

Inter-S

Versão GR IV - 2011

Guia de Referência

Sumário

1. Apresentação	2
2. Comandos.....	4
3. Instruções	6
3.1. Instruções de iniciação e declaração	6
3.2. Instruções de entrada e saída.....	7
3.3. Instruções para manipulação de tela.....	8
3.4. Controle de fluxo e estruturas de decisão e repetição.....	10
3.5. Instruções para chamadas externas	14
3.6. Instruções diversas	15
4. Funções	17
4.1. Funções matemáticas	17
4.2. Funções Diversas	17
5. Operadores	18
5.1. Operadores lógicos.....	18
5.2. Operadores matemáticos.....	18
5.3. Operadores relacionais	18
5.4. Operador de caractere	19
6. Arquivos de dados	20
6.1. Declaração de arquivo	20
6.2. Abertura de arquivos de dados	20
6.3. Leitura de registros	21
6.4. Gravação de registros.....	21
6.5. Exclusão de registros.....	21
6.6. Fechamento de arquivos.....	22
6.7. Topo do arquivo.....	22
6.8. Fim de arquivo (FDA).....	22
7. Instruções para impressão	24
8. Instruções gráficas.....	24
9. Variáveis	25

1. Apresentação

Bem-vindo ao INTER-S

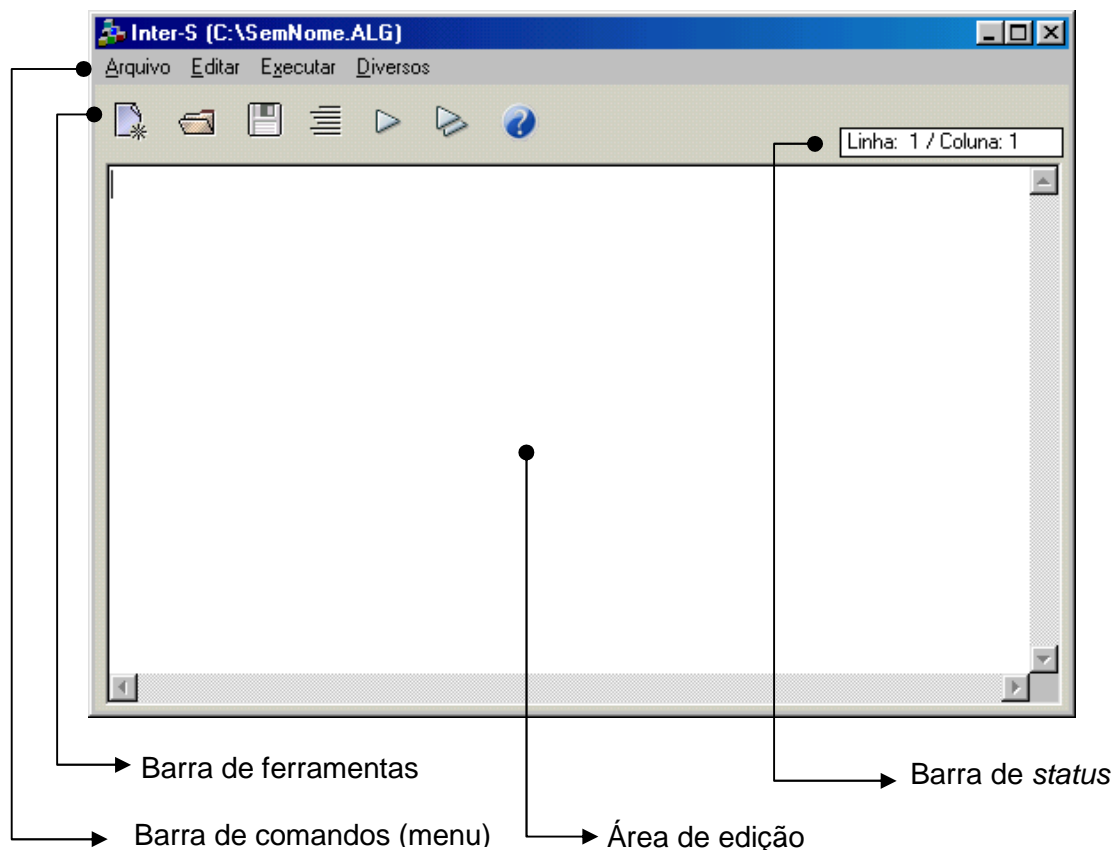
O INTER-S é um ambiente de desenvolvimento que permite a criação de programas de computador utilizando uma linguagem simples, em português, de fácil aprendizado e de fácil utilização.

Este ambiente oferece um editor de programas e ferramentas para a interpretação, execução e depuração do código, além de ferramentas auxiliares como estruturação do programa e ajuda "on-line".

A linguagem utilizada pelo INTER-S, também chamada de "Linguagem INTER-S", com todas as suas regras sintáticas e estruturais, fornece ao estudante todas as bases necessárias ao entendimento de outras linguagens de programação e, principalmente, fornece a visão lógica necessária ao desenvolvimento de programas de computador.

O ambiente de desenvolvimento

A figura abaixo retrata a tela de desenvolvimento do INTER-S:



Barra de comandos

Esta barra, como podemos observar na figura, contém os comandos **Arquivo**, **Editar**, **Executar** e **Diversos**. Para acionar um dos comandos basta clicar sobre o nome do mesmo ou utilizar a tecla ALT juntamente com a letra sublinhada do comando (observe que todos eles possuem uma letra sublinhada). O capítulo 2 deste guia oferece explicações mais detalhadas sobre os comandos.

