



**PLANO DE ENSINO**

**1. IDENTIFICAÇÃO**

<b>COMPONENTE CURRICULAR:</b> PROGRAMAÇÃO ORIENTADA A OBJETOS				
<b>UNIDADE OFERTANTE:</b> FACULDADE DE COMPUTAÇÃO				
<b>CÓDIGO:</b> FACOM39502		<b>PERÍODO/SÉRIE:</b> 5º		<b>TURMA:</b> I
<b>CARGA HORÁRIA</b>			<b>NATUREZA</b>	
<b>TEÓRICA:</b> 36	<b>PRÁTICA:</b> 36	<b>TOTAL:</b> 72	<b>OBRIGATÓRIA: ( x )</b>	<b>OPTATIVA: ( )</b>
<b>PROFESSOR(A):</b> DANIEL ANTÔNIO FURTADO			<b>ANO/SEMESTRE:</b> 2025/2	
<b>OBSERVAÇÕES:</b>				

**2. EMENTA**

Introdução dos conceitos fundamentais de programação orientada a objetos, como classes, objetos, encapsulamento, herança e polimorfismo. Conhecimento dos membros que tipicamente compõem classes: construtores, destrutores, variáveis e métodos. Entendimento e aplicação dos conceitos de orientação a objetos em linguagens de programação que suportem tal paradigma. Desenvolvimento de sistemas usando programação orientada a objetos.

**3. JUSTIFICATIVA**

Através de conceitos como classes, objetos e encapsulamento, o futuro gestor aprende a enxergar a informação não como dados isolados, mas como componentes coesos que possuem tanto atributos (características) quanto comportamentos (operações), refletindo a complexidade dos ambientes organizacionais. Essa habilidade é crucial para a especificação de requisitos de sistemas, a modelagem de bancos de dados e a compreensão da arquitetura de software, permitindo que o profissional dialogue de forma eficaz com equipes de desenvolvimento. Mais do que apenas uma técnica de programação, a POO oferece um paradigma para a resolução de problemas, capacitando o aluno a projetar soluções de informação que são modulares, escaláveis e diretamente alinhadas às necessidades do negócio.

**4. OBJETIVO**

**Objetivo Geral:**

Capacitar o aluno a analisar problemas computacionais e propor soluções utilizando conceitos de programação orientada a objetos como classes, objetos, herança e polimorfismo.

**Objetivos Específicos:**



Introduzir o paradigma de programação orientada a objetos comparando-o com o paradigma de programação procedural; apresentar os benefícios da programação orientada a objetos no contexto de aplicações de software no âmbito da gestão da informação; apresentar o conceito de herança no contexto da programação orientada a objetos e sua relação com o conceito de generalização e especialização, assim como interfaces e polimorfismo; capacitar o aluno a utilizar bibliotecas no desenvolvimento de sistemas de software utilizando programação orientada a objetos.

## **5. PROGRAMA**

1. Introdução à Programação Orientada a Objetos
  - 1.1. Histórico da programação orientada a objetos.
  - 1.2. Programação procedimental versus programação orientada a objetos.
2. Conceitos Fundamentais de Programação Orientada a Objetos
  - 2.1. Classes concretas, classes abstratas e interfaces.
  - 2.2. Objetos, atributos, métodos, construtores e destrutores.
  - 2.3. Membros de classe e membros de objetos.
  - 2.4. Métodos concretos e métodos abstratos.
  - 2.5. Pacotes, visibilidade e encapsulamento.
  - 2.6. Generalização, especialização e herança.
  - 2.7. Polimorfismo.
  - 2.8. Ligação estática e ligação dinâmica.
3. Uso de Bibliotecas em Linguagens Orientadas a Objetos
  - 3.1. Tratamento de exceções.
  - 3.2. Uso de API básica.
  - 3.3. Manipulação de arquivos.
  - 3.4. Interfaces gráficas simples.
  - 3.5. Acessos ao banco de dados.
4. Desenvolvimento de um Sistema Orientado a Objetos
  - 4.1. Implementação usando programação orientada a objetos.
  - 4.2. Documentação do sistema desenvolvido.

## **6. METODOLOGIA**

O conteúdo será consolidado de forma gradativa e integrada, de modo que o aluno possa aplicar os conhecimentos adquiridos anteriormente com os atuais.

As aulas teóricas serão ministradas em sala de aula, utilizando projetor multimídia, nas segundas-feiras das 14h50 às 16h30. As aulas práticas serão ministradas em laboratório de informática nas terças-feiras das 14h50 às 16h30. O controle de frequência será realizado por meio de chamada oral, que poderá ocorrer a qualquer momento durante o período de aula.

A complementação de carga horária será realizada por meio da aplicação de trabalhos práticos, que serão disponibilizados na forma de roteiros no website do professor ([www.furtado.prof.ufu.br](http://www.furtado.prof.ufu.br)). Tais atividades estão previstas no cronograma detalhado a seguir.



