



UFPB

**UNIVERSIDADE FEDERAL DA PARAÍBA**  
**CENTRO DE CIÊNCIAS APLICADAS E EDUCAÇÃO**  
**DEPARTAMENTO DE CIÊNCIAS EXATAS - DCX**  
**CAMPUS IV - RIO TINTO**



**PLANO DE CURSO – PERÍODO 2025.1**

[aps2025.1](#)

**Dados Gerais da Turma e Links Importantes**

Turma:	8103180 - ANALISE E PROJETOS DE SISTEMAS - Turma: 01 (2025.1)
Docente:	Matheus Barbosa de Oliveira
Carga horária:	60h
Horário:	Quinta 10h-12h / 13h-15h (Lab1)
Horários de Atendimento:	Servidor do Discord
Link para Canal Discord:	<a href="https://discord.gg/dhg5Exk2xc">https://discord.gg/dhg5Exk2xc</a>
Link para materiais da disciplina:	SIGAA
Link dos horários dos monitores:	<a href="https://docs.google.com/document/d/1_TfqDY-J26lHKWPYjDKLOsNN1MxFafu-iGAKHvm-eso/edit?usp=sharing">https://docs.google.com/document/d/1_TfqDY-J26lHKWPYjDKLOsNN1MxFafu-iGAKHvm-eso/edit?usp=sharing</a>

**Programa do Componente Curricular**

Ementa:	Técnicas avançadas de Análise e Projeto de Software empregadas no desenvolvimento de software. Técnicas utilizadas para melhorar o reuso de software usando o paradigma de Orientação a Objeto. Projeto de arquitetura de software. Padrões para atribuição de responsabilidades. Padrões de projeto.
Objetivos:	O principal objetivo da disciplina é capacitar os alunos a utilizar técnicas avançadas de análise e projeto de software no desenvolvimento de sistemas robustos, reutilizáveis e de fácil manutenção. O curso busca proporcionar uma compreensão profunda do paradigma de orientação a objetos, promovendo o uso de boas práticas como princípios SOLID, padrões GRASP e padrões de projeto, além de abordar a modelagem de arquiteturas de software. Por meio de um projeto prático integrado, os alunos aplicam os conceitos em cenários reais, consolidando o aprendizado teórico na prática.
Conteúdo:	O conteúdo da disciplina está estruturado em três unidades: (1) Revisão de Java, introdução ao desenvolvimento de projetos e padrões GRASP; (2) Princípios SOLID e fundamentos de arquitetura de software, incluindo modelos arquiteturais e modelagem com C4Model; e (3) Padrões de projeto, com foco em sua aplicação prática para resolver problemas específicos no projeto. A cada aula, são abordados conceitos teóricos seguidos de

	atividades práticas no projeto individual, abrangendo desde controle de responsabilidades até reuso e integração arquitetural."
Habilidades e Competências:	Ao final da disciplina, os alunos serão capazes de analisar requisitos de sistemas, projetar soluções orientadas a objetos utilizando padrões de responsabilidade e projeto, e desenvolver software com foco em reutilização, coesão e baixo acoplamento. Adicionalmente, terão competências para projetar arquiteturas de software em camadas e compreender os princípios fundamentais de microsserviços e APIs RESTful. O curso também desenvolve a habilidade de aplicar padrões de projeto e princípios SOLID em implementações práticas, promovendo a solução de problemas com eficiência e inovação.

### Metodologia de Ensino e Avaliação

Metodologia:	A disciplina será conduzida com atividades de laboratório realizadas em todas as aulas. A cada encontro, os alunos terão contato com conteúdos teóricos que serão imediatamente aplicados no desenvolvimento de um projeto individual. Esse projeto será incrementado ao longo do curso, permitindo a aplicação direta de conceitos como princípios de design, padrões de responsabilidade, modelagem arquitetural e padrões de projeto.
Procedimentos de avaliação da aprendizagem:	A avaliação será realizada de forma contínua e prática, com foco no progresso e na qualidade do projeto individual. Durante as apresentações agendadas, o professor solicitará que os alunos realizem alterações e implementações específicas no projeto em tempo real, avaliando sua compreensão, capacidade de adaptação e domínio dos conceitos aplicados. A avaliação considerará critérios como a correta aplicação de padrões, coesão do código, reuso, e alinhamento com boas práticas de desenvolvimento, além da clareza na apresentação das soluções propostas.

