

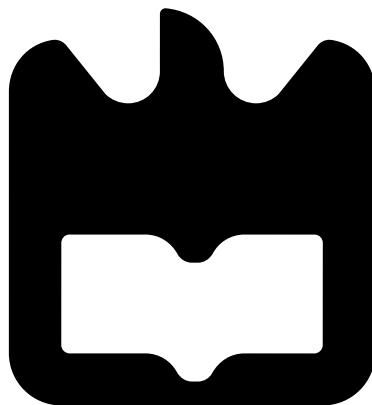
**Universidade de Aveiro**  
Departamento de Eletrónica, Telecomunicações e Informática

**Algoritmos e Estruturas de Dados**

**Relatório do Trabalho 1**  
**Análise de Algoritmos sobre Imagens**  
**RGB**

**Autores:**

Vicente Amorim Silva – 125160  
[vicenteamorimsilva@ua.pt](mailto:vicenteamorimsilva@ua.pt)  
Victor Miguel L. V. da Costa Morais – 125478  
[victorcostamoraиш@ua.pt](mailto:victorcostamoraиш@ua.pt)



**Universidade de Aveiro – DETI**

## Introdução

O presente relatório descreve a análise formal e experimental de funções de processamento de imagens desenvolvidas no âmbito do módulo **imageRGB** da unidade curricular *Algoritmos e Estruturas de Dados*. O módulo recorre a uma representação indexada composta por uma matriz de labels e uma tabela de cores (LUT), permitindo operações eficientes de manipulação e comparação de imagens RGB.

O trabalho inclui: (1) a caracterização formal da função **ImageisEqual**; (2) a avaliação experimental do comportamento desta função em vários cenários; (3) a comparação entre diferentes estratégias de segmentação de regiões, fundamentais em algoritmos de *region growing*.

O objetivo final consiste em avaliar o desempenho e a adequação destas abordagens para imagens indexadas, apoiando-se em análise teórica e evidência experimental.

## 1. Estrutura Interna do TAD **imageRGB**

O módulo **imageRGB** representa imagens através de uma matriz de labels inteiros, onde cada label corresponde a uma cor armazenada numa LUT (look-up table) de valores RGB de 24 bits. Esta abordagem reduz redundância, facilita a partilha de cores e melhora o desempenho de operações repetidas.

**A estrutura contém:**

- **width** – número de colunas da imagem
- **height** – número de linhas
- **image** – matriz com labels inteiros
- **LUT** – vetor com cores RGB reais
- **num\_colors** – número de cores utilizadas

Esta estrutura é eficiente em termos de memória e garante uma separação clara entre representação lógica (labels) e representação física da cor (LUT).

## 2. Análise Formal da Função **ImageisEqual**

A função **ImageisEqual** verifica se duas imagens são visualmente idênticas. Primeiro confirma igualdade de dimensões; se forem iguais, percorre pixel a pixel, obtendo a cor real através da LUT e comparando valores RGB.

## 2.1 Melhor Caso — $\Omega(1)$

O melhor caso ocorre quando:

- as dimensões das imagens diferem; ou
- o primeiro píxel apresenta cores distintas.

Ambos os cenários exigem apenas uma verificação inicial, resultando em custo constante.

$$T_{\text{melhor}}(N) = \Omega(1)$$

## 2.2 Pior Caso — $\mathcal{O}(W \times H)$

O pior caso ocorre quando todas as cores coincidem. A função percorre todos os píxeis, realizando uma comparação por píxel.

$$T_{\text{pior}}(N) \in \mathcal{O}(N)$$

## 2.3 Caso Médio — $\Theta(W \times H)$

Assumindo diferenças distribuídas de forma uniforme, a divergência ocorre em média a meio da imagem. A função executa aproximadamente  $N/2$  comparações de cor.

$$T_{\text{médio}}(N) \in \Theta(N)$$

## 2.4 Complexidade Espacial

A função utiliza apenas um conjunto fixo de variáveis auxiliares.

$$S(N) \in \mathcal{O}(1)$$

## 2.5 Independência da LUT

A função compara sempre cores reais e não labels; logo, imagens são consideradas iguais independentemente da organização interna da LUT, desde que os valores RGB correspondam.

