

Plano Analítico: Programação II (Orientada a Objetos)

1. Identificação da Unidade Curricular

- **Curso:** Engenharia de Redes e Telecomunicações (ERT)
 - **Ano:** 4^o | **Semestre:** 1^o
 - **Créditos:** 6.0 UC
 - **Carga Horária Total:** 90 Horas
 - **Distribuição:**
 - **Teóricas (T):** 20h
 - **Teórico-Práticas (TP):** 20h
 - **Práticas (P):** 20h
 - **Trabalho Autónomo (TA):** 22h
 - **Orientação e Tutoria (OT):** 4h
 - **Avaliação (AV):** 4h
-

1. Fundamentação

À medida que os sistemas de software em telecomunicações se tornam mais complexos (como sistemas de gestão de faturação, monitorização de rede ou aplicações de utilizador), a programação estruturada torna-se difícil de manter. A POO permite modelar elementos da rede como "objetos" reais, facilitando o reaproveitamento de código e a escalabilidade. Esta disciplina é fundamental para o desenvolvimento de software robusto e para a compreensão de APIs e frameworks modernos utilizados no setor das TIC.

2. Objectivos Instrutivos e Educativos

- **Instrutivos:** Compreender e aplicar os quatro pilares da POO (Encapsulamento, Herança, Polimorfismo e Abstração); dominar o tratamento de exceções; aprender a utilizar coleções de dados dinâmicas; introduzir a programação multithread e a interface gráfica (GUI).
- **Educativos:** Fomentar a organização modular e o design de software limpo; desenvolver a capacidade de abstração de problemas do mundo real; promover o trabalho colaborativo através de padrões de design.

3. Resultado de Aprendizagem

O estudante será capaz de:

- Desenvolver aplicações complexas utilizando classes e objetos de forma eficiente.
- Implementar hierarquias de classes para modelar sistemas de telecomunicações.
- Gerir erros de execução de forma profissional através de blocos try-catch.
- Criar programas que executam múltiplas tarefas simultaneamente (Threads), essencial para servidores de rede.
- Utilizar interfaces e classes abstratas para definir contratos de software.

4. Planeamento Temático (6 UC)

Tema	Horas (T+TP+P)	Conteúdo Programático
I. Paradigma de Objetos	10h	Classes, Objetos, Atributos e Métodos; Construtores; Modificadores de acesso (Private, Public).
II. Encapsulamento e Herança	12h	Reutilização de código; Sobrescrita de métodos; Palavra-chave super e this.
III. Polimorfismo e Abstração	14h	Classes e métodos abstratos; Interfaces; Polimorfismo dinâmico.
IV. Gestão de Erros e Coleções	12h	Exceções (Exceptions); Listas, Sets e Maps (Java Collections Framework).
V. Threads e I/O de Objetos	12h	Ciclo de vida de uma Thread; Sincronização básica; Serialização de objetos.

5. Recomendações Metodológicas

- **Laboratório de POO (20h):** Desenvolvimento de projetos em **Java** ou **C#** utilizando IDEs profissionais como **IntelliJ IDEA**, **Eclipse** ou **Visual Studio**.
- **Modelação UML:** Uso de diagramas de classe para planejar a arquitetura do software antes da codificação.

- **Exemplos Reais:** Criar uma classe EquipamentoRede com subclasses Router e Switch, demonstrando herança e polimorfismo nas funções de processamento de pacotes.

6. Sistema de Avaliação

Conforme a alocação de **4h para AV:**

- **Avaliação Contínua (60%):** Projeto prático de média complexidade (ex: Simulador de Gestão de Inventário de Rede) (40%) e mini-testes de codificação (20%).
- **Avaliação Formal (40%):** Exame final prático em laboratório para resolução de um problema de modelagem de objetos.

7. Bibliografia Principal Indicada

1. **DEITEL, Paul & DEITEL, Harvey.** *Java: Como Programar*. Pearson.
2. **SIERRA, Kathy & BATES, Bert.** *Use a Cabeça! Java*. Alta Books.
3. **STROUSTRUP, Bjarne.** *A Linguagem de Programação C++*. Bookman.