**UNIVERSIDADE FEDERAL DE VIÇOSA**

**CAMPUS FLORESTAL**

SAMUEL SENA - 3494

**TRABALHO 0**

FLORESTAL

2019

Sumário

[Sumário 2](#_Toc16965819)

[Introdução 3](#_Toc16965820)

[Tipos de Artes 7](#_Toc16965821)

[Desenvolvimento 11](#_Toc16965822)

[Conclusão 14](#_Toc16965823)

# Introdução

O trabalho apresentado a seguir entrega um algoritmo executado em modo texto, com um menu interativo pelo teclado. O algoritmo é capaz de gerar 5 tipos de quadros composto por figuras despostas de maneira completamente aleatória.

Inicialmente para se executar o programa, é necessário realizar a compilação do código fonte em C. Para isso, em algum terminal Linux execute o “*makefile”* da seguinte forma:

Para compilar:

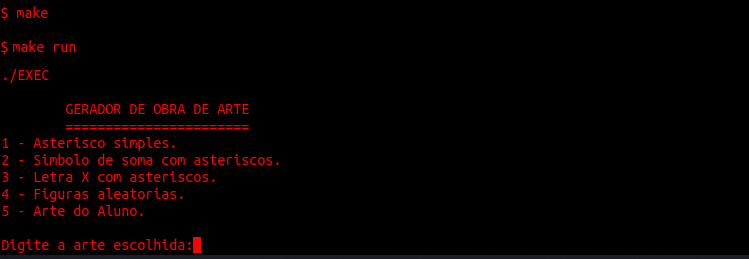
$ make

E para executar:

$ make run

A figura abaixo exemplifica o processo de compilação e execução pelo Linux:

Figura 1



Fonte: Terminal Linux

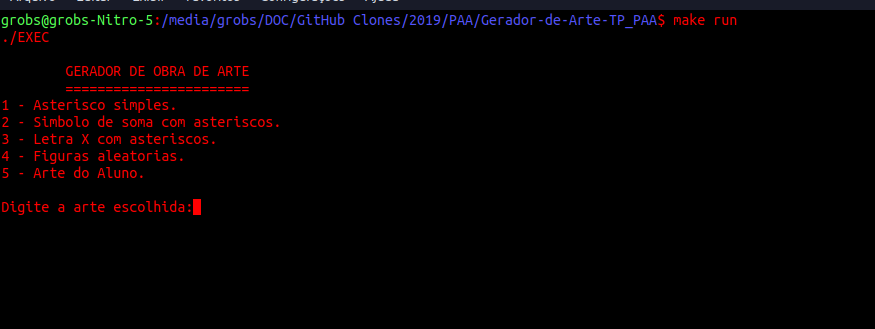
Ou no caso de execução em um sistema Windows, execute o seguinte código no Prompt de comando devidamente navegado até a pasta contendo o arquivo “main.c”:

> gcc main.c -o EXEC Sources/Gerador\_Quadro.c

E em seguida abra o executável gerado.

O programa inicialmente exibirá um menu com as seguintes opções:

Figura 2



Fonte: Terminal Linux

Primeiramente, a escolha do tipo de arte deverá ser realizada (números de 1 a 5, caso maior que 5, automaticamente a opção 5 será selecionada). Em seguida a quantidade de elementos que deveram ser combinados na arte desejada será requisitada (valores entre 0 e 100 devem ser escolhidos preferencialmente. Caso maior que 100, este automaticamente este será convertido para 100. Caso menor ou igual a 0, automaticamente um valor aleatório de 1 a 100 será computado). Ao final da exibição do quadro, será perguntado se deseja visualizar um novo quadro com as mesmas configurações selecionadas anteriormente (0 para não, 1 para sim). A figura abaixo representa bem as situações:

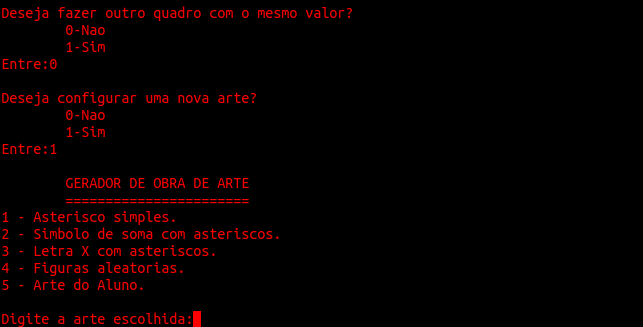
Figura 3



Fonte: Terminal Linux

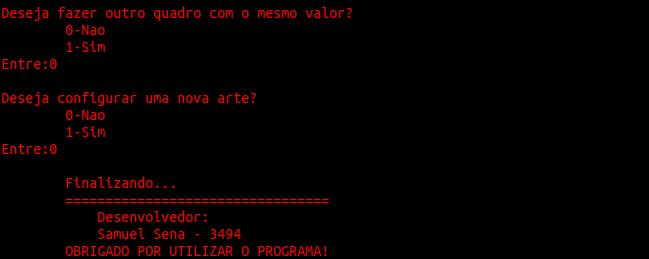
Caso em algum momento, a opção 0 seja selecionada, outra pergunta será realizada pelo programa, porem agora, ela se refere a retornar ao menu principal parar poder realizar uma nova configuração de uma arte, ou se deseja finalizar o programa. Como ilustrado a seguir nas figuras abaixo:

Figura 4



Fonte: Terminal Linux

Figura 5

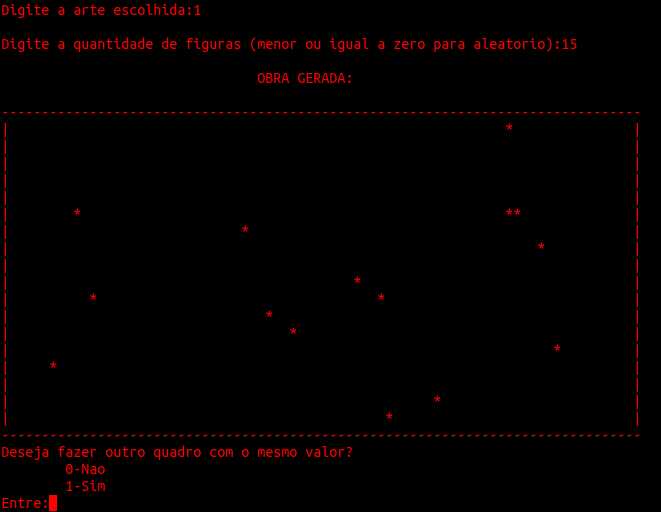


Fonte: Terminal Linux

# Tipos de Artes

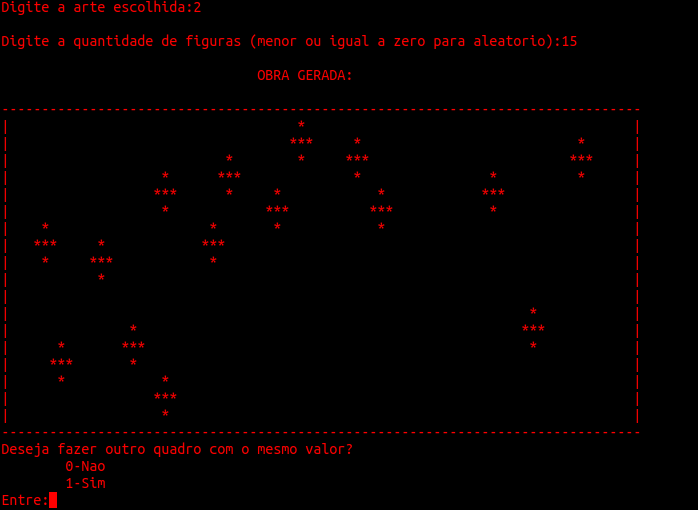
Dependendo da escolha durante a configuração da arte desejada a ser gerada, diferentes desenhos serão formados. Entre os elementos básicos que compõem as possíveis artes geradas, estão: Asterisco simples; Símbolo de soma feito de asteriscos; Letra X feito de asteriscos; Mistura aleatória dos 3 anteriores; Arte pessoal do Aluno. Para cada tipo de elemento selecionável citado, as figuras (6 a 10) abaixo ilustram quadros gerados com 15 elementos para fins de demonstração:

Figura 6



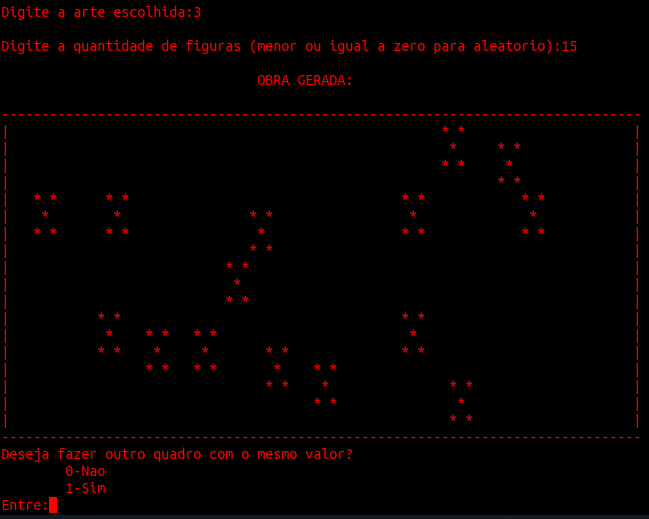
Fonte: Terminal Linux

Figura 7



Fonte: Terminal Linux

Figura 8

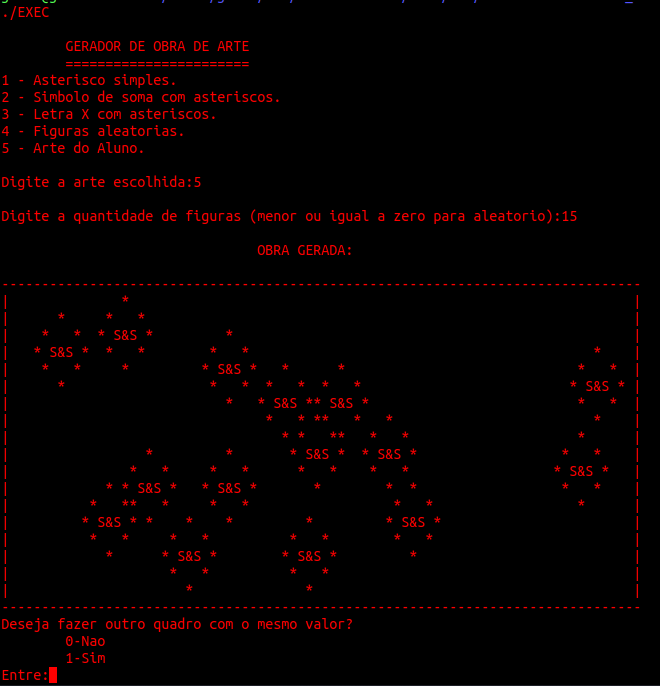


Fontes: Terminal Linux

Figura 9



Figura 10



Fonte: Terminal Linux

Se atentando a figura 10, a qual ilustra o quadro com a arte pessoal do aluno, é possível perceber que a composição central de um elemento individual conta com a disposição dos caracteres “S&S”. Estes são em referência ao primeiro e último nome do aluno. Já os contornos desse elemento, por outro lado, foram escolhidos desta maneira por possuírem certa capacidade de se misturarem bem em casos que valor da quantidade de elementos é elevado. O quadro configurado para ser gerado com a arte do aluno tem capacidade máxima de executar com 55 elementos (Caso o valor entrado seja superior, o valor será convertido para 55). Este limite foi estabelecido após testes exaustivos para garantir que o programa não entre em “loop” caso fiquei sem espaço de encaixar novos elementos na arte em questão. Como exemplificado na figura abaixo:

Figura 11

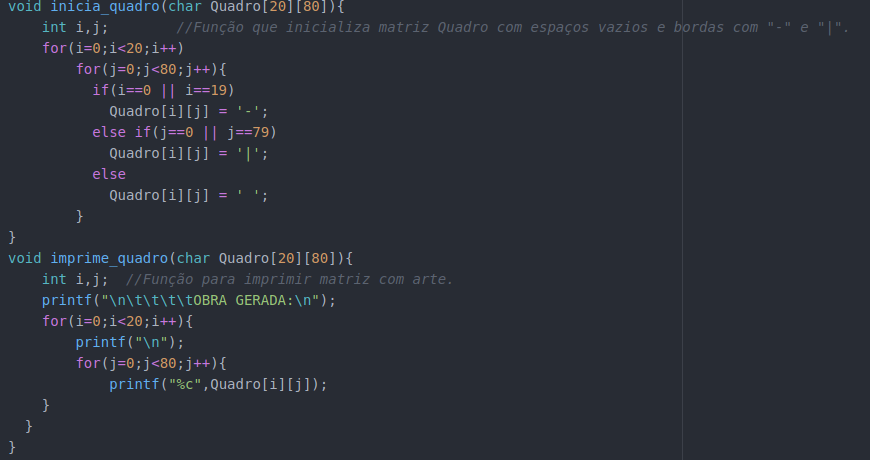


Fonte: Terminal Linux

# Desenvolvimento

O desenvolvimento do programa se consistiu basicamente na criação de uma matriz de caracteres de dimensões 20x80. Em seguida, quando a matriz é inicializada, o preenchimento com as laterais do quadro também já é realizado. Desta forma, apenas um percurso pela matriz é efetuado. A função responsável por isso é ilustrada a seguir:

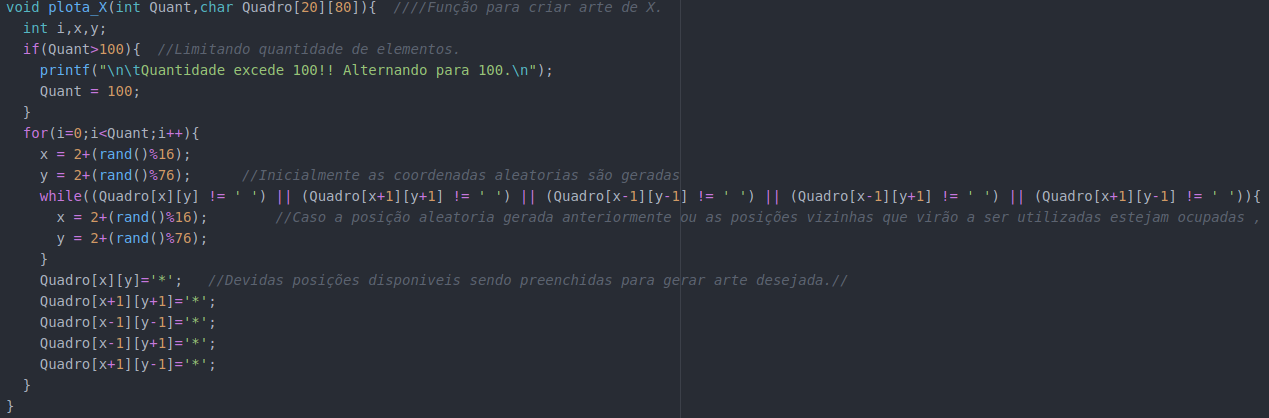
Figura 12



Fonte: Gerador\_Quadro.c

Mais à frente, após as devidas entradas desejadas serem computadas, o algoritmo realiza um loop de tamanho igual a entrada do usuário, sendo que para cada tipo de elemento, posições geradas aleatoriamente são testadas e enquanto não for encontrado uma respectiva posição vazia, o algoritmo continua a gerar novas coordenadas aleatórias. A figura a seguir demonstra o funcionamento em um dos casos de geração da arte:

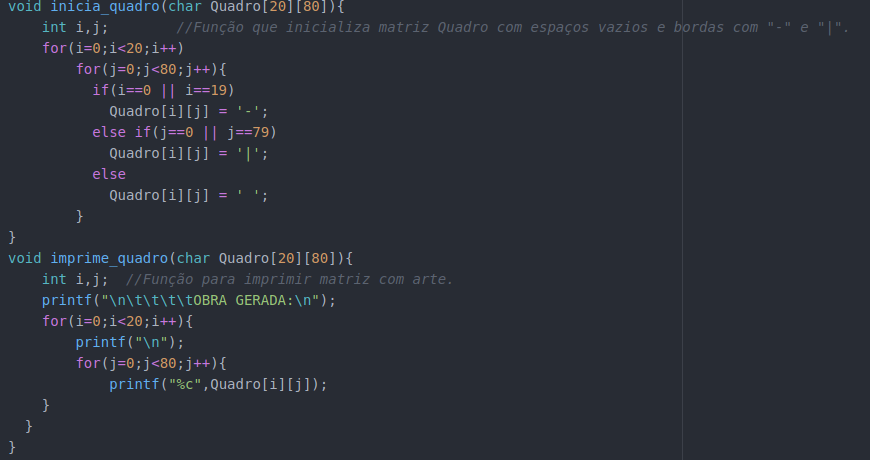
Figura 13



Fonte: Gerador\_Quadro.c

Ao final deste loop, a matriz é exibida na tela, com o uso da seguinte função:

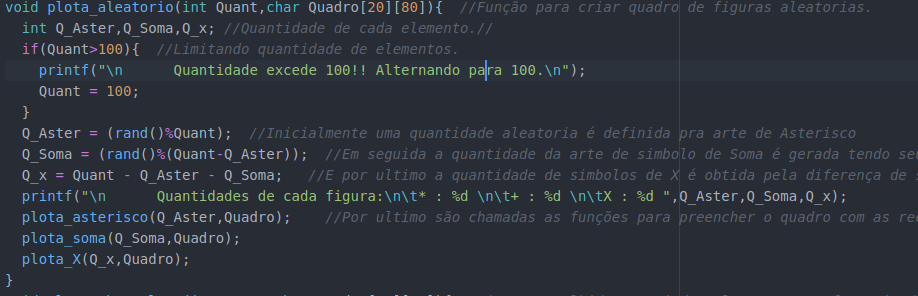
Figura 14



Fonte: Gerador\_Quadro.c

A opção de arte que gera um quadro com três elementos em quantidades aleatórias tem como principal mecanismo para se definir as quantidades de cada elemento da seguinte forma apresentada abaixo:

Figura 15



Fonte: Gerador\_Quadro.c

Sendo que “Q\_Aster” se refere a quantidade de asteriscos simples, “Q\_Soma” se refere a quantidade de símbolos de soma, e “Q\_x” se refere a quantidade de X.

O arquivo principal do programa “main.c” é responsável por organizar as chamadas de cada subprograma e possibilitar o loop do programa controlado pelo usuário. A figura abaixo ilustra uma pequena parte do código fonte do arquivo “main.c”:

Figura 16



Fonte: main.c

Vale lembrar que todo o código fonte encontra-se comentado para a fácil compreensão e entendimento do seu funcionamento.

# Conclusão

Sem dúvidas, o desenvolvimento de uma ideia abstrata para o devido cumprimento do comportamento desejado do programa requisitado pela descrição do trabalho foi a maior empecilho em comparação com os demais. A decisão de se trabalhar com uma matriz de caracteres serviu como uma espécie de “gatilho” mental para fazer com que os demais problemas que surgissem em seguida fossem facilmente contornados. Portanto, após longos testes de execução e verificações no código fonte, o programa se encontra executando da maneira desejada. E por último, mas não menos importante, é notável a suma importância com relação a aprendizagem e aperfeiçoamento de conceitos em programação, não ficando limitado apenas a *sintax* da linguagem, mas abrangendo também ao pensamento lógico que possibilita a correta programação.

Agradecimentos ao professor Daniel Mendes pela oportunidade de realização do trabalho e dúvidas sanadas.

Todo o desenvolvimento e distribuição do trabalho encontra-se hospedado na seguinte página do [GitHub](https://github.com/Globson/Gerador-de-Arte-TP_PAA).