UNIVERSIDADE DE ÉVORA

Departamento de Informática

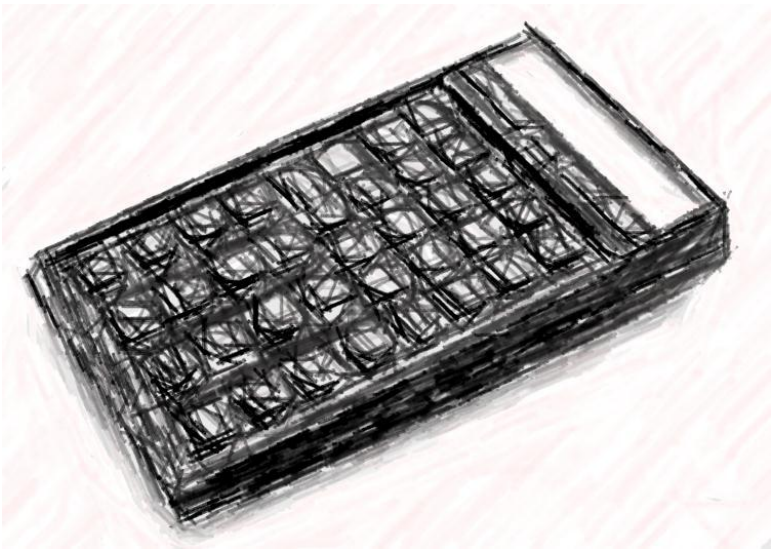
licenciatura em Engenharia Informática

- CALCULADORA RPN -

(RELATÓRIO FINAL)

(Cadeira de Arquitectura de Sistemas e Computadores I- 2º Semestre)

1ºAno



Eduardo Simão Ramos Almeida Costa Martins, 29035

Daniel Gonçalo Jesus Ramos, 29423

Évora, 2012

Índice

Introdução

Uma calculadora deste género (notação polonesa inversa) obedece a certas características que influenciam a sua programação uma vez que os operandos antecedem sempre os operadores, para qualquer que seja a operação, inclusive a conversão de números positivos para negativos não se realiza através do convencional sinal negativo antes do número, mas sim com a operação ***"neg"*** colocada depois do número que se deseja converter.

Desta forma o utilizador pode querer usar o interface de duas formas distintas.

-recorrendo ao uso de uma string sequencial com todos os operandos e operadores ordenados e separados devidamente por espaços.

```
1 1 + 5 swap – 2 + 5 * neg 50 + 5 / Enter
```

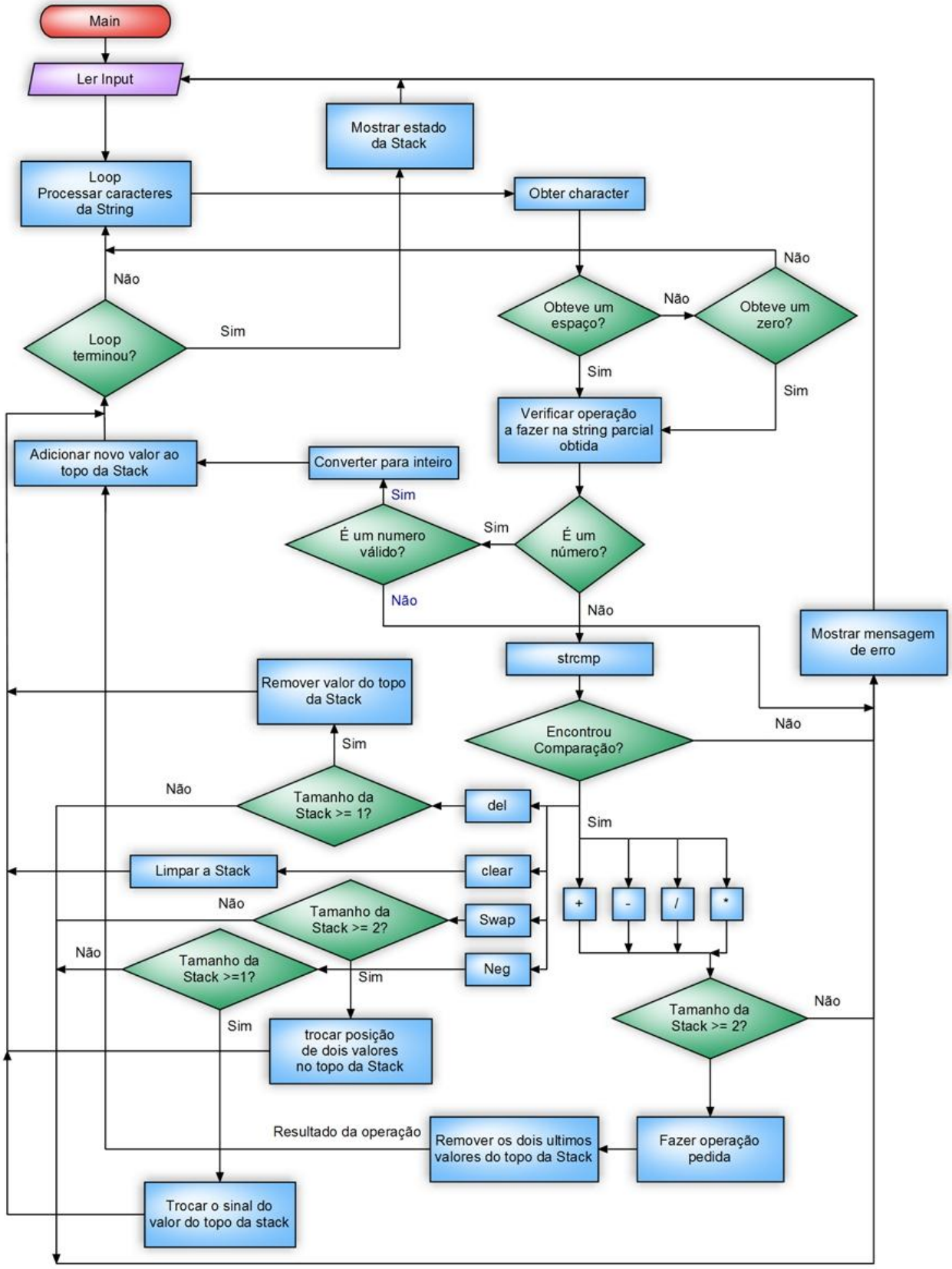
-ou enunciando operandos e operadores de maneira convencional:

```
1 Enter
1 Enter
+ Enter
5 Enter
swap Enter
- Enter
...
```

Para a programação da calculadora, vários procedimentos poderiam ser escolhidos para o tratamento da string e posterior resolução das operações. O mais eficaz seria um procedimento que conseguisse abranger ambas as formas de usar o interface. Por essa razão planeamos programar um loop que identifique os espaços em branco na string e que trabalhe com ela caracter a caracter, e quando encontrar um zero (null) será o ponto final da string, seja sequencial seja pelo modo convencional(string simples), interrompendo o loop e aguardando outro input do utlizador.

A pilha é um array que será alocado na memória dinâmica para que esteja presente durante todo o programa.

Calculadora notação polonesa inversa



Funcionamento Geral do Programa

Baseando-nos no fluxograma acima apresentado podemos delinear os principais traços do programa bem como seguir o seu funcionamento básico:

-o utilizador insere um input na consola que será colhido pelo loop principal inserido na função main, após a string ser inserida pelo utilizador é feito um jump para outra função que contém um loop responsável pelo processamento dos caracteres da string.

-a obtenção de caracteres é orquestrada pela existência de espaços na string sequencial e zero(null) no fim desta ou na simples. Por outras palavras, numa string sequencial o loop irá procurar o primeiro espaço e tratar o operador e/ou operando antes deste, passando para o próximo espaço iterativamente durante toda a string sequencial. No caso da string simples, irá ser detectado o zero(null) e usar o mesmo processo que na string sequencial com a diferença que no fim o programa não irá voltar ao loop (processar characters da string), mas sim pedir o input ao utilizador.

-se foi obtido um número este será convertido para inteiro e adicionado à pilha(stack) para posterior tratamento da operação e vizualização do utilizador, com a excepção de números negativos (-1) que são possíveis de ser convertidos mas que segundo a notação polonesa inversa não é permitido (sendo por isso *1* ***"neg"*** e não -1). Caso o utilizador tente colocar o input -1, a consola irá mostrar uma mensagem de erro, e o programa voltará a aguardar pelo input do utilizador.

-se o caracter não é um número irá ser comparado na função strcmp (string compare) e procurar uma igualdade entre a string originada na função anterior e outras pré-definidas, para o programa reconhecer a operação em causa, se não encontrar correspondência a consola irá mostrar uma mensagem de erro e irá voltar a pedir o input ao utilizador.

-se as operações detectadas forem soma, subtração, divisão e multiplicação (+,-,/,*) irá-se comparar o tamanho da pilha e se não existirem pelo menos 2 operandos, a operação não poderá ser efectuada logo a consola irá mostrar uma mensagem de erro e irá pedir de novo o input ao utilizador. Caso haja 2 ou mais operandos na pilha irá ser realizada a operação pretendida e posteriormente serão removidos os dois valores do topo da pilha, e será adicionado o resultado da operação realizada de novo ao topo da pilha, retornando á função (processar caracters da string).

-se a operação pedida pelo utilizador for ***"del"*** (delete) e existirem valores na pilha, o del irá remover o valor do topo desta e voltar á função (processar caracters da string), se a pilha estiver vazia a consola irá mostrar uma mensagem de erro e o programa volta a pedir o input ao utilizador.

-se a operação processada for ***"clear"*** o programa remove todos os valores na pilha, e volta á função (processar caracters da string).

-se na string estiver a operação ***"swap"***, na pilha terão de existir obrigatoriamente dois valores para poderem ser trocados, se existirem, trocam-se os últimos dois valores do topo da pilha e retorna-se á função (processar caracters da string), se não existirem no mínimo dois valores a consola mostrará um mensagem de erro e voltará a pedir o input ao utilizador.

-para última operação se estiver presente na string o ***"neg"*** terá de existir pelo menos um valor na pilha para se poder converter no seu simétrico, se não existir, a consola, assim como para as outras operações, irá mostrar uma mensagem de erro.

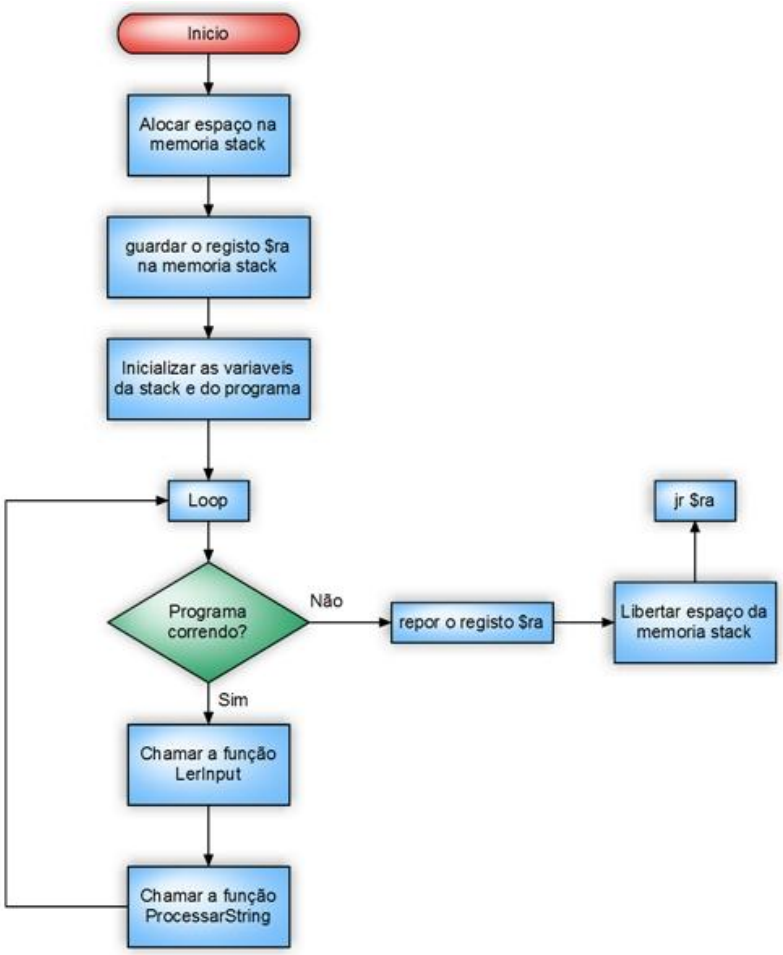
Desenvolvimento e Detalhes do Programa

Após a projecção preliminar do programa apresentada no capítulo anterior é imperativo fazer parte deste relatório, uma análise mais detalhada da organização do programa, contendo as funções que dele fazem parte, e a explicação do intuito da criação da mesma função bem como a exemplificação (recorrendo a um fluxograma) do comportamento da mesma. Com o código finalizado apuramos que existem 29 funções que passaremos a descrever.

