



## Regras para Atividade

### Analizador de imagens para quantificação bioquímica

#### 1. Cenário

Em muitos cenários científicos, os dados devem ser coletados e analisados. Dado um cenário em que um especialista digitalizou dezenas de ensaios de Bradford ao longo do dia, é preciso extrair dados quantitativos dessas imagens (ver Figura 1).

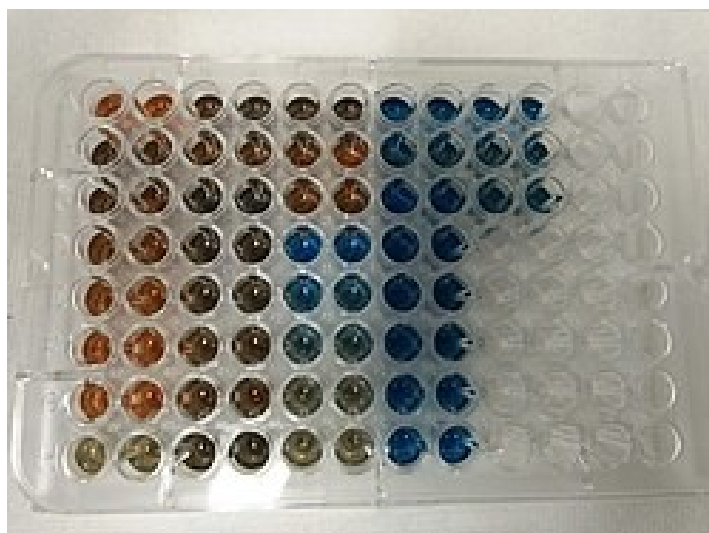


Figura 1 - Reação de cor entre proteína e reagente de Bradford

Neste projeto, os grupos deverão desenvolver um programa em Python que analisará um conjunto de imagens de ensaios de Bradford. Os estudantes deverão compor **equipes com 4 membros**. A **MATRÍCULA** e **NOME** dos estudantes (equipe) de ser enviado pelo **link “Equipes” no Moodle-UFU** até o dia **11/08/2025**.

A aplicação permitirá ao usuário carregar imagens, definir uma região de interesse (ROI), calibrar um modelo com base em imagens de amostras padrão e, em seguida, analisar imagens de amostras desconhecidas para prever a concentração de proteína. Essa é uma ótima oportunidade para explorar modelos de cores, análise quantitativa e métodos de compressão. O conjunto de dados para análise está disponibilizado no Moodle para uso neste projeto.

## 2. Objetivos

- Compreender o fluxo de trabalho de análise de dados de imagem, desde o carregamento até a extração de características.
- Desenvolver uma interface para navegação de arquivos e visualização de imagens.
- Implementar a lógica para carregar, exibir e manipular imagens estáticas com OpenCV e NumPy.
- Criar uma interface interativa para definir uma Região de Interesse (ROI) na imagem.
- Construir um pipeline de dados para calibração usando um conjunto de imagens de treinamento.
- Aplicar um modelo treinado para analisar novas imagens, individualmente ou em lote.
- Explorar modelos de cores para imagens digitais.

## 3. Descrição

O fluxo de trabalho será dividido em fases:

### A. Fase de Calibração

1. O usuário seleciona uma imagem do conjunto de calibração usando um botão "Carregar Imagem Padrão". A imagem é exibida.
2. O usuário define a Região de Interesse (ROI). Idealmente, a ROI pode ser desenhada ou movida com o mouse para um posicionamento preciso.
3. Implemente um algoritmo para detectar automaticamente os círculos no ensaio. A Transformada de Hough para Círculos (cv2.HoughCircles no OpenCV) é perfeita para isso.
4. Correção de Fundo: A iluminação pode não ser uniforme na imagem ou pode variar entre as fotos, afetando a medição da cor. Permita que o usuário defina uma segunda ROI em uma área "em branco" da imagem (um poço sem reagente ou uma área entre os poços). O valor médio de cor dessa ROI de fundo seria subtraído (ou normalizado) do valor da ROI da amostra.
5. O usuário digita a concentração conhecida correspondente àquela imagem (ex: "100") em um campo de texto.
6. Ao clicar em "Adicionar Ponto de Calibração", a aplicação calcula a cor média dentro da ROI e armazena o par (Cor, Concentração).
7. O processo é repetido para todas as imagens de calibração.
8. O botão "Calibrar Modelo" treina o modelo de Regressão Linear e o deixa pronto para uso.

