

Análise de projeto de sistemas



Alunos representantes do trabalho

- Kauan Modonez Alves
- Gabriel de Souza Santos
- Guilherme Henrique Daroz
- Luís Artur Faustinoni Ribeiro
- Pedro Lucas Aparecido Silva

Responsáveis de cada uma das funções

- Kauan Modonez Alves=(Testar e apresentar novas possibilidades de funções)
- Gabriel de Souza Santos = (Programar o código base)
- Guilherme Henrique Daroz=(Levantamento de requisitos)
- Luís Artur Faustinoni Ribeiro = (Diagrama de bloco)
- Pedro Lucas Aparecido Silva = (Levantamento de requisitos)

LEVANTAMENTO DE REQUISITOS

ALGORITMO SOBRE ALGARISMOS ROMANOS

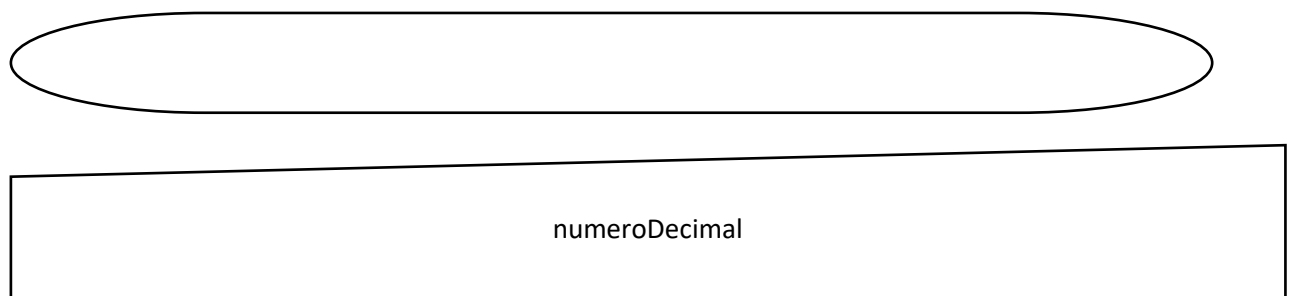
A finalidade desse programa é propor ao usuário uma maior facilidade em saber números romanos, um exemplo é que a maioria dos programas existentes não possibilitam ao usuário uma pesquisa rápida e eficaz, como nosso projeto é relacionado a entregar números romanos para o usuário com velocidade, nos

mudaremos o conceito do usuário ficar buscando informações do que ele deseja, com isso nosso projeto que no momento é relacionado a números romanos, mas pode ser ampliado, proporcionaremos ao usuário uma facilidade em sua procura, nisso vamos propor uma barra de pesquisa em que o indivíduo somente digitara o número que ele quer saber, em que atualmente o utilizador tem que ficar procurando até achar, mas não é só isso, se levarmos em conta que os números romanos vão até 9999, que é o que pretendemos estabelecer em nosso programa (podendo ser aumentado os números) , nisso o usuário não perderia seu tempo e o mais importante receberia o resultado com uma velocidade imprescindível.

Falando um pouco mais do que o usuário terá que digitar, que basicamente aparecera para o mesmo como se fosse uma barra de pesquisa, onde o mesmo digitara o numero em algarismo, com isso o programa ira ler e mostrar ao usuário em números romanos, exemplo: o utilizador digitara 123 e o programa irá apresentar em número romano, que seria CXXIII. Não só isso, ele pegara os algarismos e olhara a casa decimal que cada um está, a partir disso convertera para números romanos.

Considerando tudo isso, o programa basicamente recebera de entrada o número em algarismo decimal, seu processamento é fazer as conversões seguindo regras dos números romanos e casas decimais e pôr fim a saída que exibira ao usuário os algarismos em romanos.

Diagrama de bloco



unidade: {'','I','II','III','IV','V','VI','VII','VIII','IX'}

dezena: {'','X','XX','XXX','XL','L','LX','LXX','LXXX','XC'}

centena: {'','C','CC','CCC','CD','D','DC','DCC','DCCC','CM'}

milhar: {'','M','MM','MMM','MMMM','V','VI','VII','VIII','IX'}

string -> unidadeRomano, dezenaRomano, centenaRomano, milharRomano

se numero.caracteres >= 1:

char u = numeroDecimal[numeroDecimal.Length - 1]

char uInt = int.Parse(u)

unidadeRomano = unidade[uInt]

se numero.caracteres >= 2:

char d = numeroDecimal[numeroDecimal.Length - 2]

char dInt = int.Parse(d)

dezenaRomano = dezena[dInt]

se numero.caracteres >= 3:

char c = numeroDecimal[numeroDecimal.Length - 3]

char cInt = int.Parse(c)

centenaRomano = centena[cInt]

se numero.caracteres == 4:

char m = numeroDecimal[numeroDecimal.Length - 4]

char mInt = int.Parse(m)

milharRomano = milhar[mInt]

