

Exercício de Programação Período 2022/02**Descrição:**

Deseja-se filtrar um sinal digital, lido de um arquivo texto, aplicando-se 3 filtros do tipo FIR (*Finite Impulse Response* ou Filtro cujo o tempo de resposta ao impulso é finito). Você deve desenvolver um sistema com interface gráfica cujo layout é mostrado na Figura 1. Todo o controle da interface de ser feito usando o mouse para controlar a ação desejada.

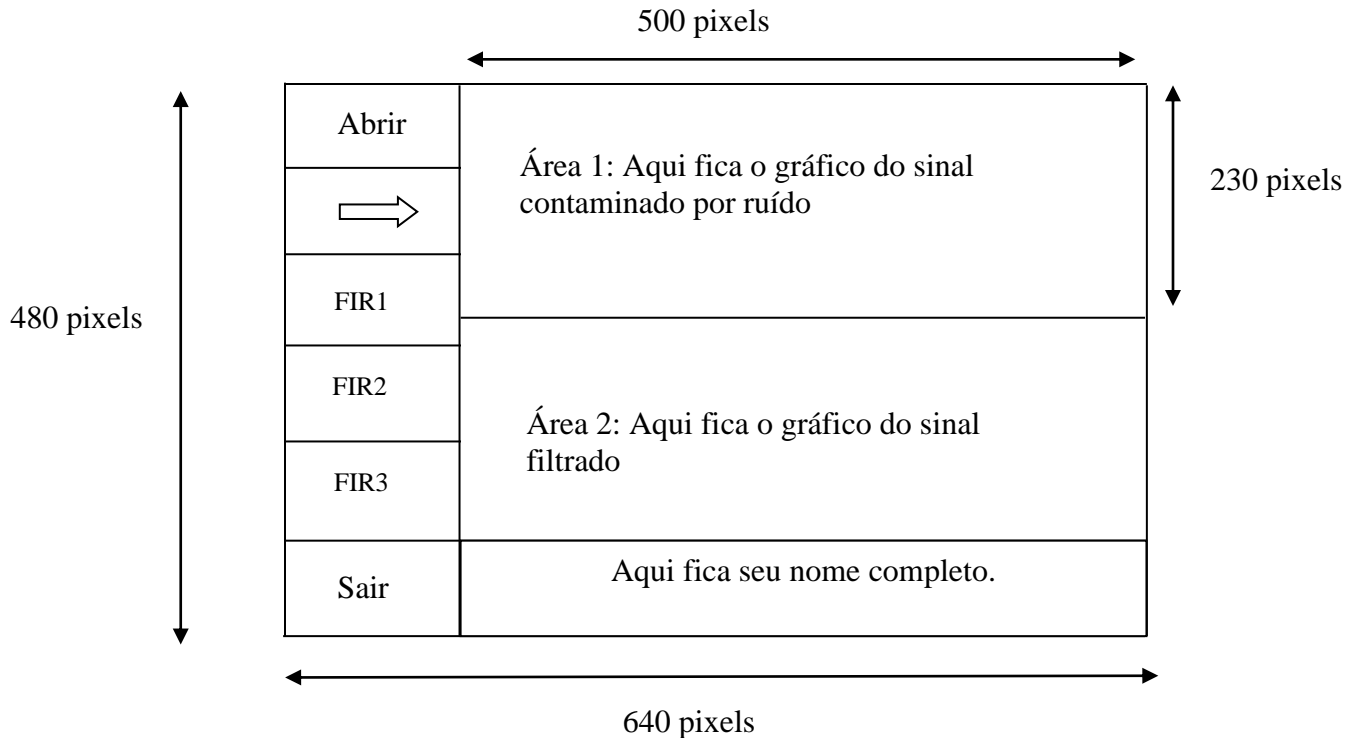


Figura 1: Layout da tela a ser desenvolvida.

Ao abrir o arquivo, que contém o sinal $x(n)$ para n discreto e representado por uma *string* ASCII, deve-se convertê-lo para um vetor e, em seguida, aplicar os filtros. O gráfico de $x(n)$ (com eixos coordenados) deve ser plotado na Área 1 e o gráfico do sinal filtrado deve ser plotado na Área 2 (vide Figura 1). Todas as linhas e textos da interface devem ser pintados na cor branca e o fundo deve ser preto. Já para o *menu* de opções (*menu* na lateral esquerda), uma vez que um deles seja selecionado, deve-se trocar a cor da legenda, operação dentro de cada retângulo, de branco para verde e assim permanecer até que uma outra opção seja selecionada. A seleção deverá ser exclusivamente feita pelo uso do *mouse*. Para tanto, deve-se posicionar o cursor do *mouse* em qualquer parte do retângulo que contiver a função desejada e clicar com o botão esquerdo do mouse para executar a operação. No *menu*, tem-se as opções:

Abrir: Abrir o arquivo contendo o sinal, aqui chamado de 'sinal.txt';

→: Como a área útil do gráfico só permite 500 pontos, cada clique aqui traz mais 500 pontos, movendo ao mesmo tempo, 500 pontos na Área 1 e Área 2. Caso fique, seguidamente, pressionando esse botão, ao final do sinal, retorna-se para o começo.

