

UNIVERSIDADE TIRADENTES - UNIT
CIÊNCIA DA COMPUTAÇÃO
PROCESSAMENTO DE IMAGENS DE C GRÁFICA

ÁTILA GOMES VIANA
JOÃO PEDRO DINIZ GUERRA ROSA CRUZ
JOÃO VINÍCIUS VIEIRA MENEZES GÓIS
MATHEUS DE SENA TABOSA FERREIRA
ROOSEVELT ANDRADE ALVES

PROJETO CONVERSÃO IMAGENS CORREÇÃO

ARACAJU - SE
2025

ÁTILA GOMES VIANA
JOÃO PEDRO DINIZ GUERRA ROSA CRUZ
JOÃO VINÍCIUS VIEIRA MENEZES GÓIS
MATHEUS DE SENA TABOSA FERREIRA
ROOSEVELT ANDRADE ALVES

PROJETO CONVERSÃO IMAGENS CORREÇÃO

ATIVIDADE sobre Conversão de imagens para escala de cinza
apresentado como requisito parcial da avaliação da disc-
Iplina Processamento de Imagens de C gráfica,
ministrada pela Prof. LAYSE SANTOS SOUZA,
no 2º semestre de 2025.

ARACAJU - SE
2025

Sumário

1. Resumo
 2. Análise e Planejamento Inicial
 - 2.1. Objetivo Geral
 - 2.2. Objetivos Específicos
 - 2.3. Público-Alvo
 3. Escopo do Projeto
 - 3.1. Funcionalidades Essenciais (MVP)
 - 3.2. Funcionalidades Extras
 4. Requisitos do Sistema
 - 4.1. Requisitos Funcionais (RF)
 - 4.2. Requisitos Não Funcionais (RNF)
 5. Planejamento de Fluxo e Responsabilidades
 - 5.1. Pesquisa de Algoritmos e Preservação de Detalhes
 - 5.2. Planejamento do Fluxo de Processamento
 - 5.3. Organização de Tarefas e Responsabilidades
 6. Cronograma de Atividades
 7. Técnicas Adicionais Aplicáveis
 8. Metodologia Aplicada no Desenvolvimento do Sistema
 9. Conclusão
-

1. Resumo

Este documento detalha o planejamento e a estrutura do projeto de desenvolvimento de uma aplicação de software para a conversão de imagens coloridas para escala de cinza. O objetivo é criar uma ferramenta eficiente que preserve a fidelidade visual da imagem original, com foco em retratos e fotos de natureza. O projeto inclui a implementação de múltiplos algoritmos de conversão, uma interface de usuário simplificada e funcionalidades como processamento em lote e análise de histograma. O público-alvo engloba profissionais da saúde, estudantes e designers. A documentação aborda os requisitos funcionais e não funcionais, o fluxo de processamento, o cronograma e a divisão de tarefas entre a equipe.

Palavras-chave: Processamento de Imagens, Escala de Cinza, Conversão de Imagens, Python, Interface Gráfica.

2. Análise e Planejamento Inicial

2.1. Objetivo Geral

Criar uma aplicação de software que converta imagens coloridas em preto e branco (escala de cinza) com qualidade e eficiência. O foco é manter a fidelidade visual, principalmente em retratos de pessoas e fotos de natureza, preservando detalhes importantes como texturas, contraste e iluminação.

2.2. Objetivos Específicos

- **Testar Diferentes Técnicas:** Implementar e disponibilizar mais de um algoritmo de conversão (como luminosidade, média e decomposição), permitindo que o usuário compare e escolha o resultado que achar melhor.
- **Preservar Detalhes:** Dar prioridade a métodos que mantenham a qualidade da imagem original.

- **Interface Simples:** Desenvolver uma interface amigável, fácil de usar e acessível.
- **Documentação:** Produzir material que explique o processo de desenvolvimento, escolhas feitas e testes realizados.

2.3. Público-Alvo

O sistema é pensado para três grupos principais:

1. Profissionais da área da saúde;
 2. Estudantes e acadêmicos;
 3. Designers e criadores de conteúdo.
-

3. Escopo do Projeto

3.1. Funcionalidades Essenciais (MVP - Produto Mínimo Viável)

- Upload de imagens em formatos comuns (JPG, PNG).
- Conversão padrão usando um método base.
- Comparação lado a lado da imagem original da imagem convertida.
- Download da imagem convertida.

