```
PImage img;

void setup() {
  img = loadImage("Lena.jpg"); // Carrega a imagem
  img.resize(256, 256); // Redimensiona para facilitar o processamento
  img.filter(GRAY); // Converte para escala de cinza

  float media = calcularMedia(img);
  float desvioPadrao = calcularDesvioPadrao(img, media);

  println("Média: " + media);
  println("Desvio padrão: " + desvioPadrao);

  PImage imgAutoEscala = autoEscala(img);
  image(img, 0, 0);
  image(imgAutoEscala, 256, 0);
}

float calcularMedia(PImage img) {
  float soma = 0;
  for (int i = 0; i < img.width; i++) {
    for (int j = 0; j < img.height; j++) {
      soma += brightness(img.get(i, j));
    }
  }
  return soma / (img.width * img.height);
}

float calcularDesvioPadrao(PImage img, float media) {
  float soma = 0;
  for (int i = 0; i < img.width; i++) {
    for (int j = 0; j < img.height; j++) {
      float brilho = brightness(img.get(i, j));
      soma += pow(brilho - media, 2);
    }
  }
  return sqrt(soma / (img.width * img.height));
}

PImage autoEscala(PImage img) {
  PImage result = img.copy();

  float min = 255;
  float max = 0;
  for (int i = 0; i < img.width; i++) {
    for (int j = 0; j < img.height; j++) {
      float brilho = brightness(img.get(i, j));
      if (brilho < min) min = brilho;
      if (brilho > max) max = brilho;
    }
  }

  for (int i = 0; i < img.width; i++) {
    for (int j = 0; j < img.height; j++) {
      float brilho = brightness(img.get(i, j));
      float ajustado = map(brilho, min, max, 0, 255);
      result.set(i, j, color(ajustado));
    }
  }

  return result;
}

void draw() {
}
```