FIR1: Faz a convolução linear de **todo o sinal** $x(n)$ com o filtro $h_1(n) = 1/11 \times [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]$

FIR2: Faz a convolução linear de **todo o sinal** $x(n)$ com o filtro $h_2(n) = 1/14 \times [1 \ 1 \ 1 \ 1 \ 2 \ 2 \ 2 \ 1 \ 1 \ 1 \ 1]$

FIR3: Faz a convolução linear de **todo o sinal** $x(n)$ com o filtro $h_3(n) = 1/9 \times [1 \ 1 \ 2 \ 2 \ -3 \ 3 \ -3 \ 2 \ 2 \ 1 \ 1]$

Procedimento

Primeiro, deve-se abrir o arquivo contendo o sinal codificado em ASCII. Ao fazê-lo, carrega-se todo o sinal em um vetor e os primeiros 500 pontos devem ser mostrados na Área 1. Ao selecionar o botão da seta, deve-se apresentar os próximos 500 pontos, até chegar ao final do sinal. Depois de chegar ao final do sinal, caso o usuário continue a selecionar a seta, essa próxima seleção vai, novamente, apresentar os 500 primeiros pontos. Ou seja, clicar na seta constantemente gera um comportamento cíclico na interface. Ao selecionar qualquer um dos filtros (FIR1, FIR2 ou FIR3), deve-se fazer a convolução completa do sinal com o respectivo filtro e, de imediato, deve ser mostrado na Área 2 os primeiros 500 pontos da filtragem. Nesse caso, ao selecionar a seta, os próximos 500 pontos, de cada uma das áreas, devem ser mostrados. Se o usuário continuar a selecionar a seta, seguidamente, o comportamento cíclico deve ocorrer, agora mostrando os próximos 500 pontos de ambas as áreas (Área 1 e Área 2). A qualquer instante você pode aplicar qualquer uma das funções do *menu* lateral. Ao clicar em ‘Abrir’ com um arquivo já aberto, seu código deve fechar o arquivo aberto para, novamente, abri-lo. As cores na tela são de acordo com a Tabela 1.

Tabela 1: Tabela de Cores modo VGA 640×480

Preto	0	Cinza	8
Azul	1	Azul_claro	9
Verde	2	Verde_claro	10
Cyan	3	Cyan_claro	11
Vermelho	4	Rosa	12
Magenta	5	Magenta_claro	13
Marrom	6	Amarelo	14
Branco	7	Branco_intenso	15

Para facilitar a programação, um programa (**LINEC.ASM**) é fornecido, contendo o básico para a mudança do modo de vídeo (gráfico, 640×480 com 16 cores), funções de plotar ponto, desenhar uma linha, posicionar o cursor e escrever um caractere. Este programa deve ser usado como referência para a programação do exercício. Para as funções gráficas, a tela pode ser vista com uma resolução de 640×480 (x e y) sendo que a origem (0,0) é o ponto inferior à esquerda da tela.

Procedimento:

O programa deverá abrir o arquivo e ler o seu conteúdo para uma variável na memória e depois fechar o arquivo. A partir desta variável os dados devem ser processados para mostrar as figuras descritas no arquivo. Utilizar as seguintes funções da INT 21H :08H,3DH,3EH,3FH, e 4CH e as interrupções da BIOS das funções de vídeo (INT 10H). Para a utilização do mouse, utilize a INT33H, cuja forma de utilização é semelhante à INT21H (veja uma descrição detalhada em http://stanislavs.org/helppc/int_33.html). O arquivo a ser lido, sinalep.txt, pode, portanto, ser aberto no Matlab, tem cada amostra codificada em uma string de ASCII. Por exemplo:

a string de ASCII 2.9000000e+01 representa 29 ,
a string de ASCII 7.0000000e+00 representa 7 ,
a string de ASCII -3.3000000e+01 representa -33 .

Cada amostra deve variar na faixa inteira entre -128 a 127. Seu programa deve abrir o arquivo, converter cada amostra de string no formato ASCII para um valor inteiro de 8 bits e com sinal. Para ver o gráfico do sinal, você pode fazer no Matlab:

```
>> load sinal.txt;  
>> plot(sinal);  
>> subplot(311); plot(conv(sinal,h1)); subplot(312);plot(conv(sinal,h2)); subplot(313);plot(conv(sinal,h3));
```

Para as respostas impulsivas dos filtros h1, h2 e h3 dadas conforme:

$$h1(n) = 1/11 \times [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1],$$

$$h2(n) = 1/14 \times [1 \ 1 \ 1 \ 1 \ 2 \ 2 \ 2 \ 1 \ 1 \ 1],$$

$$h3(n) = 1/9 \times [1 \ 1 \ 2 \ 2 \ -3 \ 3 \ -3 \ 2 \ 1 \ 1].$$

Importante: Procedimento para entrega:

Enviar para o email: evandro.salles@ufes.br com o assunto: “EP_sistemb1_2022-2”. O nome do arquivo ASM deve ser as iniciais de seu nome e deve conter, no máximo, 8 caracteres. No arquivo, não se esqueça de adicionar logo nas primeiras linhas, comentadas, o seu nome completo e a turma. Enviar até 28/11/2022, às 23:59h.

Atenção:

O exercício é individual. Qualquer cópia parcial ou total acarretará na atribuição da nota 0 (zero) para todos os envolvidos.