SERVIÇO PÚBLICO FEDERAL  
MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DE UBERLÂNDIA



Aula	Sem.	Data	Conteúdo/Descrição	Carga Hor. (hora-aula)
1	2ª	20/10/2025	Apresentação da disciplina. Revisão de programação com ling. O.O.	2
2	2ª	21/10/2025	Exercícios de revisão de programação.	2
3	3ª	28/10/2025	Fundamentos de programação utilizando linguagem OO.	2
4	3ª	--	Ativ. Complementar - exercícios de programação com linguagem OO.	2
5	4ª	03/11/2025	Introdução à POO, classes, objetos, membros de instância.	2
6	4ª	04/11/2025	Classes, objetos, membros de instância – exercícios.	2
7	5ª	10/11/2025	Construtores, modificadores de acesso, encapsulamento.	2
8	5ª	11/11/2025	Construtores, modificadores de acesso, encapsulamento – exercícios.	2
9	5ª	--	Ativ. Complementar – continuação de atividade prática.	2
10	6ª	17/11/2025	Propriedades, membros estáticos, agregação.	2
11	6ª	18/11/2025	Propriedades, membros estáticos, agregação – exercícios.	2
12	7ª	24/11/2025	Referências, garbage collector, destrutores.	2
13	7ª	25/11/2025	Referências, garbage collector, destrutores – exercícios.	2
14	8ª	01/12/2025	Pacotes, namespaces, APIs, bibliotecas.	2
15	8ª	02/12/2025	Pacotes, namespaces, APIs, bibliotecas – exercícios.	2
16	9ª	08/12/2025	Tratamento de exceções.	2
17	9ª	09/12/2025	Tratamento de exceções – exercícios.	2
18	9ª	--	Ativ. Complementar - manipulação de arquivos.	2
19	9ª	--	Ativ. Complementar - manipulação de arquivos.	2
20	10ª	15/12/2025	<b>1ª Avaliação (40 pontos)</b>	2
21	10ª	16/12/2025	Vista de atividades.	2
22	11ª	02/02/2026	Generalização, especialização e herança.	2
23	11ª	03/02/2026	Generalização, especialização e herança – exercícios.	2
24	12ª	09/02/2026	Classes e membros abstratos, polimorfismo, lig. dinâmica.	2
25	12ª	10/02/2026	Classes e membros abstratos, polimorfismo, lig. dinâmica – exerc.	2
26	13ª	23/02/2026	Herança vs interfaces (não gráfica).	2
27	13ª	24/02/2026	Herança vs interfaces (não gráfica) – exercícios.	2
28	14ª	02/03/2026	Interface gráfica simples.	2
29	14ª	03/03/2026	Inteface gráfica simples – exercícios.	2
30	14ª	--	Ativ. Complementar – desenvolvimento de sistema O.O.	2
31	14ª	--	Ativ. Complementar – desenvolvimento de sistema O.O.	2
32	14ª	--	Ativ. Complementar – desenvolvimento de sistema O.O.	2
33	15ª	09/03/2026	<b>2ª Avaliação (40 pontos)</b>	2
34	15ª	10/03/2026	Vista de atividades.	2
35	16ª	16/03/2026	<b>Avaliação de Recuperação</b>	2
36	16ª	17/03/2026	Vista final de atividades	2
<b>Carga Horária Total</b>				<b>72</b>



### Atendimento e Comunicação com os Discentes

A comunicação com a turma será realizada pelo Microsoft Teams. Durante a primeira semana de aula será enviado para o e-mail do aluno (@ufu.br) um link para entrada na equipe da turma no Microsoft Teams.

O atendimento aos alunos ocorrerá preferencialmente por chat de mensagens e/ou chamadas de áudio no ambiente do Microsoft Teams nas quintas e sextas-feiras das 9h às 11h. O atendimento também poderá ocorrer presencialmente em horário agendado com o professor.

Slides de aula, exercícios e trabalhos serão disponibilizados no website do professor no endereço:  
[www.furtado.prof.ufu.br](http://www.furtado.prof.ufu.br)

## 7. AVALIAÇÃO

O sistema de avaliação é organizado da seguinte forma:

- **40 pontos** – 1ª Avaliação teórica individual - 15/12/2025.
- **40 pontos** – 2ª Avaliação teórica individual – 09/03/2026.
- **20 pontos** - Trabalhos semanais e desenvolvimento de sistema orientado a objetos.

### Avaliação de Recuperação

Para os alunos que não atingirem 60 pontos e que tenham frequência mínima de 75%, será aplicada uma avaliação de recuperação sobre todo o conteúdo no valor de 80 pontos, a qual substituirá as duas primeiras avaliações semestrais, mas a nota final ficará limitada a 60 pontos.

Data da avaliação de recuperação: **16/03/2026**.

## 8. BIBLIOGRAFIA E MATERIAL DE APOIO

### Bibliografia Básica

1. BARNES, D. J. Programação orientada a objetos com Java: uma introdução prática usando o BlueJ. São Paulo: Prentice Hall, 2009.
2. CARDOSO, C. Orientação a objetos na prática: aprendendo orientação a objetos com Java. Rio de Janeiro: Ciência Moderna, 2006.
3. DEITEL, H. M.; DEITEL P. J. Java: como programar. 6. ed. São Paulo: Prentice Hall, 2010.

### Bibliografia Complementar

1. BOOCH, G., RUMBAUGH, J., JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Campus, 2000.
2. HORSTMANN, C. S. Padrões e projeto orientados a objetos. Porto Alegre: Bookman, 2007.
3. GOSLING, J. The java application programming interface. Reading: Addison-Wesley, 1996.
4. SEDGEWICK, R. Introduction to programming in Java: an interdisciplinary approach. Boston: Addison-Wesley, 2008.
5. SIERRA, K. Use a cabeça!: Java. Rio de Janeiro: Alta Books, 2005.

### Material de Apoio Online

Materiais de apoio complementares e materiais de aula como slides, exercícios e trabalhos, serão disponibilizados no website do professor no endereço [www.furtado.prof.ufu.br](http://www.furtado.prof.ufu.br)



**SERVIÇO PÚBLICO FEDERAL  
MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DE UBERLÂNDIA**



**9. APROVAÇÃO**

Aprovado em reunião do Colegiado realizada em: \_\_\_\_/\_\_\_\_/\_\_\_\_

Coordenação do Curso de Graduação em: \_\_\_\_\_