

**MO620 - Engenharia de Software II - turma B**  
**MC976 – Tópicos em Engenharia de Software II - turma A**  
**Instituto de Computação - UNICAMP**  
**2o. Semestre de 2018**  
**Profa. Cecilia M.F.Rubira**  
**sala 13,cmrubira@ic.unicamp.br, IC1**

**1. Critério de Avaliação**

Os alunos farão 1 prova: dia 31/10/2018 (quarta), 1 projeto e 1 seminário de pesquisa.

A prova e o seminário são atividades individuais enquanto que o projeto será desenvolvido em grupo.

As apresentações dos seminários começam a partir de 5 de novembro.

Existe uma nota relativa à participação em sala do aluno (*Participa*) que inclui exercícios feitos em sala, listas, exercícios extras, etc.

Prova vale 4.0.

Projeto vale 2.0.

Semin vale 3.0.

Participa vale 1.0.

A média  $M$  da disciplina será calculada por:

$$M = Prova + Projeto + Semin + Participa.$$

**2. Programa do Curso**

Os seguintes tópicos serão cobertos neste curso:

- Modelo de Objetos:
  - conceitos do modelo de objetos e notação UML,
  - diagramas de casos de uso,
  - modelagem da estrutura estática das classes e seus relacionamentos,
  - definição de subsistemas e seus interrelacionamentos,
  - modelagem da estrutura dinâmica do sistema,
  - diagramas de seqüência e de colaborações,
  - diagramas de pacotes.
- Arquitetura de Software
  - arquitetura de software: componentes e conectores, configurações.
  - estilos e padrões de arquitetura,
  - padrões de projeto e frameworks orientados a objetos,
- Tolerância a Falhas e Tratamento de Exceções,
- Reutilização de Software, Desenvolvimento baseado em componentes e Linhas de Produto de Software.

**3. Temas dos Seminários**

Abaixo encontram-se alguns temas sugeridos. Os alunos podem definir temas de interesse próprio próximos dos seus temas de pesquisa.

“Comparação entre Métodos Ágeis e Tradicionais”  
“Desenvolvimento Orientado a Aspectos”  
“Testes em Linhas de Produtos de Software”  
“Arquitetura de Software orientada a serviços”  
“Desenvolvimento Orientado por Modelos”  
“Engenharia de software experimental”  
“Ecosistemas em Engenharia de Software”  
“Visualização de software”

#### 4. Tema Sugerido para o Projeto

Um sistema de gestão acadêmico para o Programa de Pós-graduação do IC-UNICAMP.

#### 5. Bibliografia Recomendada

##### Engenharia de Software

I. Sommerville, *Software Engineering (9th edition)*, Pearson (Addison-Wesley), 2011.  
I. Sommerville, *Software Engineering: (Update) (8th edition)*, Addison Wesley, 8th edition, 2007.  
Roger S. Pressman, *Engenharia de Software*, 6a. edição, 2006, McGraw-Hill, São Paulo.  
P. Jalote, *An Integrated Approach to Software Engineering*, Springer, 2nd edition, 1997.  
Ghezzi, C., Jazayeri, M. & Mandrioli, D., *Fundamentals of Software Engineering*, Prentice Hall, 1991.  
E.J. Braude, *Software Engineering: An OO Perspective*, John Wiley & Sons, INC., 2001.  
W.P. Paula Filho, *Engenharia de Software: Fundamentos, Métodos e Padrões*, Segunda Edição, Editora LTC, 2001.

##### Modelagem e Projeto Orientados a Objetos

Rumbaugh, J. et al., *Object-Oriented Modeling and Design*, Prentice Hall, 1991.  
Booch, G., *Object-Oriented Design with Applications*, Benjamin-Cummings, 1991.  
G. Booch, J. Rumbaugh & I. Jacobson, *The Unified Modeling Language User Guide*, Addison Wesley, 1999.  
H. Eriksson & M. Penker, *UML Toolkit*, Wiley, 1998.  
P. Muller, *Instant UML*, Wrox, 1997.  
C. Larman, *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process*, Second Edition, Prentice-Hall, 2002.  
P. Harmon & M. Watson, *Understanding UML*, Morgan Kaufmann, 1998.  
C. Richter, *Designing Flexible OO Systems with UML*, MTP, 2001.  
M. Page-Jones, *Fundamentals of OO Design in UML*, Addison-Wesley, 2000.  
P. Stevens, *Using UML: software engineering with objects and components*, Addison-Wesley, 1999.  
Y. Lau, *The Art of Objects: OO design and architecture*, Addison-Wesley, 2001.  
A. Carvalho & T. Chioffi, *Introdução à Engenharia de Software*, Editora da UNICAMP, 2001.  
C.M.F. Rubira, *Apostila Introdução à Análise Orientada a Objetos*, IC-UNICAMP, 2003.

