

INSTITUTO SUPERIOR POLITECNICO DE TECNOLOGIA E CIÊNCIA

DEPARTAMENTO DE ENGENHARIA

ENGENHARIA INFORMATICA

COMPUTAÇÃO GRÁFICA

António Kipanda Cardoso

Edivaldo Maciel de Almeida Andrade

Fanuel Henriques de Oliveira Quimbango

**IMPLEMENTAÇÃO DE UM CONVERSOR DE CORES UTILIZANDO O  
PROCESSING**

LUANDA  
2020

# RESUMO

Este projecto apresenta o desenvolvimento de um Conversor de Cores, usando os conhecimentos adquiridos em Processing na disciplina de Computação Gráfica.

A conversão baseia-se na passagem de uma cor em um dos sistemas (RGB, HSV ou CMYK), e converte para estes mesmos sistemas.

Primeiramente será apresentada uma breve introdução, à seguir, os conceitos do Processing e sobre as cores. No capítulo seguinte é apresentado a implementação do projecto e logo em seguida, a secção dos testes realizados.

**Palavras-chave:** Conversor de cores. Processing, Computação Gráfica. Implementação.

# SUMÁRIO

1.	CONCEITOS GERAIS.....	1
1.1.	PROCESSING.....	1
1.1.1.	CARACTERÍSTICAS .....	1
1.1.2.	PORQUÊ O PROCESSING?.....	1
1.2.	CMYK.....	2
1.3.	HSV .....	3
1.4.	RGB .....	3
2.	IMPLEMENTAÇÃO.....	4
2.1.	CONVERSÃO DE CMYK PARA RGB .....	4
2.2.	CONVERSÃO DE RGB PARA CMYK.....	5
2.3.	CONVERSÃO DE HSV PARA RGB.....	6
2.4.	CONVERSÃO DE RGB PARA HSV .....	7
4.	TESTES E RESULTADOS .....	9
	REFERÊNCIAS .....	12

# SUMÁRIO DE QUADROS E FIGURAS

Figura 1 - Variáveis globais do conversor.....	4
Figura 2 - Função responsável pela conversão CMYK/RGB.....	5
Figura 3 - Função responsável pela conversão RGB/CMYK.....	6
Figura 4 - Função responsável pela conversão HSV/RGB.....	7
Figura 5 - Função responsável pela conversão RGB/ HSV.....	8
Figura 6: Tela inicial do conversor de cores.....	9
Figura 7: Inserindo valores RGB e pressionando em “converter RGB” ele converte para os dois tipos de valores aceitáveis para cores.....	9
Figura 8: Tela mostrando a conversão para outros formatos... à direita podemos ver uma detalhe da cor que foi formada a partir dos valores descritos .....	10
Figura 9: Convertendo de CMKY para todos os outros .....	10
Figura 10: Resultado da conversão de CMYK para todos os outros, e mais uma vez ao lado vimos o resultado da conversão.....	10
Figura 11: Convertendo de HSV para todos os outros .....	11
Figura 12: Resultado da conversão de HSV para todos os outros, e mais uma vez ao lado vimos o resultado da conversão.....	11

# 1.CONCEITOS GERAIS

## 1.1. PROCESSING

Processing é uma linguagem de programação de código aberto e ambiente de desenvolvimento integrado (IDE), construído para as artes eletrônicas e comunidades de projetos visuais com o objetivo de ensinar noções básicas de programação de computador em um contexto visual e para servir como base para cadernos eletrônicos. O projeto foi iniciado em 2001 por Casey Reas e Ben Fry, ambos ex-membros do Grupo de Computação do MIT Media Lab.

Um dos objetivos do Processing é atuar como uma ferramenta para não-programadores iniciados com a programação, através da satisfação imediata com um retorno visual. A linguagem tem por base as capacidades gráficas da linguagem de programação Java, simplificando características e criar alguns novos.

### 1.1.1. CARACTERÍSTICAS

Processing é considerado um sketchbook, uma alternativa de organização de projetos sem ser o um IDE padrão.

Cada esboço (sketch) de Processing é realmente uma subclasse do Java PApplet classe que implementa a maioria das funcionalidades da Linguagem Processing.

