



UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Faculdade de Computação

Av. João Naves de Ávila, nº 2121, Bloco 1A - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902

Telefone: (34) 3239-4144 - <http://www.portal.facom.ufu.br/> facom@ufu.br



PLANO DE ENSINO

1. IDENTIFICAÇÃO

Componente Curricular:	PROGRAMAÇÃO ORIENTADA A OBJETOS 2						
Unidade Ofertante:	FACOM						
Código:	GBC055	Período/Série:	5	Turma:			
Carga Horária:			Natureza:				
Teórica:	36	Prática:	36	Total:	72	Obrigatória: (X)	Optativa: ()
Professor(A):	Fabiano Azevedo Dorça				Ano/Semestre:	2021-2	
Observações:							

2. EMENTA

Conceitos avançados de programação orientada a objetos, polimorfismo e ligação dinâmica. Reuso de software, frameworks de aplicação e tecnologia de componentes. Padrões de projeto e de arquitetura, bad smells e refatoração de software. API avançadas: interfaces gráficas, concorrência e distribuição.

3. JUSTIFICATIVA

Disciplina de fundamental relevância a formação do profissional considerando-se o perfil do egresso desejado constante no PPC do curso. A disciplina provê conceitos indispensáveis à formação das capacidades avançadas de projeto e desenvolvimento de sistemas de informação complexos considerando-se conceitos de qualidade de software tais como manutenibilidade e reusabilidade.

4. OBJETIVO

Objetivo Geral:

Propiciar ao aluno formação das capacidades avançadas de projeto e desenvolvimento de sistemas de informação complexos considerando-se conceitos de qualidade de software tais como manutenibilidade e reusabilidade.

Objetivos Específicos:

Capacitar o aluno a:

1. Implementar sistemas complexos usando programação orientada a objetos
2. Reconhecer problemas e aplicar padrões de projeto específicos na solução destes problemas
3. Refatorar sistemas orientados a objetos de acordo com a necessidade
4. Usar bibliotecas avançadas no desenvolvimento de sistemas orientados a objetos

5. PROGRAMA

1. Revisão dos conceitos fundamentais de programação orientada a objetos

1.1. Classes, objetos, atributos e métodos

1.2. Herança, polimorfismo e ligação dinâmica

2. Introdução a reuso e frameworks de aplicação

3. Idiomas e padrões de programação orientada a objetos

3.1. Padrões de projeto (GoF)

- 3.2. Padrões de arquitetura (MVC e Broker)
- 4. Refatoração de programas orientados a objetos
 - 4.1. Introdução a bad smells
 - 4.2. Refatorações de programas (Fowler)
- 5. Elementos avançados de programação orientada a objetos
 - 5.1. Interfaces gráficas (Tree, Drag'n'drop, etc.)
 - 5.2. Programação concorrente (Thread)
 - 5.3. Programação distribuída (RMI e Sockets)
- 6. Desenvolvimento de um sistema orientado a objetos
 - 6.1. Implementação do sistema usando padrões de projeto e de arquitetura
 - 6.2. Documentação do sistema desenvolvido

6. **METODOLOGIA**

Aulas expositivas as quintas-feiras das 08:50 as 10:40 (2 horas/aula)

Aulas práticas as quintas-feiras das 10:40 as 12:20 (2 horas/aula)

Detalhamento do cronograma de atividades de ensino:

Aula 01 - Teórica

Revisão dos conceitos fundamentais de programação orientada a objetos 1.1. Classes, objetos, atributos e métodos

Aula 02 - Prática

Desenvolvimento da 1a. iteração do estudo de caso prático

Aula 03 - Teórica

Síncrona 1.2. Herança, polimorfismo e ligação dinâmica

Aula 04 - Prática

Desenvolvimento da 2a. iteração do estudo de caso prático

Aula 05 - Teórica

2. Introdução a reuso e frameworks de aplicação

Aula 06 - Prática

Desenvolvimento da 3a. iteração do estudo de caso prático

Aula 07 - Teórica

3. Idiomas e padrões de programação orientada a objetos

3.1. Padrões de projeto (GoF)

Aula 08 - Prática

Desenvolvimento da 4a. iteração do estudo de caso prático

Aula 09 - Teórica

3.1. Padrões de projeto (GoF)

Aula 10 - Prática

Desenvolvimento da 5a. iteração do estudo de caso prático

Aula 11 - Teórica

3.1. Padrões de projeto (GoF)

Aula 12 - Atividade assíncrona Desenvolvimento da 6a. iteração do estudo de caso prático