### Distribuição do Conteúdo

12/06/25 a 17/07/25	Unidade 1: Revisão de Java, Projeto e Padrões GRASP 1.1 Revisão de Orientação a Objetos 1.2 Uso do Framework de Coleções de Java e Generics 1.3 Composição x Herança 1.4 Enums e Annotations 1.5 Documentação de Código 1.6 Teste Automático de Software utilizando JUnit 1.7. Uso de ambientes integrados de desenvolvimento (IDEs), git e maven.
13/02/25 a 20/03/25	Unidade 2: SOLID e Arquitetura de Software
25/03/25 a 01/05/25	Unidade 3: Padrões de Projeto

### Cronograma de Aulas

12/06/25	Apresentação do curso, revisão de orientação a objeto. Classes x objetos
19/06/25	<b>Feriado de corpus christi</b>
26/06/25	Revisão de Java básico, Git e atividade inicial. Introdução ao ambiente de desenvolvimento e controle de versão. Atividade prática inicial com Java e Git. Modelagem OO de sistemas. Visibilidade e Encapsulamento.
03/07/25	Herança x Composição. Herança e uso de interfaces. Uso de diagramas de classe UML. Uso do Framework de Coleções de Java. Enums e Annotations.

	Classes e métodos abstratos. Documentação de código com Javadoc. Sobrescrita de métodos (@Override). equals e compareTo
10/07/25	Tratamento/lançamento de exceções. Eventos. Classes internas. Listeners de eventos. Teste Automático de Software
17/07/25	1a. Avaliação
24/07/25	Padrões GRASP; Separação de interesses, princípios para atribuição de responsabilidades e modularização;
31/07/25	Refatoração.
07/08/25	Arquitetura: camadas, microserviços e RESTful APIs. Discussão teórica e exemplos práticos; Apresentação do projeto
14/08/25	Princípios SOLID: LSP e ISP. Discussão teórica e análise de exemplos. DIP e injeção de dependências
21/08/25	2a. Avaliação
28/08/25	
04/09/25	Introdução aos padrões criacionais. Padrão Singleton, Abstract Factory
11/09/25	Introdução aos padrões estruturais. Padrão Facede, Composite e Decorator
18/09/25	Introdução aos padrões comportamentais.; Padrão Strategy, Template Method e Observer
25/09/25	3a. Avaliação
02/10/25	Reposição
09/10/25	Prova Final

Avaliações Programadas	
1a. Avaliação	17/07/2025
2a. Avaliação	21/08/2025
3a. Avaliação (entrega do projeto)	25/09/2025
Reposição (1a.)	02/10/2025
Exame Final	09/10/2025

Recursos e ferramentas digitais utilizados	
Recursos:	Para realizar as atividades de programação propostas na disciplina é necessário que o(a) aluno(a) disponha de computador (pessoal ou da universidade) por ao menos 6h semanais.
Ferramentas:	Para participar de atendimentos virtuais síncronos, o(a) aluno(a) precisará utilizar o Google Meet. É importante que todos os alunos tenham instalado em sua máquina alguma IDE (ambiente integrado de desenvolvimento) para Java, sendo indicado principalmente o IntelliJ ( <a href="https://www.jetbrains.com/pt-br/idea/">https://www.jetbrains.com/pt-br/idea/</a> ). Pode ser utilizado também o VSCode ( <a href="https://code.visualstudio.com/">https://code.visualstudio.com/</a> ou Eclipse ( <a href="https://www.eclipse.org/downloads/">https://www.eclipse.org/downloads/</a> ). O Kit de desenvolvimento Java (JDK) ( <a href="https://jdk.java.net/archive/">https://jdk.java.net/archive/</a> ) deve também estar instalado na máquina. A disciplina será também apoiada por grupo no WhatsApp (para integração da turma e avisos rápidos), mas avisos gerais oficiais serão enviados via Sigaa.