Main

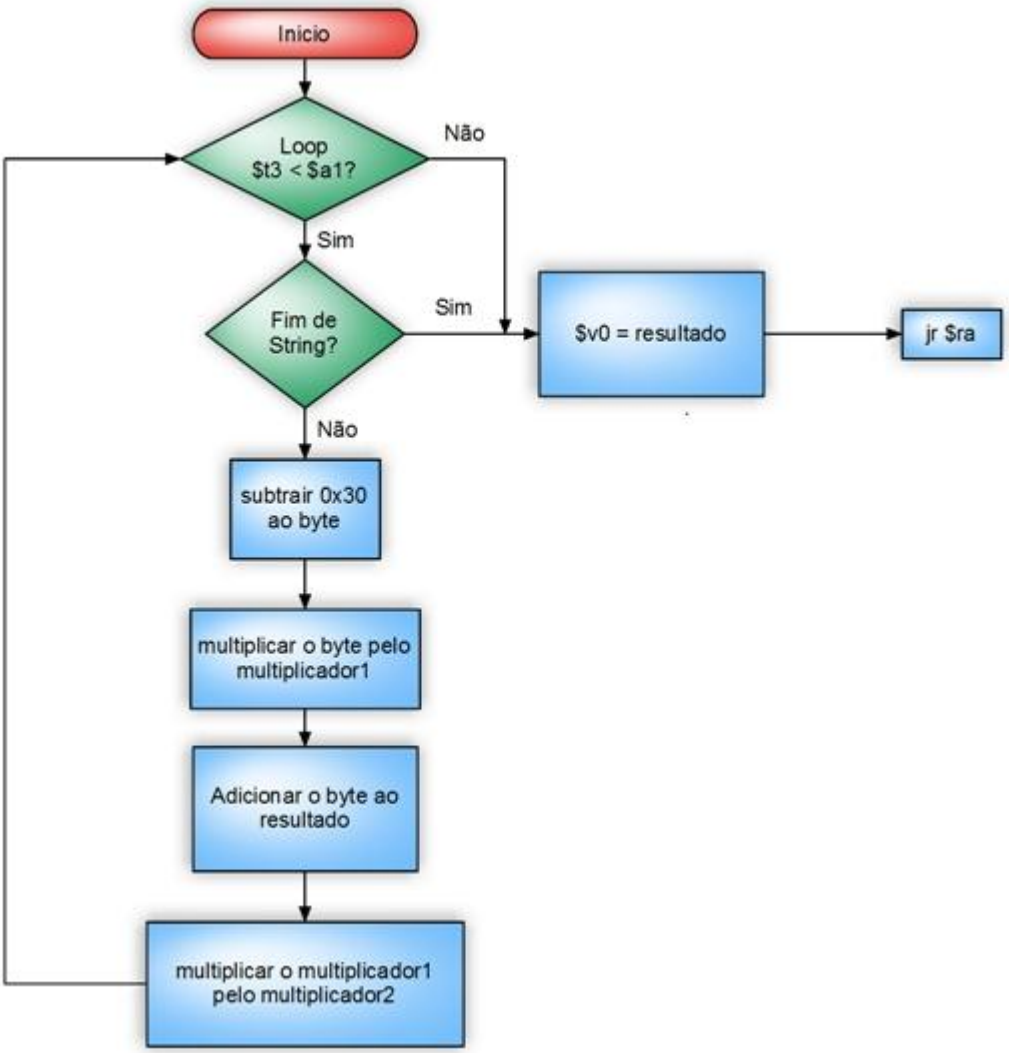
Função que aloca a memória dinâmica para guardar os valores da pilha, para além de incializar os valores do tamanho e da posição da stack.

Esta função aguarda pelo input do utilizador chamando a função ProcessarString.



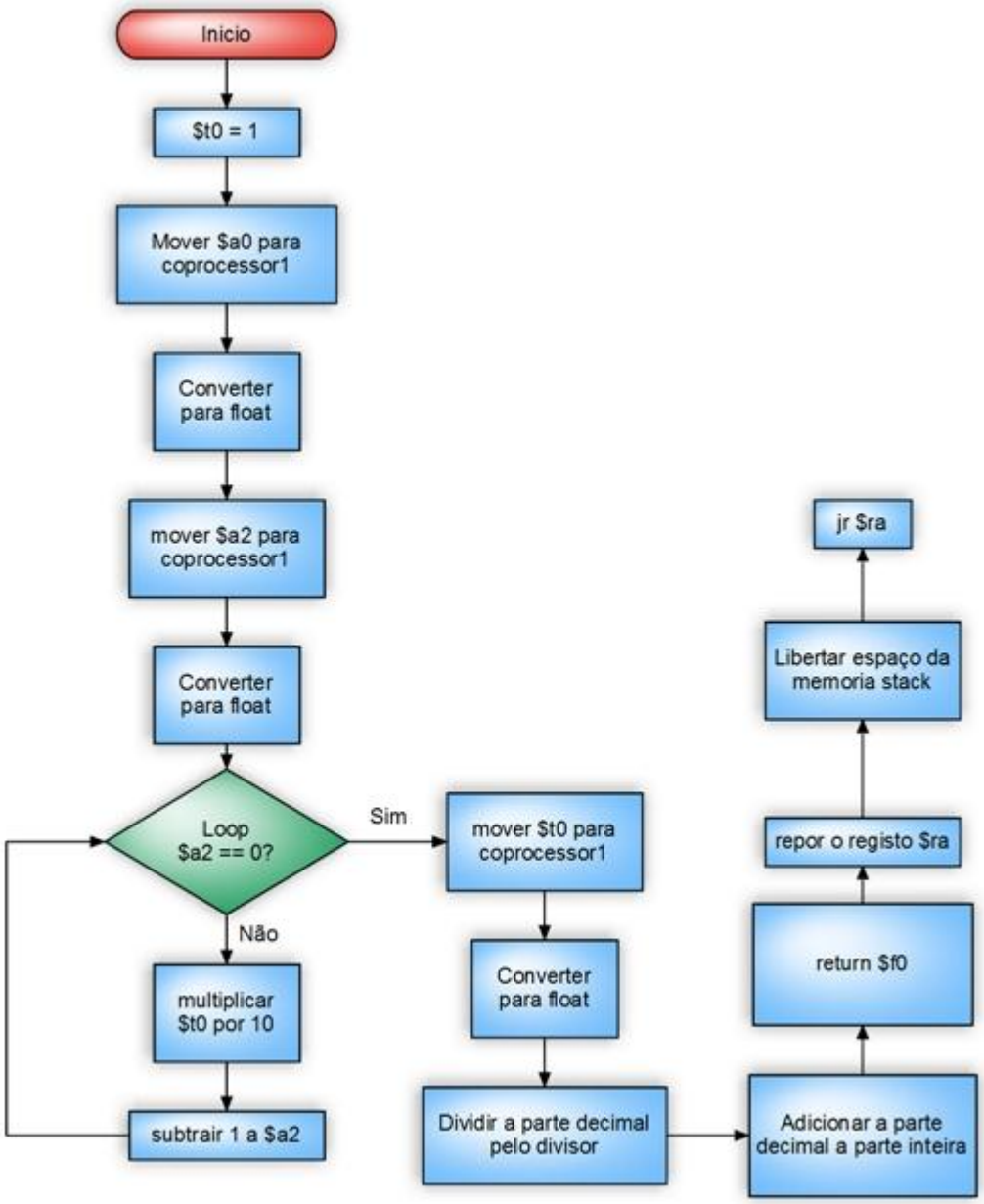
ConverterParaInteiro

Converte uma string para número inteiro, retornando o valor convertido.



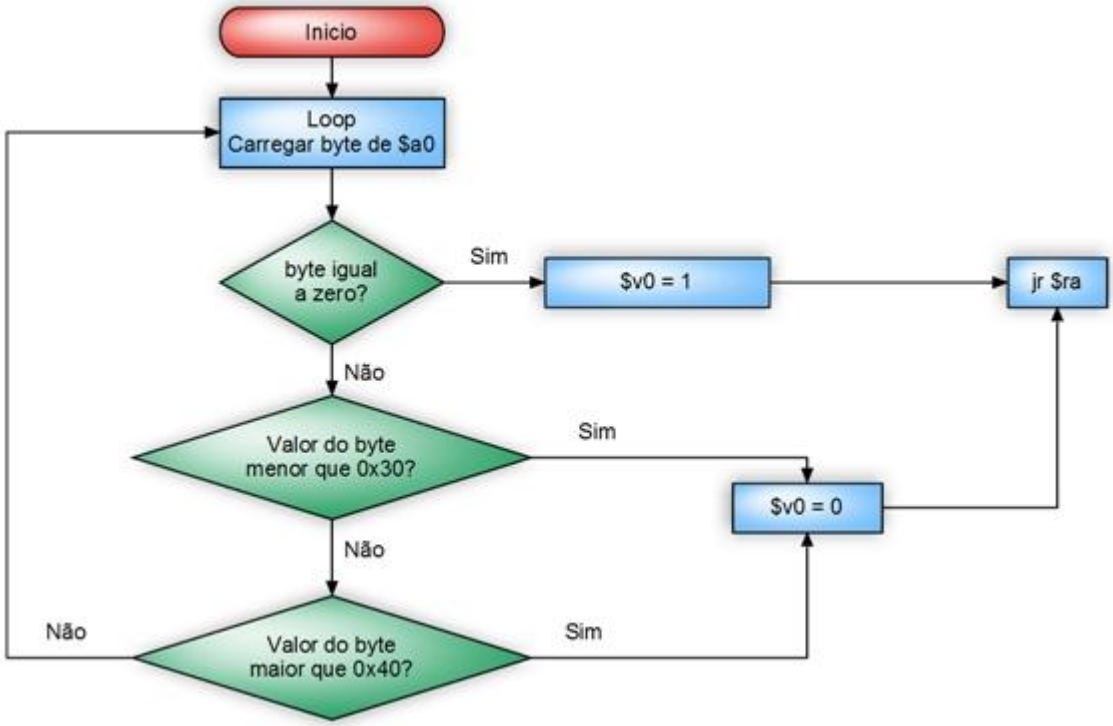
ConverterParaVF

Responsável por converter a parte inteira e decimal para vírgula flutuante. Retornando o valor convertido em float.



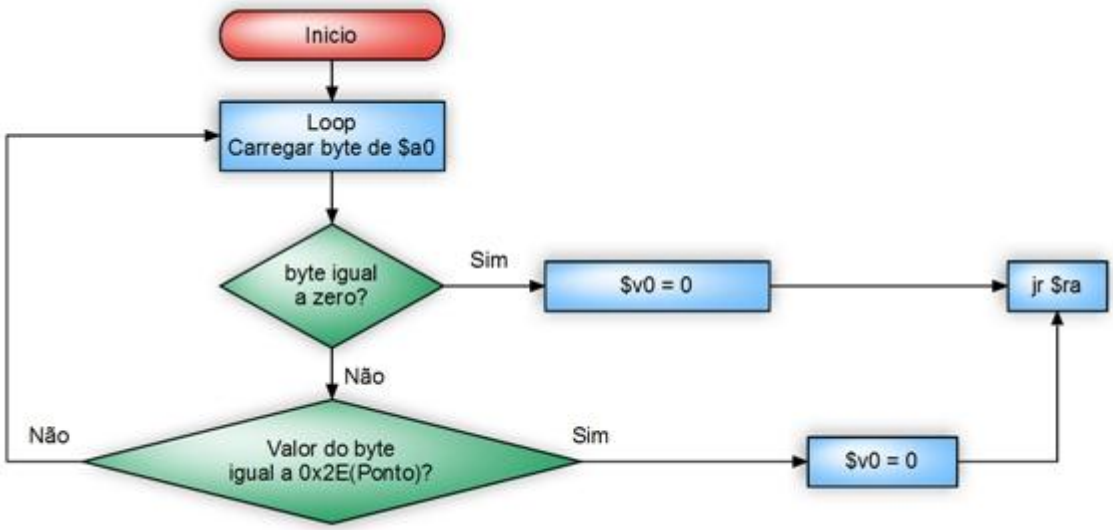
EDigitoInteiro

Verifica se uma dada string contém apenas digítos, retornando verdadeiro se tiver apenas dígitos ou retornando falso para o caso contrário.



