Janeiro: Elsevier, 2007.
- 2.Sommerville, I. Software Engineering. Addison-Wesley, 9th edition, 2011. ISBN-13: 978-0137035151.
- 3.BUSHMANN, F., STAL, M., Meunier, R., SOMMERLAD, P., Pattern-Oriented Software Architecture: a System of Patterns. Editora Wiley, 1996.
- 4.LARMAN, C.; Salvador, L.M.A, Utilizando UML e Padrões: Uma Introdução à Análise e ao Projeto Orientado a Objetos. Porto Alegre: Bookman, 2000.

5. LADDAD, R. AspectJ in action: practical aspect-oriented programming. Greenwich: Manning Publications Co., 2003. ISBN 1930110936.
6. GAMMA, E.; HELM, R. Design Patterns: Elements Of Reusable Object-Oriented Software. Boston: Addison-Wesley, 2002
7. WAZLAWICK, R. S. Análise e Projeto de Sistemas de Informação Orientados a Objetos, Editora Campus, 2004. [20]
8. GUERRA, E. Design Patterns com Java: Projeto orientado a objetos guiado por padrões. Casa do Código, 2013.