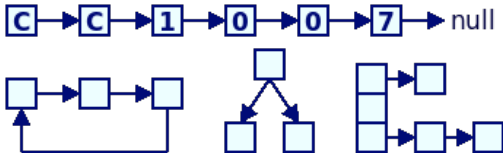


# Estruturas de Dados

P. Ribeiro e M. Areias

DCC/FCUP

2022/2023





## Recursos Principais

- **Website:**

<http://www.dcc.fc.up.pt/~pribeiro/aulas/edados2223/>

-  **slack** - **Slack:** Slack: <http://edados2232.slack.com>  
(usado para a comunicação nesta UC)

-  **Mooshak:**  
<http://mooshak.dcc.fc.up.pt/~edados/>  
(submissão de código e resposta a quizzes)

-  **YouTube** - **YouTube:**  
<https://www.dcc.fc.up.pt/~pribeiro/aulas/edados2223/videos/>  
(vídeos de teóricas de outras edições)

- **Aulas Teóricas:** *(1h+1h por semana, em anfiteatro)*

- ▶ Miguel Areias
- ▶ Pedro Ribeiro

*Planeamento previsto de quem dá qual aula:*

<https://www.dcc.fc.up.pt/~pribeiro/aulas/edados2223/plan.html>

- **Aulas Práticas:** *(2h por semana, em laboratório)*

- ▶ Hélder Oliveira (1 turma)
- ▶ Maria Pedroto (2 turmas)
- ▶ Miguel Areias (2 turmas)



**Nome:** Miguel Areias

**Gabinete:** FC6 1.01 (DCC/FCUP)

**Website:** <https://www.dcc.fc.up.pt/~miguel-areias/>

## Principais Interesses de Investigação

- Computação Paralela e Distribuída
- Estruturas de Dados Concorrentes
- Implementação de Linguagens
- Programação em Lógica



**Name:** Pedro Ribeiro

**Gabinete:** FC6 1.47 (DCC/FCUP)

**Website:** <https://www.dcc.fc.up.pt/~pribeiro/>

## Principais Interesses de Investigação

- Redes Complexas, *Network Science*, *Graph Mining*, Ciência de Dados
- Algoritmos e Estruturas de Dados, Complexidade
- Computação Paralela e Distribuída
- Educação em Ciência de Computadores e Concursos de Programação

# Programação Competitiva - Pedro Ribeiro

Estou envolvido em muitos **concursos de programação algorítmicos**:  
(*organização, criação e resolução de problemas, treino de estudantes, ...*)

- Alunos Pré-Universitários (Básico e Secundário)
  - ▶ **Olimpíadas** de Informática Nacionais e Internacionais (ex: ONI, IOI)
  - ▶ **Bebras** - Desafio Internacional de Pensamento Computacional
- Alunos Universitários
  - ▶ Concursos nacionais e internacionais **ICPC** (ex: MIUP, SWERC)



Para obter frequência é necessário:

- Não exceder limite de faltas nos **quizzes** online (máximo de 5 faltas)
  - ▶ Estão previstos 11 quizzes
  - ▶ São obrigatórios, mas não contam para nota
  - ▶ Cada um estará online para efeitos de frequência durante uma semana (00:01 de Domingo a 23:59 de Sábado)  
(depois ficam disponíveis para treino, sem contar para frequência)
  - ▶ São constituídos por perguntas de escolha múltipla
  - ▶ Podem ver os resultados, saber que opção estava correcta e voltar a submeter quantas vezes quiserem

# Fórmula de Cálculo da Avaliação

- **P:** nota prática, valendo **30%** da nota final, obtida através de 3 componentes:
  - ▶ 2 testes práticos de programação (2.5 valores cada)
  - ▶ resolução de exercícios ao longo do semestre (1 valor).

Nota mínima:  $P \geq 1.5$  (escala da nota: 0 a 6).

