

# UC: Programação Funcional

Introdução da UC

2024-2025

# Unidade Curricular

- Moodle para disponibilização do conteúdo da UC.
- Canal do Teams **24-25\_(Agregação) PROGRAMAÇÃO FUNCIONAL - General** para a gestão contínua da UC de Programação Funcional (PF).
- Dúvidas sobre a Unidade Curricular?
  - Nas aulas (preferencial)
  - No Teams
  - Enviar um email com o assunto 'PF: <título>' onde '<título>' é o título da dúvida em questão.

# Docentes da UC

- André de Matos Pedro (TE1, TE2 e PL1)
  - Email: matos.pedro@ubi.pt
- Luís Horta (PL2 e PL3)
  - Email: luis.horta@ubi.pt
- Tiago Filipe Dias dos Santos Roxo (PL4, PL5, PL6 e PL7)
  - Email: tiago.roxo@ubi.pt

# Objetivos da UC

- Existem diferentes paradigmas de programação que se adequam a diferentes tipos de resolução de problemas por um computador.
- Para compreender o paradigma funcional, estudaremos:
  - os fundamentos de programação funcional para resolver problemas de natureza computacional;
  - as diferenças entre os paradigmas de programação imperativa e funcional;
  - os conceitos básicos de programação funcional com recurso a uma linguagem funcional;
  - algoritmos sobre estruturas de dados como listas e árvores;
  - algoritmos de procura e de ordenação;
  - técnicas algorítmicas: estratégia gulosa, programação dinâmica ou programação com retrocesso (*backtracking*).

# Conteúdo da UC

- História da Programação Funcional
- Tipos Básicos, Entrada/Saída, Estruturas de Controlo e Funções, Recursividade e Funções de Ordem Superior.
- Polimorfismo, Tipos de dados algébricos: tipos produtos, enumerados, soma e estruturados.
- Módulos e Estruturas de dados abstratas.
- Noções Gerais de análise de programas: complexidade computacional e correção funcional.
- Estruturas sequenciais: vetores (redimensionáveis), tabelas de Hash, listas, pilhas e filas.
- Estruturas não sequenciais : conjuntos, dicionários, árvores, cordas, amontoados, árvores binária equilibradas.
- Ordenação, indexação e pesquisa.
- Algoritmia: dividir-e-conquistar, algoritmos gulosos, por retrocesso, programação dinâmica, memoização.

# Resultados da UC

- O aluno deverá ser capaz de
  - Definir funções usando equações com padrões;
  - Codificar algoritmos recursivos elementares sobre estruturas de dados fundamentais (e.g. listas e árvores, etc.);
  - Definir novos tipos algébricos para representar dados;
  - Decompor problemas de programação usando os mecanismos próprios da programação funcional;
  - Saber desenhar uma solução programática que envolva estrutura de dados (sequências ou arborescentes) e algoritmos básicos (ordenação ou pesquisa) e para um problema computacional;
  - Saber usar técnicas algorítmicas como dividir-e-conquistar, a programação dinâmica, algoritmia gulosa ou ainda programação com retrocesso.

# Critérios de Avaliação da UC

- É constituída por duas componentes:
  1. a componente prática - exercício prático e o enunciado do exercício será disponibilizado no Mooshak;
  2. a componente teórica - provas escritas.
- Avaliação:
  - 70% da nota da média das três Frequências ou Exame
  - 30% da nota do exercício prático (a nota transita para Exame)

# Datas e Notas Importantes da UC

- Frequência 1 – **19 de Março às 18.30h na sala 4.02**
- Frequência 2 – **16 de Abril às 18.30h na sala 4.02**
- Frequência 3 – **28 de Maio às 18.30h na sala 4.02**
- Exercício Prático – **para submeter no Mooshak** em <https://ocaml.di.ubi.pt/mooshak/>
- Exame – Datas a anunciar no Balcão Virtual
- Condição para Exame:
  - A componente prática ser **superior ou igual a 6**
- **Avaliação:**
  - 70% da média dos testes Teórico/Práticos (Frequência) ou Exame
  - 30% da nota do exercício prático (a componente prática fica inalterada e transita para Exame)



# Referências Bibliográficas da UC

- Yaron Minsky, Anil Madhavapeddy, Jason Hickey. Real World OCaml, Functional Programming for the masses. 2nd Edition. O'Reilly. [Versão online](#).
- Michael R. Clarkson. OCaml Programming: Correct + Efficient + Beautiful. [Versão online](#).
- John Whittington. OCaml From the Very Beginning. [Versão online](#).
- Jean-Christophe Filliâtre, Sylvain Conchon. Apprendre à programmer avec OCaml, Eyrolle, 2014. ISBN-13: 978-2212136784