Barra de ferramentas

Na barra de ferramentas você encontrará um botão para cada um dos comandos mais utilizados. Estes botões funcionam como um atalho, dispensando o acesso à barra de comandos. Veja maiores detalhes no capítulo 2 (Comandos).

Barra ou linha de *status*

Esta barra mostra o número da linha e o número da coluna onde se encontra o cursor na área de edição. Estas informações são muito úteis no momento da edição e depuração do programa.

Área de edição

É nesta área que o programa deverá ser digitado. Possui uma limitação de 1000 (mil) linhas, ou seja, o programa editado não poderá ter mais de mil linhas.

2. Comandos



Arquivo/Novo

Este comando cria um arquivo em 'branco' (novo arquivo), e entra no modo de edição. Pode ser acionado através da barra de comandos ou através da barra de ferramentas (botão mostrado acima).

O novo arquivo será chamado de 'SemNome.ALG' (nome padrão), que pode ser alterado em ocasião da gravação do mesmo.

Se este comando for acionado antes da gravação do programa que está sendo editado, o Inter-S permitirá a gravação do programa antes da abertura do novo arquivo.



Arquivo/Abrir

Permite a abertura de um arquivo para edição. Será exibida uma janela onde você deverá procurar o arquivo desejado.

O comando 'Abrir' pode ser acionado pela barra de comandos ou pela barra de ferramentas, através do botão mostrado acima.

Se este comando for acionado antes da gravação do programa que está sendo editado, o Inter-S permitirá a gravação do programa antes da abertura do arquivo desejado.



Arquivo/Salvar

Processa a gravação do arquivo que está sendo editado. Será exibida uma janela onde você digitará o nome desejado para o arquivo (na primeira gravação o Inter-S sugere o nome 'SemNome.ALG'), você deve indicar também a unidade e a pasta onde o arquivo será gravado.

Procure dar nomes sugestivos aos arquivos, isto facilitará a sua localização quando necessário.

O comando 'Salvar' pode ser acionado pela barra de comandos ou pela barra de ferramentas, através do botão mostrado acima.



Editar/Iniciar estruturação

Processa a estruturação do programa, ou seja, faz o deslocamento da margem esquerda (reco) das instruções que fazem parte de uma estrutura (instruções entre as palavras ROTINA e FIMROTINA e instruções que fazem parte de estruturas de decisão ou repetição).

A estruturação de um programa não é necessária para o bom funcionamento do mesmo, porém, é de grande utilidade para a documentação, dando maior clareza ao programa.

O recuo padrão é de 3 colunas à direita, mas este número pode ser alterado através do comando 'Editar/Alterar margem'.

O comando 'Editar/Iniciar estruturação' pode ser acionado pela barra de comandos ou pela barra de ferramentas, através do botão mostrado acima.

Editar/Alterar margem

Este comando permite a definição de uma nova margem para o processo de estruturação do programa.

A margem deve ser um número entre 0 (zero) e 6 (seis). Este comando não possui um botão na barra de ferramentas, portanto, para acioná-lo você deve acessar a barra de comandos.

▶ Executar/Executar

Este comando inicia a interpretação e execução do algoritmo. Surgirá uma tela em branco com um botão no canto esquerdo superior, clique sobre este botão para iniciar a execução do programa.

O processo de execução faz a verificação de cada uma das instruções do programa e, no caso de encontrar algum erro, será emitida uma mensagem indicando o número da linha errada e o tipo de erro encontrado. Neste caso você deverá corrigir a linha apontada e repetir o processo de execução.

O comando Executar/Executar pode ser acionado através do botão mostrado acima ou através da barra de comandos.

▶ Executar/Passo-a-passo

Este comando, assim como o comando Executar/Executar, faz a interpretação e execução do programa, porém, cada linha executada será exibida em uma janela inferior, permitindo que o programador visualize toda a seqüência de execução.

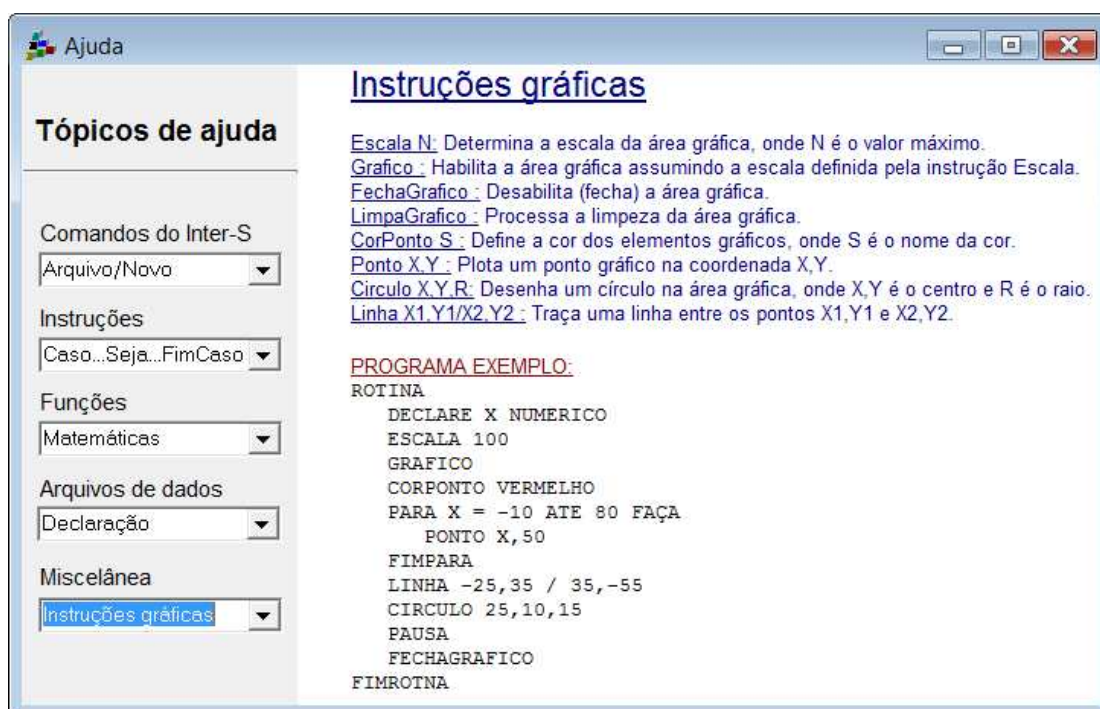
Este comando é muito útil para a depuração do algoritmo, onde o programador poderá analisar e descobrir eventuais erros.