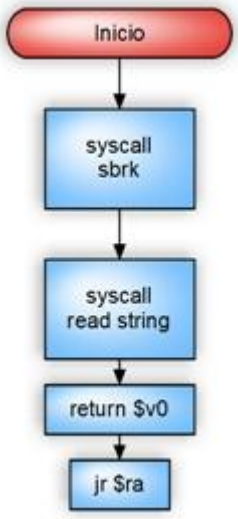
EdigitoVF

Detecta se a string é um número de vírgula flutuante.



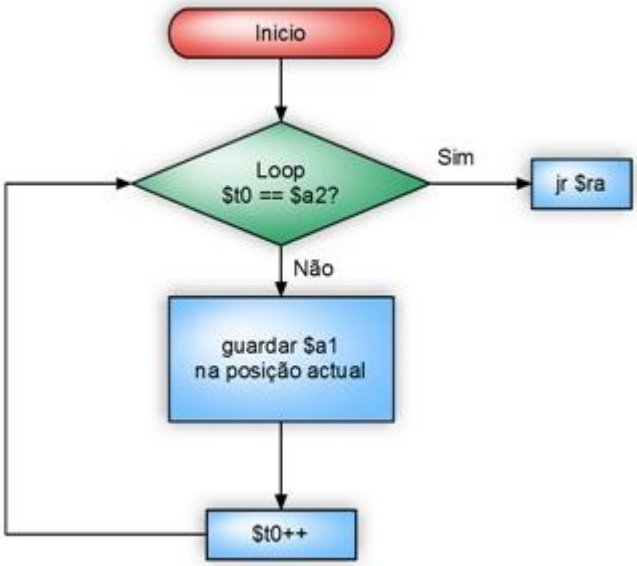
LerInput

Aloca memória e usa-a para guardar o resultado do Input do utilizador.



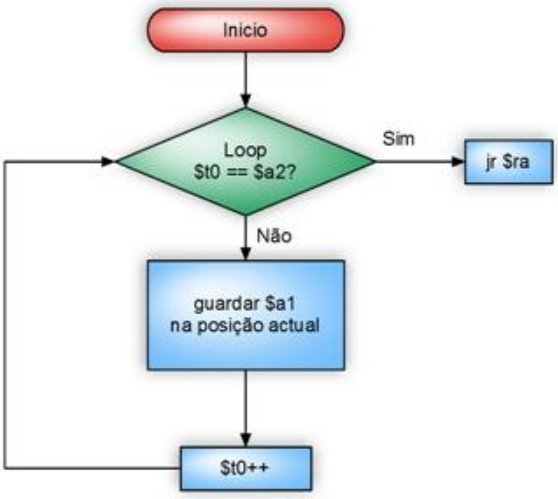
memset

Preenche a memória com um valor.



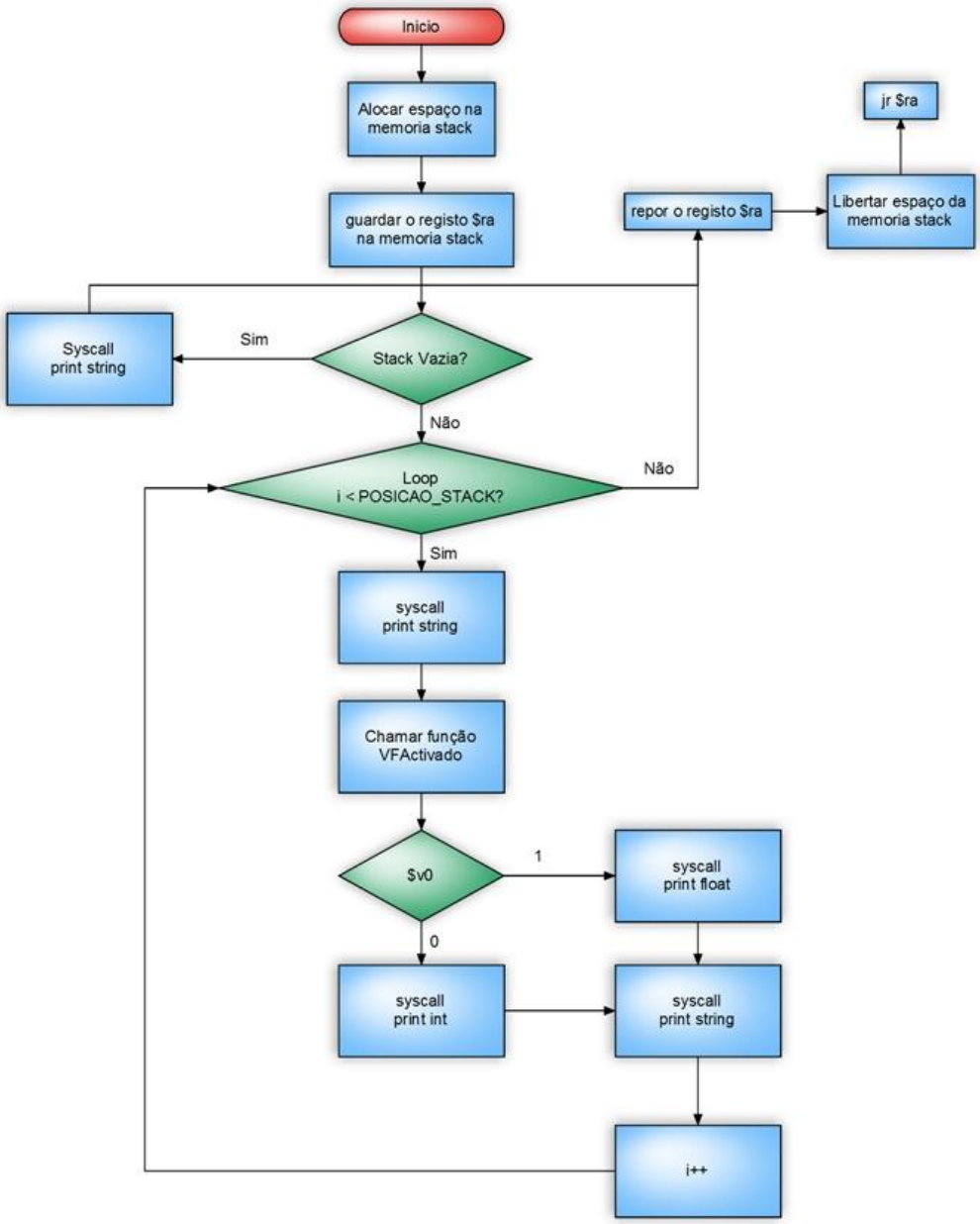
MostrarErro

Responsável por fazer um print do erro que está contido no argumento a0.



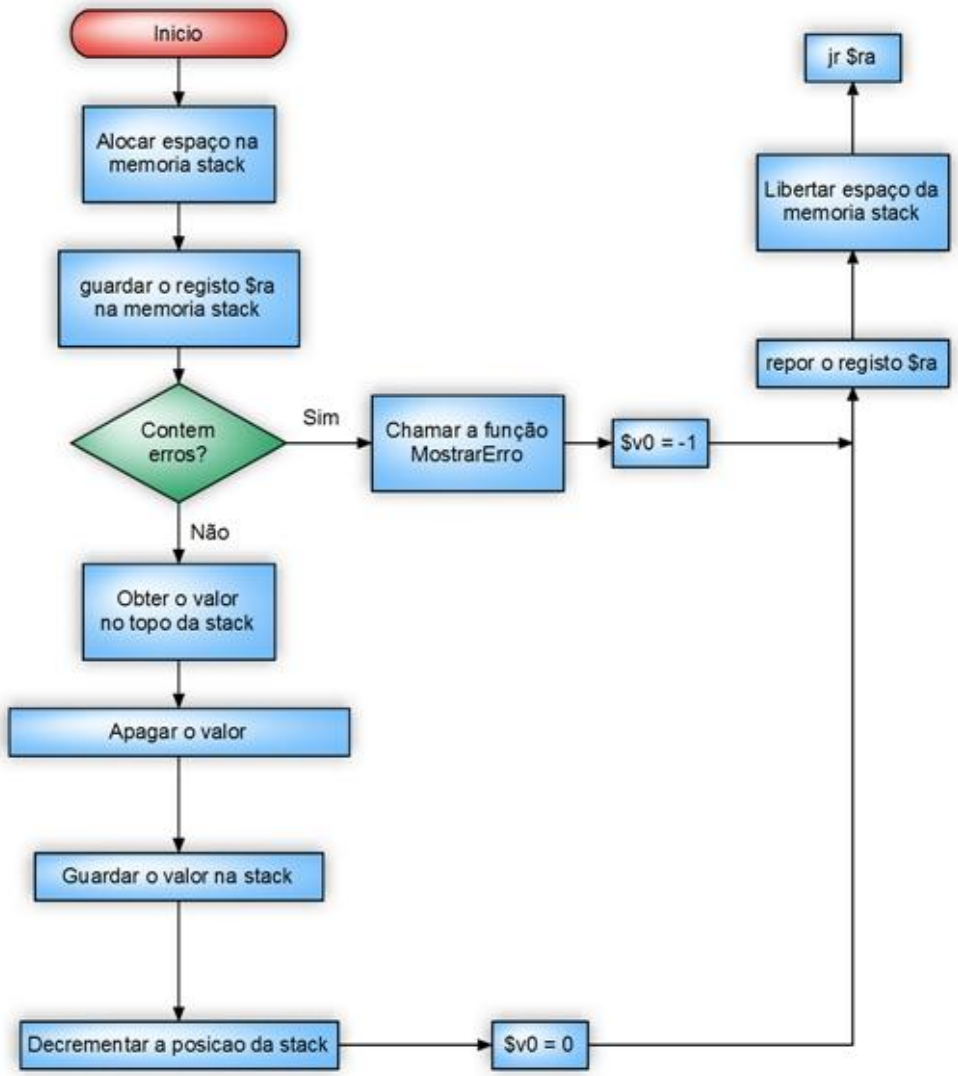
MostrarStack

Mostra o estado actual da pilha, caso o processamento de vírgula flutuante esteja activado esta mostra o conteúdo, mas em vírgula flutuante, para o caso contrário mostra o conteúdo como números inteiros.



