



UNIVERSIDADE FEDERAL DE UBERLÂNDIA



## FICHA DE COMPONENTE CURRICULAR

<b>CÓDIGO:</b> FACOM33408	<b>COMPONENTE CURRICULAR:</b> Programação Orientada a Objetos II	
<b>UNIDADE ACADÊMICA OFERTANTE:</b> Faculdade de Computação		<b>SIGLA:</b> FACOM
<b>CH TOTAL TEÓRICA:</b> 30 horas	<b>CH TOTAL PRÁTICA:</b> 30 horas	<b>CH TOTAL:</b> 60 horas

1. **OBJETIVOS**

**Geral:** Capacitar o aluno na aplicação de técnicas avançadas de análise e projeto de software, enfatizando formas de melhorar o reuso através do paradigma de desenvolvimento orientado a objetos.

**Específicos:**

- Compreender conceitos mais avançados de linguagens de Programação Orientada a Objetos;
- Desenvolver software usando técnicas avançadas de análise e projeto Orientado a Objetos;
- Compreender alguns padrões típicos de análise, i.e., modelos conceituais de objetos reutilizáveis, de forma a entender o papel dos padrões na reutilização de colaborações entre classes e objetos;
- Compreender, analisar e decidir os Padrões de Projeto mais apropriados ao software a ser desenvolvido;
- Compreender e identificar anti-padrões;
- Projetar uma arquitetura de software usando padrões arquiteturais;
- Compreender, analisar e experimentar frameworks e como eles permitem reutilizar a análise de problemas e o projeto de soluções, permitindo assim escrever aplicações relacionadas com eficácia para diversos domínios;
- Identificar e entender a aplicação dos princípios de projeto, padrões e arquiteturas nas implementações dos frameworks estudados.

2. **EMENTA**

Tópicos Avançados em Programação Orientada a Objetos; Análise Orientada a Objetos e Princípios de Projeto; Padrões de Projeto e Arquitetura; Estudo de frameworks e suas aplicações.

3. **PROGRAMA**

1. Tópicos Avançados em Programação Orientada a Objetos:
  1. Classes e métodos genéricos (ou parametrizados);

2. Comparação, ordenação e operações com objetos genéricos;
  3. Reuso através de bibliotecas com estruturas de dados parametrizadas: listas, conjuntos, dicionários, árvores;
  4. Programação de classes e métodos genéricos;
  5. Expressões e funções Lambda;
  6. Multithreading;
  7. Anotações.
2. Análise Orientada a Objetos e Princípios de Projeto:
    1. Padrões de análise: Party, Organization Hierarchy, Accountability, Knowledge Level, Quantity, Range, Temporal Patterns, Accounting Patterns;
    2. Princípios de projeto de classes (SOLID): Single responsibility principle (SRP); Open-closed principle (OCP); Liskov substitution principle (LSP); Interface Segregation Principle (ISP); Dependency Inversion Principle (DIP);
    3. Princípios de coesão e acoplamento de pacotes: Reuse/Release Equivalency Principle (REP); Common Closure Principle (CCP); Common Reuse Principle (CRP); Acyclic Dependencies principle (ADP); Stable Dependencies Principle (SDP); Stable Abstraction Principle (SAP).
    4. Mapeamento de elementos em Modelos de Domínio (espaço do problema) para elementos de Projeto (espaço da solução).
  3. Padrões de Projeto e Arquitetura:
    1. Padrões de projeto clássicos (GoF) e anti-padrões;
    2. Padrões arquiteturais clássicos: Camadas, Microkernel, MVC, Blackboard, Broker;
    3. Padrões de arquiteturas corporativas: Hexagonal (portas e adaptadores), Orientada a Serviços, Micro-serviços, REST, CQRS, Orientada a Eventos; Fábrica de Dados e Computação Distribuída baseada em Grid;
    4. Aplicações e evolução no uso de padrões em softwares modernos.
  4. Estudo de frameworks e suas aplicações:
    1. Interfaces Gráficas;
    2. Persistência e Mapeamento Objeto-Relacional;
    3. Relatórios e Gráficos;
    4. Testes.

#### 4. BIBLIOGRAFIA BÁSICA

FREEMAN, Eric; FREEMAN, Elisabeth; BATES, Bert. **Use a cabeça!:** padrões de projeto. 2. ed. Rio de Janeiro: Atlas Books, 2009.

MARTIN, Robert C.; MARTIN, Micah. **Princípios, padrões e práticas ágeis em C#.** Porto Alegre: Bookman, 2011. *E-book.* Disponível em: <https://www.sistemas.ufu.br/biblioteca-gateway/minhabiblioteca/9788577808427>. Acesso em: 22 jul. 2022.

MCLAUGHLIN, Brett; POLLICE, Gary; WEST, David. **Use a cabeça!:** análise e projeto orientado ao objeto. Rio de Janeiro: Alta Books, 2007.

#### 5. BIBLIOGRAFIA COMPLEMENTAR

BLAHA, Michael; RUMBAUGH, James. **Modelagem e projetos baseados em objetos com UML 2**. 2. ed. Rio de Janeiro: Campus, 2006

BOOCH, Grady *et al.* **Object-oriented analysis and design with applications**. 3. ed. Upper Saddle River: Addison-Wesley, 2007.

EVANS, Eric. **Domain-driven design: atacando as complexidades no coração do software**. 2. ed. Rio de Janeiro: Alta Books, 2010.

FOWLER, Martin. **Padrões de arquitetura de aplicações corporativas**. Porto Alegre: Bookman, 2006. *E-book*. Disponível em: <https://www.sistemas.ufu.br/biblioteca-gateway/minhabiblioteca/9788577800643>. Acesso em: 11 jul. 2022.

GAMMA, Erich *et al.* **Design patterns: elements of reusable object-oriented software**. Reading: Addison-Wesley, 1995.

## 6. APROVAÇÃO

Prof. Dr. Rafael Dias Araújo  
Coordenador do Curso de Sistemas de Informação  
Campus Monte Carmelo

Prof. Dr. Mauricio Cunha Escarpinati  
Diretor da Faculdade de Computação



Documento assinado eletronicamente por **Rafael Dias Araújo, Coordenador(a)**, em 25/01/2023, às 22:13, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Mauricio Cunha Escarpinati, Diretor(a)**, em 26/01/2023, às 17:18, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [https://www.sei.ufu.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **4053294** e o código CRC **206E2FAD**.