UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - CÂMPUS CURITIBA - SEDE CENTRO
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Giovanni Mioto

Alexandre Alberto Menon

ICSF13 - 2023-2 - PROJETO 2

CURITIBA
2023

SUMÁRIO

1.INTRODUÇÃO

Esse relatório refere-se ao projeto 2 da disciplina de Fundamentos de Programação, atividade cuja qual abordou lógicas de programação referente à digitalização de áudio, estudando vetores de forma aplicada. Para isso, a atividade foi dividida em 5 questões de manipulação de vetores e, consequentemente, do áudio que se refere ao vetor trabalhado.

2. Se ambientando com a estrutura do projeto

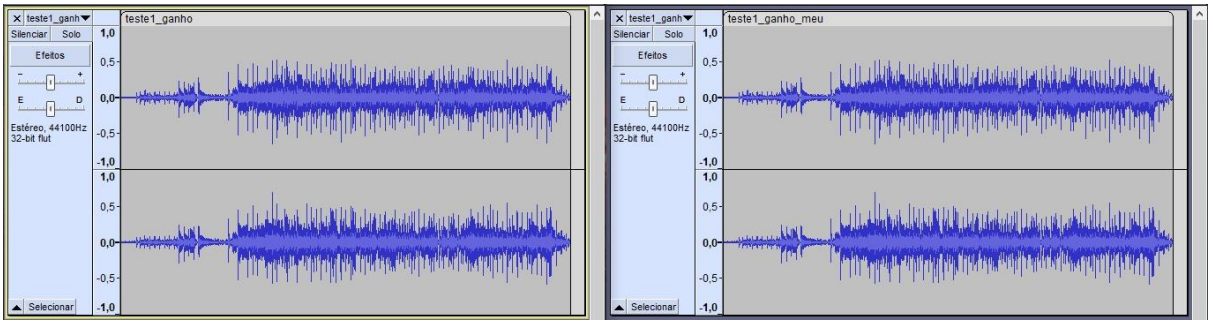
Antes de começar as questões do projeto, procuramos entender os arquivos disponibilizados e também como executar um teste básico de funcionamento. Logo quando executamos o primeiro teste, entendemos que os arquivos de testes deveriam ser inseridos um por vez, cada um se referindo a uma questão específica.

Entendido isso, criamos os escopos das funções no nosso arquivo de trabalho, e executamos um “printf” para entender como seria manipulado o vetor “dados” que estava sendo passado como parâmetro para as funções, e, depois desse passo, foi possível começar a trabalhar nas questões pedidas.

3.1 Questão 1

A questão 1 consistia em modificar o ganho (volume) de um sinal, e de certa forma, foi uma questão bastante simples, ao passo que só seria necessário multiplicar cada posição do vetor por algum fator comum. Mesmo assim, só conseguimos ter dimensão do funcionamento da questão 1 ouvindo o áudio, pois só iríamos ver os resultados em um software de produção de áudio nas últimas duas questões do trabalho. Esse erro poderia ter comprometido em maior proporção o trabalho, entretanto o funcionamento da questão 1 foi concluída com êxito, mesmo que só concluímos isso, de fato, no final do projeto.

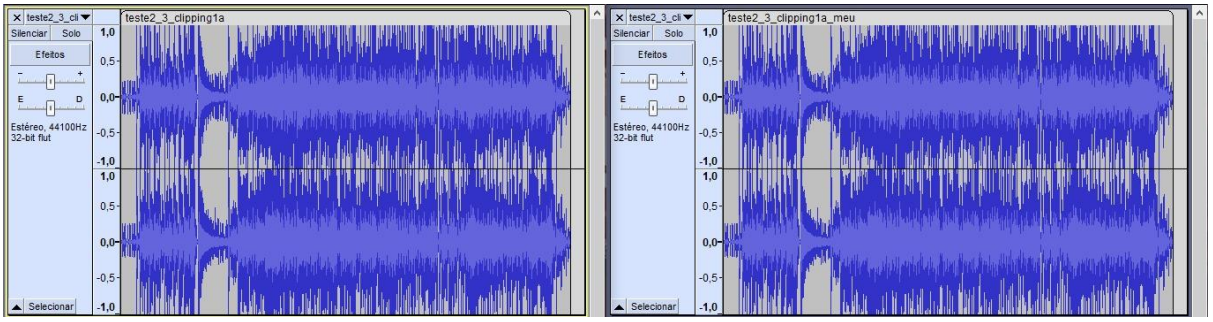
Resultados obtidos no Audacity



3.2 Questão 2

A segunda questão foi destinada para contar a quantidade de saturações de um áudio, o que bastou mapear no vetor as posições cujas quais o valor era maior que -1 ou maior que 1. Essa questão também foi relativamente simples de se resolver, mas também foi bastante importante para dar sequência na questão 3.