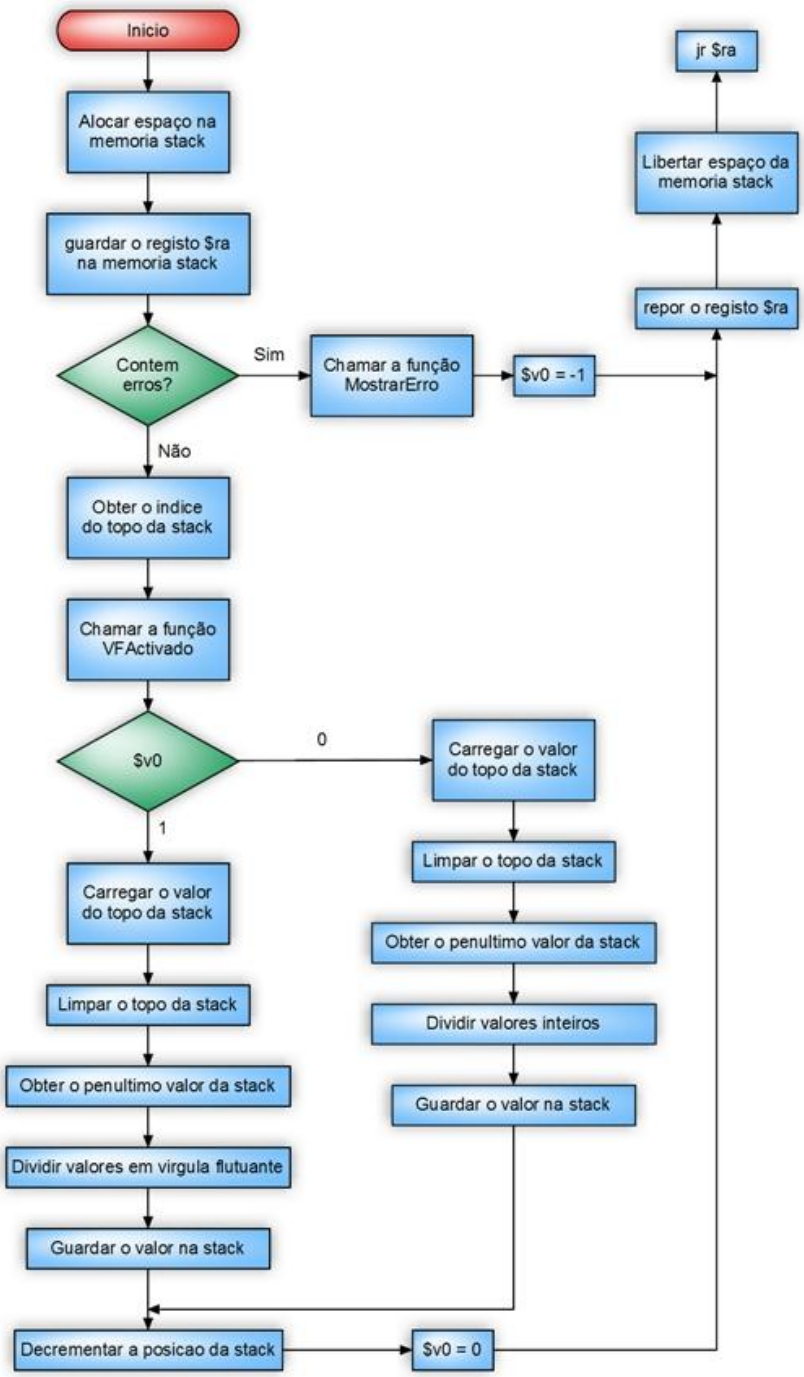
OperadorDel

Apaga o último valor do topo da stack. Retornando se houve ou não erros. Retorna -1 se existe erros ou 0 para o caso contrário.



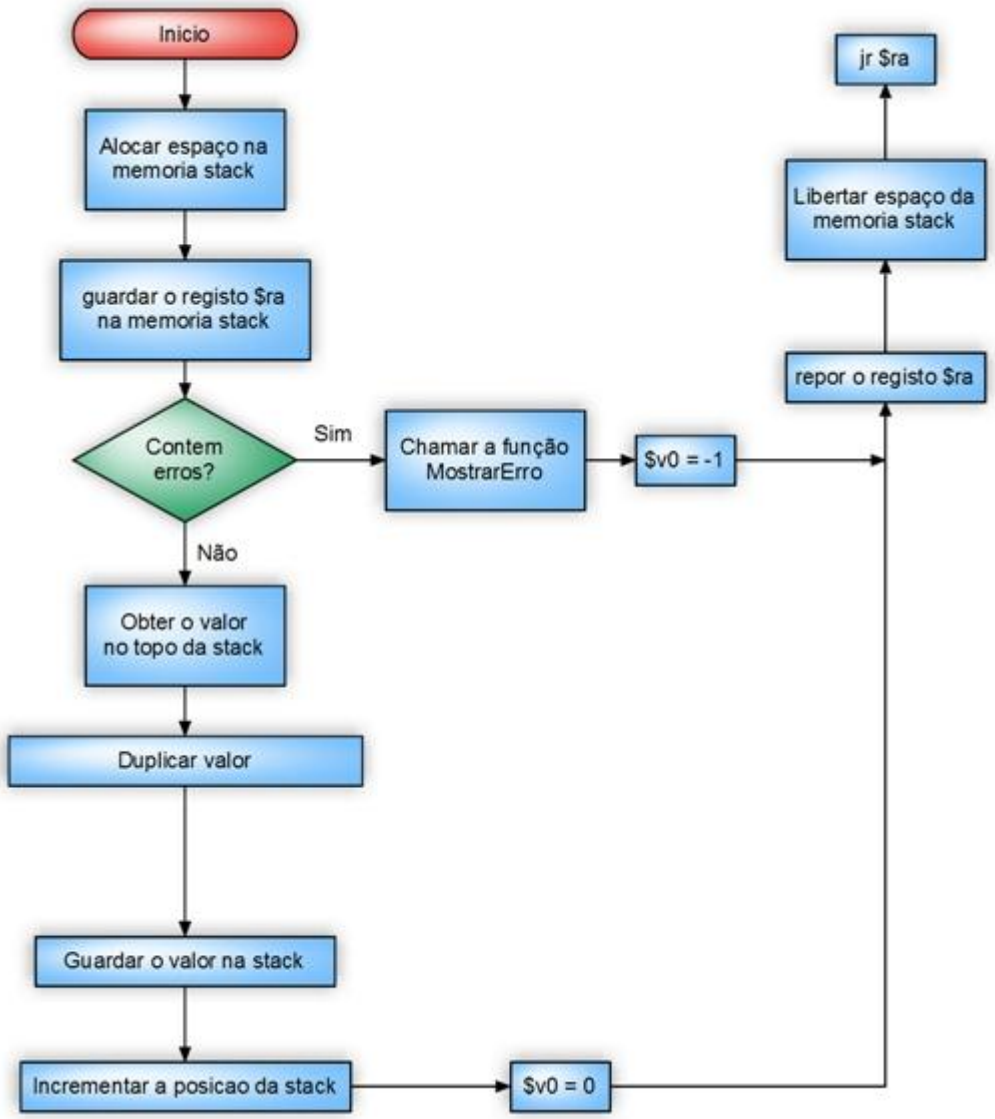
OperacaoDivisao

Faz a divisão dos valores do topo da stack. Retornando se houve ou não erros. Retorna -1 se existe erros ou 0 para o caso contrário.



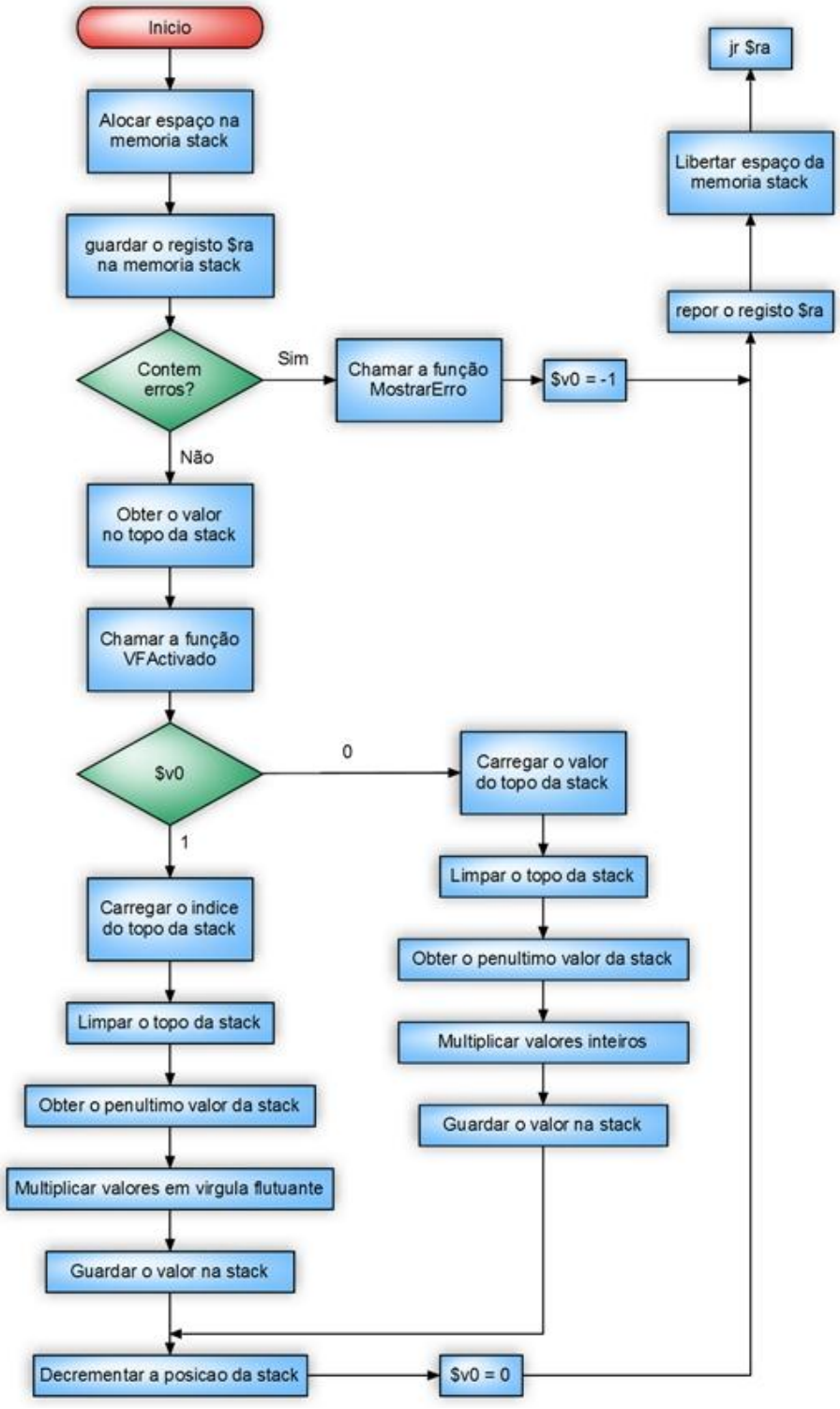
OperadorDup

Duplica o último valor da stack. Retorna -1 se existe erros ou 0 para o caso contrário.



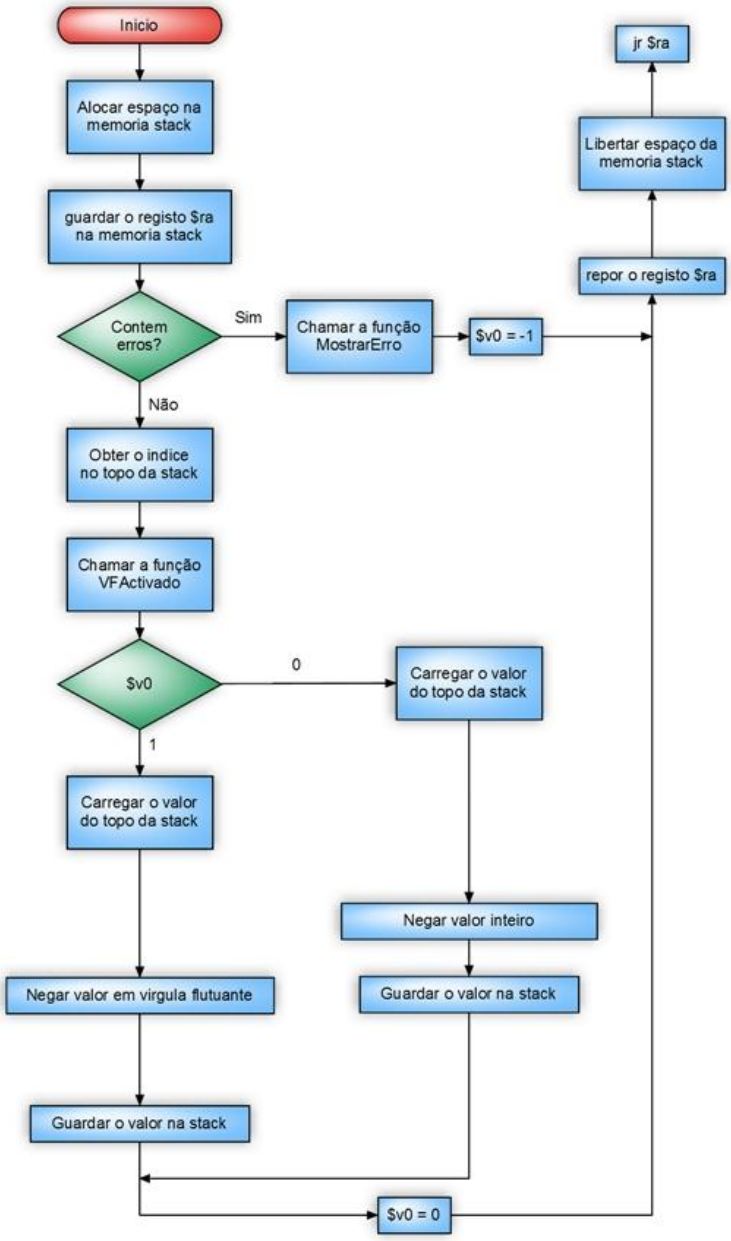
Operacaomultiplicacao

Faz a multiplicacao dos valores do topo da stack. Retornando se houve ou não erros. Retorna -1 se existe erros ou 0 para o caso contrário.