O comando Executar/Passo-a-passo pode ser acionado através do botão mostrado acima ou através da barra de comandos.

🔍 Diversos/Ajuda (F1)

O Inter-S oferece um importante serviço de ajuda. Trata-se de um guia para consultas rápidas que proporciona informações sobre todos os comandos, instruções e demais recursos do ambiente, com vários programas exemplos. Através dos tópicos de ajuda, compostos por caixas de combinação, o usuário poderá navegar por todo o catálogo de ajuda.

O comando Diversos/Ajuda pode ser acionado através da barra de comandos ou através da tecla F1. Veja abaixo o exemplo de um dos tópicos de ajuda:



3. Instruções

Instruções são formações de palavras (ou frases) que, obedecendo às regras sintáticas da linguagem, determinam as ações dentro de um programa de computador.

Neste capítulo você encontra uma relação com todas as instruções suportadas pelo INTER-S, classificadas por categoria, com explicações e exemplos de uso.

Algumas coisas que você deve saber para compreender os comentários e exemplos deste capítulo:

- Em alguns comentários você encontrará os termos *algoritmo* e *pseudocódigo*, que, dentro do nosso contexto, correspondem a *programa em linguagem INTER-S*;
- Todo programa escrito em INTER-S deve ser iniciado com a instrução **ROTINA** e finalizado com a instrução **FIMROTINA**;
- Declarações de variáveis devem ser feitas logo após a instrução **ROTINA** e antes de qualquer outra instrução.

3.1. Instruções de iniciação e declaração

Rotina...FimRotina

Todo o algoritmo Inter-S deve ser iniciado com a instrução ROTINA e finalizado com a instrução FIMROTINA. A ausência de uma destas instruções fará com que o programa não funcione.

Programa exemplo:

```
ROTINA
  DECLARE NOME CARACTERE
  RECEBA "DIGITE SEU NOME: ",NOME
  POSICIONE 23,1
  ESCREVA "SEU NOME É: ",NOME
FIMROTINA
```

Note que as instruções entre ROTINA e FIMROTINA estão dispostas em uma margem mais à direita. Isto não é necessário para que o programa funcione adequadamente, porém, trata-se de uma boa prática para garantir a legibilidade da rotina.

A estruturação do programa pode ser feita automaticamente com o uso do comando "**Editar/Iniciar estruturação**".

Declare

Declara as variáveis que serão utilizadas no programa.

Sintaxe:

```
DECLARE Nome-da-Variável Tipo
```

Exemplo:

```
DECLARE VALOR NUMERICO
```

Notas:

Em uma mesma instrução de declaração podemos também declarar uma lista de variáveis, desde que todas elas sejam do mesmo tipo (todas numéricas ou todas caracteres). Veja exemplos:

```
DECLARE VALOR, TOTAL, CUSTO NUMERICAS
DECLARE NOME, ENDERECO CARACTERES
```

Na primeira instrução estamos declarando as variáveis VALOR, TOTAL e CUSTO como numéricas, ou seja, dentro do programa elas só poderão receber e tratar dados numéricos. Na segunda instrução declaramos NOME e ENDEREÇO como variáveis do tipo caractere: receberão textos.

Declare Arranjo

Permite a declaração de arranjos de variáveis. Um arranjo corresponde a uma matriz de dados (*array*) onde podemos armazenar diversos dados, simultaneamente, em uma única variável, e acessar estes dados através de um índice.

Sintaxe:

```
DECLARE ARRANJO Variável[qtde] Tipo
```

Exemplo:

```
DECLARE ARRANJO R[4] CARACTERE
```

Notas:

Devemos informar a quantidade máxima de elementos que o arranjo poderá utilizar. Esta informação deve ser colocada entre colchetes, como demonstra o exemplo acima. Neste exemplo estamos declarando o arranjo R com 4 elementos.

Após esta declaração é possível usar cada um dos elementos do arranjo como se fossem variáveis comuns, simplesmente indicando entre os colchetes o número do elemento (índice) desejado. Exemplos:

```
RECEBA R[1]
R[2]=R[1]*4
A=3
R[4]=R[A]
```

Observações:

- Não é possível declarar mais que um arranjo em uma única instrução DECLARE ARRANJO.
- A quantidade máxima de elementos por arranjo é 100.

3.2. Instruções de entrada e saída

Receba

Recebe dados pelo teclado e os armazena em variáveis.

Sintaxe:

```
RECEBA [mensagem,] variável
```

Exemplo:

```
RECEBA "DIGITE SEU NOME:",NOME
```

A instrução RECEBA permite que se coloque, opcionalmente, uma mensagem entre aspas que será exibida antes do recebimento do dado, conforme mostra o exemplo acima.