Resultados obtidos no Audacity



3.3 Questão 3

Na questão 3 deveria ser tratada as saturações, o que bastou incluir um passo a mais na questão 2, justamente para fazer com que as posições do vetor que os valores passaram de -1 ou de 1 voltassem para esse limite. Ainda sim, foi necessário mudar um pouco da lógica da questão 2, pois em vezes de verificar se os valores passavam do intervalo de -1 a 1 dentro de uma só estrutura condicional, tivemos que separar em duas estruturas, pois seria necessário voltar para -1 aqueles valores que passaram de -1, ou voltar para 1 os valores que passarem de 1 de forma separada para não interferir no sinal daquela posição.

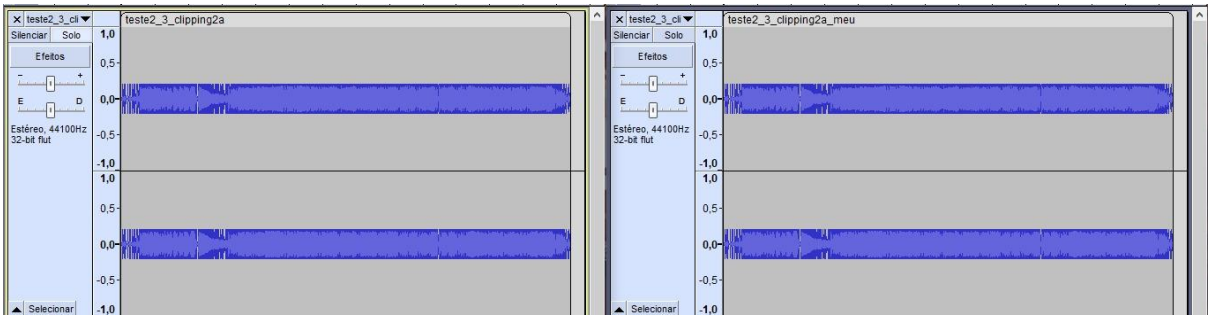
Estrutura condicional da questão 2

```
if(dados[i]>1 || dados[i]<-1)
{
    cont_saturacoes++;
}
```

Estrutura condicional da questão 3

```
if(dados[i]>limite)
{
    dados[i]=limite;
    cont_alteracoes++;
}
else if(dados[i]<(limite*(-1)))
{
    dados[i]=limite*(-1);
    cont_alteracoes++;
}
```

Resultados obtidos no Audacity



3.4 Questão 4

A questão 4 nos encarregou da tarefa de limitar os sinais, assim como na questão 3, porém, sendo necessário guardar o fator multiplicador de um valor que passasse do intervalo [-1,1] para entrar nesse intervalo. Por exemplo, se em alguma posição do vetor o valor fosse 2, o fator multiplicador para retornar o valor para o intervalo máximo seria de 0.5. Além disso, esse fator multiplicador seria usado para interferir nos valores vizinhos daquela posição. O índice de aumento refere-se o quanto é acrescentado no fator multiplicador das posições vizinhas, e foi calculado da seguinte forma:

O fator multiplicador da posição que saiu do intervalo é subtraído de 1, e, após saber esse resultado, é dividido pela quantidade de posições vizinhas que serão afetadas somadas de um.

Dessa maneira, temos o fator multiplicador das posições vizinhas, que irá crescendo proporcionalmente com a distância da posição que saiu do intervalo de [-1,1] somando o índice de aumento.

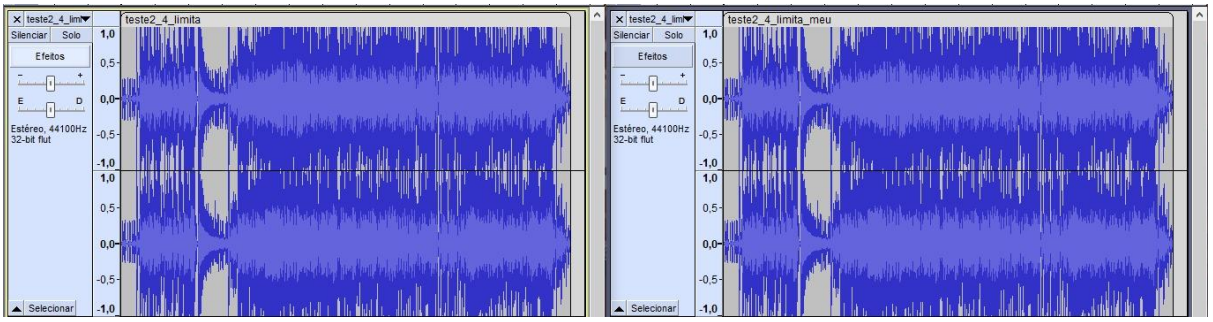
No exemplo, o índice de aumento é 0.1, então uma casa à direita e uma casa à esquerda da posição do valor que era 2 será multiplicada pelo fator 0.6, uma casa à direita e uma casa à esquerda a mais serão multiplicadas pelo fator 0.7 e assim por diante.