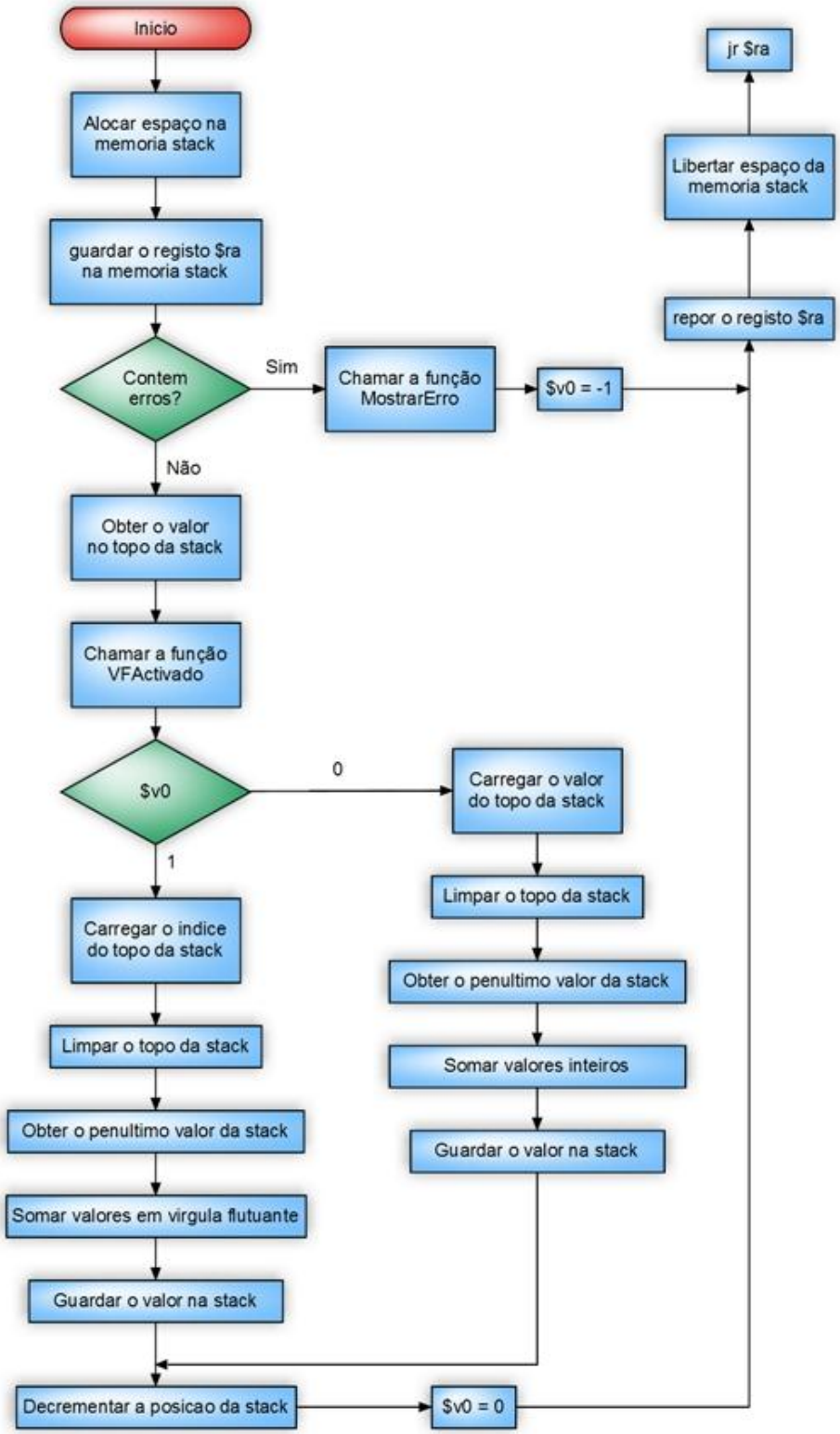
OperadorNeg

Transforma o valor do topo da stack no seu simétrico. Retornando se houve ou não erros. Retorna -1 se existe erros ou 0 para o caso contrário.



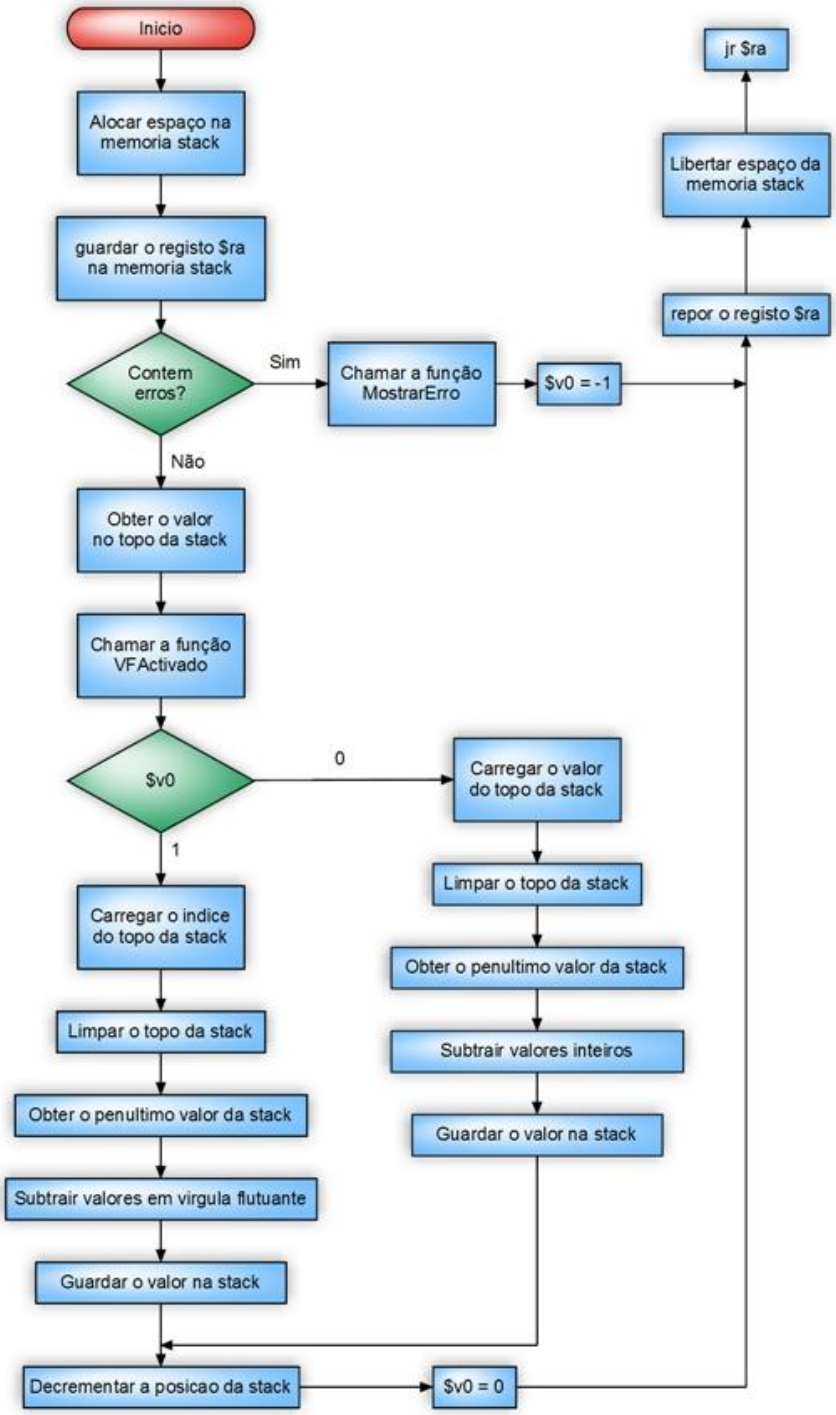
OperacaoSoma

Faz a soma dos valores do topo da stack. Retornando se houve ou não erros. Retorna -1 se existe erros ou 0 para o caso contrário.



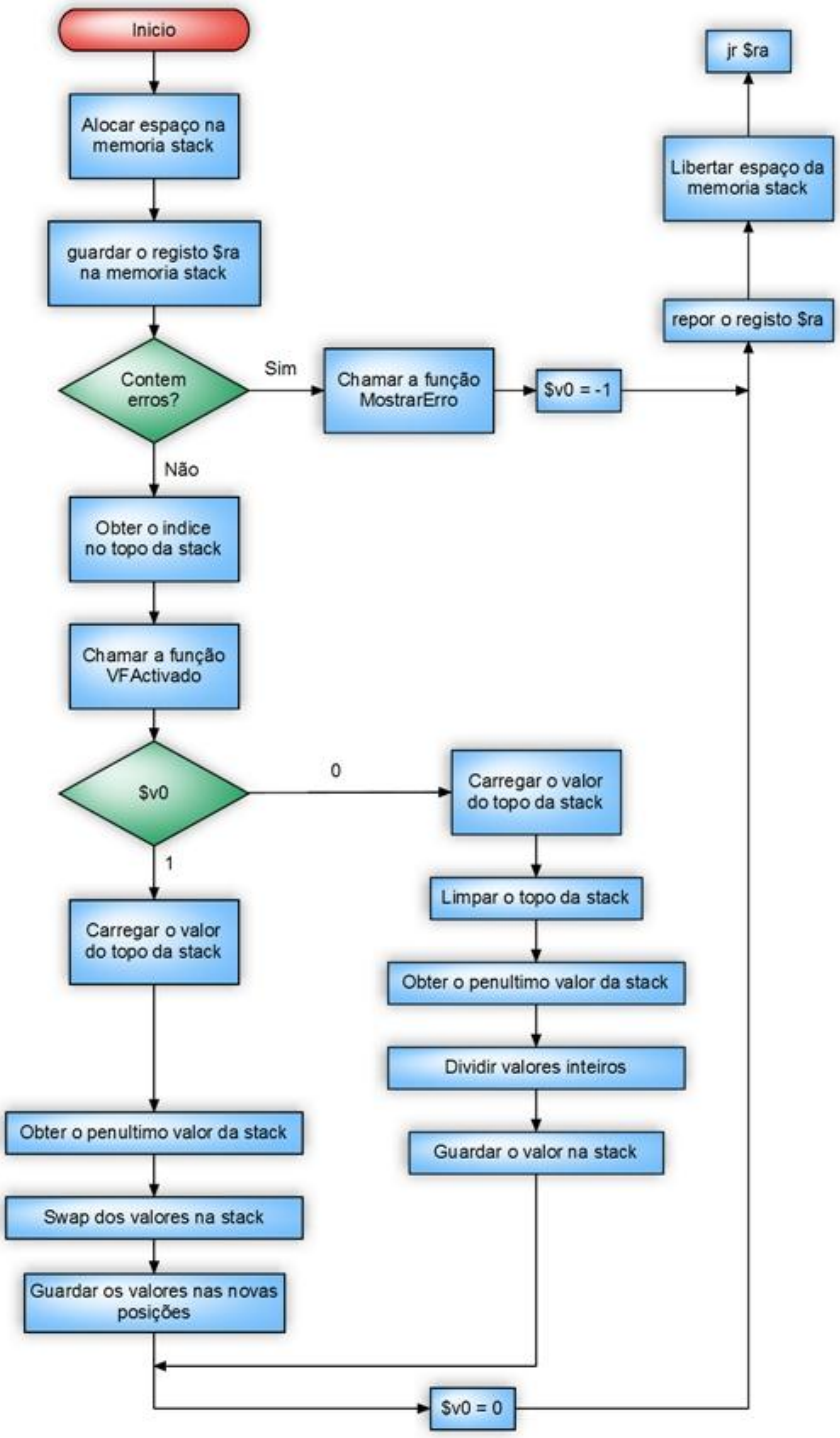
OperacaoSubtraccao

Faz a subtraccao dos valores do topo da stack. Retornando se houve ou não erros. Retorna -1 se existe erros ou 0 para o caso contrário.



