Plano Analítico: Programação II (Programação Orientada a Objetos)

1. Identificação da Unidade Curricular

- **Instituição:** Instituto Superior Politécnico de Ciências e Tecnologia (INSUTEC)
- **Curso:** Engenharia de Informática e Sistemas de Informação (EISI)
- **Classificação:** Disciplina Específica (Nuclear)
- **Ano:** 3º | **Semestre:** 2º (6º Semestre)
- **Créditos:** 8.0 UC
- **Carga Horária Total:** 120 Horas (90h de Contacto | 30h de Trabalho Complementar)

2. Apresentação e Justificação

A disciplina de Programação II marca a transição da programação estruturada para o paradigma de Orientação a Objetos (POO). Este modelo de desenvolvimento é o padrão da indústria de software moderna, permitindo a criação de sistemas modulares, reutilizáveis e de fácil manutenção. O curso foca na abstração de problemas do mundo real através de classes e objetos, garantindo que o estudante adquira as competências necessárias para o desenvolvimento de software empresarial, conforme o **Decreto Presidencial 193/18**.

3. Competências a Desenvolver (Decreto 193/18)

3.1 Competências Instrumentais (Saber)

- Compreender os quatro pilares da POO: Abstração, Encapsulamento, Herança e Polimorfismo.
- Dominar o conceito de classes, objetos, atributos e métodos.
- Entender a gestão de exceções e a utilização de interfaces e classes abstratas.

3.2 Competências Técnicas e Operacionais (Saber Fazer)

- **Modelagem e Codificação:** Implementar sistemas utilizando linguagens orientadas a objetos (ex: **Java** ou **C#**).
- **Desenvolvimento Modular:** Utilizar herança e polimorfismo para criar código extensível e evitar redundância.
- **Persistência e Interface:** Criar interfaces gráficas básicas e realizar a conexão de objetos a fontes de dados.

3.3 Competências Atitudinais (Saber Ser/Estar)

- Aplicar padrões de projeto (*Design Patterns*) simples para resolver problemas recorrentes de engenharia de software.
- Demonstrar organização e capacidade de abstração na arquitetura de sistemas complexos.

4. Conteúdo Temático (Estrutura de 120 Horas)

1. **Introdução ao Paradigma de Objetos:** Objetos vs. Programação Estruturada; O ambiente de desenvolvimento (JDK/IDE).
2. **Classes e Objetos:** Construtores, métodos acessores (Getters/Setters) e modificadores de visibilidade (Private, Public, Protected).
3. **Encapsulamento e Composição:** Proteção de dados e relacionamento entre classes.
4. **Herança e Polimorfismo:** Especialização de classes, sobrescrita de métodos (*Override*) e ligação dinâmica.
5. **Classes Abstratas e Interfaces:** Contratos de programação e design de sistemas flexíveis.
6. **Tratamento de Exceções:** Robustez de software com blocos Try-Catch-Finally.
7. **Coleções de Objetos (Generics):** Uso de Listas, Conjuntos e Mapas dinâmicos para armazenamento de objetos.

5. Regime de Avaliação (Disciplina Específica)

- **Avaliação Contínua (40%):**
    - 1ª Frequência (Fundamentos, Classes e Objetos): 13%
    - 2ª Frequência (Herança, Polimorfismo e Exceções): 14%
    - **Projeto de Software Orientado a Objetos:** Desenvolvimento de uma aplicação funcional: 13%
- **Exame Normal (60%):** Prova global teórica e prática (implementação de diagrama de classes).

6. Referências Bibliográficas (APA 7ª Ed.)

- Deitel, P., & Deitel, H. (2017). *Java: Como programar* (11ª ed.). Pearson.
- Sierra, K., & Bates, B. (2008). *Use a cabeça!: Java*. Alta Books.
- Eckel, B. (2006). *Thinking in Java* (4th ed.). Prentice Hall.
- Gamma, E., et al. (1994). *Design patterns: Elements of reusable object-oriented software*. Addison-Wesley.