Equação para descobrir o índice de aumento no fator multiplicador

```
aux=1/dados[i];
indice_aumento=1-aux;
indice_aumento/=n_passos+1;
dados[i]*=1/dados[i];
```

Essa questão elevou o nível do trabalho, pois demoramos um considerável tempo para entender o padrão da equação do índice de aumento, sendo necessário a consulta com o monitor Kahuê Bandeira para adaptar nossa lógica. Nesse momento, utilizamos o software de produção de áudio, Audacity, para nos certificarmos que a lógica foi implementada corretamente e alcançasse os resultados esperados, e também aproveitamos para conferir as questões anteriores.

Resultados obtidos no Audacity



3.5 Questão 5

A quinta questão consistia em trabalhar no vetor para que fossem produzidas as chamadas ondas quadradas. Em geral, deveríamos calcular um intervalo, que é calculado pela taxa de amostragem de um áudio sobre a frequência, e dividir metade desse intervalo para ser inserido o valor 1, e outra metade para o valor -1, o que nos indica quantos ciclos completos devem ocorrer em 1 segundo.

Para essa questão, geramos um loop para inserir os valores 1 e -1 no vetor para caso o intervalo, que chamamos de meio período, fosse um valor inteiro. Até esse ponto foi relativamente fácil e conseguimos atribuir essa condição.

```
if(meio_ciclo==(int)meio_ciclo)
{
    for(i=0; i<n_amostras; i+=(int)meio_ciclo)
    {
        for(j=i; j<(int)meio_ciclo+i; j++)
        {
            dados[j]=inverte_sinal;
        }
        inverte_sinal*=-1.0;
    }
}
```

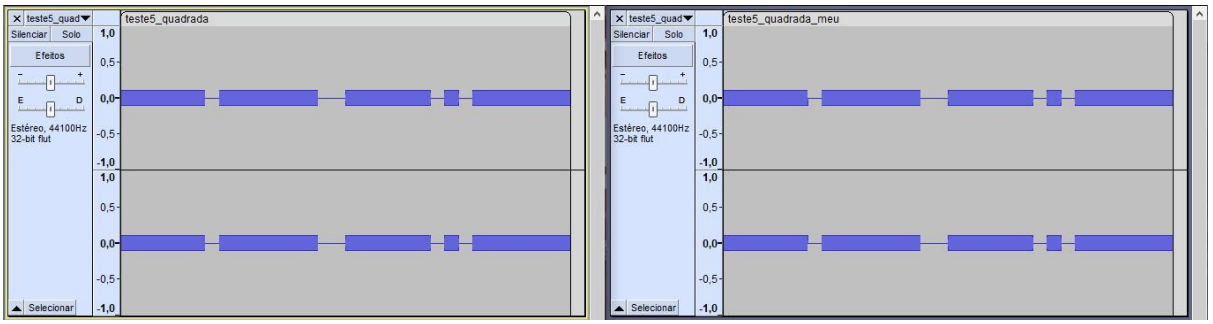
Outrossim, era necessário implementar outro loop para caso o meio período não fosse um valor inteiro. Dessa maneira, estudamos o padrão gerado no exemplo para entender como era calculado o meio período se esse valor fosse “quebrado”, e com isso, foi possível perceber que com base na taxa de erro, que se refere ao valor do meio período considerado subtraído da sua parte inteira, seria calculado quando o valor teria o número do “meio ciclo” de vezes dos valores de 1 ou -1, ou se teria o valor do “meio ciclo + 1” de vezes dos valores de 1 ou -1.

Equação codificada da quantidade de vezes que fosse inserido 1 ou -1

```
erro=meio_ciclo_atualizado-(int)meio_ciclo_atualizado;
meio_ciclo_atualizado=meio_ciclo+erro;
```

Mesmo que o formato das ondas do gabarito diferiram em alguns momentos, concluímos que chegamos em um bom resultado em relação ao áudio que era esperado, que as diferenças dos formatos das ondas na maioria tinha relação com a posição dos vales e picos da onda.

Resultados obtidos no Audacity



4. Considerações finais

Vale ressaltar que buscamos nossa forma de solucionar o problema da questão 5, tentando identificar um padrão que acontece em períodos de meio ciclo com valor não inteiro, e que discordamos em parte do padrão reconhecido pelo monitor consultado, Kahuê, que apontou que os períodos de um meio ciclo seriam alterados da seguinte forma:

Um meio período de vezes com os valores 1 ou -1

Dois meio período de vezes + 1 com os valores de 1 ou -1

Exemplo com meio período de 112.50:

112 vezes o valor 1, 113 vezes o valor -1, 113 vezes o valor -1 e assim segue.

Concluímos não seguir por este caminho ao passo que os resultados em relação ao áudio obtido e o formato da onda não conseguiram satisfazer as expectativas.