**Estruturas de Dados**

**Exercícios**

**- Recursividade**

1) Faça as funções recursivas que:

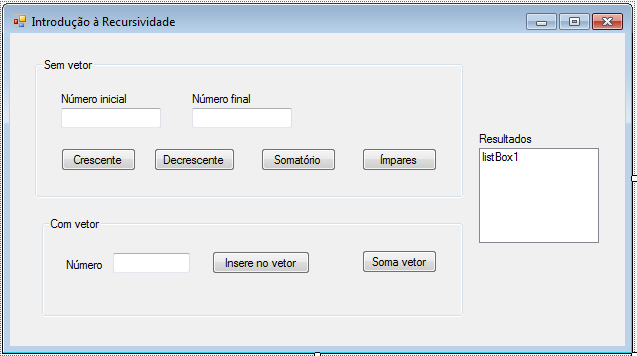
a) informe os números inteiros entre um número inicial e final (crescente)

b) informe os números inteiros entre um número inicial e final (decrescente)

c) calcule o somatório dos números inteiros entre um número inicial e final

d) informe os números inteiros ímpares entre um número inicial e final

e) some os números inteiros de um vetor



2) Escreva uma função recursiva, potencia(x, y), que devolva x elevado a potência y.

3) Implemente uma versão recursiva da seguinte função:

void cubos (int n)

{

for (int i = 1; i <= n; i++)

exibe i \* i \* i;

}

4) Baseado no algoritmo de Euclides, implemente uma função recursiva para determinar o máximo divisor comum (mdc) entre dois números inteiros x e y.

Algoritmo de Euclides:

se (x = y) retorna x

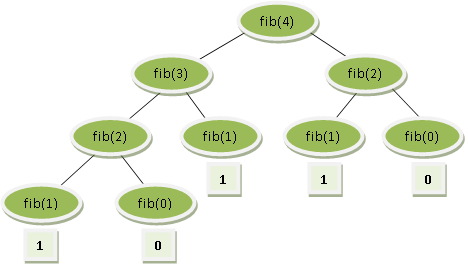
senão se (x < y) retorna mdc(y, x)

senão retorna mdc(x - y, y)

5) A Série de Fibonacci pode ser definida da seguinte maneira:

se (n == 0 ou n == 1) retorna n

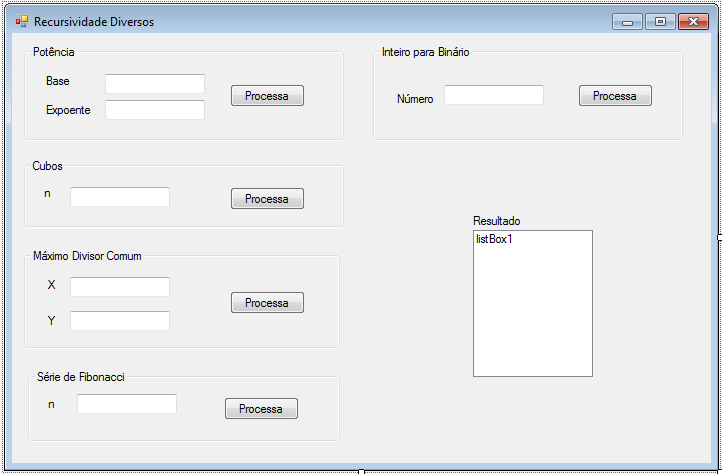
se (n >= 2) retorna fib(n – 1) + fib(n – 2)



Seja fib(n) uma função que retorna o n-ésimo termo da série de Fibonacci, implemente uma versão recursiva e outra iterativa. Observe que a partir de um determinado número, a função recursiva começa a ficar mais lenta que a iterativa.

6) Defina uma função recursiva que converte um número inteiro para a base binária.