Escreva

Escreve informações na tela (conteúdos de variáveis ou constantes).

Sintaxe:

```
ESCREVA variável / constante
```

Exemplo:

```
ESCREVA "Olá Mundo!"
```

Notas:

Podemos, em uma mesma instrução de escrita, enviar mais de uma informação para a tela, para isso devemos usar a vírgula como separador dos dados enviados. Exemplos:

```
ESCREVA "VALOR:",VL  
ESCREVA VALOR,TOTAL,CUSTO
```

Na primeira instrução estamos escrevendo na tela a expressão 'VALOR' e em seguida o conteúdo da variável VL. Na segunda instrução estamos escrevendo apenas os conteúdos das variáveis VALOR, TOTAL e CUSTO.

...Escreva

Esta instrução é semelhante à instrução ESCREVA, a diferença é que as informações enviadas para a tela serão escritas na mesma linha da última instrução de escrita, ou seja, não haverá salto de linha.

Sintaxe:

```
...ESCREVA variável / constante
```

Exemplo:

```
...ESCREVA "Mesma linha!"
```

3.3. Instruções para manipulação de tela

Limpa

Limpa (apaga) a tela.

Sintaxe:

```
LIMPA
```

Exemplo:

```
LIMPA
```

Programa exemplo:

```
ROTINA  
  DECLARE NOME CARACTERE  
  RECEBA "DIGITE SEU NOME:",NOME
```

```
LIMPA
ESCREVA "SEU NOME É: ",NOME
FIMROTINA
```

Posicione

Posiciona o cursor em uma determinada coordenada da tela.

Sintaxe:

```
POSICIONE linha, coluna
```

Exemplo:

```
POSICIONE 10,15
```

Programa exemplo:

```
ROTINA
  DECLARE NOME CARACTERE
  RECEBA "DIGITE SEU NOME: ",NOME
  POSICIONE 23,1
  ESCREVA "SEU NOME É: ",NOME
FIMROTINA
```

As coordenadas (linhas e colunas) podem ser representadas por variáveis declaradas como numéricas.

CorFundo

Altera a cor de fundo da tela.

Sintaxe:

```
CORFUNDO nome-da-cor
```

Exemplo:

```
CORFUNDO azul
```

Notas:

A instrução 'Corfundo' reconhece 16 cores básicas, conforme tabela abaixo:

PRETO	VERMELHO	MAGENTA	MAGENTA+
CINZA	VERDE	MARROM	VERDE+
BRANCO	AMARELO	AZUL+	CELESTE+
AZUL	CELESTE	VERMELHO+	BRANCO+

O sinal de mais (+), colocado à direita do nome da cor, faz com que seja acionada a cor mais brilhante.

CorTexto

Define uma nova cor para as letras (texto), que será usada nas próximas instruções de escrita, porém não altera a cor do conteúdo já existente na tela.

Sintaxe:

```
CORTEXTO nome-da-cor
```

Exemplo:

CORTEXTO branco+

Notas:

A instrução 'Cortexto' reconhece as mesmas 16 cores básicas mostradas na tabela da instrução 'CorFundo'.

Moldura

Desenha uma moldura nas coordenadas indicadas.

Sintaxe:

```
MOLDURA L1,C1,L2,C2,cor
```

Exemplo:

```
MOLDURA 2,1,10,20,AZUL
```

Observação:

A instrução MOLDURA desenha um retângulo a partir da linha e coluna indicadas por 'L1' e 'C1' (canto esquerdo superior) até a linha 'L2' e coluna 'C2' (canto direito inferior), utilizando a cor de preenchimento determinada por 'cor'.

(Vide tabela de cores na descrição da instrução "CorFundo")

MolduraDupla

Desenha uma moldura dupla nas coordenadas indicadas.

Sintaxe:

```
MOLDURADUPLA L1,C1,L2,C2,cor
```

Exemplo:

```
MOLDURADUPLA 2,1,10,20,VERDE+
```

Observação:

A instrução MOLDURADUPLA é semelhante à instrução MOLDURA, a diferença é que o retângulo será desenhado com linhas duplas.

3.4. Controle de fluxo e estruturas de decisão e repetição

VaPara

Processa um desvio incondicional para um determinado ponto do programa.

Sintaxe:

```
VAPARA rótulo
```

Programa exemplo:

```
ROTINA
  DECLARE CONTA NUMERICO
  CONTA=0
  [ INICIO ]
  CONTA=CONTA+1
  ESCREVA CONTA
```

```
SE CONTA<100 ENTAO
  VAPARA INICIO
FIMSE
FIMROTINA
```

O rótulo é um ponto no programa que devemos indicar através de um nome entre colchetes ([]). Note que o rótulo utilizado neste exemplo é [INICIO].

Se...FimSe

Esta estrutura de decisão permite que uma expressão seja avaliada e, se for verdadeira, executa o bloco de instruções que se encontra entre a palavras ENTAO e FIMSE.

Programa exemplo:

```
ROTINA
  DECLARE VL NUMERICO
  RECEBA "DIGITE O VALOR:",VL
  SE VL<100 ENTAO
    ESCREVA "O VALOR É MENOR QUE 100"
  FIMSE
FIMROTINA
```

Neste exemplo, instruímos para que seja recebido um valor pelo teclado, o qual será armazenado na variável VL. Em seguida estamos verificando se o valor digitado é menor que 100: (SE VL<100 ENTAO), caso positivo será executada a instrução que está entre as palavras ENTAO e FIMSE, caso contrário o processamento pulará para a instrução após o FIMSE, que neste caso é FIMROTINA.

