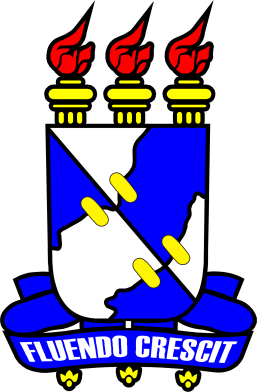
**UFS - UNIVERSIDADE FEDERAL DE SERGIPE** **CCET - CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
 DCOMP - DEPARTAMENTO DE COMPUTAÇÃO  
 CIÊNCIA DA COMPUTAÇÃO**

**PROGRAMAÇÃO ORIENTADA A OBJETOS**

**PAINT**

**PROF. ANDRE BRITTO DE CARVALHO**

**MURILO DOS ANJOS MONTINO - 201600017098**

**São Cristóvão - SE**

**Terça-Feira, 24 de Maio de 2017**

**MURILO DOS ANJOS MONTINO - 201600017098**

**PAINT**

**São Cristóvão – SE**

**2017**

**SUMÁRIO**

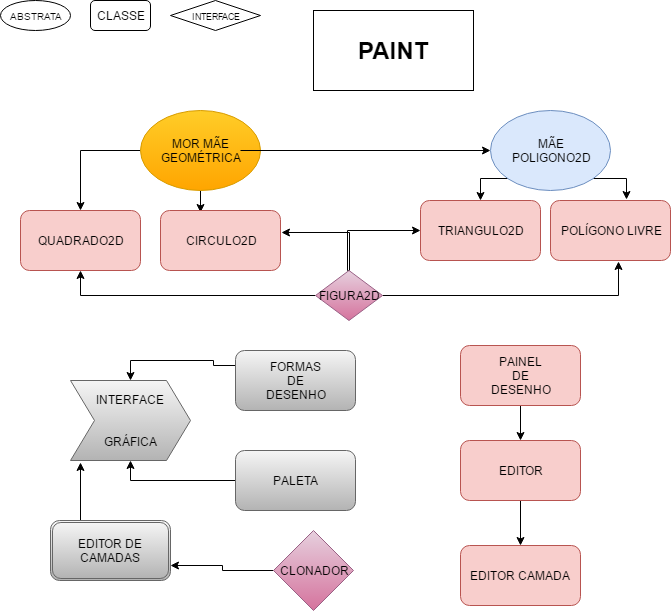
[1- Diagrama de Classes](#_Toc320880443) 4

[2- Classes e Principais Métodos](#_Toc320880444) 5

[3- Problemas, Soluções e como funcionam as classes](#_Toc320880445) 7

[4- Adições de recursos ao Projeto](#_Toc320880445) 9

# Diagrama de Classes



# Classes e principais métodos

1.0 FORMAS GEOMETRICAS

1.1 Geometrica

- AtualizarMOUSE

- setCor

- set Borda

1.2 Figura2D (Interface)

- draw

1.3 Circulo2D

- draw

- PontoAtualizado

- setRaio1

- setRaio2

1.4 Triangulo2D

- draw

1.5 Quadrado2D

- draw

- AtualizarMouse

1.6 PoligonoLivre

- draw

- AtualizarMouse

1.7 Poligono2D

- getVertices

2.0 AUXILIARES

2.1 Clonador

- Clone

- Desclonagem

- ExClonadorMaster

2.2 Editor

- clear

- SalvarFiguras

- carregarFiguras

- setDiretorio

- getDiretorio

- getDados

2.3 Painel Desenho

- addFigura

2.4 Editor Camada

3.0 JANELAS

3.1 Interface

- SalvarComo

- abrirArquivo

-As linhas de código que constrõem o desenho estão no construtor

3.2 Formas De Desenho

- Métodos Acessores e modificadores

3.3 Paleta

- Métodos Acessores e modificadores

3.4 Camadas

- AtualizarCor

- ATUALIZAR

- AtualizarOk

# Problemas, Soluções e como funcionam as classes

1.1   **Problema**: *Salvar e carregar arquivos*

As primeiras tentativas de solução buscavam a serialização da Classe Editor que continha todas estas variáveis, porém não era possível carregá-las novamente, pois salvava-se uma referência, logo para a solução encontrada foi:

Salva-se imagem por imagem como extensão “fig” enumeradas de um até o total. Utilizando-se de um contador de figuras, que é atualizado a medida que se desenha e salva essa quantidade em um arquivo txt. Depois salva as figuras delimitando onde terminar pelo contador de figuras.