Interface para os exercícios 2, 3, 4, 5 e 6

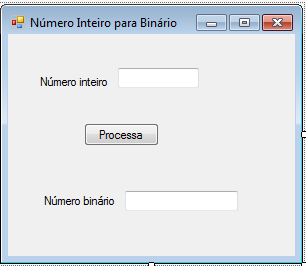


**- Lista linear sequencial: Pilha**

7) Mostre a situação da pilha (memória), inicialmente vazia, após a execução de cada uma das operações a seguir:

1. Insere(pilha, 20);
2. Insere(pilha, 30);
3. Insere(pilha, 40);
4. v = Remove(pilha);
5. Insere(pilha, 50);
6. Insere(pilha, v);
7. Remove(pilha);
8. Remove(pilha);

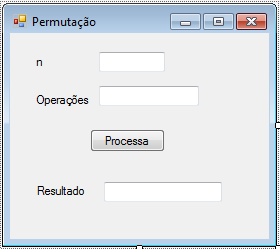
8) Faça um programa que converta um número inteiro da base decimal para a base binária.



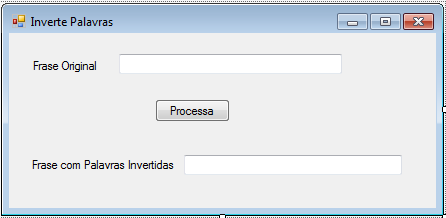
9) Seja 1, 2, ..., n uma sequência de elementos que serão inseridos e posteriormente removidos de uma pilha P, um por vez. A ordem de inclusão dos elementos na pilha é 1, 2, ..., n, enquanto a de remoção depende das operações realizadas.

Por exemplo, com n=3 e representado as operações de Inserir com a letra I e Remover com a letra R, a sequência de operações IIRIRR, produzirá a permutação 2,3,1.

Determinar a permutação correspondente a sequência de operações IIIRRIRR, com n=4.



10) Escreva um programa para conhecer uma frase e exibi-la com as palavras invertidas. Exemplo: a frase “Jose da Silva” deve sair “esoJ ad avliS”.



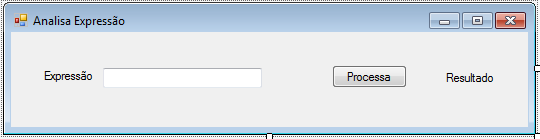
11) Elabore um programa que verifica se uma dada expressão é válida em relação aos abre e fecha parênteses, colchetes e chaves. Exemplos:

a) a = b + (c -d) \* (e – f)

b) b = [c – d)

c) while (m < (n[8] + o)) { m = m + 1; }

d) for (i = 1; i <= tl; i++



**- Lista linear sequencial: Fila**

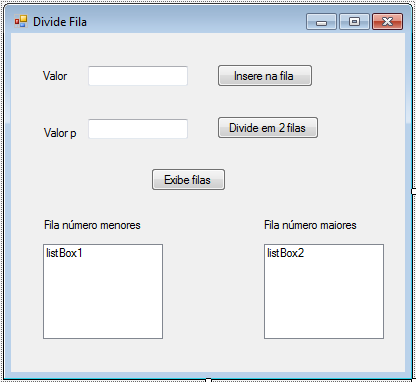
12) Mostre a situação da fila (memória), inicialmente vazia, após a execução de cada uma das seguintes operações:

1. Insere(fila, 20);
2. Insere(fila, 30);
3. Insere(fila, 40);
4. Remove(fila);
5. Remove(fila);
6. Insere(fila, 50);
7. Remove(fila);
8. Insere(fila, 60);
9. Insere(fila, 70);
10. v = Remove(fila)
11. Insere(fila, v);

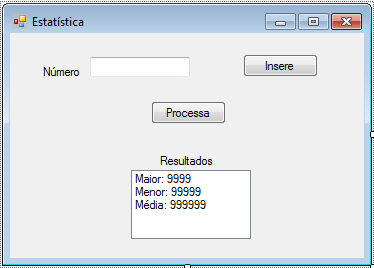
13) Escreva um programa que tenha uma fila cujo valores indicam prioridade, quanto menor o valor, maior a prioridade.

