

## Pontifícia Universidade Católica do Paraná

### Plano de Ensino

Escola/ Campus:	Politécnica			
Cursos:	Tecnologia em Jogos Digitais Bacharelado em Jogos Digitais Cibersegurança Ciência da Computação Sistemas de Informação Engenharia de Software		Ano/Semestre: 2023/1	
Código/Nome da disciplina:	Programação Orientada a Objetos (POO)			
Carga Horária:	120 horas-aula (90 horas-relógio)			
Requisitos:	Não há			
Créditos:	6	Período: 3º	Turma: B	Turno: Manhã
Professor Responsável:	Edson Emilio Scalabrin			

### 1. Ementa

A disciplina, de natureza teórico-prática, é ofertada por todos os cursos da área de Computação da Escola Politécnica. O estudante aplica conceitos da orientação a objetos na resolução de problemas estruturados. Ao término da disciplina, é capaz de codificar produtos de software utilizando linguagem de programação orientada a objetos de forma sistematizada e aderente às especificações. Os estudantes trabalham em equipes para verificar os benefícios de orientação a objetos no desenvolvimento de sistemas de grande porte. Para frequentar a disciplina é desejável conhecimento em algoritmos.

## 2. Relação com disciplinas precedentes e posteriores

Esta disciplina é parte integrante do eixo de programação dos cursos. Não há pré-requisitos, mas é desejável conhecimento prévio de algoritmos e programação estruturada. A disciplina fornece a base conceitual e tecnológica necessária para a construção de programas orientados a objetos. Contribui com resultados de aprendizagem para as seguintes disciplinas:

<b>Curso</b>	<b>Disciplinas Precedentes</b>	<b>Disciplinas Posteriores</b>
Ciência da Computação	<ul style="list-style-type: none"><li>• Raciocínio Algorítmico (1º Período);</li><li>• Programação Imperativa (2º Período);</li><li>• Arquitetura de Banco de Dados (2º Período);</li></ul>	<ul style="list-style-type: none"><li>• Resolução de Problemas Estruturados em Computação (4º Período);</li><li>• Modelagem de Sistemas Computacionais (4º Período);</li><li>• Programação Distribuída (6º Período);</li><li>• Experiência Criativa: Projeto Transformador I (7º Período);</li><li>• Experiência Criativa: Projeto Transformador II (8º Período);</li></ul>
Bacharelado em Engenharia de Software (BES)	<ul style="list-style-type: none"><li>• Raciocínio Algorítmico (1º Período);</li><li>• Programação Web (2º Período);</li><li>• Banco de Dados (2º Período);</li></ul>	<ul style="list-style-type: none"><li>• Resolução de Problemas Estruturados em Computação (4º Período);</li><li>• Design de Software (4º Período);</li><li>• Verificação e Validação (4º Período);</li><li>• Desenvolvimento Orientado a Reuso (6º Período);</li><li>• Projeto Final I (7º Período) e II (8º Período).</li></ul>
Tecnologia em Jogos Digitais		
Cibersegurança		
Sistemas de Informação		
Engenharia de Software		

### 3. Temas de estudo

- TE1: Encapsulamento (classe, método e atributo)
- TE2: Objeto (instanciação, referência, interação, teste e coleção de objetos)
- TE3: Herança entre classes (generalização, especialização e polimorfismo)
- TE4: Classe abstrata e interface
- TE5: Tratamento de exceção
- TE6: Persistência de objeto

### 4. Resultados de Aprendizagem

Resultados de Aprendizagem	Temas de Estudo	Elemento de Competência
<b>Competência:</b> Implementar softwares para problemas estruturados de solução algorítmica, utilizando técnicas e ferramentas de desenvolvimento, de forma autônoma e cooperativa.		
<b>RA1:</b> Codificar programas baseados em interação entre objetos instanciados a partir de classes associadas, em conformidade com o princípio de encapsulamento, com a validação do seu comportamento, de forma sistematizada.	<b>TE1:</b> Encapsulamento (classe, método e atributo) <b>TE2:</b> Objeto (instanciação, referência, interação, teste e coleção de objetos)	Definir a sequência lógica a partir da análise das especificações de software, (Internaliza) Testar produtos de software em relação aos requisitos e às especificações de forma sistematizada, integrada e autorregulada (Internaliza)
<b>RA2:</b> Codificar programas baseados em objetos definidos por hierarquia de classes.	<b>TE3:</b> Herança entre classes (generalização, especialização e polimorfismo) <b>TE4:</b> Classe abstrata e interface <b>TE5:</b> Tratamento de exceção	Codificar produtos de software utilizando boas práticas de programação, conforme os requisitos e as especificações, de forma sistematizada e colaborativa. (Internaliza)
<b>RA3:</b> Codificar programas baseados em objetos que interajam com componentes de entrada e saída de dados, em conformidade com a especificação.	<b>TE6:</b> Persistência de objeto	Criar modelos de soluções computacionais inovadores, utilizando a linguagem mais apropriada ao contexto para dados estruturados e não estruturados. (Internaliza)

## 5. Mapa Mental



## 6. Metodologia e Avaliação

Tabela de Resultados de Aprendizagem (RA), correspondentes Indicadores de Desempenho (ID), métodos de ensino e aprendizagem e processos de avaliação.

