

**Resumo da experiência de monitoria na disciplina Programação 2 no semestre de 2024.1.**

**Monitor:** José Gustavo De Oliveira Cunha.

**Prof(a). orientador(a):** Jackson Raniel Florencio da Silva.

Universidade de Pernambuco – Campus Caruaru  
Rodovia BR 104, KM 55 - Eventos (Polo comercial - Nova Caruaru, PE, 55002-917

{gustavo.cunha, jackson.florencio}@upe.br

**Resumo.** A disciplina de Programação 2 explora o paradigma de Orientação a Objetos (OO) no desenvolvimento de software. Tem como foco conceitos fundamentais, como classes, objetos, métodos, herança, encapsulamento, abstração, polimorfismo, composição e coesão. Esses conceitos são essenciais para a construção de softwares robustos, flexíveis e reutilizáveis. O encapsulamento protege os dados e métodos das classes, enquanto a herança permite a reutilização de código. O polimorfismo possibilita que objetos de diferentes classes sejam tratados de maneira uniforme, promovendo a extensibilidade do código. A abstração simplifica a complexidade ao focar nos aspectos mais relevantes. Com o progresso da disciplina, espera-se que os alunos adquiram uma compreensão sólida dos principais princípios da Orientação a Objetos. Eles devem ser capazes de aplicar esses conceitos na resolução de problemas complexos de software, utilizando boas práticas de desenvolvimento, como encapsulamento adequado, criação de classes coesas, utilização correta de herança e composição, além de desenvolver sistemas que façam uso eficiente de polimorfismo. Além disso, os alunos aprenderão a aplicar testes unitários para garantir a qualidade do software, utilizando ferramentas de gerenciamento de dependências, como o Maven, e controle de versão, como o Git. Também devem desenvolver a habilidade de identificar e solucionar problemas de arquitetura de software, projetando e implementando soluções com base no paradigma de Orientação a Objetos. Além disso, espera-se que os alunos consigam gerenciar de maneira eficiente o versionamento de código e dependências externas por meio de ferramentas como Git e Maven. A prática constante de testes unitários deve ser incorporada à rotina de desenvolvimento, garantindo a entrega de código mais estável e confiável. Essas habilidades são indispensáveis para qualquer programador que queira atuar em projetos de desenvolvimento de software. A monitoria desempenhou um papel fundamental ao aprofundar os conceitos abordados em sala de aula, além de oferecer suporte aos alunos em suas dúvidas. O monitor, ao interagir diretamente com os alunos, ofereceu um suporte personalizado, especialmente no que diz respeito a temas como configuração de ambientes, testes unitários, versionamento de código e conceitos fundamentais de OO. A monitoria proporcionou uma abordagem mais personalizada para resolver as dificuldades dos alunos, permitindo um ambiente com menos participantes e maior atenção individual. Durante as atividades, foram abordados temas como encapsulamento, herança e polimorfismo, com demonstrações práticas em Java. O uso de testes unitários foi reforçado, mostrando sua importância na garantia da qualidade do software. Por fim, a monitoria ofereceu aos alunos a oportunidade de praticar o uso de ferramentas de controle de versão e gerenciamento de dependências. A experiência foi bem avaliada pelo professor orientador, destacando sua importância no processo de aprendizado. O monitor demonstrou responsabilidade e comprometimento. A monitoria foi essencial para

*aprofundar os conteúdos da disciplina, proporcionando suporte extra para a aplicação prática dos conceitos de Orientação a Objetos no desenvolvimento de software. Embora a adesão tenha sido menor do que o esperado, os alunos que participaram aproveitaram bem as sessões, esclarecendo dúvidas e consolidando seus conhecimentos. O monitor também destacou a importância da monitoria como uma oportunidade de aperfeiçoamento de suas próprias habilidades didáticas e técnicas, além de contribuir para o desenvolvimento acadêmico dos alunos.*

**Palavras-chave:** *Orientação a Objetos, Java, Monitoria, boas práticas.*