Seu programa deve inserir n valores numa fila e depois dividi-la em duas novas filas, uma fila com valores menores ou igual a um valor p fornecido pelo usuário e outra fila com os valores maiores que p.

Por fim, informe os valores das duas novas filas.



14) Escreva um programa que insira vários números numa fila. Após a digitação dos números, seu programa deve encontrar o maior, o menor e a média aritmética dos números da fila. Por fim, informe os resultados encontrados.



15) Escreva um programa que simule o controle de uma pista de decolagem de aviões em um aeroporto. Os aviões são identificados pelos números digitados pelo usuário. Neste programa, o usuário deve ser capaz de realizar as seguintes operações:

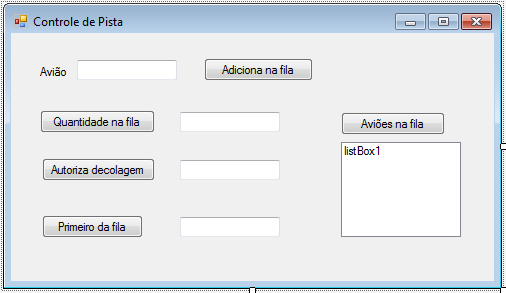
a) Adicionar um avião à fila de espera para decolagem;

b) Exibir a quantidade de aviões aguardando na fila;

c) Autorizar a decolagem do primeiro avião da fila;

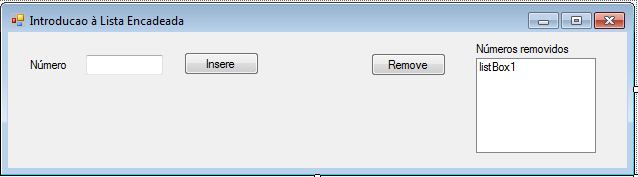
d) Listar os números de todos os aviões na fila;

e) Exibir o número do primeiro avião da fila.



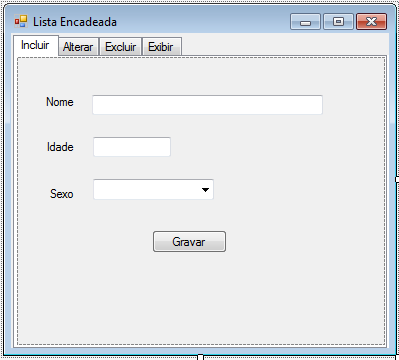
**- Lista linear encadeada**

16) Faça uma implementação que construa uma lista encadeada. Seu programa deve ter as opções de inserção e remoção dos elementos. Após remover um elemento da lista, exiba-o na tela.

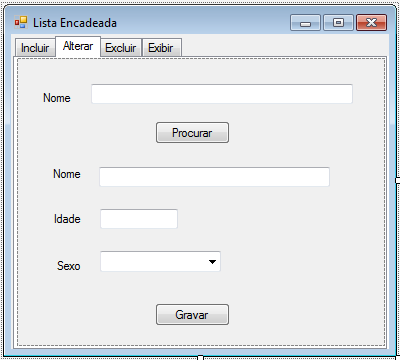


17) Faça uma implementação de lista encadeada que tenha as seguintes interfaces de usuário. O nó deve conter os atributos: nome, idade, sexo e prox.

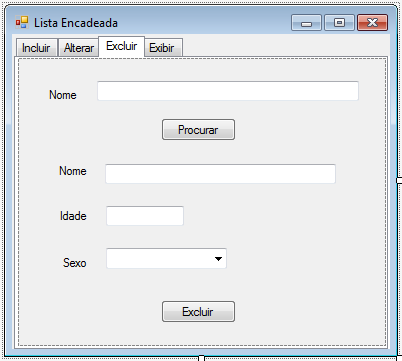
1. Incluir conforme apresentado em aulas



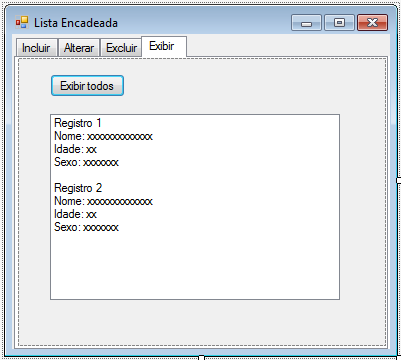
1. Para alterar, consulte pelo nome. Se encontrar, exiba os valores atuais e permita a alteração.