Se...Senao...FimSe

Este conjunto de instruções é chamado de 'Estrutura de decisão composta'. Permite que uma condição seja avaliada e, de acordo com o resultado da avaliação, executa um ou outro bloco de instruções.

Programa exemplo:

```
ROTINA
  DECLARE VL NUMERICO
  RECEBA "DIGITE O VALOR:",VL
  SE VL<100 ENTAO
    ESCREVA "O VALOR É MENOR QUE 100"
  SENAO
    ESCREVA "O VALOR NÃO É MENOR QUE 100"
  FIMSE
FIMROTINA
```

Neste exemplo, instruímos para que seja recebido um valor pelo teclado, o qual será armazenado na variável VL. Em seguida estamos verificando se o valor digitado é menor que 100: (SE VL<100 ENTAO), caso positivo será executada a instrução que vem após o ENTAO, caso contrário será executada a instrução que vem após o SENAO. O FIMSE finaliza a estrutura.

Caso...Seja...FimCaso

Este conjunto de instruções também é uma 'estrutura de decisão'. A instrução 'CASO' requer uma expressão que será avaliada; as instruções 'SEJA' contêm os comandos que deverão ser executados para cada um dos casos.

Programa exemplo:

```

ROTINA
  DECLARE V NUMERICO
  RECEBA "DIGITE O VALOR:",V
  CASO V
    SEJA < 0 FACa ESCREVA "O VALOR É NEGATIVO"
    SEJA = 1 FACa ESCREVA "O VALOR É UM"
    SEJA = 2 FACa ESCREVA "O VALOR É DOIS"
    CASONAO FACa ESCREVA "O VALOR É MAIOR QUE DOIS"
  FIMCASO
FIMROTINA

```

Neste exemplo, instruímos para que seja recebido um valor pelo teclado, o qual será armazenado na variável V. Em seguida estamos avaliando o valor de V através de uma estrutura 'CASO' (CASO V). As instruções 'SEJA' farão a execução do comando caso a condição seja verdadeira. Será executado o comando da instrução 'CASONAO' se nenhuma das condições anteriores forem verdadeiras.

Esta estrutura não permite aninhamentos.

Enquanto...FimEnquanto

Esta é uma estrutura de repetição que processa o bloco de instruções existente entre as palavras ENQUANTO e FIMENQUANTO, repetidamente, enquanto uma determinada condição for verdadeira.

Programa exemplo:

```

ROTINA
  DECLARE N NUMERICO
  RECEBA "DIGITE UM N°:",N
  ENQUANTO N>0 FACa
    ESCREVA N
    N=N-1
  FIMENQUANTO
FIMROTINA

```

Neste exemplo, instruímos para que seja recebido um valor pelo teclado, o qual será armazenado na variável N. Em seguida será processada a estrutura de repetição enquanto o valor de N for maior que 0, ou seja, enquanto a condição (N>0) for verdadeira. Se o valor digitado inicialmente não atender à condição, a estrutura não será executada nenhuma vez.

Para...FimPara

Esta é uma estrutura de repetição que processa o bloco de instruções existente entre as instruções PARA e FIMPARA um número determinado de vezes, controlado por uma variável chamada de 'Contador'.

Programa exemplo:

```

ROTINA

```

```

DECLARE I NUMERICO
PARA I=1 ATE 10 FACA
  ESCREVA I
FIMPARA
FIMROTINA

```

Neste exemplo, a estrutura de repetição determina que a variável I é o contador de repetições, que será iniciado com o valor 1 (PARA I=1), sendo que seu conteúdo deverá chegar ao valor 10 (ATE 10), momento em que a estrutura será finalizada. A cada repetição o contador I será incrementado em 1.

Para alterar o valor do incremento, podemos usar a instrução PASSO antes do início da estrutura de repetição PARA. Exemplos:

```

ROTINA
  DECLARE I NUMERICO
  PASSO 2
  PARA I=1 ATE 10 FACA
    ESCREVA I
  FIMPARA
FIMROTINA

```

No exemplo acima, o contador I será incrementado em 2 a cada ciclo, até atingir o valor 10. Portanto, esta rotina escreverá na tela os valores 1, 3, 5, 7 e 9, pois, o contador é iniciado em 1 e a cada ciclo é acrescido de 2.

```

ROTINA
  DECLARE I NUMERICO
  PASSO -1
  PARA I=10 ATE 1 FACA
    ESCREVA I
  FIMPARA
FIMROTINA

```

No exemplo acima, o contador I será incrementado em -1 a cada ciclo, até atingir o valor 1. Note que o incremento negativo (-1) é na realidade um decremento. Portanto, esta rotina escreverá na tela os valores 10, 9, 8, 7, 6, 5, 4, 3, 2 e 1, pois, o contador é iniciado em 10 e a cada será diminuído em 1.

Nota: o valor do incremento além de poder ser positivo ou negativo, pode também ser um número fracionário.

Repita...AteQue

Esta é uma estrutura de repetição que processa o bloco de instruções existente entre as palavras REPITA e ATEQUE, repetidamente, até que uma determinada condição seja verdadeira.

Programa exemplo:

```

ROTINA
  DECLARE N NUMERICO
  RECEBA "DIGITE UM N°: ",N
  REPITA
    ESCREVA N
    N=N-1
  ATEQUE N<=0
FIMROTINA

```

Neste exemplo, instruímos para que seja recebido um valor pelo teclado, o qual será armazenado na variável N. Em seguida será processada a estrutura de repetição até que o valor de N seja menor ou igual a 0, ou seja, até que a condição ($N \leq 0$) seja verdadeira. Observe que esta estrutura processará o bloco de instruções pelo menos uma vez.