Ao programar em Processing, todas classes adicionais definidas serão tratados como classes internas quando o código é traduzido para Java puro antes de compilar. Isso significa que o uso de variáveis e métodos estáticos em classes é proibido a menos que você diga que deseja o processamento para o código no modo de Java puro.

### 1.1.2. PORQUÊ O PROCESSING?

- Open Source
- Versão Beta (ainda) o que o torna ideal
- Multi-plataforma: MacOS, Win, Unix.
- Poder do JAVA
  - Simplicidade
  - Adequa-se ao processo de ensino\*

- Faz uma boa transição entre linguagens de baixo nível e scripting de muito alto nível;
- Documentação extensa.
- Há diversos websites dedicados à programação, especialmente em JAVA e muitos dedicados ao Processing – ver a secção de links;
- Extensível
- Exporta Executáveis e Applets para a Web
- Comunidade

## 1.2. CMYK

CMYK é a abreviatura do sistema de cores subtrativas formado por Ciano (Cyan), Magenta (Magenta), Amarelo (Yellow) e Preto (Black (Key para não confundir com o B de "Blue" no padrão Hi-Fi com RGB)).

O CMYK funciona devido à absorção de luz, pelo fato de que as colorações que são vistas vêm da parte da luz que não é absorvida. Este sistema é empregado por impressoras, impressoras e fotocopiadoras para reproduzir a maioria das cores do espectro visível, e é conhecido como quadricromia. É o sistema subtrativo de cores, em contraposição ao sistema aditivo, o RGB.

Ciano é a cor oposta ao vermelho, o que significa que actua como um filtro que absorve a dita cor ( $-R +G +B$ ). Da mesma forma, magenta é a oposta ao verde ( $+R -G +B$ ) e amarelo é a oposta ao azul ( $+R +G -B$ ). Assim, magenta mais amarelo produzirá vermelho, magenta mais ciano produzirá azul e ciano mais amarelo produzirá verde.

O preto pode ser produzido misturando os três pigmentos primários, mas por várias razões, é preciso adicionar tinta preta ao sistema:

- O preto que se cria misturando os três pigmentos primários não é puro, devido às impurezas encontradas neles;
- Empregar 100% das tintas ciano, magenta e amarelo produz uma camada que, dependendo do tipo de papel, pode não secar ou ainda romper a folha se muito fina;
- Os textos imprimem-se geralmente no preto pois incluem detalhes muito finos que seriam complicados de conseguir mediante a superposição de três tintas;

- O pigmento preto é o mais barato de todos, razão pela qual criar preto com três tintas seria muito mais caro.

### 1.3. HSV

HSV é a abreviatura para o sistema de cores formadas pelas componentes hue (matiz), saturation (saturação) e value (valor). O HSV também é conhecido como HSB (hue, saturation e brightness — matiz, saturação e brilho, respectivamente). Esse sistema de cores define o espaço de cor conforme descrito abaixo, utilizando seus três parâmetros:

- Matiz (tonalidade): Verifica o tipo de cor, abrangendo todas as cores do espectro, desde o vermelho até o violeta, mais o magenta. Atinge valores de 0 a 360, mas para algumas aplicações, esse valor é normalizado de 0 a 100%.
- Saturação: Também chamado de "pureza". Quanto menor esse valor, mais com tom de cinza aparecerá a imagem. Quanto maior o valor, mais "pura" é a imagem. Atinge valores de 0 a 100%.
- Valor (brilho): Define o brilho da cor. Atinge valores de 0 a 100%.

Este sistema foi inventado no ano de 1974, por Alvy Ray Smith. É caracterizada por ser uma transformação não-linear do sistema de cores RGB.