1. Para excluir, procure pelo nome. Se encontrar, exiba os valores atuais e permita a exclusão.



1. Na opção exibir, exiba todos os registros.



18) Escreva uma função recursiva que copie os números de um vetor para uma lista encadeada. Após copiar os números do vetor para a lista, percorra a lista e encontre o menor, o maior e a média dos números.

19) Implemente a lógica da inserção em uma lista duplamente encadeada (nó com ponteiro para o anterior e o próximo elemento).

20) Implemente a lógica da remoção em uma lista duplamente encadeada. Faça a procura pelo nó a ser excluído.

21) Implemente uma lista encadeada com um ponteiro para o início e outro para o fim da lista. Insira no fim e remova do início.

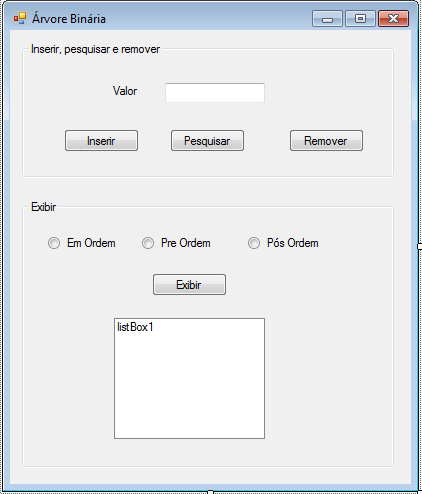
**- Lista não linear: Árvore Binária**

22) Seja uma árvore binária inicialmente vazia, esquematize o seu estado final após terem sido inseridos os elementos 4, 1, 0, 5, 3, 7, 2, 6, 9 e 8, nesta ordem.

23) Considerando a árvore esquematizada a seguir, responda:

1. Quais são os nós folhas
2. O nível de cada nó não terminal
3. A altura da árvore
4. O grau da árvore
5. Os descendentes do nó c

24) Desenvolva a seguinte interface de usuário para inserir, pesquisar, remover e exibir os valores de uma árvore binária.



1. Inserir um valor digitado pelo usuário
2. Pesquisar um valor digitado pelo usuário. Exiba uma mensagem informando se encontrou ou não
3. Remover um valor digitado pelo usuário. Exiba a mensagem se removido com sucesso ou não encontrado
4. Exibir todos os valores da árvore em ordem, pré ordem ou pós ordem

25) Baseado no atravessamento “em ordem” desenvolva:

a) um algoritmo para percorrer os nós e exibir apenas os valores armazenados nas folhas.

b) uma rotina que determina se uma árvore é estritamente binária. OBS: Dizemos que uma árvore é estritamente binária se todo nó pertencente a ela é folha ou tem os dois filhos.

