



DIRETORIA ACADÊMICA

PLANO DE ENSINO

Curso: Análise e Desenvolvimento de Sistemas		
Unidade Curricular: Programação Orientada a Objetos		Carga Horária: 80h
Modalidade: Presencial (X)	Semipresencial ()	EAD ()
Período: 3º		Ano/ Semestre Letivo: 2026.1
Professor (es): Walter Travassos Sarinho		

1. EMENTA

Introdução à orientação a objetos. Linguagens típicas orientadas a objetos. Programação Orientada a Objeto em Java. Conceitos básicos e terminologias de Programação orientada a objetos. Classe, Objetos, Atributos, Métodos, Construtores e sobrecarga. Instanciação e Referência de objetos. Envio de mensagens. Ciclo de vida de um objeto. Abstração e encapsulamento. Herança. Criação e uso de hierarquia de classes. Classes abstratas e Interfaces. Relacionamento entre classes. Polimorfismo. Ligação dinâmica (dynamic binding). Tratamento de exceções.

2. OBJETIVO (S)

2.1. Geral:

- Capacitar os alunos a compreender e aplicar os conceitos fundamentais relacionados à orientação a objetos em Java, visando o desenvolvimento de sistemas e aplicações de software estruturados, modularizados e reutilizáveis.

2.2. Específicos:

- Realizar análise e entender os conceitos de orientação a objetos.
- Analisar um problema do mundo real e projetar a solução utilizando Orientação a Objetos.
- Entender a UML necessária para desenvolver sistemas utilizando a tecnologia Java.
- Entender o conceito de abstração de dados, modelagem de sistemas utilizando-se classe/objeto, encapsulamento, herança, composição, polimorfismo, tratamento de erros e arquivo, aplicando estes conceitos em situações práticas com uma linguagem de programação orientada a objeto.
- Construir interfaces utilizando bibliotecas gráficas.
- Compreender os conceitos e aplicações do paradigma da Programação Orientada a Objetos, por meio do desenvolvimento de programas.

3. CONTEÚDO PROGRAMÁTICO									
U	Conteúdos	Conhecimentos	Habilidades	Atitudes	Nº Horas /Aulas				
					T	P	L	SP	EAD
I	Introdução à Programação em Java - Apresentação da disciplina e conceitos iniciais. - Histórico e evolução da linguagem Java. - Arquitetura da tecnologia Java. - Características da linguagem. - Estado da arte em ambientes de desenvolvimento e execução. - Escrevendo, compilando e executando aplicações Java. - Entrada padrão de dados. - Saída padrão de dados. - Tipos de dados e variáveis em Java. - Métodos de conversão entre tipos de dados. - Operadores básicos em Java. - Classes para manipulação de datas (Date, Calendar e DateTime). - Estruturas condicionais (if, else, switch). - Estruturas de repetição (for, while, do-while). - Classe String e operações de manipulação de Strings.	- Compreensão dos conceitos iniciais da linguagem Java, incluindo sua história, arquitetura, características e evolução. - Conhecimento sobre como escrever, compilar e executar aplicações Java básicas. - Conhecimento dos tipos de dados, variáveis e operadores básicos em Java, além da capacidade de realizar conversões entre tipos de dados. - Compreensão das estruturas condicionais (if, else, switch) e de repetição (for, while, do-while) em Java. - Conhecimento sobre como manipular Strings e realizar operações com datas utilizando as classes disponíveis em Java. - Conhecimento sobre como realizar entrada e saída padrão de dados em Java, além de trabalhar com dados de entrada via GUI.	- Capacidade de escrever, compilar e executar código Java de forma eficiente e correta. - Habilidade para utilizar operadores básicos em Java para realizar operações matemáticas e lógicas. - Capacidade de implementar estruturas condicionais (if, else, switch) e de repetição (for, while, do-while) de acordo com a lógica do programa. - Habilidade para manipular Strings, realizar concatenação, substituição e outras operações de manipulação de texto. - Capacidade de utilizar as classes Date, Calendar e DateTime para manipular e trabalhar com datas e horas em Java.	- Demonstração de interesse e curiosidade em aprender sobre a linguagem Java e suas funcionalidades. - Capacidade de organizar e estruturar o código de forma clara e organizada, seguindo boas práticas de programação. - Disposição para resolver problemas e desafios, buscando soluções e aprimorando o conhecimento. - Atitude colaborativa em trabalhos em grupo, compartilhando conhecimentos e experiências com os colegas. - Capacidade de analisar criticamente o próprio código, identificar possíveis melhorias e corrigir eventuais erros.	20	20			