Aula 13 - Teórica

3.1. Padrões de projeto (GoF)

Aula 14 - Prática

Desenvolvimento da 7a. iteração do estudo de caso prático

Aula 15 - Teórica

3.1. Padrões de projeto (GoF)

Aula 16 - Prática

Desenvolvimento da 8a. iteração do estudo de caso prático

Aula 17 - Teórica

3.1. Padrões de projeto (GoF)

Aula 18 - Prática

1a. Avaliação Individual

Aula 19 - Teórica

3.2. Padrões de arquitetura (MVC e Broker)

Aula 20 - Prática

Desenvolvimento da 9a. iteração do estudo de caso prático

Aula 21 - Teórica

4. Refatoração de programas orientados a objetos

4.1. Introdução a bad smells

Aula 22 - Prática

Desenvolvimento da 10a. iteração do estudo de caso prático

Aula 23 - Teórica

4.1. Introdução a bad smells

Aula 24 - Prática

Desenvolvimento da 11a. iteração do estudo de caso prático

Aula 25 - Teórica

4.2. Refatorações de programas (Fowler)

Aula 26 - Prática

Desenvolvimento da 12a. iteração do estudo de caso prático

Aula 27 - Teórica

4.2. Refatorações de programas (Fowler)

Aula 28 - Prática

Desenvolvimento da 13a. iteração do estudo de caso prático

Aula 29 - Teórica

5. Elementos avançados de programação orientada a objetos

5.1. Interfaces gráficas (Tree, Drag'n'drop, etc.)

Aula 30 - Prática

Desenvolvimento da 14a. iteração do estudo de caso prático

Aula 31 - Teórica

5.2. Programação concorrente (Thread)

Aula 32 - Prática

Desenvolvimento da 15a. iteração do estudo de caso prático

Aula 33 - Teórica

5.3. Programação distribuída (RMI e Sockets)

Aula 34 - Prática

Desenvolvimento da 16a. iteração do estudo de caso prático

Aula 35 - Teórica

6. Desenvolvimento de um sistema orientado a objetos

6.1. Implementação do sistema usando padrões de projeto e de arquitetura

6.2. Documentação do sistema desenvolvido

Aula 36 - Prática

2a. Avaliação Individual

Entrega final do projeto prático

7. AVALIAÇÃO

- Avaliações individuais sem consulta (02 avaliações, conforme indicado no cronograma) - 50 pontos

- Atividades práticas semanais em grupo (conforme indicado no cronograma) - 50 pontos

- Recuperação de nota: Avaliação individual de 25 pontos substitutiva à avaliação de menor nota obtida pelo discente.

8. BIBLIOGRAFIA

Básica

BUSCHMAN, F.; MEUNIER, R.; ROHNERT, H.; SOMMERLAND, P.; STAL, M. Pattern-Oriented Software Architecture: A System of Patterns, John Wiley & Sons, 1996.

FOWLER, M. Refactoring: Improving the Design of Existing Code, Addison-Wesley, 2000.

GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995

Complementar

BOOCH, G. Object-Oriented Analysis and Design with Applications, 3a Edição. Addison-Wesley, 2007.

BOOCH, G., RUMBAUGH, J., JACOBSON, I. UML, Guia do Usuário. Rio de Janeiro: Campus, 2000.

DEITEL, H. M.; DEITEL P. J. Java: Como Programar, 6a. Edição. Pearson, 2005.

ECKEL B. Thinking in Java, 2a Edição. Prentice Hall, 2000.

HEMRAJANI, A. Desenvolvimento Ágil em Java com Spring, Hibernate e Eclipse. Prentice Hall, 2007.

LARMAN, C. Utilizando UML e Padrões: Uma Introdução à Análise e ao Projeto Orientado a Objetos. Porto Alegre: Bookmann, 2001.

RUMBAUGH, J.; BLAHA, M. Modelagem e Projetos Baseados em Objetos com UML 2, 1a Edição. Editora Campus, 2006.

Alternativa (bibliografia alternativa disponível de forma online e aberta)

Padrões de Projeto. Disponível em: <https://padroesdeprojeto.com.br/>

Padrões de Projeto. Disponível em <https://www.devmedia.com.br/conheca-os-padroes-de-projeto/957>

Design Patterns. Disponível em: <https://refactoring.guru/pt-br/design-patterns>

Princípios de Projeto. Disponível em: <https://engsoftmoderna.info/cap5.html>

9. **APROVAÇÃO**

Aprovado em reunião do Colegiado realizada em: ____/____/____

Coordenação do Curso de Graduação: _____