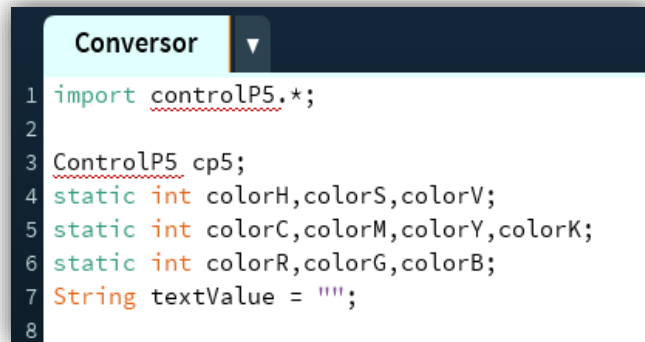
### 1.4. RGB

RGB é a abreviatura de um sistema de cores aditivas em que o Vermelho (Red), o Verde (Green) e o Azul (Blue) são combinados de várias formas de modo a reproduzir um largo espectro cromático. O propósito principal do sistema RGB é a reprodução de cores em dispositivos eletrônicos como monitores de TV e computador, retroprojetores, scanners e câmeras digitais, assim como na fotografia tradicional.

O modelo de cores RGB é baseado na teoria de visão colorida tricromática, de Young-Helmholtz, e no triângulo de cores de Maxwell. O uso do modelo RGB como padrão para apresentação de cores na Internet tem suas raízes nos padrões de cores de televisões RCA de 1953 e no uso do padrão RGB nas câmeras Land/Polaroid, pós Edwin Land.

## 2.IMPLEMENTAÇÃO

Para a implementação prática do conversor foram utilizadas as seguintes estruturas:



```

1 import controlP5.*;
2
3 ControlP5 cp5;
4 static int colorH,colorS,colorV;
5 static int colorC,colorM,colorY,colorK;
6 static int colorR,colorG,colorB;
7 String textValue = "";
8

```

*Figura 1: Variáveis globais do conversor.*

A biblioteca **controlP5** foi utilizada para a construção da interface, fazendo uso de diversos componentes, como caixas de textos e botões.

Para os 3 diferentes tipos de sistemas de cores, temos as variáveis do tipo *int* que descrevem a propriedade de cada um dos sistemas:

- CMYK – colorC, colorM, colorY, colorK;
- HSV – colorH, colorS, colorV;
- RGB – colorR, colorG, colorB.

### 2.1. CONVERSÃO DE CMYK PARA RGB

Os valores, vermelho, verde e azul são dados no intervalo de [0, 255], a cor vermelha(R) é calculada a partir das cores ciano(C) e preta(K), a cor verde(G) é calculada a partir da cor magenta(M) e preta(K), a cor azul(B) é calculada a partir das cores amarela(Y) e preta(K). Abaixo está a fórmula em CMYK para a conversão RGB.

$$\text{Red} = 255 \times (1 - \text{Cyan} \div 100) \times (1 - \text{Black} \div 100)$$

$$\text{Green} = 255 \times (1 - \text{Magenta} \div 100) \times (1 - \text{Black} \div 100)$$

$$\text{Blue} = 255 \times (1 - \text{Yellow} \div 100) \times (1 - \text{Black} \div 100)$$



```

74 public static void CMYKtoRGB(double c, double m, double y, double black){
75     c=c/100;
76     m=m/100;
77     y=y/100;
78     black=black/100;
79     colorR=(int)Math.round(255 * ( 1 - c) * ( 1 - black));
80     colorG=(int)Math.round(255 * ( 1 - m) * ( 1 - black));
81     colorB=(int)Math.round(255 * ( 1 - y) * ( 1 - black));
82 }

```

*Figura 2: Função responsável pela conversão CMYK/RGB.*

## 2.2. CONVERSÃO DE RGB PARA CMYK

O número máximo para os valores R, G e B é 255. Primeiramente, dividimos-os para ter um número que varia de 0.0 a 1.0, esta proporção será usada no cálculo.

$$Rc = R \div 255$$

$$Gc = G \div 255$$

$$Bc = B \div 255$$

A cor chave preta(K) pode fornecer muitos resultados. Quando assumimos um valor de chave preta, as outras três cores (ciano, magenta, amarela) podem ser calculadas. Podemos calculá-lo a partir da coloração vermelho, verde e azul, o número máximo da chave preta deve ser:

$$K = 1 - \max(Rc, Gc, Bc);$$

Ou podemos supor que acabamos a tinta preta, precisamos usar as restantes outras tintas de três cores para concluir o trabalho de impressão.