Alinhamento Construtivo			
Resultado de aprendizagem	Indicadores de desempenho	Processos de Avaliação	Métodos ou técnicas empregados**
<b>RA1:</b> Codificar programas baseados em interação entre objetos instanciados a partir de classes associadas, em conformidade com o princípio de encapsulamento, com a validação do seu comportamento, de forma sistematizada.	<b>ID1.1:</b> Codifica classes, definindo seus membros (atributos e métodos), e aplicando proteção a seus membros (encapsulamento e sobreposição).	[Formativo] Exercícios  [Somativo] PBL, PjBL e prova	Sala de aula invertida  Feedback coletivo (em aula)  Feedback individual (em aula e remoto)  Interação: ambiente virtual de aprendizagem.
	<b>ID1.2:</b> Codifica métodos de classe, com parâmetros, referência a objeto, variáveis locais e invocação de outros métodos (estáticos ou não), validando seu comportamento por meio de testes unitários (invariante, pré-condição e pós-condição), de forma sistematizada.  <b>ID1.3:</b> Codifica coleção de objetos, utilizado agregação ou composição.		
<b>RA2:</b> Codificar programas baseados em objetos definidos por hierarquia de classes	<b>ID2.1:</b> Codifica modelos de hierarquia de classes para compartilhamento de atributos e métodos, de forma colaborativa.	[Formativo] Exercícios  [Somativo] PBL, PjBL e prova	Sala de aula invertida  Feedback coletivo (em aula)  Feedback individual (em aula e remoto)  Interação: ambiente virtual de aprendizagem.
	<b>ID2.2:</b> Codifica polimorfismo no contexto de hierarquia entre classes.  <b>ID2.3:</b> Codifica classes abstratas e interfaces, no contexto de hierarquia entre classes.  <b>ID2.4:</b> Aplica mecanismo de tratamento de exceção para tratamento de erro.		

<b>RA3:</b> Codificar programas baseados em objetos que interajam com componentes de entrada e saída de dados, em conformidade com a especificação.	<b>ID3.1:</b> Manipula arquivo para recuperar, alterar e armazenar dados.  <b>ID3.2:</b> Codifica persistência de objeto usando serialização.	[Formativo] Exercícios  [Somativo] PjBL	Sala de aula invertida  Feedback coletivo (em aula)  Feedback individual (em aula e remoto)  Interação: ambiente virtual de aprendizagem.
---	---	---	---

**PjBL:** Project-Based Learning == trabalho prático de longa duração, ao longo de 3 semanas.

**PjBL: Equipes de 2 a 4 estudantes**

#### Feedback

PjBL	Feedback no ato da defesa do trabalho realizado individualmente
Prova	Feedback por meio de devolutiva da prova

## Avaliação Somativa:

RA	Prova		PjBL (em equipe)
	I	II	
1	30%	0	0
2	0	30%	0
3	0	0	40%

**Critério de aprovação:** nota do semestre igual ou superior a 7 (sete) e frequência mínima de **75%**.

### Recuperação Parcial:

- A recuperação parcial dos RAs 1 e 2 ocorrerão durante o semestre letivo por meio de provas específicas, conforme o cronograma.
- É elegível para realizar a recuperação parcial o estudante que obteve nota menor que 7 no RA.
- A nota máxima do estudante no respectivo RA após a recuperação parcial será igual a 7,0 (sete). Assim, os estudantes que conseguirem uma nota entre 7,0 e 10,0 na recuperação terão a nota máxima igual a 7,0 (sete).

### Recuperação Final:

- A Semana de Recuperação Estendida permitirá recuperar os RAs 1, 2 e 3.
- Somente o estudante que obteve nota semestral maior ou igual a 4 e menor que 7 poderá fazer a recuperação da Semana de Recuperação Estendida.
- Somente os RAs com nota inferior a 7 poderão ser recuperados na Semana de Recuperação Estendida.
- A média semestral para o estudante que realizar avaliações na Semana de Recuperação Estendida é limitada a 7 (sete).

