

## Algoritmos e Estruturas de Dados

2025/2026 — 1º Semestre

### 1º Trabalho — O TAD imageRGB

Data limite de entrega: 28 de novembro de 2025, às 18 horas

#### 1. Completar o desenvolvimento e testar funções básicas do TAD imageRGB

Esse TAD permite instanciar e operar com **imagens** cujos *pixels* podem tomar valores de intensidade na escala de cor **RGB** (Red, Green, Blue), com uma gama de 8 *bits* para cada canal.

Cada imagem é representada internamente usando uma matriz 2D e uma **LUT (Look-Up Table)**. A LUT contém a **informação de cor** (tuplos RGB) e cada elemento da matriz (*pixel* da imagem) **indexa** o correspondente elemento da LUT, i.e., associa a cada pixel a sua cor representada na escala RGB. Por exemplo, a cor dos *pixels* de uma **imagem binária BW (black-and-white)** será representada usando os tuplos RGB (255, 255, 255) (*branco*) e (0, 0, 0) (*preto*).

Analise a **estrutura de dados** que permite representar os **atributos** e o conteúdo de uma **imagem**, a LUT que lhe está associada, bem como as **funcionalidades** definidas.

As **funções básicas** que deve **completar** estão assinaladas no ficheiro **imageRGB.c**

- Image ImageCopy(img)
- int ImageIsEqual(img1, img2)
- Image ImageRotate90CW(img)
- Image ImageRotate180CW(img)

#### 2. Complexidade computacional da função ImageIsEqual(img1, img2)

Deve analisar a **complexidade computacional** da função **ImageIsEqual(img1, img2)** em função do número de **comparações** efetuadas envolvendo os **pixels** das imagens.

Identifique as situações correspondentes ao **melhor caso** e ao **pior caso**, e indique o **número de comparações** efetuadas. Execute **testes computacionais** para imagens com diferentes tamanhos e conteúdos, registe e analise o número de comparações efetuadas.

#### 3. Completar o desenvolvimento e testar as funções que permitem identificar regiões de uma imagem

As **funções** que deve **completar** estão assinaladas no ficheiro **imageRGB.c**

- int ImageRegionFillingRecursive(img, u, v, color)
- int ImageRegionFillingWithQUEUE(img, u, v, color)
- int ImageRegionFillingwWithSTACK(img, u, v, color)
- int ImageSegmentation(img, fillFunct)

As primeiras funções, que permitem atribuir um dado índice da LUT (i.e., uma cor) aos *pixels* de uma região da imagem, fazem-no usando o algoritmo de **Region Growing by Flood Filling**.

Na sua versão recursiva (**ImageRegionFillingRecursive**), o algoritmo pode ser apresentado do seguinte modo:

```
FloodFill(Img, u, v, color)
    // (u, v) são as coordenadas de um pixel do background
    // em que se inicia o preenchimento de uma região
    // color é o índice da cor que se vai atribuir aos pixels da região
    // propagando-se a partir do pixel (u, v)
    If ValidPixel(u, v) and Img(u, v) == background
        Img(u, v) = color
        FloodFill(Img, u + 1, v, color)
        FloodFill(Img, u, v + 1, color)
        FloodFill(Img, u, v - 1, color)
        FloodFill(Img, u - 1, v, color)
```

É possível desenvolver versões iterativas (**ImageRegionFillingWithQUEUE** e **ImageRegionFillingWithSTACK**) deste algoritmo, usando as estruturas de dados auxiliares disponibilizadas.

A função **ImageSegmentation** permite preencher todas os *pixels* das regiões de uma imagem com diferentes índices da LUT (i.e., cores), efetuando um **varrimento sequencial** da imagem a partir do pixel (0, 0) e executando, para cada região ainda não identificada, uma função de **RegionGrowing by Flood Filling**.

**Atenção:**

## 1 – Desenvolvimento do código

- O ficheiro de interface **imageRGB.h** especifica as diferentes funções do TAD e **não deve ser alterado**.
- Para cada função são indicadas as eventuais **pré-condições** e **pós-condições**, que deverão ser obrigatoriamente tidas em conta.
- Caso facilite o desenvolvimento do código, poderão ser definidas e implementadas **funções auxiliares adicionais** (**static**) no ficheiro **imageRGB.c**. Deve assinalá-las com **comentários** apropriados.
- O **código** desenvolvido deverá ser **claro** e **comentado**: os identificadores escolhidos para as variáveis e a estrutura do código, bem como os eventuais comentários, deverão ser suficientes para a sua compreensão.
- **Não deve alterar** qualquer um dos ficheiros associados aos outros **TADs auxiliares** fornecidos.
- Não é necessário entregar qualquer relatório relativo ao desenvolvimento do TAD.
- Deverá apenas ser entregue o ficheiro **imageRGB.c** — com a **identificação do(s) seu(s) autor(es)**.

## 2 – Análise da complexidade computacional da função ImageIsEqual

1. Realize testes computacionais com imagens de diferentes tamanhos e conteúdos. Registe e analise o número de **operações** efectuadas envolvendo os valores dos *pixels* das imagens.
2. Efetue uma **análise formal** da complexidade do algoritmo.
3. Compare os resultados obtidos nas duas tarefas anteriores.

## 3 – Escrita de um relatório sucinto (máx. 6 págs.)

### O relatório deverá incluir:

- Uma tabela com os **resultados dos testes** efetuados para a função ImageIsEqual.
- A **análise formal da sua complexidade**.
- Para as funções que efetuam a segmentação de uma região será **valorizada** a comparação das suas **diferentes estratégias**.

## 4 – Critérios de Avaliação

- **Colaboração na Avaliação entre Pares** (10%)
  - Testar o funcionamento do código de colegas, avaliar a sua qualidade e clareza (2 trabalhos)
- **Desenvolvimento e teste das funções pedidas** (45%)
  - Será avaliada a qualidade e clareza do código e comentários
  - Será verificada a existência de eventuais fugas de memória
- **Relatório:** (45%)
  - Aspectos Gerais/Apresentação/Conclusão
  - Análise da função **ImageIsEqual(img1, img2)**
    - Análise experimental
    - Análise Formal
    - Comparação: análise experimental vs formal
  - Comparação das estratégias de **Region Growing by Flood Filling**

A **nota atribuída ao desenvolvimento e teste das funções pedidas** será obtida pela média da classificação atribuída pelo docente e da classificação atribuída pelos colegas (avaliação entre pares, cada trabalho será avaliado por 3 a 4 estudantes):

A **avaliação entre pares está sujeita a validação pelos docentes**, podendo ser descartada nos casos em que se verifique que corresponde a uma avaliação manifestamente incorreta do trabalho apresentado.

### Atenção:

- O trabalho deve ser realizado em grupos de 2 alunos.
- Poderão vir a ser disponibilizados ficheiros adicionais no **Moodle**, para poderem testar o funcionamento correto das funções à medida que forem sendo desenvolvidas.

- A entrega do trabalho (**ficheiro imageRGB.c + relatório**) será feita através do Moodle.
- Após a conclusão das submissões, cada aluno receberá informação sobre os dois trabalhos que deverá rever. Essa tarefa é parte integrante deste trabalho.