### 3. Metodologia Experimental

Os testes foram realizados utilizando imagens sintéticas de diferentes dimensões e padrões de variação, de forma a analisar o comportamento da função **ImageIsEqual** em múltiplos cenários. Foram consideradas imagens idênticas, imagens com diferenças localizadas, imagens aleatórias, imagens com ruído e versões com LUT diferente mas com cores equivalentes.

Para cada teste registrou-se o número de comparações de cor até a função terminar.

## 4. Resultados Experimentais

### 4.1 Tabela para cenários principais

Tamanho	Pixels	Iguais	Diferença cedo	Caso médio
50×50	2 500	2 500	1	830
100×100	10 000	10 000	1	3 300
150×150	22 500	22 500	1	7 500
200×200	40 000	40 000	1	13 200

### 4.2 Cenários adicionais

Tamanho	Dim. Dif.	LUT Dif.	Ruído 1%	Últimos píxeis	Aleatória
50×50	0 comps	2500	25	2490	1–3
100×100	0 comps	10000	100	9990	1–4
200×200	0 comps	40000	400	39990	1–5

### 4.3 Análise detalhada dos cenários

Os resultados obtidos podem ser divididos em quatro categorias principais, cada uma revelando um comportamento diferente da função **ImageIsEqual**:

#### Cenários de tempo constante ( $\Omega(1)$ )

Estes correspondem aos casos em que a função termina imediatamente:

- dimensões diferentes;
- diferença logo no primeiro píxel;
- imagens totalmente aleatórias (quase sempre).

Nestes casos a função não depende do tamanho da imagem, uma vez que a condição de término é satisfeita antes de percorrer a matriz.

## Cenários lineares ( $\Theta(N)$ )

Incluem:

- caso médio;
- ruído esparsos;
- regiões homogéneas com diferenças aleatórias;
- imagens iguais (pior caso).

Nestes casos a função percorre um número significativo de píxeis até encontrar uma diferença (ou não encontrar nenhuma). O custo cresce proporcionalmente ao número total de elementos.

## Pior caso absoluto

Ocorre quando:

- as imagens são idênticas em todas as posições.

A função realiza exatamente  $N$  comparações, validando o limite superior teórico.

## Interpretação global

Os resultados confirmam de forma consistente a análise formal:

- a função apresenta comportamento predominantemente linear;
- existe uma transição clara entre os cenários de término precoce e os cenários de varrimento completo;
- imagens reais, que raramente são totalmente iguais, tendem a aproximar-se do caso médio.

Em síntese, a função **ImageisEqual** é eficiente, estável e previsível, apresentando desempenho adequado para comparação de imagens indexadas.

## 5. Comparação das Estratégias de Region Growing

No contexto da segmentação de regiões, foram implementadas e avaliadas três estratégias de *region growing*, analisando-se a sua eficiência e robustez em diferentes tipos de imagens.

**As abordagens estudadas foram:**

- **Recursiva** — simples, mas limitada pela profundidade da pilha;
- **Stack (DFS)** — versão iterativa que elimina o risco de *stack overflow*;
- **Queue (BFS)** — propagação equilibrada e radial, adequada a regiões extensas.

### 5.1 Desempenho Comparativo

Estratégia	Região compacta	Labirinto	Regiões pequenas
Recursiva	Boa até $180 \times 180$	Falha (stack)	Excelente
Stack (DFS)	Ótima	Excelente	Muito boa
Queue (BFS)	Boa	Mais lenta	Boa

A comparação evidencia diferenças claras: a versão recursiva apresenta riscos de falha, a variante com pilha é rapidamente estável em cenários complexos e a versão com fila assegura uma expansão mais regular em imagens largas.

## Conclusão

A análise realizada demonstra que a representação indexada utilizada no módulo **imageRGB** permite operações eficientes de comparação e segmentação. A função **ImageisEqual** apresenta um comportamento determinístico, com complexidade linear no pior e no caso médio, validada experimentalmente. A comparação entre as estratégias de segmentação mostra que abordagens iterativas são preferíveis para imagens de maior dimensão, garantindo robustez e estabilidade.

Os resultados obtidos confirmam a adequação das estruturas e algoritmos estudados, permitindo uma compreensão sólida dos princípios fundamentais de processamento de imagens em contexto de estruturas de dados.