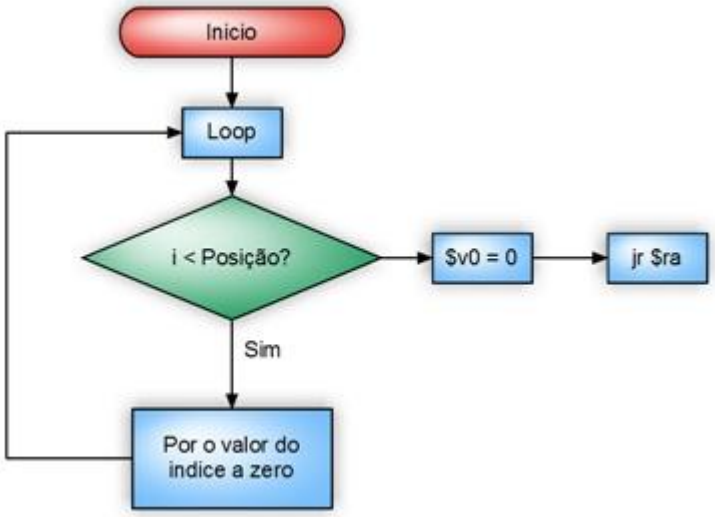
OperadorSwap

Faz a troca dos valores no topo da stack. Retornando se houve ou não erros. Retorna -1 se existe erros ou 0 para o caso contrário.



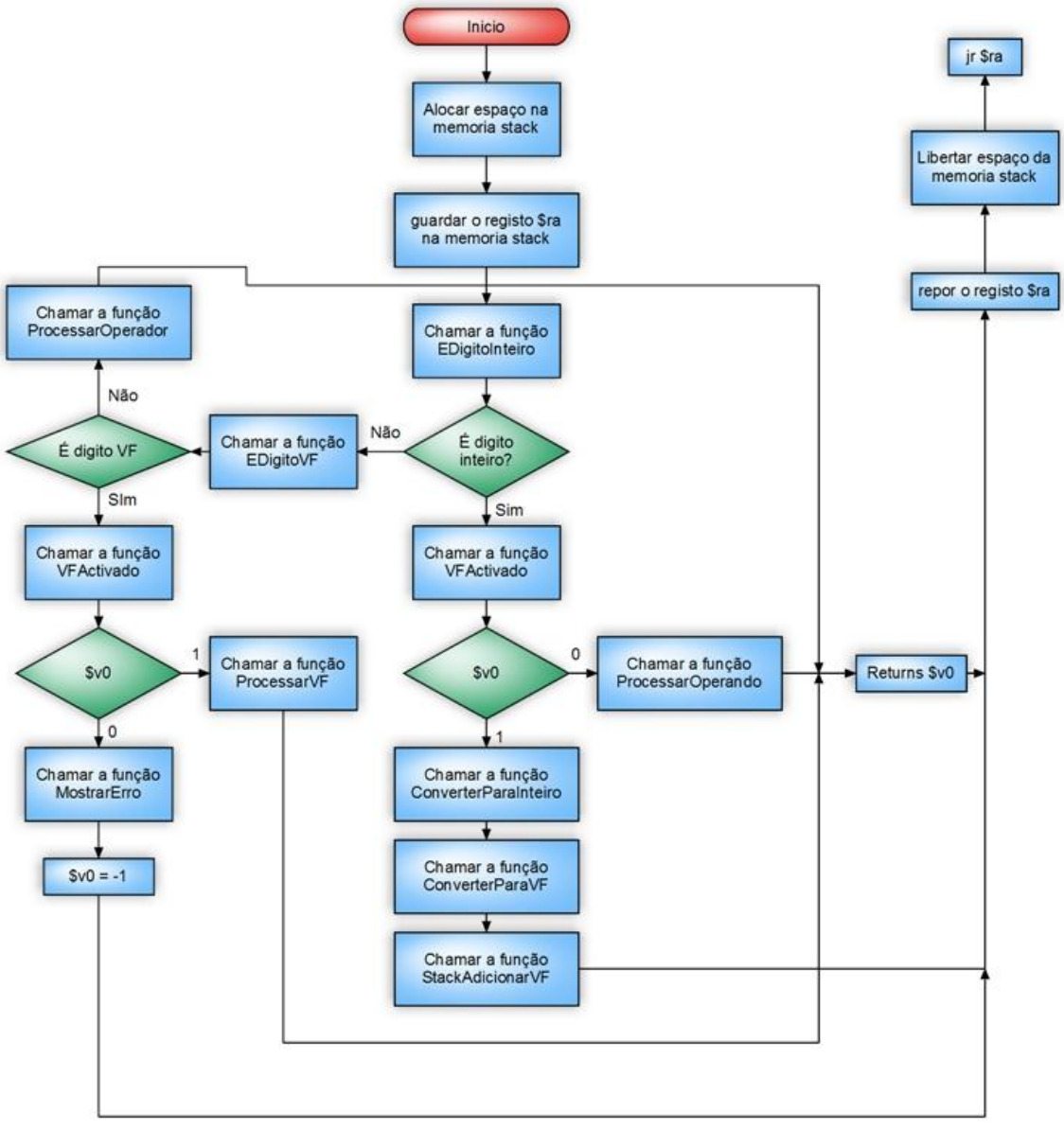
OperacaoClear

Limpa a stack na sua totalidade, colocando a array a 0. Retornando se houve ou não erros. Retorna -1 se existe erros ou 0 para o caso contrário.



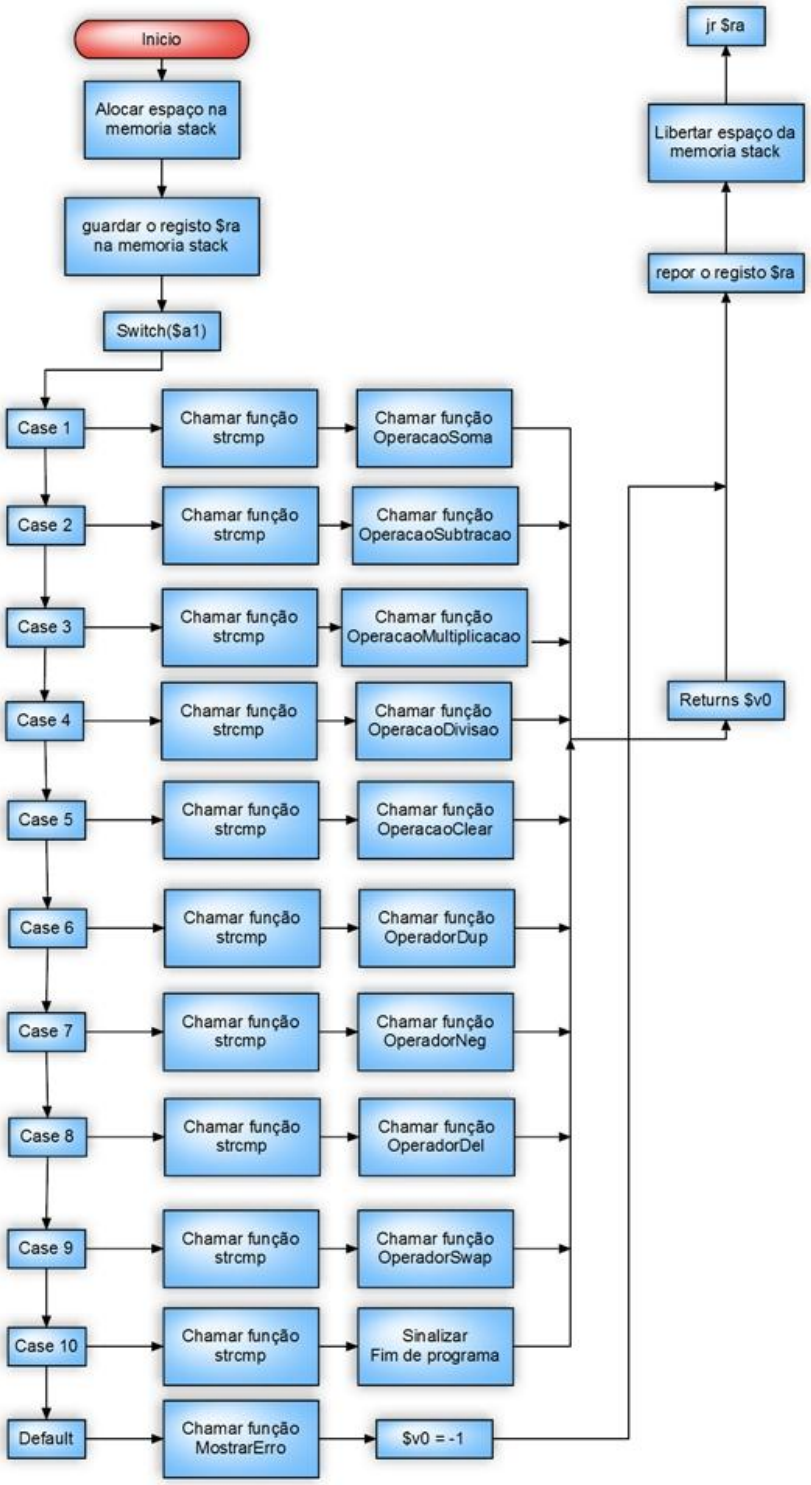
ProcessarComando

Processa a string parcial, independentemente se é operador ou operando.



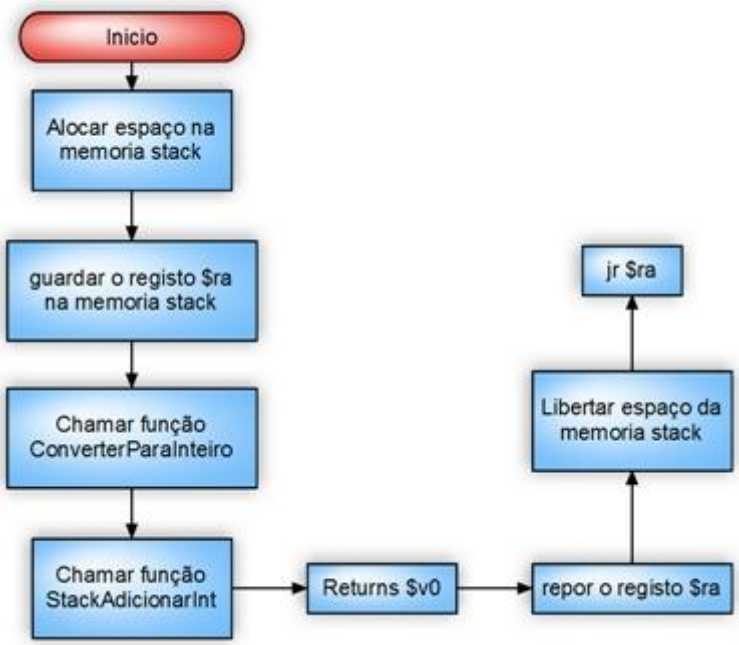
ProcessarOperador

Processa o operador determinando qual a operação a efectuar. Caso não exista operação invoca um erro, retornando -1 caso exista erro e 0 caso não exista.