## 7. Cronograma de atividades

Período		Semana	RA	Atividades pedagógicas	Em aula / TDE	Carga horária da atividade
MARÇO	10/03	1	1	[Teoria & prática] Expressões sobre dados de tipos primitivos. [Teoria & prática] Comandos de desvio e repetição.	Em aula	6 h/a
	17/03	2	1	[Teoria & prática] Classes, encapsulamento e interação entre objetos <i>Git</i>	Em aula <b>TDE</b>	6 h/a 14 h/a
	24/03	3	1	[Teoria & prática] Coleção de Objetos.	Em aula	6 h/a
	31/03	4	1	[Exercícios] Revisão I – RA1	Em aula	4 h/a
				[Avaliação formativa] Teste I (simulado em dupla) – RA1	Em aula	2 h/a
ABRIL	07/04					
	14/04	5	1	[Avaliação Somativa] Prova I – RA1	Em aula	2 h/a
	21/04		2	[Teoria & prática] Hierarquia de Classes	Em aula	4 h/a
	28/04	6	2	[Teoria & prática] Polimorfismo	Em aula	6 h/a
MAIO	05/05	7	2	[Avaliação Somativa] Recuperação parcial RA1	Em aula	2 h/a
				[Teoria & prática] Classes abstratas, métodos abstratos e interface.	Em aula	4 h/a
	12/05	8	2	[Exercícios] Revisão II – RA2	Em aula	4 h/a
				[Avaliação formativa] Teste II (simulado em dupla) – RA2	Em aula	2 h/a
	19/05	9	2	[Avaliação Somativa] Prova II – R2	Em aula	4 h/a
				[Teoria & prática] Tratamento de Exceções.	Em aula	2 h/a
JUNHO	26/05	10	3	[Teoria & prática] Persistência de Objeto [Prática Equipe] PjBL – Desenvolvimento Interface Gráfica	Em aula Em aula <b>TDE</b>	2 h/a 4 h/a 14 h/a
	02/06	11	3	[Avaliação Somativa] Recuperação parcial RA2	Em aula	2 h/a
				[Prática Equipe] PjBL – Desenvolvimento	Em aula	4 h/a
				Armazenamento de dados em Arquivos	<b>TDE</b>	14 h/a
	09/06 (TDE)	12	3	[Exercícios] Revisão III – RA3	Em aula	2 h/a
				[Avaliação formativa] Teste III (simulado em dupla) – RA3	Em aula	2 h/a
				[Prática Equipe] PjBL – Desenvolvimento	Em aula	2 h/a



	16/06	13	3	<b>[Avaliação Somativa]</b> PjBL – Entrega de projeto final da disciplina)	Em aula	6 h/a
	23/06	14	3	<b>[Avaliação Somativa]</b> Recuperação parcial RA3	Em aula	6 h/a
	30/06	15	1,2,3	<b>Semana estendida de recuperação</b>	Em aula	6 h/a

## 8. Referências

### Básica:

Todas atividades contarão com o apoio de material preparado pelo professor sob a forma de slides, vídeos e exercícios implementados em linguagem de programação disponibilizados no ambiente virtual de aprendizagem. Os seguintes livros poderão ser consultados na biblioteca da PUCPR pelos estudantes como fonte complementar:

1. DEITEL and DEITEL; Java Como Programar, Editora Bookman, 2007
2. SANTOS, Rafael; Introdução à Programação Orientada a Objetos Usando Java. Editora Campus. 2003.
3. HORSTMANN, Cay S; Cornell, Gary. Core Java, Vol.1 e 2. Makron, 2001.

### Complementar

4. MEYER, Bertrand. OBJECT-ORIENTED SOFTWARE CONSTRUCTION. 2nd ed. Upper Saddle River: Prentice Hall PTR, c1997. 1254 p.1997 ISBN 0-13-629155-4.
5. TREMBLAY, Jean-Paul; CHESTON, Grant A. DATA STRUCTURES AND SOFTWARE DEVELOPMENT IN AN OBJECT-ORIENTED DOMAIN. Upper Saddle River: Prentice Hall, 2001. 1037 p. ISBN: 0-13-787946-6.
6. ARNOLD, Ken; GOSLING, James; HOLMES, David. A LINGUAGEM DE PROGRAMAÇÃO JAVA. Porto Alegre: Bookman, 2007. 799 p. ISBN 978-85-60031-64-1.
7. FLANAGAN, David. JAVA: O GUIA ESSENCIAL. tradução [da 5.ed. original de Edson Furmankiewicz.]. Porto Alegre: Bookman, 2006. 1099 p. ISBN 85-60031-07-3.
8. PREISS, Bruno R. ESTRUTURAS DE DADOS E ALGORITMOS: PADRÕES DE PROJETOS ORIENTADOS A OBJETO COM JAVA. [tradução Elizabeth Ferreira Gouvêa]. Rio de Janeiro: Campus, 2001. 566 p. ISBN: 85-7110-0693-0.

## 9. Acessibilidade\*\*

Não houve necessidade de adaptação.

*\*\* conforme nota técnica conjunta número 17/2020 CGLNRS/DPR/SERES/SERES*