3.2. Funcionalidades Extras (para depois do MVP)

- Escolha de diferentes algoritmos de conversão.
 - Processamento em lote.
 - Exibição de informações básicas da imagem.
-

4. Requisitos do Sistema

4.1. Requisitos Funcionais (RF)

- **RF01:** O sistema deve permitir que o usuário faça o upload de imagens nos formatos JPG, JPEG e PNG.
- **RF02:** O sistema deve exibir a imagem original (colorida) após o upload.
- **RF03:** O sistema deve converter a imagem carregada para escala de cinza.
- **RF04:** O sistema deve exibir a imagem convertida (em escala de cinza) ao lado da imagem original para comparação.
- **RF05:** O sistema deve permitir que o usuário faça o download da imagem convertida em seu formato original.
- **RF06:** O sistema deve fornecer uma opção para o usuário selecionar entre pelo menos três algoritmos de conversão diferentes (Logaritmo, Quadrado e Exponencial).
- **RF07:** O sistema deve atualizar a imagem em escala de cinza em tempo real quando um novo algoritmo de conversão for selecionado.
- **RF08:** O sistema deve permitir o upload de múltiplas imagens para processamento em lote.
- **RF09:** O sistema deve permitir que o usuário visualize individualmente os canais de cor da imagem original e da convertida.
- **RF10:** O sistema deve incluir uma função para inverter as cores da imagem em escala de cinza (efeito de negativo).
- **RF11:** O sistema deve exibir um histograma de tons de cinza para a imagem convertida.
- **RF12:** O sistema deve exibir as dimensões (largura e altura em pixels) da imagem carregada.
- **RF13:** O sistema deve ser capaz de processar imagens de retratos e de natureza, conforme especificado no escopo.
- **RF14:** O sistema deve exibir uma mensagem de erro clara se o usuário tentar carregar um formato de arquivo não suportado.
- **RF15:** O sistema deve ter uma função para limpar a tela, removendo as imagens atuais e permitindo um novo upload.

4.2. Requisitos Não Funcionais (RNF)

- **RNF01:** O processamento principal deve ser concluído em menos de 5 segundos para imagens de até 10MB.
 - **RNF02:** A interface deve permitir conversão em no máximo 4 cliques.
 - **RNF03:** Compatibilidade com as últimas versões dos navegadores Chrome, Firefox e Edge.
 - **RNF04:** O sistema deve lidar com erros de upload exibindo mensagens claras.
 - **RNF05:** A imagem convertida não deve apresentar artefatos visuais significativos.
 - **RNF06:** O código deve ser otimizado para ambiente Colab sem esgotar memória.
 - **RNF07:** O código deve seguir padrões e ser comentado para manutenção.
 - **RNF08:** O sistema deve ser implantável em diferentes servidores.
 - **RNF09:** O notebook não deve salvar permanentemente imagens, apenas temporariamente.
 - **RNF10:** A documentação deve seguir normas ABNT.
-

5. Planejamento de Fluxo e Responsabilidades

5.1. Pesquisa de Algoritmos e Preservação de Detalhes

Para atender ao objetivo de "manter a fidelidade visual", foi realizada uma análise sobre como diferentes algoritmos impactam o resultado final. A escolha de múltiplos métodos de conversão é fundamental, pois cada um interpreta as cores de forma distinta, influenciando diretamente o contraste, a textura e a percepção de profundidade:

- **Método de Luminosidade:** Utiliza pesos diferentes para os canais de cor (Vermelho, Verde e Azul), aproximando-se da percepção humana. Tende a preservar melhor o contraste e os detalhes em retratos e paisagens, sendo a escolha prioritária para o requisito de qualidade.
- **Método da Média:** Calcula a média simples dos valores dos três canais de cor. É computacionalmente mais rápido, mas pode resultar em perda de contraste em áreas onde o brilho percebido é mais importante que a cor.

- **Métodos de Decomposição (Mínimo/Máximo):** Usam o valor mínimo ou máximo entre os canais de cor. São menos comuns para fins de fidelidade, mas podem ser úteis para análises específicas ou efeitos artísticos.

Essa abordagem permitirá ao usuário final escolher o algoritmo que melhor se adapta à sua necessidade, seja para análise técnica ou para fins estéticos.

5.2. Planejamento do Fluxo de Processamento

O fluxo de processamento foi detalhado para esclarecer as etapas técnicas dentro de cada célula de execução (Upload, Processamento, Interatividade e Download) e segue as bibliotecas para implementação.

Fluxograma de Processamento

- **Início**
- **Upload da Imagem (RF01, RF14)**
- **Exibir Imagem Original (RF02)**
- **Converter p/ Escala de Cinza (RF03, RF06)**
- **Exibir Comparaçāo (RF04, RF12)**
- **Opções Extras: Histograma, Negativo, Canais (RF07-RF11)**
- **Download da Imagem (RF05)**
- **Fim**

Bibliotecas

- **OpenCV**
- **Matplotlib**
- **NumPy**
- **Pillow**
- **IPython Widgets**
- **Scikit-image**

5.3. Organização de Tarefas e Responsabilidades

Com uma equipe de cinco integrantes, as responsabilidades foram divididas de forma específica para otimizar o desenvolvimento, testes e documentação:

- **João Vinícius Vieira Menezes Góis (Desenvolvedor de Backend):** Responsável por pesquisar, implementar e otimizar os algoritmos de conversão para escala de cinza (ex: luminosidade, média) e as lógicas de processamento de imagem, como a geração de histograma e o efeito negativo.
- **João Pedro Diniz Guerra Rosa Cruz (Desenvolvedor de Frontend/Interface):** Focado no desenvolvimento da interface do usuário no ambiente Colab, incluindo os componentes de upload, a exibição lado a lado das imagens, os widgets interativos para a seleção de algoritmos e os botões de download.
- **Átila Gomes Viana (Desenvolvedor de Funcionalidades Adicionais):** Encarregado de implementar as funcionalidades extras, como o processamento em lote (RF08) e a lógica para a visualização dos canais de cor individuais da imagem (RF09).
- **Matheus de Sena Tabosa Ferreira (Analista de Qualidade e Testes):** Responsável por validar todos os requisitos funcionais e não funcionais, executar os testes de desempenho (RNF01), usabilidade (RNF02) e compatibilidade entre navegadores (RNF03).
- **Roosevelt Andrade Alves (Líder de Projeto e Documentador):** Encarregado de supervisionar o cronograma, garantir a coesão do projeto e liderar a elaboração da documentação técnica e final, assegurando a conformidade com os padrões definidos (RNF07, RNF10).

6. Cronograma de Atividades

- **Semana 1:** Análise e Planejamento inicial.
- **Semana 2:** Modelagem do sistema e funções.

- **Semana 3:** Documentação parcial e Apresentação.
 - **Semana 4:** Documentação e análise dos resultados
-

7. Técnicas Adicionais Aplicáveis

- Dithering (Floyd-Steinberg)
 - Detecção de Bordas (Sobel ou Canny)
 - Correção Gama
 - Binarização (Thresholding)
 - Filtros de Nitidez (Unsharp Mask)
-

8. Metodologia Aplicada no Desenvolvimento do Sistema

Metodologia de Desenvolvimento: O projeto foi estruturado em módulos independentes dentro de um notebook interativo do Google Colab, utilizando a linguagem Python e bibliotecas de código aberto para processamento e análise de imagens. Cada módulo corresponde a uma etapa da pipeline de pré e pós-processamento de imagens, com foco em métricas objetivas de qualidade (Ruído, Foco e SSIM) e transformações controladas de contraste.

Módulo 0 – Conversão de Escala de Cinza: Implementou-se uma comparação entre diferentes métodos de conversão (padrão OpenCV, média e canais RGB), avaliando o desempenho por meio de métricas quantitativas (Contraste RMS, Michelson e Entropia). Essa abordagem garante a escolha do método mais adequado para cada imagem analisada.

Módulo 1 – Filtros de Suavização: Foram aplicados filtros Gaussiano, Mediano e Bilateral, avaliando a redução de ruído e preservação de foco com base nas métricas de variância do Laplaciano e estimativa de ruído via scikit-image. A comparação de

resultados foi automatizada em uma tabela interativa com pontuação ponderada (“Score Final”), determinando o filtro mais eficaz.

Módulo 2 – Teste de Estresse (Ruído Sintético): Introduziu-se ruído artificial (Gaussiano e Sal e Pimenta) em diferentes intensidades, simulando condições adversas de captura. Os filtros foram novamente avaliados quanto à robustez, mantendo a mesma estrutura de comparação quantitativa. Essa etapa valida a estabilidade do sistema frente a variações externas.

Módulo 3 – Transformações de Contraste e Manipulações: Aplicaram-se transformações de intensidade (Logarítmica, Quadrática, Exponencial e Equalização de Histograma), permitindo ao usuário analisar o impacto no brilho e contraste médio. Também foi incluída a opção de inversão de cores, simulando efeitos negativos. As análises foram acompanhadas de histogramas e tabelas de variação percentual.

Módulo 4 – Download em Lote (ZIP): Implementou-se uma rotina automatizada para exportar todas as versões processadas de uma imagem (original, convertida, filtrada, ruidosa e transformada) em um arquivo ZIP. Essa função permite reprodutibilidade e organização dos resultados de forma padronizada.

Bibliotecas e Ferramentas Utilizadas: O sistema foi construído utilizando as bibliotecas OpenCV (processamento de imagens), NumPy (operações matriciais), Matplotlib (visualização), Pandas (análise de dados), scikit-image (métricas e ruído sintético), e ipywidgets (para interatividade). O uso do ambiente Google Colab foi escolhido pela compatibilidade e facilidade de execução em nuvem.

9. Conclusão

A metodologia adotada garantiu um ambiente experimental completo, combinando usabilidade, rigor técnico e clareza visual. O sistema permite analisar com precisão diferentes aspectos do processamento digital de imagens, constituindo uma plataforma de referência para estudos e experimentos futuros na área.