Bibliografia

### **Bibliografia Básica:**

1. BLOCH, Joshua. Java Efetivo. 3.ed. rev. Rio de Janeiro: Alta Books, 2019.
2. DEITEL, Harvey M.; DEITEL, Paul J. Java: como programar. 10a. ed. São Paulo: Pearson Prentice Hall, 2017. Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/39590>.
3. FOWLER, Martin. Refatoração - Aperfeiçoando o Projeto de Código Existente. Bookman. 2004. Disponível em: <https://integrada.minhabiblioteca.com.br/books/9788577804153>
4. GAMMA, Erich et al. Padrões de projeto: soluções reutilizáveis de software orientado a objetos. Porto Alegre: Bookman, 2000, 2008. 364p. ISBN: 8573076100, 9788573076103. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788577800469>
5. HORSTMANN, Cay. Conceitos de Computação em Java. 5a. ed. Bookman, 2009. Disponível me: <https://integrada.minhabiblioteca.com.br/#/books/9788577804078>
6. LARMAN, C. Utilizando UML e Padrões. 3a. ed. Porto Alegre: Bookman, 2007. Disponível em : <https://integrada.minhabiblioteca.com.br/reader/books/9788577800476/>.
7. SCHILDT, Herbert. Java para Iniciantes. 6a. ed. Bookman, 2015. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788582603376>.
8. Site do JUnit. Disponível em: <http://junit.org>
9. Tutorial Java da Oracle. Disponível em: <https://docs.oracle.com/javase/tutorial/>
10. WINDER, Russel; GRAHAM, Roberts. Desenvolvendo Software em Java. 3a. ed. Rio de Janeiro: LTC, 2009. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/978-85-216-1994-9>.

### **Bibliografia Complementar:**

1. BARNES, David J; KÖLLING, Michael. Programação orientada a objetos com Java. 4a. ed. São Paulo: Pearson Prentice Hall, 2009. Disponível em: <https://plataforma.bvirtual.com.br/Acervo/Publicacao/434>.
2. BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML: guia do usuário. 2.ed.rev.atual. Rio de Janeiro: Elsevier, 2006.
3. DENNIS, Alan; WIXOM, Barbara Haley; ROTH, Roberta M. Análise e Projeto de Sistemas. 5a. ed. Rio de Janeiro: LTC, 2014. ISBN: 978-85-216-2633-6. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/978-85-216-2634-3>
4. FREEMAN, Eric et al. Use a cabeça!: padrões e projetos. 2. ed. rev.. Rio de Janeiro: Alta Books, c2007. 495p. ISBN: 9788576081746.
5. LANGR, Jeff; HUNT, Andy; THOMAS, Dave. Pragmatic Unit Testing in Java 8 with JUnit. 1a. Pragmatic Bookshelf. 2015
6. KERVESKY, Joshua. Refatoração para Padrões. 1a. ed. Porto Alegre: Bookman, 2008. ISBN: 978-85-7780-303-3 Disponível em: <https://integrada.minhabiblioteca.com.br/books/9788577803033>
7. MARTIN, Robert C. Código limpo: Habilidades práticas do Agile Software. Editora Alta Books, 2009. E-book. ISBN 9788550816043. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788550816043/>. Acesso em: 30 jun. 2024.
8. MESZAROS, Gerard. xUnit test patterns: refactoring test code. Upper Saddle River, NJ: Addison-Wesley, c2007. lx, 883 p. (The Addison-Wesley signature series) ISBN: 97801314950500131495054.
9. ORAM, Andrew; WILSON, Greg. Beautiful code. Beijing Cambridge Farnham Köln Sebastopol Taipei Tokyo: O'Reilly, c2007. 593p. (Theory in practice series) ISBN: 9780596510046.
10. RUMBAUGH, James et al. UML: guia do usuário. 2. ed. rev. e atual. Rio de Janeiro: Elsevier, 2006. 474 p. ISBN: 9788535217841.
11. SCHILDT, Herbert; CAPITANIO, Bianca; ESPÍNDOLA, Wallace. Java: referência completa. 8a edição. Rio de Janeiro: Editora Alta Books, 2014.

12. SIERRA, Kathy; BATES, Bert. Use a cabeça!: java. 2a. edição revisada. Editora Alta Books, 2007.
13. Site W3Schools. Disponível em: <https://www.w3schools.com>
14. URMA, Raoul-Gabriel; WARBURTON, Richard. Desenvolvimento Real de Software: Um guia de projetos para fundamentos em Java. 1a edição. Editora Alta Books, 2021. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9786555202021>
15. VINCENZI, Auri Marcelo Rizzo; DELAMARO, Márcio Eduardo; DIAS NETO, Arilo; FABBRI, Sandra Camargo Pinto Ferraz; JINO, Mario; MALDONADO, José Carlos. Automatização de Teste de Software com Ferramentas de Software Livre. 1. ed. - Rio de Janeiro : Elsevier, 2018. il. ; 24 cm. Disponível em: <https://integrada.minhabiblioteca.com.br/books/9788595155305>