$$K = 0;$$

A cor ciano(C) é calculada a partir das cores vermelha e preta.

$$C = (1 - Rc - K) \div (1 - K)$$

A cor magenta(M) é calculada a partir das cores verde e preta.

$$M = (1 - Gc - K) \div (1 - K)$$

A cor amarela(Y) é calculada a partir das cores azul e preta.

$$Y = (1 - Bc - K) \div (1 - K)$$

```

84 public static void RGBtoCMYK(double r,double g,double b){
85     double rc,gc,bc,k,c,m,y;
86
87     rc=r/255;
88     gc=g/255;
89     bc=b/255;
90     k=1-max(rc,gc,bc);
91     c = (1 - rc - k) / (1 - k);
92
93     m = (1 - gc - k) / (1 - k);
94
95     y = (1 - bc - k) / (1 - k);
96
97     colorC=(int)Math.round(c*100);
98     colorM=(int)Math.round(m*100);
99     colorY=(int)Math.round(y*100);
100    colorK=(int)Math.round(k*100);
101 }
102

```

Figura 3: Função responsável pela conversão CMYK/RGB.

## 2.3. CONVERSÃO DE HSV PARA RGB

Seja uma cor definida por (H, S, V), onde H, varia de 0.0 a 360.0, informando o ângulo, em graus, no círculo onde esse parâmetro está definido, e com S e V variando de 0.0 a 1.0. A transformação para os parâmetros (R, G, B) desta cor podem ser calculados conforme as fórmulas abaixo:

Primeiramente, se  $S = 0$ , o resultado será cinza. Para este caso, os valores de R, G e B são iguais a V e, o valor de H é irrelevante.

Para S diferente de zero, as fórmulas abaixo são aplicáveis:

$$\begin{aligned}
 H_i &= \left\lfloor \frac{H}{60} \right\rfloor \bmod 6 \\
 f &= \frac{H}{60} - H_i \\
 p &= V(1 - S) \\
 q &= V(1 - fS) \\
 t &= V(1 - (1 - f)S) \\
 \text{if } H_i = 0 &\rightarrow R = V, G = t, B = p \\
 \text{if } H_i = 1 &\rightarrow R = q, G = V, B = p \\
 \text{if } H_i = 2 &\rightarrow R = p, G = V, B = t \\
 \text{if } H_i = 3 &\rightarrow R = p, G = q, B = V \\
 \text{if } H_i = 4 &\rightarrow R = t, G = p, B = V \\
 \text{if } H_i = 5 &\rightarrow R = V, G = p, B = q
 \end{aligned}$$

Estas fórmulas, dão R, G e B variando de 0.0 a 1.0.

```
public static void HSVtoRGB(double h,double s,double v){
    s=s/100;
    v=v/100;
    double c=v*s;
    double x=c*(1-Math.abs((h/60)%2-1));
    double m=v-c;
    double rc,gc,bc;
    if(h<60){
        rc=c; gc=x; bc=0;
    }else if(h<120){
        rc=x; gc=c; bc=0;
    }else if(h<180){
        rc=0; gc=c; bc=x;
    }else if(h<240){
        rc=0; gc=x; bc=c;
    }else if(h<300){
        rc=x; gc=0; bc=c;
    }else{
        rc=c; gc=0; bc=x;
    }

    colorR=(int) Math.round((rc+m)*255);
    colorG=(int) Math.round((gc+m)*255);
    colorB=(int) Math.round((bc+m)*255);
}
```

Figura 4: Função responsável pela conversão HSV/RGB.

## 2.4. CONVERSÃO DE RGB PARA HSV

Seja uma cor definida por (R, G, B), onde R, G e B estão entre 0.0 e 1.0, onde 0.0 e 1.0 são, respectivamente, o maior e o menor valor possível para cada. A transformação para os parâmetros (H, S, V) dessa cor pode ser determinada pelas fórmulas abaixo.