II	Coleções em Java - Introdução a vetores e matrizes. - Estrutura de repetição for each. - Utilização do ArrayList e suas funcionalidades	- Entendimento dos conceitos básicos de coleções em Java, incluindo vetores, matrizes e ArrayList. - Conhecimento sobre como criar e manipular vetores e matrizes em Java, incluindo alocação de memória, acesso e manipulação dos elementos. - Compreensão da estrutura de repetição for-each em Java e sua aplicação na iteração de elementos de coleções. - Conhecimento sobre como utilizar o ArrayList em Java, incluindo adição, remoção, busca e ordenação de elementos.	- Habilidade para manipular e trabalhar com coleções em Java, incluindo inserção, remoção e busca de elementos. - Capacidade de utilizar vetores e matrizes de forma eficiente, otimizando o uso de memória e processamento. - Habilidade para implementar a estrutura de repetição for-each de forma correta e eficiente. - Capacidade de gerenciar e manipular coleções utilizando o ArrayList, aplicando suas funcionalidades de forma adequada.	- Postura organizada na manipulação de coleções, mantendo uma estrutura clara e legível no código. - Disposição para resolver desafios relacionados à manipulação de coleções, buscando soluções eficientes e corrigindo eventuais erros. - Atitude colaborativa ao trabalhar em equipe na manipulação de coleções, compartilhando conhecimentos e experiências. - Interesse em explorar novas funcionalidades e aprimorar as habilidades relacionadas à manipulação de coleções em Java.	4	4			
III	Programação Orientada a Objetos em Java - Conceitos fundamentais da POO (objetos, classes, métodos, atributos) - Métodos getters e setters. - Construção de métodos. - Métodos construtores e sobrecarga de métodos. - Princípios da POO: abstração, encapsulamento, herança, polimorfismo. - Classes abstratas e interfaces.	- Entendimento dos princípios básicos da Programação Orientada a Objetos, incluindo objetos, classes, métodos e atributos. - Conhecimento sobre a utilização de métodos getters e setters para acesso e modificação de atributos de uma classe. - Compreensão de como construir métodos em Java para realizar operações específicas. - Conhecimento sobre métodos construtores e sua utilização, além da capacidade	- Habilidade para implementar classes em Java, definindo atributos, métodos, construtores e aplicando encapsulamento. - Capacidade de criar e utilizar métodos getters e setters para garantir o encapsulamento dos atributos de uma classe. - Habilidade para criar e utilizar métodos em Java, aplicando boas práticas de programação. - Capacidade de implementar métodos construtores e aplicar sobrecarga de métodos de forma adequada. - Capacidade de aplicar os princípios da Programação	- Postura organizada na construção de classes e métodos, mantendo uma estrutura clara e legível no código. - Disposição para resolver desafios relacionados à implementação de classes e métodos em Java, buscando soluções eficientes e corrigindo eventuais erros. - Atitude colaborativa ao trabalhar em equipe na implementação de classes e métodos, compartilhando conhecimentos e experiências.	12	12			

		de aplicar sobrecarga de métodos. - Compreensão dos princípios fundamentais da Programação Orientada a Objetos, incluindo abstração, encapsulamento, herança e polimorfismo. - Conhecimento sobre classes abstratas e interfaces em Java, suas características e diferenças.	Orientada a Objetos (abstração, encapsulamento, herança, polimorfismo) na criação e organização de classes em Java. - Habilidade para criar e utilizar classes abstratas e interfaces, implementando métodos e definindo comportamentos específicos.	- Interesse em explorar novos conceitos e aprimorar as habilidades relacionadas à Programação Orientada a Objetos em Java.					
IV	Desenvolvimento de Interfaces Gráficas em Java - Introdução à construção de interfaces gráficas com Swing. - Layouts e componentes gráficos básicos. - Eventos e tratamento de eventos em interfaces gráficas.	- Compreensão dos conceitos básicos da construção de interfaces gráficas utilizando a biblioteca Swing em Java. - Conhecimento sobre os diferentes tipos de layouts (FlowLayout, BorderLayout, GridLayout, etc.) e sua aplicação na organização dos componentes gráficos. - Conhecimento sobre os principais componentes gráficos básicos, como botões, campos de texto, caixas de seleção, listas, entre outros. - Compreensão dos eventos gerados pelos componentes gráficos e sua manipulação no código Java.	- Habilidade para criar interfaces gráficas funcionais e esteticamente agradáveis utilizando a biblioteca Swing. - Capacidade de escolher e aplicar o layout mais adequado para cada situação em interfaces gráficas, garantindo uma organização eficiente dos componentes. - Habilidade para adicionar, configurar e manipular componentes gráficos básicos em uma interface Swing. - Capacidade de implementar tratamentos de eventos para responder às interações do usuário com os componentes gráficos.	- Postura organizada na criação e organização das interfaces gráficas, buscando uma apresentação visual clara e agradável. - Disposição para aprender e explorar novos recursos e técnicas na construção de interfaces gráficas em Java. - Atitude colaborativa ao trabalhar em equipe na criação e aprimoramento das interfaces gráficas, compartilhando conhecimentos e experiências. - Capacidade de pensar de forma criativa e inovadora na criação de interfaces gráficas, buscando soluções que atendam às necessidades dos usuários de forma eficiente.	4	4			
Subtotal da Carga Horária					40	40			
Total da Carga Horária					80				