### B. Fase de Análise

1. O usuário carrega uma imagem de amostra desconhecida.
2. A aplicação aplica automaticamente a mesma ROI definida durante a calibração.
3. Ela calcula a cor média e usa o modelo treinado para prever a concentração.
4. O resultado da imagem atual é exibido na tela.
5. O programa deve permitir ao usuário selecionar uma pasta inteira de imagens desconhecidas.
6. A aplicação processa todas elas, uma por uma, e gera um arquivo de resultados (ex: .csv) com o nome de cada imagem e sua concentração prevista.

**C. Modelo de Cor:** O usuário pode escolher qual modelo de cor usar para a análise:

1. RGB
2. HSV
3. L\*a\*b\*

**D. Características (Features):** Para os modelos HSV e L\*a\*b\*, será permitido escolher quais componentes usar para treinar o modelo e avaliar:

4. Apenas o Matiz (H)
5. Apenas o eixo b\*
6. Todos os três componentes (ex: H, S e V)

**E. Compressão:** Cria novas versões dessas imagens, salvas com diferentes níveis de compressão JPEG. Organiza as novas imagens em pastas separadas por nível de qualidade (ex: jpeg\_95, jpeg\_75, jpeg\_50, jpeg\_25). A compressão JPEG geralmente opera no espaço de cor YCbCr, que separa luminância (brilho) de crominância (cor) e comprime muito mais os canais de cor do que o de brilho.

**F. Nanômetros:** Crie uma opção que converta valores RGB para nanômetros (nm) no espectro visível da luz (aproximadamente de 380 a 780 nm) e apresente o gráfico de espectral de intensidade.

## 4. Bibliotecas em Python

- **PySimpleGUI:** para criar a interface, incluindo seletores de arquivos e elementos de exibição de imagem.
- **OpenCV (opencv-python).** Será usado para carregar imagens de arquivos (cv2.imread), converter espaços de cor e desenhar a ROI na imagem (cv2.rectangle).
- **NumPy** para extrair a ROI e calcular a cor média.
- **Scikit-learn** para uso de regressão linear (LinearRegression) para definição de modelo.
- **Matplotlib** para visualizar a curva de calibração.

## 5. Diretrizes e o Uso Ético

Transparência é Obrigatória: Deve incluir uma seção no relatório final chamada "Uso de Ferramentas de IA Generativa". Nela, eles devem detalhar:

1. Quais LLMs usaram (ChatGPT-4, Claude, Co-pilot, etc.).
2. Para quais tarefas (geração de código, debugging, redação, etc.).
3. Incluir os prompts que criaram e que foram particularmente úteis.
4. Deixar marcado no fonte as partes implementadas pelas LLMs e as adaptações necessárias para funcionamento do código.

## Avaliação

## 6. Entregáveis

1. **Software:**

- O código-fonte completo do projeto Python (arquivo .py e requirements.txt).
  - Um conjunto de imagens de amostra para teste (ex: 3 imagens de calibração e 3 de análise).
2. **Vídeo curto (pitch):** mostrando o fluxo de trabalho: carregar uma imagem, definir a ROI, calibrar com várias imagens e, finalmente, analisar uma imagem desconhecida.
  3. **Relatório:** descrevendo o protocolo experimental, como as imagens e os erros calculados. Apresentar tabelas com os dados de erro e o gráfico final que mostra a relação entre o tamanho do arquivo e o erro de medição, interpretando o gráfico e fornecendo uma recomendação clara sobre o nível de compressão aceitável, modelo de cores para esta aplicação específica.
  4. **Apresentação:** os grupos deverão apresentar o trabalho em **10/09/2025** com tempo de 15 minutos para mostrar o desenvolvimento e análise.