ProcessarOperando

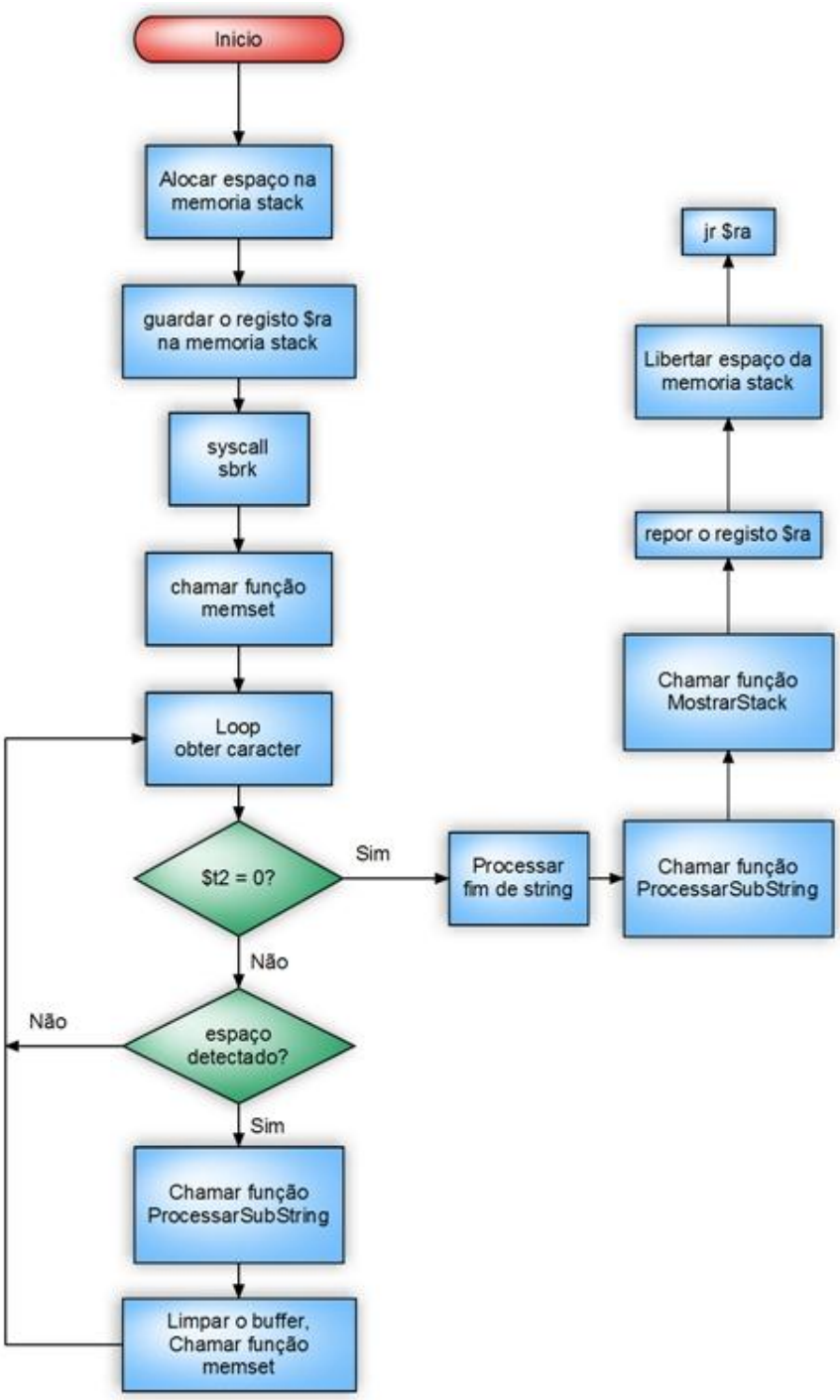
Processa o operando, convertendo-o para inteiro e adicionando-o à stack.



ProcessarString

Processa a string obtida pelo input do utilizador formando strings parciais caso seja necessário e processando as mesmas, permitindo escolher a operação a efectuar.

Retornando erro caso exista, se existir erro retorna -1 se existir retorna 0.

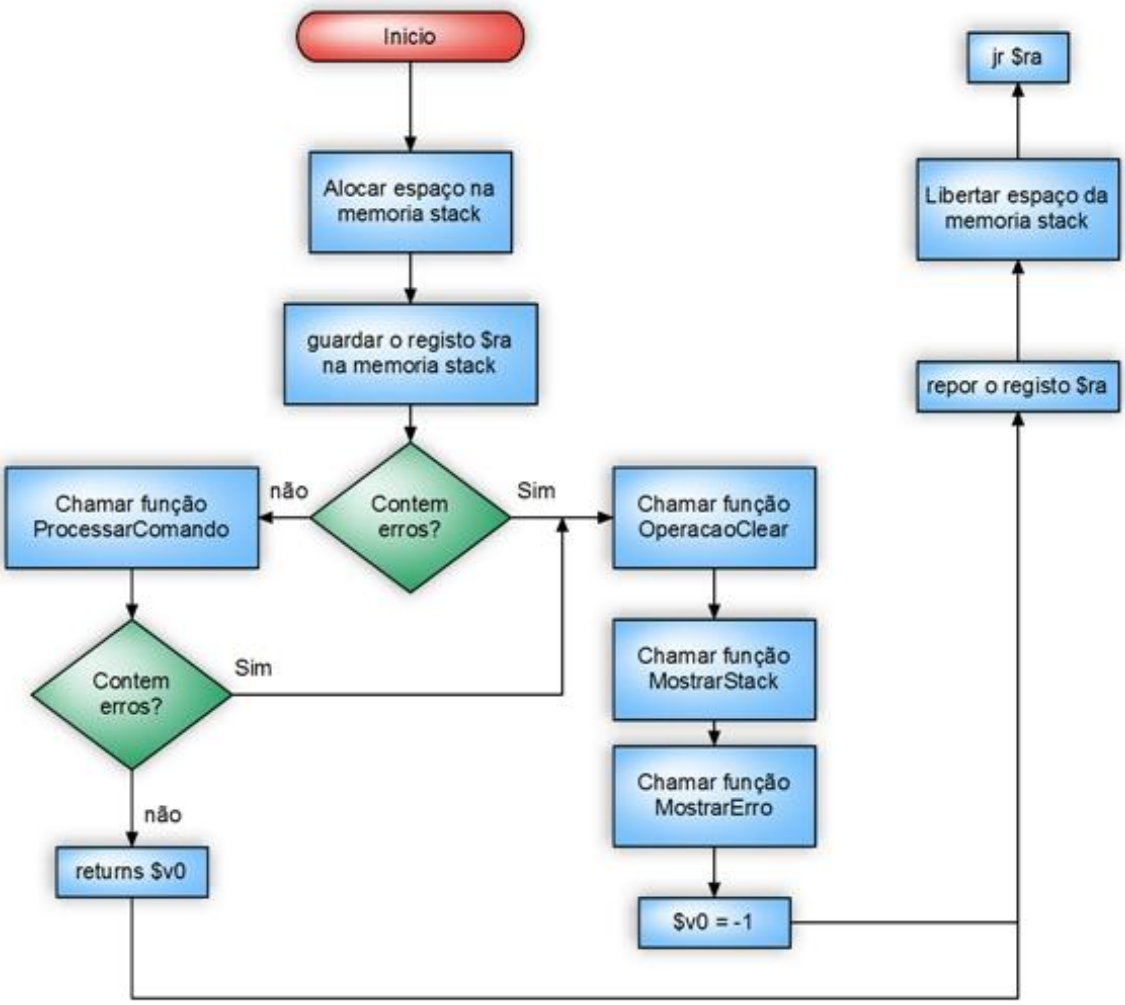


ProcessarSubString

Processa string parcial gerada e verifica qual a operação a efectuar.

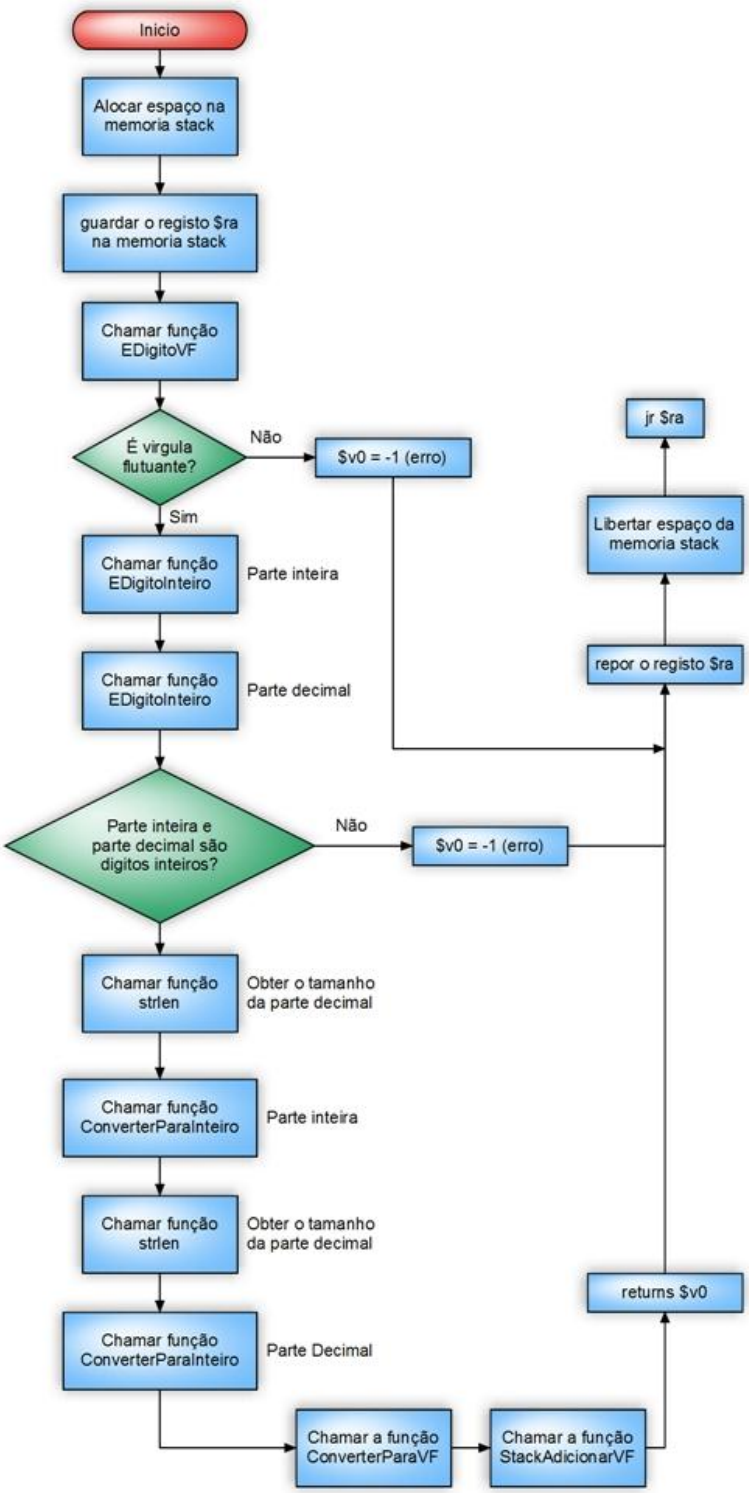
Caso exista erro limpa a stack, mostrando-a, e submete um erro na consola.

Retornando -1 caso exista erro, e caso não exista retorna 0.