##### Metodologias OO

I. Jacobson, G. Booch & J. Rumbaugh, *The Unified Software Development Process*, Addison Wesley, 1999.  
Krüchten, P., *The Rational Unified Process: An Introduction*, Second Edition, Addison-Wesley, 2000.  
J. Cheesman & J. Daniels, *UML Components: A Simple Process for Specifying Component-Based Software*, Addison-Wesley, 2001.  
D. D'Souza & A. Wills, *Objects, Components and frameworks with UML: The Catalysis Approach*, Addison-Wesley, 1999.  
S.R. Palmer & J.M. Felsing, *A Practical Guide to Feature-driven development*, The Coad Series.

##### Linguagens de Programação OO

- K. Arnold & J. Gosling, *The Java Programming Language*, second edition, Addison-Wesley, 1997.  
 Flanagan, D. *Java in a Nutshell*, O'Reilly & Associates, 1996.  
 K. Beck, *Extreme Programming: embrace change*, Addison-Wesley, 2000.

#### Casos de Uso

- G. Schneider & J.P. Winters *Applying Use Cases: A Practical Guide*, Second Edition, Addison-Wesley, 2001.  
 A. Cockburn *Writing Effective Use Cases*, Addison-Wesley, 2001.  
 D. Kulak & E. Guiney *Use Cases: Requirements in Context*, Addison-Wesley, 2000.  
 I. Jacobson, *OO Software Engineering: A Use Case Driven Approach*, Addison-Wesley, 1992.

#### Design Patterns

- M. Fowler, *Analysis Patterns: Reusable Object Models*, Addison-Wesley, 1997.  
 M. Grand, *Patterns in Java*, Wiley, 1998.  
 E. Gamma et al., *Design Patterns: Elements of reusable OO Software*, Addison-Wesley, 1995.  
 Pattern Languages of Program Design, 1, 2, 3 e 4, Software Patterns Series.

#### Frameworks Orientados a Objetos

- W. Pree, *Design Patterns for OO Software Development*, Addison-Wesley, 1995.  
 M. Fayad & R. E. Johnson, *Domain-Specific Application Frameworks*, Wiley, 2000.  
 M. Fayad, D. C. Schmidt & R.E. Johnson, *Implementing Application Frameworks*, Wiley, 1999.  
 M. Fayad, D. C. Schmidt & R.E. Johnson, *Building Application Frameworks*, Wiley, 1999.

#### Arquitetura de Software

- F. Buschmann et al., *A System of Patterns: Pattern-Oriented Software Architecture*, Wiley, 1996.  
 L. Bass, P. Clements & R. Kazman, *Software Architecture in Practice*, Second Edition, Addison-Wesley, 2003, SEI Series in Software Engineering.  
 C. Hofmeister et al., *Applied Software Architecture*, Addison-Wesley, 2000.  
 M. Shaw & D. Garlan. *Software Architecture: Perspectives on an emerging discipline*, Prentice Hall, 1996.  
 J. Bosch, *Design and use of software architecture*, Addison-Wesley, 2000.  
 L. Barroca et al. *Software Architectures: advances and applications*, Springer, 2000.  
 L. Hohmann, *Beyond Software Architecture: creating and sustaining winning solutions*, Addison-Wesley, 2003.  
 P.C.Clements & L.Northrop. *Software Architecture: an executive overview*. technical report CMU/SEI-96-TR-003, Software Engineering Institute, Carnegie-Mellon University, February 1996.  
 F. Bachman et al., *The architecture-based design method*, technical report CMU/SEI-2000-TR-001, Software Engineering Institute, Carnegie-Mellon University, January 2000.  
 P. Krutchen. *The 4+1 view model of software architecture*. IEEE Software, pages 42–50, November 1995.  
 M. Barbacci et al., *Steps in architecture tradeoff analysis method: quality attribute models and analysis*. technical report CMU/SEI-97-TR-029, Software Engineering Institute, Carnegie-Mellon University, May 1998.

#### Desenvolvimento Baseado em Componentes

- C. Szyperski, *Component software: beyond OO programming*, Addison-Wesley, 1998.  
 G.T.Leavens & M.Sitaraman, *Foundations of Component-based Systems*, Cambridge University press, Cambridge, UK, 2000.  
 F. Bachman et al., *Volume II: Technical concepts of component-based software engineering*, technical report CMU/SEI-2000-TR-008, Software Engineering Institute, Carnegie-Mellon University, April 2000.  
 J. Sametinger, *Software Engineering with reusable components*, Springer, 1997.