Carrega primeiramente o arquivo txt, para delimitar o limite do for e carrega as figuras individualmente.

Como salva assim e para não ficar arquivos dispersos, é criada uma nova pasta com o nome a ser escolhido pelo usuário.

Uma variável string guarda o nome do diretório.

Uma variável int guarda a quantidade de figuras.

1.2   *Edição de camadas*

Usou-se uma classe auxiliar chamada Clonador, para clonar as informações e reduzir seu tamanho a níveis redimensionados e para aplicar as alterações somente foi necessário desclonar e trocar a referência do objeto principal para objetos desclonados do auxiliar.

Cada botão é uma edição diferente e o atualizar é pegar as informações da tela principal novamente mas repetindo-se o processo de clonagem com redimensionamento.

Na edição de camadas a cada modificação de variáveis, atualiza-se toda a lista.

Um editor auxiliar guarda as informações redimensionadas do editor principal.

Um Arraylist guarda as informações do editor principal sem redimensioná-las.

Ambas são clones.

1.3   *Para desenhar*

Foi criado uma classe abstrata(geométrica) com vários gets e sets e um método abstrato para movimentar a posição central do desenho(no editor de camadas) e utilizou-se de uma interface para desenho(Editor). Cada filho da classe principal(Geométrica) representa uma figura desenhavel, tirando Poligono2D que é outra classe abstrata, porém filha.

1.4   *Organização das Janelas*

Por não se saber como organizar paleta, formas de desenho e editor de camadas na interface principal, decidiu-se colocá-las individualmente.

1.5   **Problema**: *Pintando os desenhos na tela*

É feita na interface do seguinte modo, ele recebe um valor de qual figura é, valor recebido de um get sobre a classe Formas de desenho, e o coloca em seu controlador e a medida que é clicada na tela, ele pinta a imagem desejada. Mas não somente essa informação, estão inclusas: Raio, Altura, base, cor e cor de borda(caso o usuário queira).

1.6   *Paleta*

Usou-se um Jpainel para representar cada cor, e caixas RGB se o usuário desejar.

A cada click com o mouse na cor desejada, é enviada a informação a uma variável Color.

A interface chama essa variável quando é necessário desenhar.

1.7   *Formas de desenho*

A uma variável de controle e baseado na opção que o usuário deseja ela molda o valor.

A interface recebe esse valor e com isso dado as instruções pinta na tela o desenho.

1.8   *Clonador*

Classe auxiliar utilizada pela classe Editor Camada. Serve para clonar dados e estes possam ser modificados, pois simplesmente seta-los em novas variáveis não seriam o suficiente com os dados do ArrayList. Foi necessário a criação dessa camada pois o método do Collection, não resolvia.

1.9   *Editor*

Classe que salva, carrega, adicionar e é um Jcomponent, sendo utilizado na classe principal como um jPanel.

1.10 *Geometrica*

Classe mãe das figuras, com ela têm-se as informações de como deverá ser desenhado. Suas filhas incluem todas as formas de desenho possíveis.

1.11 **Problema**: *Para editar posição*

Foi criado um método abstrato na classe geométrica chamado AtualizacaoMouse, que pega valores de x e y (posições do mouse) e recondicionam-às seguindo seus critérios individuais de cada classe. Salva-se essas alterações no arraylist auxiliar. Este método também é utilizado para que o centro da figura fique no click do mouse.

1. **Adições de recursos ao Projeto**

4.1 Paleta de cores com cor de borda embutida:

Caso o usuário queira que seus desenhos tenham borda, basta apenas ativá-la e selecionar sua cor com o botão direito do mouse.

4.2 Largura da borda

Na janela Formas de desenho é possível aumentar a largura da borda ou diminuir.

4.3 Desenhos Livres

Foram adicionados o modo de quadrilatero, triângulo e poligono livres de padrões, ou seja não é preciso demarcar seus limites de tamanho antes. Basta apenas clicar na tela que ela salvará os pontos como vertices e assim criará o desenho.

Poligono Livre como não há limites, também em vértices, o usuário precisa determinar onde deve parar e assim pintar o desenho na tela, para isso basta clicar com o botão direito do mouse.

4.4 Pintura de Tela

Caso o usuário queria começar o desenho com outra cor de tela, é possível pinta-la toltamente

4.5 Desenho Consecutivo

A opção Mágico permite a capacidade de fazer desde retas a quadrilateros consecutivos, bastando decidir qual opção deseja e para parar a consecutividade basta clicar com o botão direto do mouse.