

Universidade Federal do Ceará

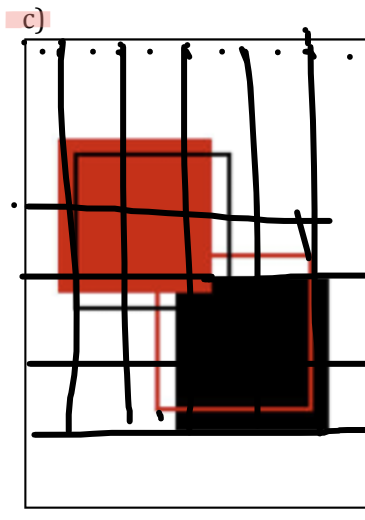
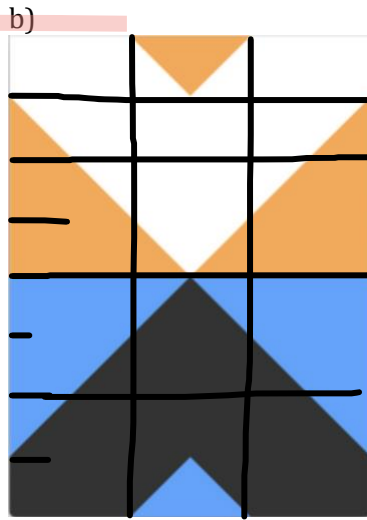
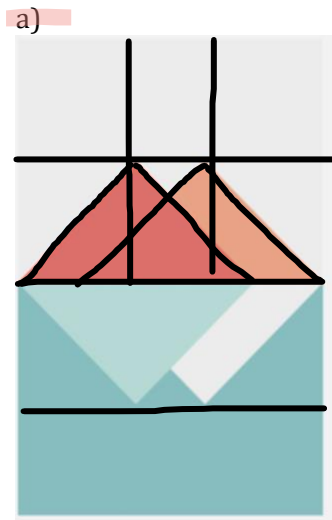
Sistemas e Mídias Digitais

Lista 1 – Exercícios de Programação I

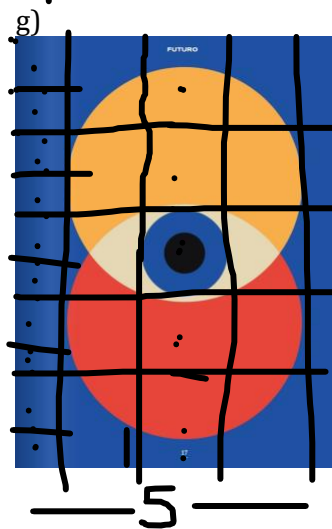
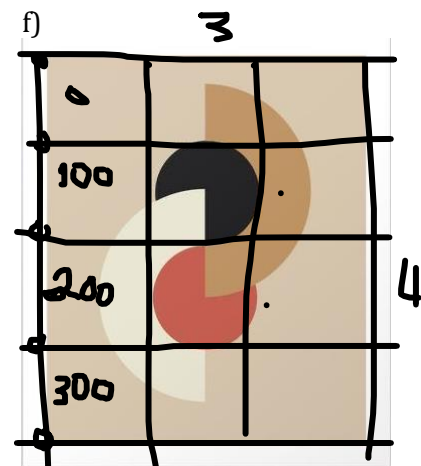
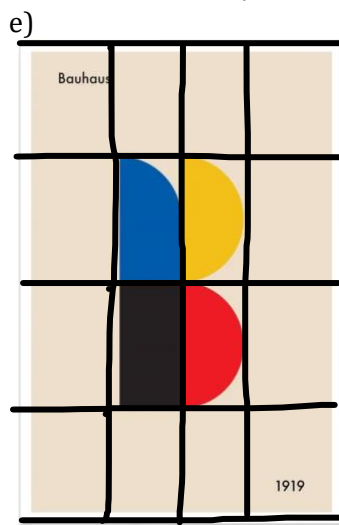
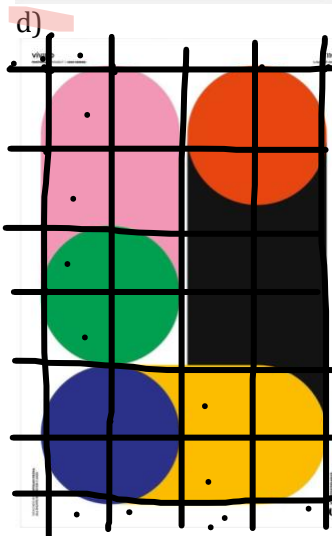
Prof. George Gomes

1. **Objetivo: Da Análise à Sintaxe.** O objetivo principal é traduzir uma análise visual em código. Primeiro, decompõe a imagem em suas formas geométricas básicas. Em seguida, use as primitivas do Processing (`rect()`, `ellipse()`, `triangle()`, etc.) para recriá-las. Para posicionar tudo corretamente, será fundamental dominar o sistema de coordenadas da tela (com o eixo Y invertido) e pensar nas proporções e alinhamento dos elementos.
2. **A Ordem do Desenho Define a Sobreposição.** O Processing desenha as formas na sequência em que aparecem no código. Elementos desenhados por último sempre ficarão na frente. Planeje a ordem dos seus comandos para garantir que as camadas e sobreposições fiquem corretas, como no exercício g).
3. **Cores e Contornos.** A fidelidade visual depende do uso correto das cores. Utilize `fill()` para o preenchimento e `stroke()` para o contorno. Caso uma forma não tenha borda, use `noStroke()`. Uma boa dica é usar um seletor de cores (*color picker*) para capturar os tons exatos das imagens de referência.
4. **Pense em Proporções, Não em Pixels.** Evite usar valores fixos (ex: 100, 250, etc.). Em vez disso, programe de forma proporcional usando as variáveis `width` e `height` para definir posições e tamanhos (ex: `ellipse(width/2, height/2, ...)`). Isso torna seu desenho mais adaptável a diferentes tamanhos de tela.
5. **Compartilhe!** Desenvolva cada item em um sketch distinto na IDE Processing. Compartilhe as soluções no Discord.

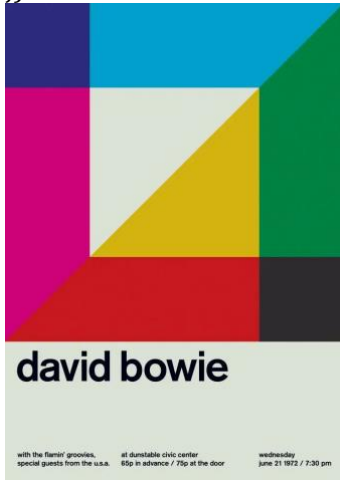
Recrie as seguintes peças gráficas utilizando as primitivas geométricas: `rect()`, `ellipse()`, `triangle()`, `arc()`, etc.



500, 600



j)



l)



m)