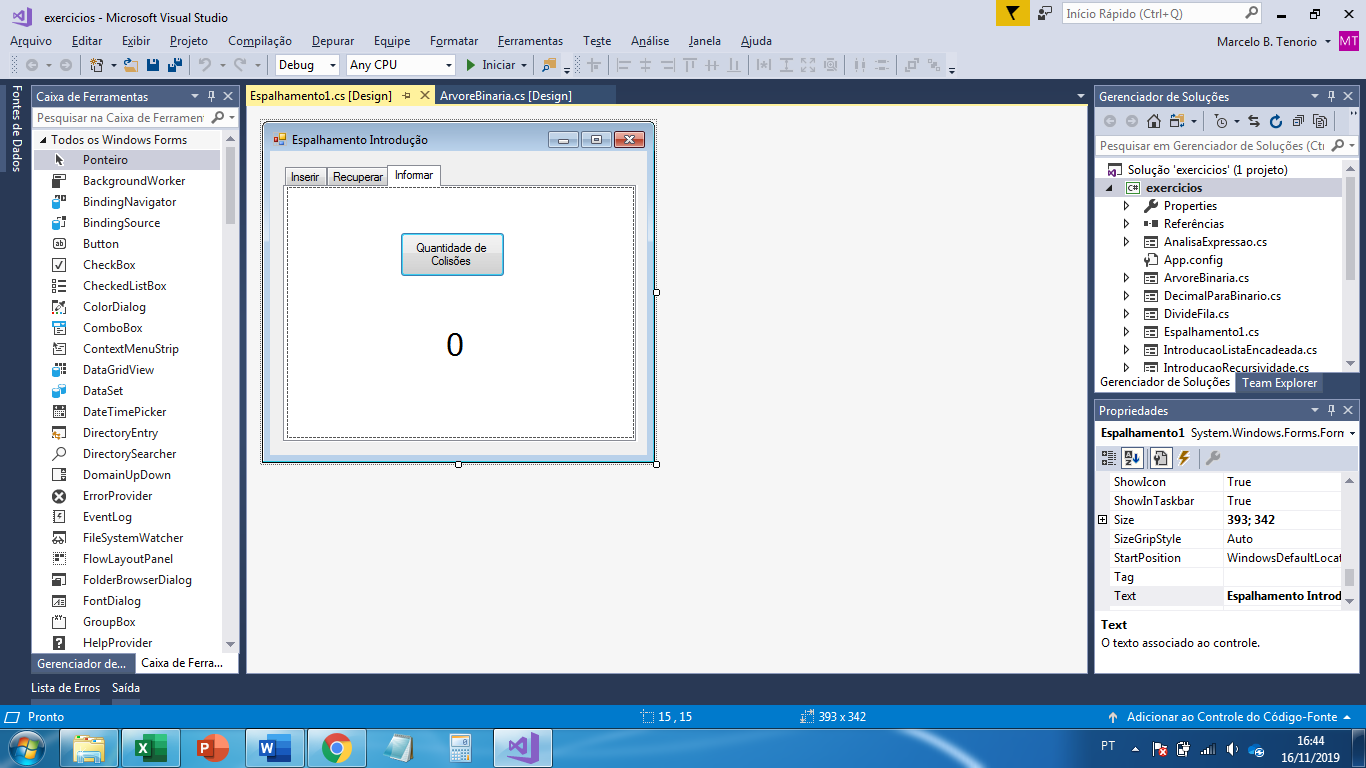
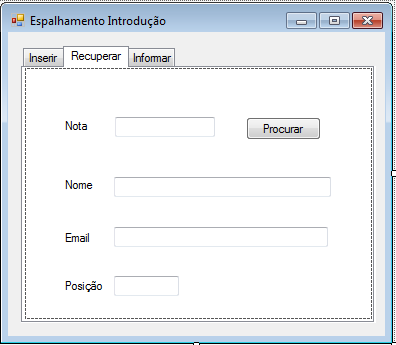
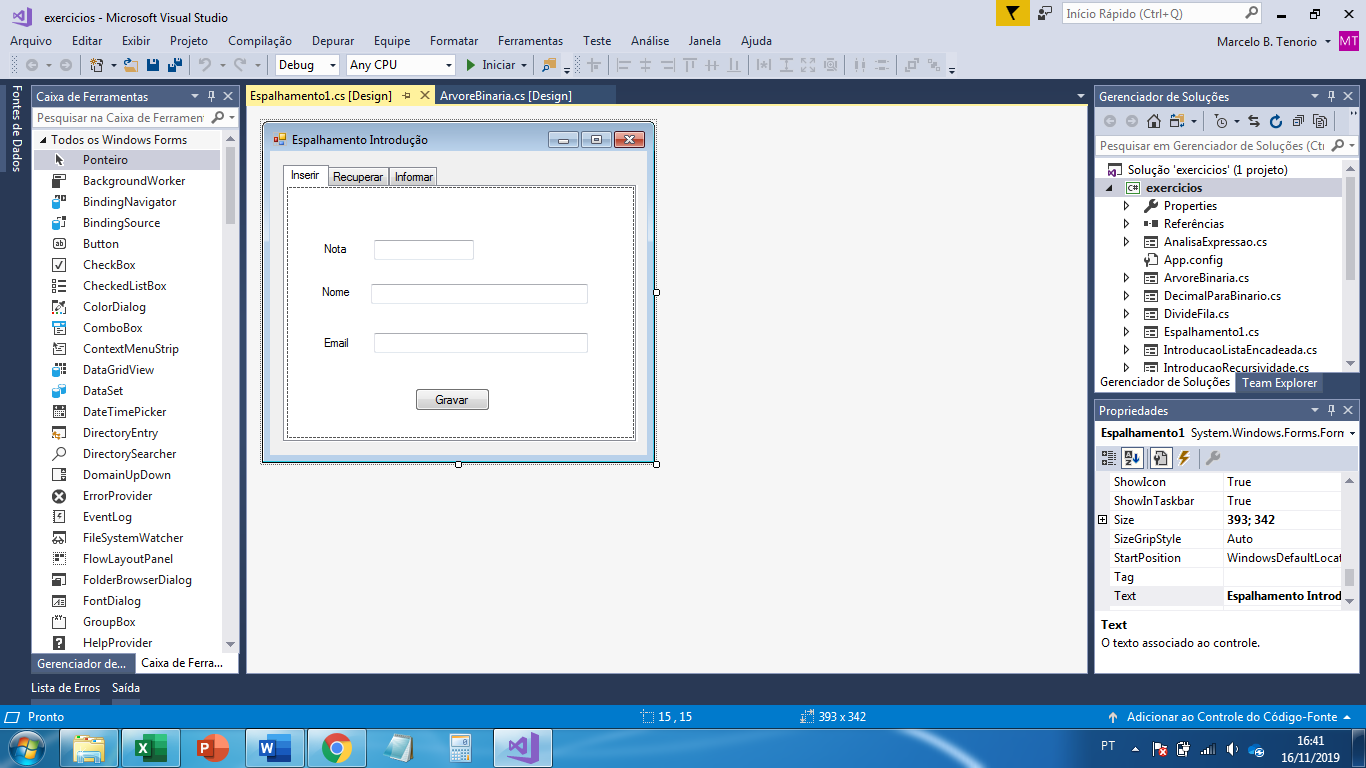
c) um algoritmo para contar o número de nós não-terminais.

**- Espalhamento**

26) Implemente um programa que conte a quantidade de colisões ocorridas durante o processo de inserção. Utilize o tratamento de colisão linear.

A interface de usuário deve conter as seguintes páginas: Inserir, Recuperar e Informar. A página Informar, informa a quantidade de colisões.

O vetor deve ser de um tipo abstrato de dado composto por nota, nome e email. A nota deve ser um número inteiro e corresponderá a chave.



27) Implemente um programa com as seguintes páginas: Sem tratamento de colisão, Tratamento de colisão Linear e Tratamento de colisão com Lista Encadeada.

Dentro de cada página deve haver as funcionalidades: Gravar/Consultar/Alterar e Relatório.

O vetor deve ser de um tipo abstrato de dado composto por idade, nome e sexo.

Para inserir um novo registro, solicite a idade, nome e sexo. Utilize a idade como chave.

Para consultar solicite a idade (chave) e informe o nome e o sexo da idade encontrada. Após a consulta, o usuário pode atualizar o nome e/ou o sexo, e gravar a alteração.

Para o Relatório, percorra o vetor do inicio ao fim e informe todos os dados.