Seja MAX e MIN os valores máximo e mínimo, respectivamente, dos valores (R, G, B):

$$H = \begin{cases} 60 \times \frac{G-B}{MAX-MIN} + 0, & \text{if } MAX = R \\ & \text{and } G \geq B \\ 60 \times \frac{G-B}{MAX-MIN} + 360, & \text{if } MAX = R \\ & \text{and } G < B \\ 60 \times \frac{B-R}{MAX-MIN} + 120, & \text{if } MAX = G \\ 60 \times \frac{R-G}{MAX-MIN} + 240, & \text{if } MAX = B \end{cases}$$

$$S = \begin{cases} \frac{MAX-MIN}{MAX} & \text{if } MAX > 0 \\ 0 & \text{if } MAX = 0 \end{cases}$$

$$V = MAX$$

Os resultados dão a tonalidade variando de 0 a 360, indicando o ângulo no círculo aonde a tonalidade (H) está definido, e a saturação e o brilho variando de 0.0 a 1.0, representando o menor e o maior valor possível.

```
public static void RGBtoHSV(double r,double g,double b){
    double maxV,minV, h,s,v;

    maxV=max(r/255, g/255, b/255);
    minV=min(r/255, g/255, b/255);

    double delta=(maxV-minV);

    if(maxV!=0)
        s=delta/maxV;
    else
        s=0;

    v=maxV;

    if(delta==0){
        h=0;
    }else if(maxV==(r/255))
        h=60*((g/255 - b/255)/delta)%6;
    else if(maxV==(g/255))
        h=60*((b/255 - r/255)/delta)+2;
    else
        h=60*((r/255 - g/255)/delta)+4;
    //123 111 235
    colorH=(int)Math.round(h);
    colorS=(int)Math.round(s*100);
    colorV=(int)Math.round(v*100);
}
```

*Figura 5: Função responsável pela conversão RGB/HSV.*

## 4. TESTES E RESULTADOS

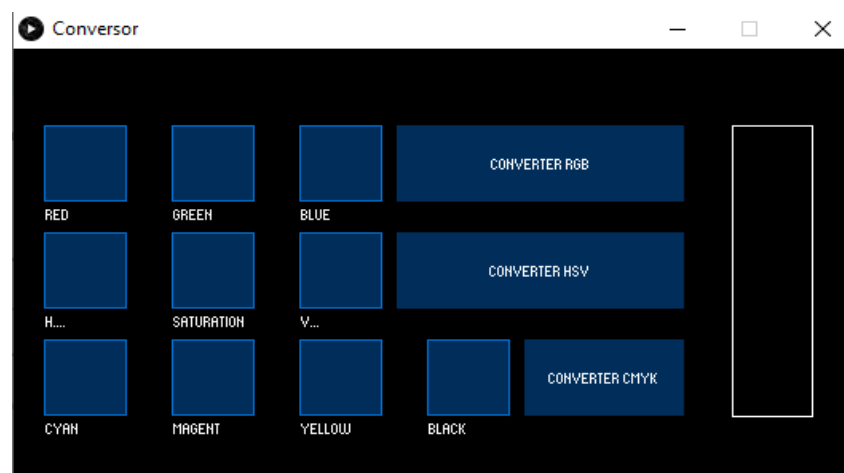


Figura 6: Tela inicial do conversor de cores

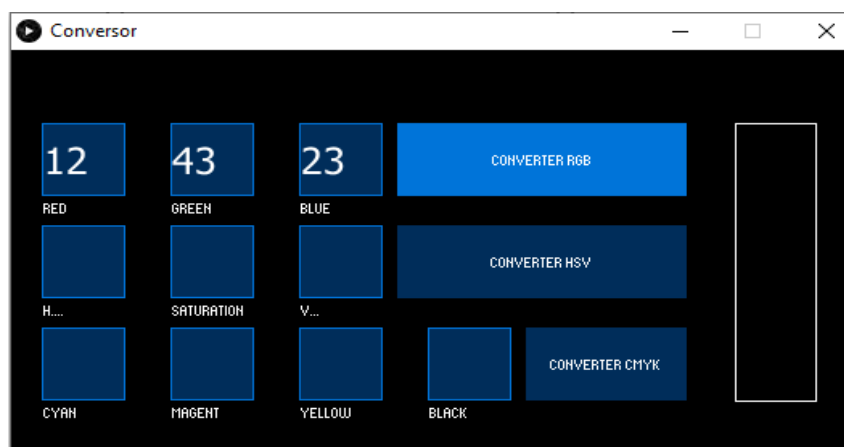


Figura 7: Inserindo valores RGB e pressionando em “converter RGB” ele converte para os dois tipos de valores aceitáveis para cores