3.5. Instruções para chamadas externas

Chame

Esta instrução permite a chamada e execução de um algoritmo externo, transferindo o processamento para o mesmo.

As variáveis não são transferidas para o algoritmo chamado, ou seja, a memória é liberada para o novo processamento.

Programa exemplo:

```

ROTINA
  DECLARE OP NUMERICO
  RECEBA "DIGITE A OPCAO:",OP
  CASO OP
    SEJA = 1 FACA CHAME C:\PROG1.ALG
    SEJA = 2 FACA CHAME C:\PROG2.ALG
  FIMCASO
FIMROTINA

```

Neste exemplo, instruímos para que seja recebido um valor pelo teclado, o qual será armazenado na variável OP. Em seguida estamos avaliando o valor de OP através de uma estrutura 'CASO' (CASO OP). Se OP for igual a 1, será chamado o algoritmo PROG1.ALG; se OP for igual a 2, será chamado o algoritmo PROG2.ALG.

SubRotina...FimSubRotina

Uma sub-rotina é um procedimento escrito após o final do programa (após o 'FimRotina') que pode ser executado através de uma ou mais chamadas dentro da rotina principal. Toda sub-rotina deve ter um identificador (nome) o qual será utilizado pela instrução 'Processe' em ocasião das chamadas.

Programa exemplo:

```

ROTINA
  DECLARE V, R NUMERICOS
  RECEBA "DIGITE O VALOR:",V
  PROCESSE CALCULO
  ESCREVA "RESULTADO:",R
FIMROTINA
SUBROTINA CALCULO
  SE V < 1000 ENTAO
    R = V * 1.15
  SENAO
    R = V * 1.10
  FIMSE
FIMSUBROTINA

```

Veja a instrução 'Processe'.

Processe

A instrução 'Processe' faz a execução de uma sub-rotina escrita após a rotina principal (veja instrução 'SubRotina').

Após a execução da sub-rotina, o processamento retorna para a linha imediatamente após à linha de sua chamada, ou seja, retorna para a instrução que vem após o 'Processe'.

Programa exemplo:

```
ROTINA
  DECLARE V NUMERICO
  PROCESSE TELA
  RECEBA "DIGITE O VALOR:",V
  V = V ^ 2
  ESCREVA "QUADRADO:",V
  PAUSA
  PROCESSE TELA
FIMROTINA
SUBROTINA TELA
  LIMPA
  CORTEXTO BRANCO+
  MOLDURA 1,1,25,80,AZUL
FIMSUBROTINA
```

Uma sub-rotina pode ser chamada várias vezes, como mostra o exemplo acima.

3.6. Instruções diversas

Ftecla

Esta instrução faz a leitura do teclado e coloca em uma variável o código da tecla pressionada. Se nenhuma tecla estiver pressionada naquele momento, será colocado valor 0 (zero) na variável. A variável receptora deve ser declarada como numérica (através da instrução DECLARE).

Sintaxe:

```
variável = FTECLA
```

Exemplo:

```
TEC=FTECLA
```

Interrompa

Interrompe (finaliza) a execução do programa.

Sintaxe:

```
INTERROMPA
```

Exemplo:

```
INTERROMPA
```

Programa exemplo:

```
ROTINA
  DECLARE TEC NUMERICO
```

```

[ INICIO ]
TEC=FTECLA
SE TEC=27 ENTÃO
    INTERROMPA
FIMSE
ESCREVA "CÓDIGO DA TECLA:", TEC
VAPARA INICIO
FIMROTINA

```

Pausa

Gera uma pausa até que uma tecla seja pressionada.

Sintaxe:

```
PAUSA
```

Exemplo:

```
PAUSA
```

Programa exemplo:

```

ROTINA
  DECLARE NOME CARACTERE
  RECEBA "DIGITE SEU NOME:", NOME
  ESCREVA "TECLE ALGO PARA CONTINUAR."
  PAUSA
  LIMPA
  ESCREVA "SEU NOME É:", NOME
FIMROTINA

```

// (comentários)

Para incluir comentários (linhas que não serão interpretadas) em seu programa, você deve iniciar a linha com duas barras (//), desta forma você poderá escrever qualquer informação em qualquer lugar do programa.

Programa exemplo:

```

// Programa PAR
// Autor: Zé Mané
ROTINA
  DECLARE A, RESTO NUMERICO
  RECEBA "DIGITE UM NÚMERO:", A
  RESTO=A%2
  SE RESTO=0 ENTÃO
    ESCREVA "O NÚMERO É PAR"
  SENÃO
    ESCREVA "O NÚMERO É ÍMPAR"
  FIMSE
FIMROTINA
// Fim do programa

```

4. Funções

4.1. Funções matemáticas

São rotinas internas que executam uma determinada tarefa (função) e devolvem o resultado desta tarefa. A tabela abaixo mostra as funções matemáticas existentes no Inter-S:

Trunca(N).....	Retorna a parte inteira de um número fracionário (N).
Atn(A).....	Retorna o arco tangente do ângulo informado em radianos (A).
Tan(A).....	Retorna a tangente do ângulo informado em radianos (A).
Log(N).....	Retorna o logaritmo natural (neperiano - base e) de N.
Exp(N).....	Retorna e (a base de logaritmos naturais) à potência de N
Cos(A).....	Retorna o cosseno do ângulo informado em radianos (A).
Seno(A).....	Retorna o seno do ângulo informado em radianos (A).
Arredonda(N)	Retorna o próximo inteiro do valor fracionário N.
Arred1(N)	Retorna o valor N arredondado para 1 casa decimal.
Arred2(N)	Retorna o valor N arredondado para 2 casas decimais.
Arred3(N)	Retorna o valor N arredondado para 3 casas decimais.
Arred4(N)	Retorna o valor N arredondado para 4 casas decimais.
Raiz(N).....	Retorna a raiz quadrada do número informado (N).