- **EN:** nota do exame de época normal, valendo 70% da nota final, obtida através de um exame escrito (presencial) com nota de 0 a 20
- **ER:** na época de recurso será feito um único exame (presencial), valendo 70% da nota final, não sendo possível repetir ou melhorar a componente prática

**Classificação da época normal:**  $C = EN * 0.7 + P \geq 9.5$

**Classificação da época de recurso:**  $C = ER * 0.7 + P \geq 9.5$



# Sobre a componente prática

- Será usada a linguagem **Java**, e terão acesso a um computador
- Terão objectivos específicos divulgados para os testes
- Terão acesso a (algum) código nosso

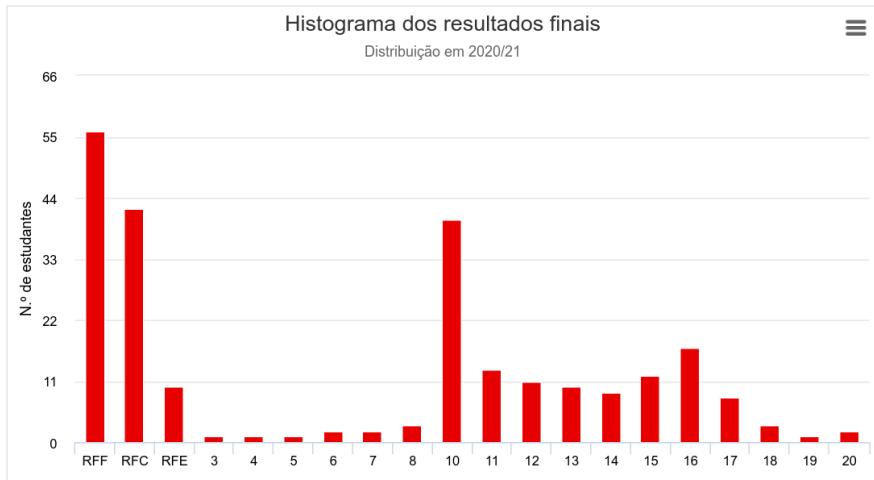
Datas dos testes práticos de programação (duração de 2h):

- **1º teste:** 16 de Novembro (quarta-feira) - 2 turnos (14:30 e 16:30)
- **2º teste:** 14 de Dezembro (quarta-feira) - 2 turnos (14:30 e 16:30)

## Estruturas de Dados nos últimos 5 anos:

- Total de Alunos:  
**285** (17/18), **259** (18/19), **236** (19/20), **244** (20/21), **49** (21/22)
- Alunos aprovados:  
**133** (17/18), **135** (18/19), **132** (19/20), **127** (20/21), **22** (21/22)

## Distribuição de notas em 20/21:



## Estruturas de Dados 2022/2023

- Total de Alunos inscritos: **144**
- Nº de inscrições: **127** (1<sup>a</sup>), **4** (2<sup>a</sup>), **8** (3<sup>a</sup>) **3** (4<sup>a</sup>), **1** (5<sup>a</sup>), **1** (7<sup>a</sup>)
- Total por curso: **66** (L:CC), **54** (L:IACD), **16** (L:M)  
**3** (L:F), **3** (L:B), **2** (Ext.)

## Programação Imperativa 2021/2022

- Total de Alunos: **164**
- Total de Avaliados: **118**
- Total de Aprovados: **101**

# Objectivos da Unidade Curricular

## Competências de **programação**

- Proeficiência na linguagem **Java**
- Paradigma de **Programação Orientada a Objectos**

## Conhecimentos de **estruturas de dados e algoritmos** associados

- Principais estruturas de dados básicas: **arrays, matrizes, listas ligadas e árvores binárias.**
- Principais tipos abstractos de dados e suas implementações: **filas, pilhas, conjuntos, dicionários e filas de prioridade.**

## Competências básicas na concepção e **análise de algoritmos**

- Competência básica na análise de **complexidade de algoritmos** e e compreensão das principais **classes de complexidade.**
- Enriquecimento do conhecimento sobre técnicas algorítmicas como **recursividade, pesquisa com retrocesso e dividir para conquistar.**
- Experiência prática de aplicação a **problemas concretos.**

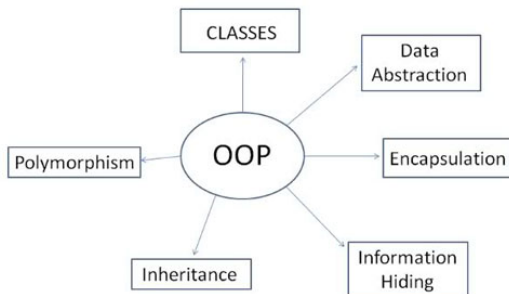
## Conceitos fundamentais de Java:

- Classes, objectos, atributos e métodos
- Tipos primitivos, strings, wrappers, arrays e tipos enumerados
- Expressões, operadores e instruções de controle de fluxo
- Input/Output e a classe Scanner
- Pacotes e biblioteca padrão do Java
- Princípios de desenvolvimento de software, estilo e documentação



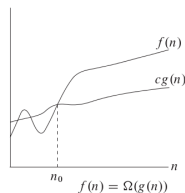
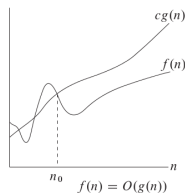
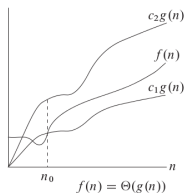
## Programação Orientada a Objectos

- Objectivos, princípio, padrões e mecanismo de herança
- Interfaces e Tipos Abstractos de Dados (TADs)
- Uso de genéricos e de iteradores



## Conceitos de Análise Assintótica

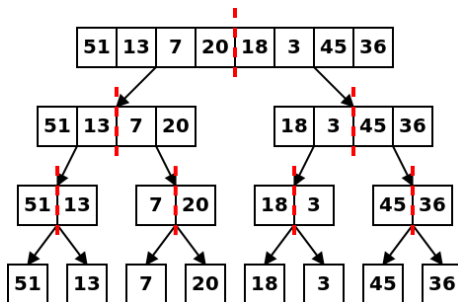
- Noções de análise assintótica
- Classes de complexidades típicas e sua comparação
- Exemplos de análise de algoritmos





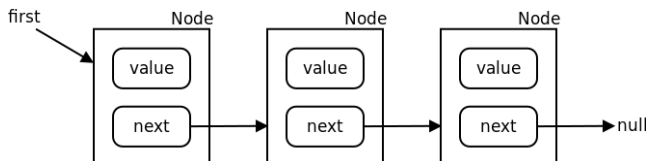
## Técnicas de Desenho de Algoritmos

- Programação estruturada
- Recursividade
- Pesquisa exaustiva e *backtracking*
- Dividir para conquistar



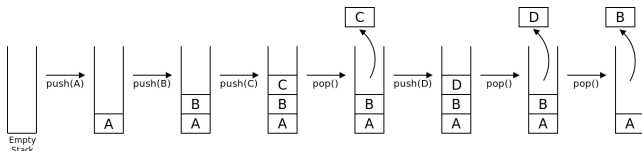
## Estruturas de Dados Fundamentais

- Arrays e matrizes
- Listas ligadas simples, circulares e duplamente ligadas
- Árvores binárias, árvores de pesquisa e heaps



## Tipos Abstratos de Dados e suas possíveis Implementações

- Sequências, pilhas, filas e dequeus
- Contentores associativos: conjuntos e dicionários
- Filas de prioridade



# Funcionamento das aulas

- **Teóricas:** presenciais (+ vídeos pré-gravados) com uso de: slides + livecoding + quadro + visualizações + ... (+ quizzes no Mooshak)
- **Práticas:** guião com exercícios (html) + implementação em código (Java) (+ submissões no Mooshak)
- **Material auxiliar:** divulgado no site, em português e/ou inglês videos, slides, apontamentos, animações, ...

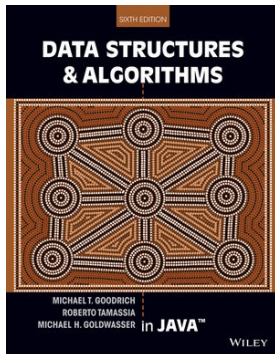
## Importante trabalhar fora das aulas!

As aulas são "apenas" as **horas de contacto**. É importante estar presente mas não chega apenas lembrar-se de E. Dados durante as aula práticas....

Livro Principal:

- **Data Structures and Algorithms in Java**

M Goodrich, R Tamassia and M Goldwasser, 6th Edition, Wiley, 2014.



Outros Livros Recomendados:

- **Introduction to Algorithms**

TH Cormen, CE Leiserson, RL Rivest and C Stein, 3rd Edition, MITPress, 2009.

- **Introduction to Programming in Java: An Interdisciplinary Approach**

R. Sedgewick, K Wayne, 2nd Edition, Addison-Wesley, 2017.

- **Think Java: How To Think Like a Computer Scientist**

A Downey, O'Reilly, 2016.

- **Think Data Structures**

A Downey, O'Reilly, 2017.