Legenda: U - unidade T – quantidade de aula(s) teórica(s) por unidade(s); P- quantidade de aula(s) práticas(s) por unidade(s); L - quantidade de aula(s) por unidade(s) de Prática Pedagógica (exclusivo para o Curso de Educação Física - Licenciatura); SP - quantidade de aulas(s) semipresenciais; EAD - quantidade de aula(s) à distância.

4. METODOLOGIA

O componente curricular será ministrado de forma dinâmica, agregando estratégias de metodologias ativas: aulas expositivas e dialogadas; exibição de vídeos; discussão de casos; experiências e situações do mercado de trabalho, através da metodologia de situações-problema; projetos de software; atividades práticas; atividade integrada (Prática Integradora), seguindo o modelo proposto para o período letivo do curso.

5. RECURSOS DIDÁTICOS

Projetor multimídia, vídeos, documentários, quadro branco e pincel que subsidiarão as aulas expositivo-dialogadas, os debates, as discussões e a atividade integrada. Para as sessões de filmes e vídeos, projetor de mídia e notebook. Para as atividades de estudo e pesquisa, livros didáticos disponíveis na biblioteca ou disponibilizados pelo docente, bem como artigos científicos disponíveis nas Bases de Dados (EBSCO e Portal CAPES). Ainda, será utilizado o Ambiente Virtual de Aprendizagem (AVA) do SECTRAS *On Line*.

6. AVALIAÇÃO (ÕES)

A avaliação do discente será contínua e processual. Se dará mediante o somatório de diferentes atividades, que constituirão uma nota final para cada uma das três principais verificações de aprendizagem (uma por UNIDADE) previstas pela Instituição. A média final será obtida a partir dos resultados de cada Unidade. Estão previstas 3 avaliações (formativas e somativas), dispostas da seguinte forma:

- 1ª Unidade: avaliação escrita (10,00).
- 2ª Unidade: avaliação escrita (5,00) e projeto de software (5,00).
- 3ª Unidade: avaliação escrita (5,00), projeto de software (3,00) e Prática Integradora (2,00).

Além disso, haverá 6 avaliações diagnósticas para obter juízo de valor relativo ao rendimento do aluno sobre os temas e nas atividades práticas.

7. BIBLIOGRAFIA

7.1. Básica:

DEITEL, P; DEITEL, H. **Java: como programar**. 10. ed. São Paulo: Pearson, 2017. [acervo digital]

FELIX, R. **Programação Orientada a Objetos**. 1. ed. São Paulo: Pearson, 2016. [acervo digital]

PUGA, S; RISSETTI, G. **Lógica de programação e estrutura de dados, com aplicações em JAVA**. 3. ed. São Paulo: Pearson, 2016. [acervo digital]

7.2. Complementar:

ASCENCIO, AFG, ARAUJO, GS. **Estrutura de Dados: Algoritmos, análise da complexidade e implementações em Java e C/C++**. São Paulo: Pearson Education do Brasil, 2010. [acervo digital]

KÖLLING, M.; BARNES, D. J. **Programação orientada a objetos com Java: uma introdução prática usando o Bluej**. 4. ed. São Paulo, SP: Pearson, 2008. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 10 dez. 2024.

SINTES, Anthony. **Aprenda programação orientada a objetos em 21 dias**. 1. ed. São Paulo: Pearson, 2002. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 10 dez. 2024.

TURINI, Rodrigo. **Desbravando Java e orientação a objetos: um guia para o iniciante da linguagem**. São Paulo, SP: Casa do Código, 2014. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 10 dez. 2024.

KALINOVSKY, Alex. **Java secreto: técnicas de descompilação, patching e engenharia reversa**. São Paulo: Pearson, 2005. E-book. Disponível em: <https://plataforma.bvirtual.com.br>. Acesso em: 10 dez. 2024.