Exemplos:

```
A=RAIZ ( B^2+C^2 )
X=ARREDONDA ( A/B )
Y=SENO ( 45 )
IF TRUNCA ( W ) = 3 ENTAO . . .
```

4.2. Funções Diversas

O Inter-S oferece algumas funções úteis para o tratamento de dados caracteres e numéricos. Estas funções estão relacionadas abaixo:

Valor(C)	Retorna o valor numérico da variável caractere (C).
Texto(N).....	Retorna o valor caractere da variável numérica (N).
Parte(C,i,t)	Retorna a parte da variável caractere (C) que se inicia na posição indicada (i), com tamanho igual a t (t). NOTA: i e t podem ser variáveis.
Tamanho(C).....	Retorna o tamanho da variável caractere (C).
Hoje()	Retorna a data atual no formato dd/mm/aaaa (a variável receptora deve ser caractere).
Randomico(N).....	Retorna um número entre 1 e N, onde N pode ser uma variável

Exemplos:

```
A=VALOR ( C )
C=TEXTO ( A )
DATA=HOJE ( )
DIA=PARTE ( DATA , 1 , 2 )
```

Estas funções só podem ser utilizadas em instruções de atribuição simples. Utilizá-las com as instruções 'Escreva', 'Se', 'Enquanto', 'Caso', 'Seja', ou dentro de uma expressão numérica ou caractere, gerará um erro. Utilize-as apenas em atribuições simples como demonstram os exemplos acima.

5. Operadores

5.1. Operadores lógicos

São palavras utilizadas para a construção de expressões condicionais compostas, ou seja, são operadores booleanos que nos permitem avaliar duas expressões simultaneamente. O Inter-S oferece 2 operadores lógicos: E e OU.

EXEMPLOS:

SE A<10 E B>5 ENTAO...

Neste exemplo a condição será verdadeira somente se as duas expressões forem verdadeiras (A<10 e B>5).

SE C<1 OU C>10 ENTAO...

Neste exemplo a condição será verdadeira se apenasuma das expressões for verdadeira, ou se ambas forem verdadeiras.

5.2. Operadores matemáticos

São símbolos que representam as operações em uma expressão matemática. A tabela abaixo os apresenta na ordem de precedência (prioridade em que são efetuadas as operações):

Símbolo	Operação
MOD	Resto da divisão (podemos usar MOD ou %)
DIV	Inteiro da divisão (podemos usar DIV ou \)
^	Exponenciação
*	Multiplicação
/	Divisão
+	Adição
-	Subtração

Podemos utilizar parênteses para alterar a prioridade de uma determinada operação, exemplo:

$R = (A * (35 - B)) \text{ MOD } 2$

5.3. Operadores relacionais

São símbolos que representam as operações de comparação em expressões condicionais. A tabela abaixo apresenta os operadores relacionais:

Símbolo	Operação
=	Igual a
#	Diferente de
>	Maior que
<	Menor que
>=	Maior ou igual a
<=	Menor ou igual a

5.4. Operador de caractere

Para dados do tipo caractere (textos) temos apenas um operador representado pelo símbolo & ('E' comercial).

Este operador permite a concatenação, ou seja, a união de dados tipo caractere.

EXEMPLO:

```
A= "OLÁ"  
B= " PESSOAL"  
C= A&B&"!"
```

Neste exemplo a variável caractere 'C' terá como conteúdo final a expressão: 'OLÁ PESSOAL!'

6. Arquivos de dados

6.1. Declaração de arquivo

Antes de efetuarmos qualquer operação com um arquivo de dados devemos declará-lo utilizando a instrução 'Declare arquivo'.

Através desta instrução informamos o caminho e o nome, o campo chave e a estrutura do registro (nome e tipo dos campos).

SINTAXE:

```
DECLARE ARQUIVO(nome-do-arquivo) CHAVE(campo-chave)
REGISTRO(nome-do-campo1 tipo-do-campo1 [tamanho]
          nome-do-campo2 tipo-do-campo2 [tamanho]
          ...
          nome-do-campoN tipo-do-campoN [tamanho])
```

NOTAS

- O nome do arquivo pode conter o caminho, exemplo: C:\Dados\Arq;
- Não são permitidos espaços antes dos parênteses;
- A instrução REGISTRO deve aparecer imediatamente após a linha da instrução de declaração do arquivo.

Exemplo de declaração:

```
DECLARE ARQUIVO(C:\INTER-S\CLIENTES) CHAVE(CODIGO)
REGISTRO(CODIGO NUMERICO [4]
          NOME CARACTERE [35]
          ENDERECO CARACTERE [40])
```

Obs.: O Inter-S permite a declaração de até 3 (três) arquivos.

6.2. Abertura de arquivos de dados

Para abrirmos um arquivo de dados devemos antes declará-lo através da instrução 'Declare arquivo'. A instrução de abertura faz com que o arquivo se torne acessível para as operações de leitura, gravação e exclusão de registros.