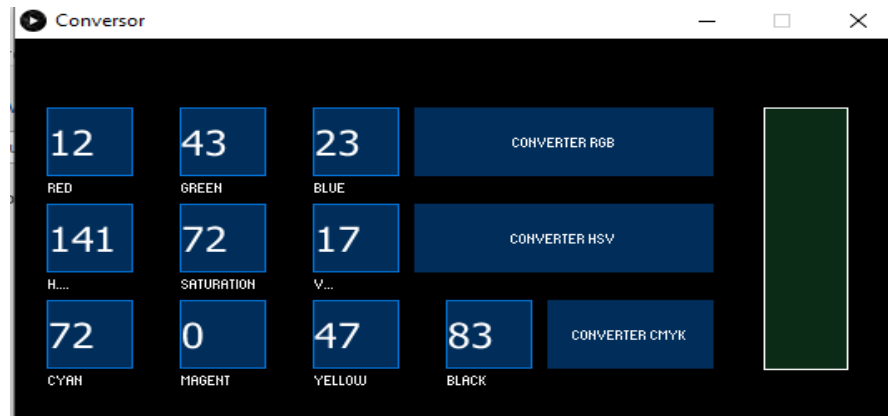


Figura 8: Tela mostrando a conversão para outros formatos... à direita podemos ver uma detalhe da cor que foi formada a partir dos valores descritos

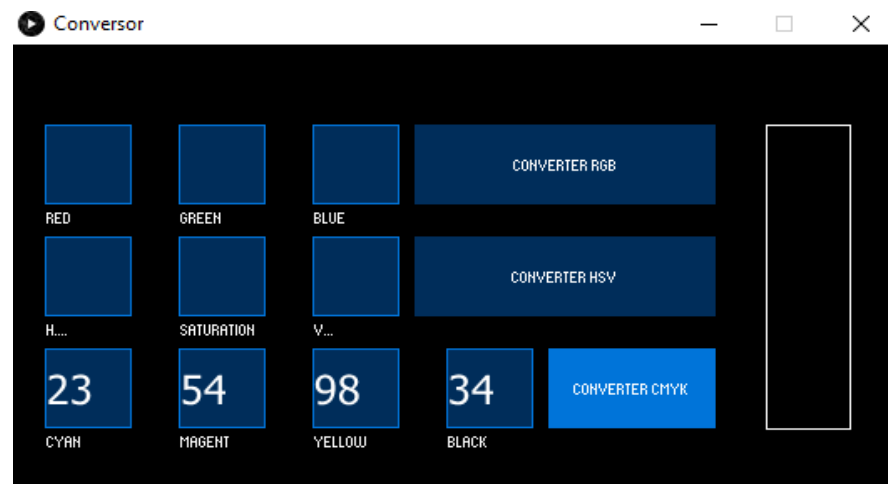


Figura 9: Convertendo de CMYK para todos os outros

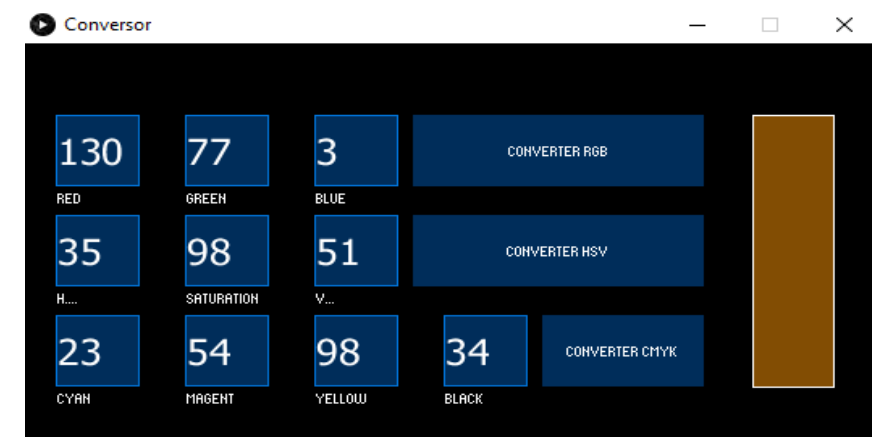


Figura 10: Resultado da conversão de CMYK para todos os outros, e mais uma vez ao lado vimos o resultado da conversão

