

# Programação Orientada a Objetos

LESI|LESIPL

## Trabalho Prático

Luís G. Ferreira\* & Ernesto Casanova†

EST-IPCA  
Barcelos

30 de outubro de 2025

### Resumo

Este trabalho da Unidade Curricular (UC) de Programação Orientada a Objetos (POO) foca a análise de problemas reais simples e a aplicação do Paradigma Orientado a Objetos na implementação de possíveis soluções.

---

\*Email: lufer@ipca.pt; Gab: 5

†Email: ecasanova@ipca.pt; Gab: A

## 1 Motivação

Pretende-se que sejam desenvolvidas soluções em C# para problemas reais de complexidade moderada. Serão identificadas classes, definidas estruturas de dados e implementados os principais processos que permitam suportar essas soluções. Pretende-se ainda contribuir para a boa redação de relatórios.

## 2 Objetivos

- Consolidar conceitos basilares do Paradigma Orientado a Objectos;
- Analisar problemas reais;
- Desenvolver capacidades de programação em C#;
- Potenciar a experiência no desenvolvimento de software;
- Assimilar o conteúdo da Unidade Curricular.

## 3 Regras do jogo

- O trabalho deverá ser feito individualmente;
- O trabalho deverá ser entregue em duas fases. Em cada fase do trabalho realizado, deverá ser submetido um ficheiro compactado, na plataforma *moodle*, cujo nome deverá conter o número do aluno e número que identifica a fase. Exemplo para a entrega da Fase 1 do aluno 1234:

- *trabalhoPOO\_1234\_Fase1.zip*

As datas previstas para cada fase e as metas a atingir em cada uma, são:

- **Fase 1 (15-11-2025)**

- Estrutura de Classes identificadas
  - Implementação essencial das classes
  - Estruturas de dados a utilizar
  - Relatório do trabalho desenvolvido até à data
  - Cumprimento dos prazos

- **Fase 2 (19-12-2025)**

- Implementação final das classes e serviços
  - Aplicação demonstradora dos serviços implementados
  - Relatório final do trabalho realizado
  - Cumprimento dos prazos

## 4 Problemas a explorar

O tema de trabalho que o aluno pretenda explorar deve ser indicado na plataforma Onedrive - para LESIN ou na plataforma Onedrive - para LESI O aluno pode propor o seu próprio tema ou escolher uma das seguintes sugestões:

- (i) Gerir de Atividades de Socorro: sistema que permita registar ocorrências de pedidos de ajuda e gerir equipamentos e pessoas para prestar auxilio.  
**keywords:** Proteção Civil; Equipamentos; INEM; Enfermeiros, Médicos; Bombeiros,
- (ii) Gerir obra de construção civil: sistema que permita controlar e gerir os custos de uma determinada obra.  
**keywords:** obras, materiais, armazéns, stocks, viaturas, serviços, mão de obra (contratada e subcontratada), orçamentos, documentos.
- (iii) Gestão de condomínios: sistema que permita fazer a gestão de condomínios de uma empresa.  
**keywords:** condomínios, condóminos (proprietários, inquilinos), permilagens, quotas, despesas, gestão de pagamentos, agendamento de reuniões, atas, documentos.
- (iv) Gestão de rendas/imóveis: sistema que permita a uma empresa/proprietário gerir os seus imóveis e as respetivas rendas mensais;  
**keywords:** senhorios, inquilinos, imóveis (apartamento, vivenda, terreno), recibos, contratos, despesas, documentos, distritos (código e descrição nas finanças), concelhos (código e descrição nas finanças), freguesias (código e descrição nas finanças), estados.
- (v) Gestão de alojamentos turísticos: sistema que permita a gestão de alojamentos turísticos.  
**keywords:** registos, consultas, reservas, check-in, clientes, alojamentos.
- (vi) Helpdesk: Sistema de gestão e apoio a assistências por telefone (call center).  
**keywords:** assistência, tipo de assistência, estado da assistência, operador, cliente, produtos, problemas conhecidos, tutoriais de resolução de problema, documentos, avaliação da assistência (e.g. se o problema foi resolvido e avaliação de 1 a 10).
- (vii) Comércio eletrónico: sistema que permita a gestão de uma loja online.  
**keywords:** produtos, categorias, garantias, stocks, clientes, campanhas, vendas, marcas.
- (viii) Gestão de jardim zoológico: sistema que permita a gestão das tarefas de um jardim zoológico.  
**keywords:** animais, informações, assistência veterinária, alimentação, calendários, tipos de comida, limpeza de jaulas, espetáculos, bilhetes.
- (ix) Gestão de Centro de Saúde: sistema que permita gerir um Centro de Saúde.  
**keywords:** staff, categorias, especialidades, consultas, camas, pacientes, diagnósticos, exames, custos.

## 5 Critérios de Avaliação

### 5.1 Qualidade do Relatório

- Estrutura, Clareza e Expressividade

- Capacidade de síntese e qualidade da escrita
- Uso de diagramas de apoio (UML ou semelhantes) para ilustrar desenho da solução.

## 5.2 Qualidade da solução desenvolvida

- Estrutura coerente da solução; utilização de boas práticas e padrões de design (Design Patterns); nomes de ficheiros e classes consistentes; conformidade com norma CLS.
- Reutilização de código através de bibliotecas (DLL), promovendo modularidade e reutilização.
- Estruturação adequada dos pilares de programação orientada a objetos, com aplicação de Interfaces, Herança, Abstração, Polimorfismo e Encapsulamento tudo o que promova reutilização, extensibilidade e desacoplamento.
- Qualidade dos algoritmos aplicados, promovendo eficiência, clareza. Deve incluir testes unitários com cobertura mínima de 50%.
- Utilização de exceções customizadas (extensão de Exception) com tratamento adequado de exceções.
- Seleção e uso adequado de estruturas de dados, consoante o problema (listas, filas, pilhas, dicionários, árvores, grafos, etc.).
- Tratamento adequado de logs, uso consistente e estruturado de logs, associado a padrões de design quando aplicável.
- Persistência de dados implementação com recurso a ficheiros (ou opcionalmente bases de dados simples).
- Programação por camadas, com aplicação de boas práticas na separação de responsabilidades (NTier, MVC, MVVM, ou equivalente).
- Integração Contínua e Deploy Contínuo (CI/CD) com Azure DevOps, GitHub Actions, GitLab CI, ou outra ferramenta à escolha.

## 5.3 Outras valências

- Utilização de Funções Lambda e LINQ.
- Aplicação adequada de outros padrões de desenho (Design Patterns) relevantes (ex: Singleton, Factory, Repository, Observer, Strategy, etc.).
- Exploração de outras valências: Windows MAUI; WPF; Forms;

## 6 Avaliação

- Fase 1 - 25% da nota final do trabalho (TP.C - Desenho da Solução refletido no relatório, (Opcionalmente) Componente código)
- Fase 2 - 75% da nota final do trabalho (TP.C - Componente código)
- O trabalho será defendido individualmente - consultar metodologia e avaliação.

## 6.1 Notas importantes

- A utilização de qualquer abordagem que não consiga justificar e ou explicar, será penalizado.
- A utilização ou inclusão duma abordagem que se desvie dos pontos obrigatórios na entrega, será penalizado”, (**por exemplo, focarem-se demasiado na UI, ou explorarem outras valências, sem entregarem um projeto bem estruturado;**
- Os alunos só após cumprir todos os pontos obrigatórios, é que devem tentar explorar melhorias que possam valorizar o projeto como produto, como é o caso da UI);

Bom trabalho.