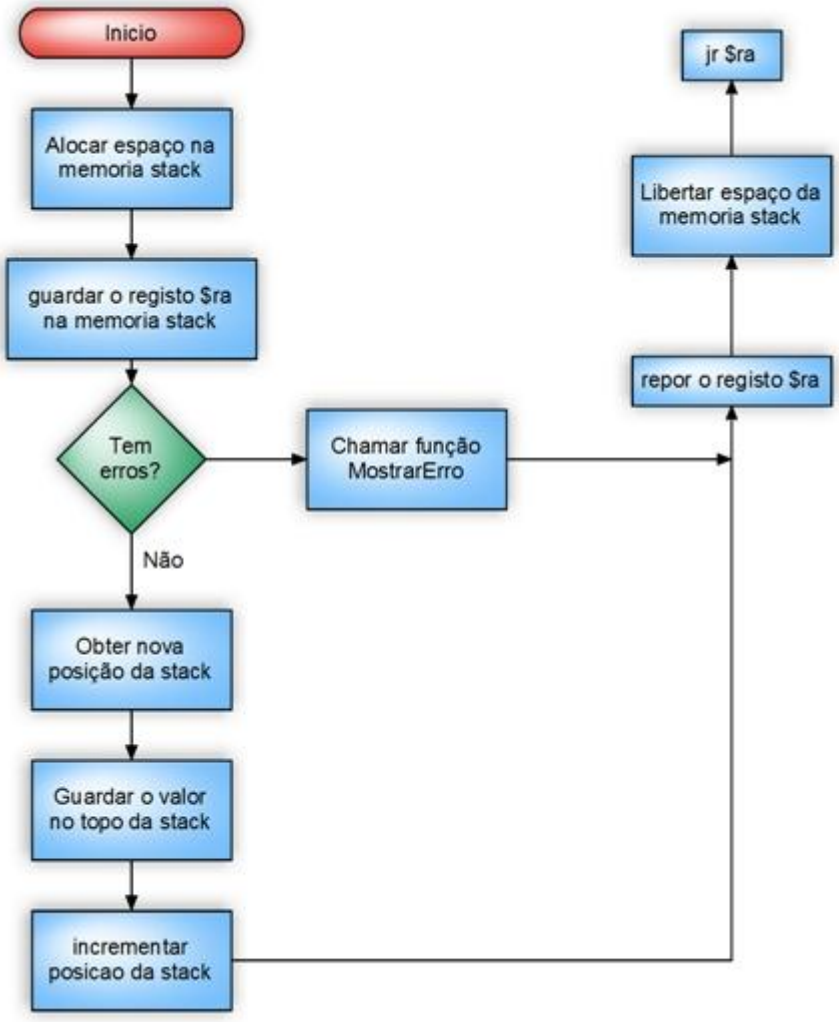
ProcessarVF

Processa a string convertendo-a para vírgula flutuante. Retornando se existe erros. Retorna -1 se existe erros ou 0 para o caso contrário.



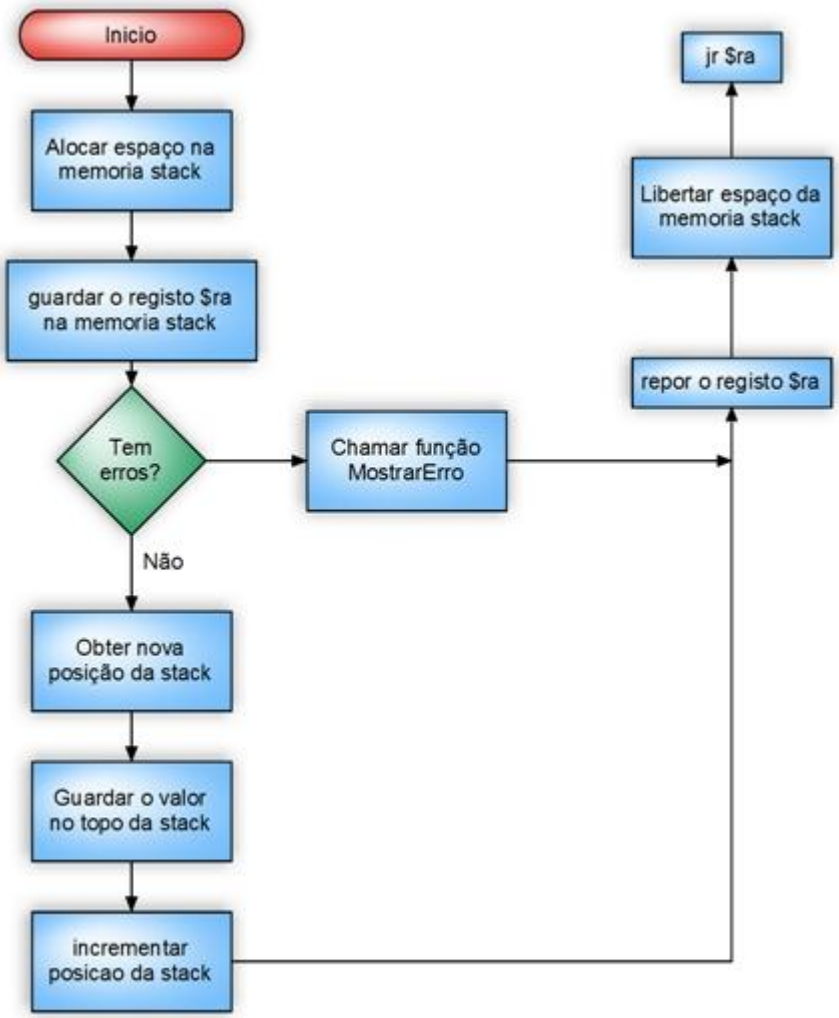
StackAdicionarInt

Adiciona um número inteiro ao topo da stack. Retorna a existência de erros, se existir retorna -1 se não existir retorna 0.



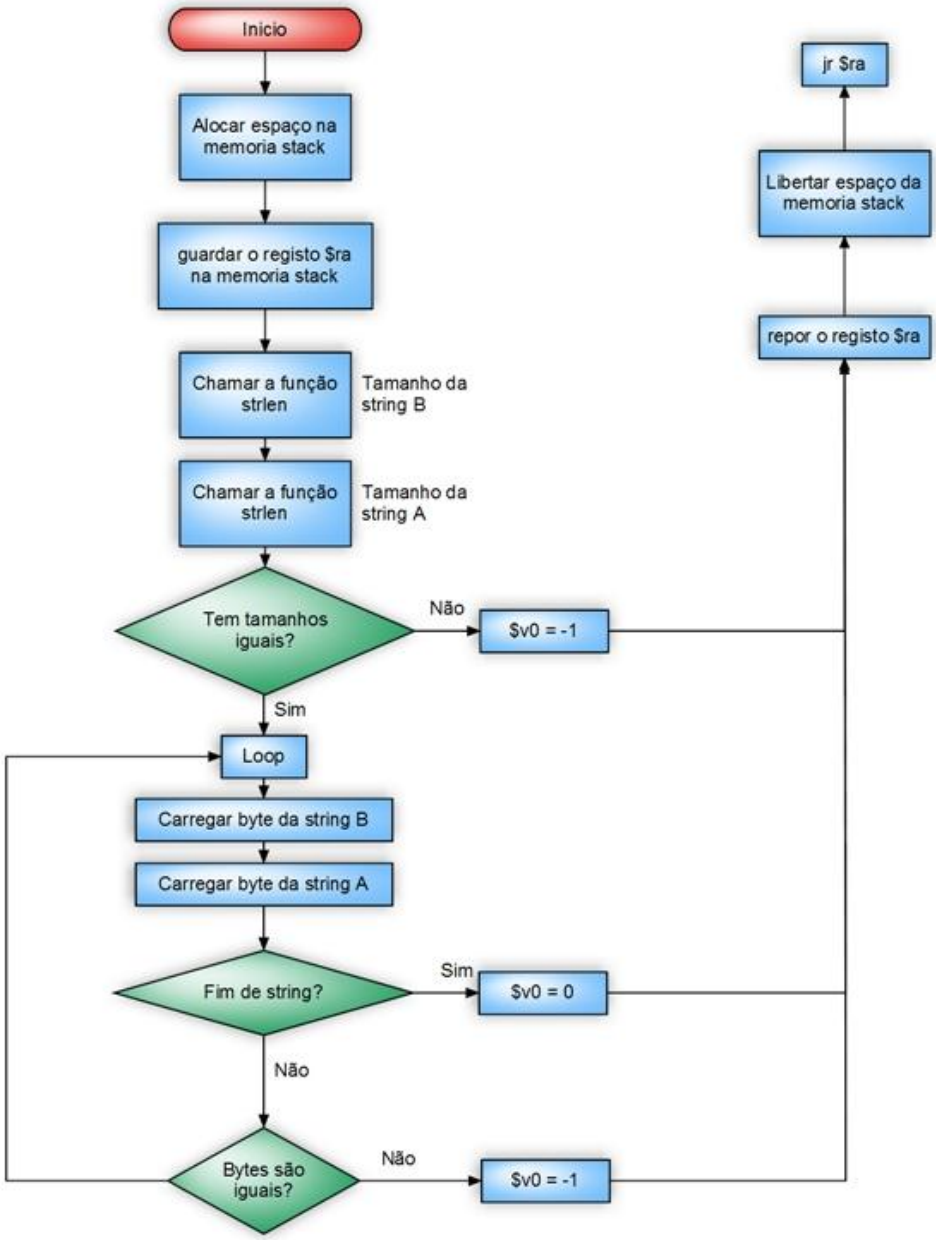
StackAdicionarVF

Adiciona um número em vírgula flutuante ao topo da stack. Retorna a existência de erros, se existir retorna -1 se não existir retorna 0.



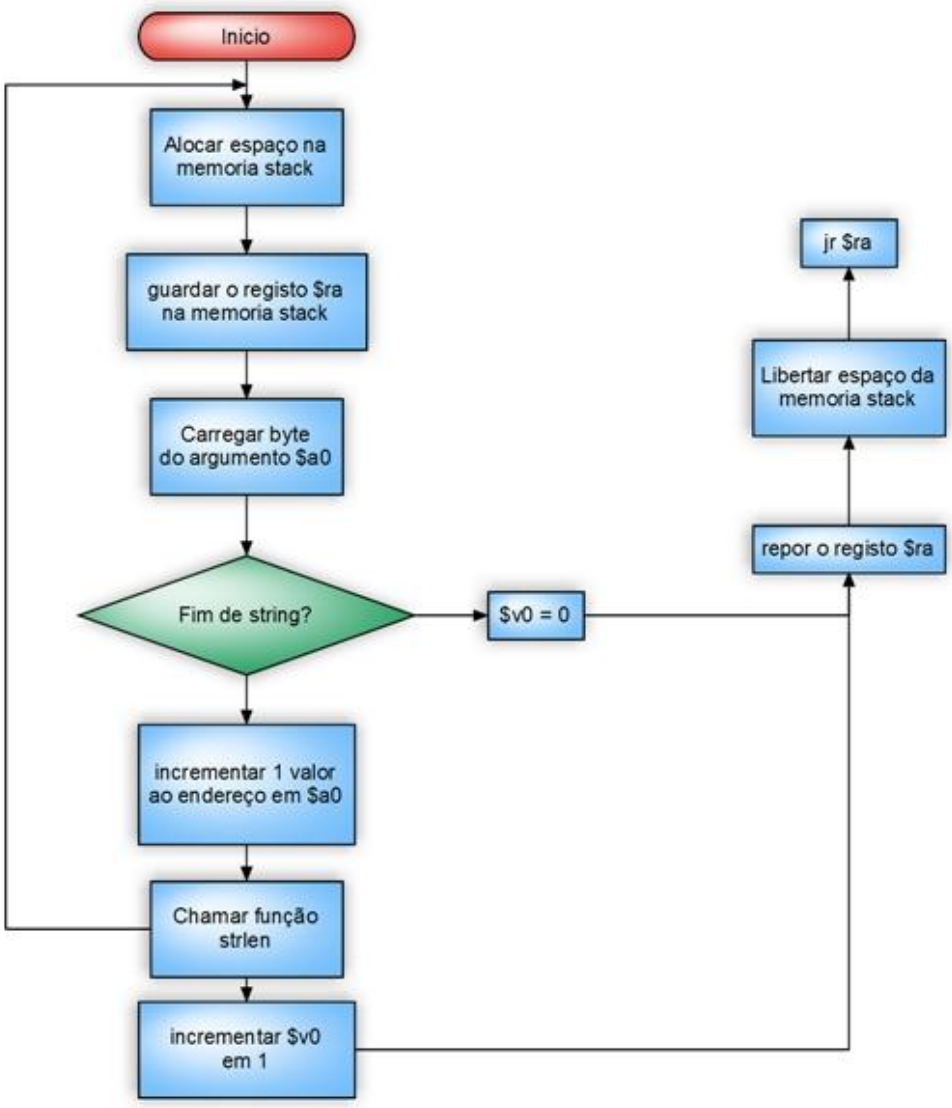
strcmp

Compara duas strings, retornando 0 se forem iguais ou -1 se forem diferentes



strlen

Calcula o tamanho da string retornando o tamanho.



VFActivado

Verifica se o processamento de vírgula flutuante está activado. Retorna Verdadeiro se estiver, e Falso para o contrário.