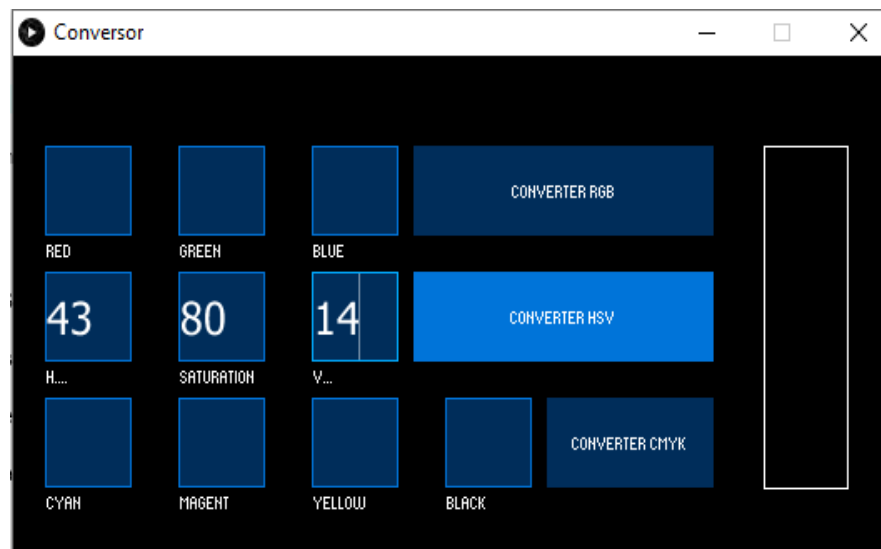


Figura 11: Convertendo de HSV para todos os outros

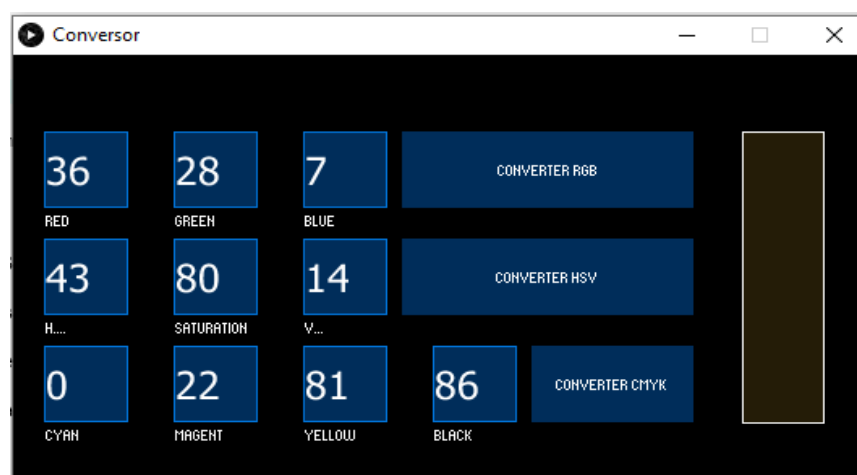


Figura 12: Resultado da conversão de HSV para todos os outros, e mais uma vez, ao lado vimos o resultado da conversão

## REFERÊNCIAS

- Processing (linguagem de programação). Disponível em: [https://pt.wikipedia.org/wiki/Processing\\_\(linguagem\\_de\\_programa%C3%A7%C3%A3o\)](https://pt.wikipedia.org/wiki/Processing_(linguagem_de_programa%C3%A7%C3%A3o))
- AMADO, Pedro. INTRODUÇÃO À PROGRAMAÇÃO GRÁFICA (USANDO PROCESSING). Versão 1.61. Porto: Creative Commons, 2006