SINTAXE:

```
ABRA nome-do-arquivo
```

NOTAS

O nome do arquivo deve ser o mesmo utilizado na instrução de declaração. Se foi informado um caminho, este deve ser agregado ao nome. Exemplo de declaração e abertura de um arquivo:

```
DECLARE ARQUIVO(C:\INTER-S\CLIENTES) CHAVE(CODIGO)
REGISTRO(CODIGO NUMERICO [4]
          NOME CARACTERE [35]
          ENDERECO CARACTERE [40])
...
ABRA C:\INTER-S\CLIENTES
```

6.3. Leitura de registros

Temos 2 formas para a leitura de um arquivo de dados: Direta e Seqüencial. A primeira forma (direta) exige que se informe um valor que será o item de procura (campo chave), se este item não for encontrado o Inter-S retornará um registro em branco e um controle FDA positivo (vide tópico 'Fim de Arquivo').

A segunda forma (seqüencial) lê sempre o próximo registro do arquivo, não necessitando de um valor para a procura. Quando for atingido o final do arquivo, o controle FDA será positivo.

SINTAXE (Forma Direta):

```
LEIA ITEM[nome-variável] nome-do-arquivo
```

SINTAXE (Forma Seqüencial):

```
LEIA PROXIMO REGISTRO nome-do-arquivo
```

NOTAS

O nome do arquivo deve ser o mesmo utilizado na instrução de declaração. Se foi informado um caminho, este deve ser agregado ao nome. Na forma direta, o valor do item de procura deve ser informado por intermédio de uma variável. Exemplos:

```
LEIA ITEM[CODIGO] C:\INTER-S\CLIENTES  
LEIA PROXIMO REGISTRO C:\INTER-S\CLIENTES
```

6.4. Gravação de registros

Para gravarmos um registro devemos antes preparar os campos do mesmo, atribuindo-lhes os valores desejados, e em seguida devemos utilizar a instrução 'GRAVE', informando o valor do item correspondente ao campo chave.

SINTAXE:

```
GRAVE ITEM[nome-variável] nome-do-arquivo
```

NOTAS

O nome do arquivo deve ser o mesmo utilizado na instrução de declaração. Se foi informado um caminho, este deve ser agregado ao nome. O valor do item de procura deve ser informado por intermédio de uma variável, que pode ser a mesma que representa o campo chave do arquivo. Caso a chave informada não exista no arquivo, o Inter-S criará um novo registro, caso contrário o registro encontrado será atualizado. Exemplo de gravação:

```
GRAVE ITEM[CODIGO] C:\INTER-S\CLIENTES
```

6.5. Exclusão de registros

Para excluirmos, ou seja, apagarmos um determinado registro do arquivo de dados, devemos utilizar a instrução 'APAGUE', informando o valor do item correspondente ao campo chave, cujo registro desejamos apagar.

SINTAXE:

```
APAGUE ITEM[nome-variável] nome-do-arquivo
```

NOTAS

O nome do arquivo deve ser o mesmo utilizado na instrução de declaração. Se foi informado um caminho, este deve ser agregado ao nome. O valor do item de procura deve ser informado por intermédio de uma variável, que pode ser a mesma que representa o campo chave do arquivo. Exemplo:

```
APAGUE ITEM[CODIGO] C:\INTER-S\CLIENTES
```

6.6. Fechamento de arquivos

Após a utilização de um arquivo de dados devemos processar o fechamento do mesmo. A instrução 'FECHE' exige que se informe o nome do arquivo que será fechado, tornando-o inacessível. Esta instrução, normalmente, deve ser dada no final do programa.

SINTAXE:

```
FECHE nome-do-arquivo
```

NOTAS

O nome do arquivo deve ser o mesmo utilizado na instrução de declaração. Se foi informado um caminho, este deve ser agregado ao nome. Exemplo:

```
FECHE C:\INTER-S\CLIENTES
```

6.7. Topo do arquivo

É necessário, às vezes, reposicionar o ponteiro de leitura de um arquivo de dados no primeiro registro. Isto é comum em programas que utilizam o acesso seqüencial, onde a leitura do arquivo deve ser iniciada no primeiro registro. Neste caso devemos reposicionar o ponteiro antes da leitura do arquivo, utilizando a instrução 'TOPO'.

SINTAXE:

```
TOPO nome-do-arquivo
```

NOTAS

O nome do arquivo deve ser o mesmo utilizado na instrução de declaração. Se foi informado um caminho, este deve ser agregado ao nome. Exemplo:

```
TOPO C:\INTER-S\CLIENTES
```

6.8. Fim de arquivo (FDA)

Existe uma variável interna de controle denominada FDA (Fim de Arquivo). Esta variável informa se o ponteiro de leitura do último arquivo acessado atingiu ou não o final. O FDA não pode ser manipulado pelo programador, pois é um controle interno do Inter-S, o programador pode apenas ler o seu conteúdo.

Este controle pode ser utilizado apenas nas instruções 'SE' e 'ENQUANTO'.

EXEMPLOS:

SE FDA ENTÃO...
SE NÃO FDA ENTÃO...
ENQUANTO NÃO FDA FAÇA..

Observe que podemos utilizar a forma negativa do FDA, colocando a palavra 'NÃO' à esquerda: 'SE NÃO FDA ENTÃO', isto é o mesmo que avaliar o 'não' fim de arquivo. A instrução 'ENQUANTO' admite apenas o uso da forma negativa.

7. Instruções para impressão

As instruções abaixo permitem a criação de relatórios impressos, utilizando a impressora padrão de seu sistema.

<u>INSTRUÇÃO</u>	<u>DESCRIÇÃO</u>
IMPRIMA	Envia para a impressora uma expressão ou lista de expressões
PULAPAGINA	Finaliza a impressão de uma página e avança para a próxima
IMPLETRA	Define o tamanho da letra
IMPCOR	Define a cor da letra
IMPFIM	Finaliza a impressão e libera a página

PROGRAMA EXEMPLO:

```

ROTINA
  DECLARE PAG NUMERICO
  PAG=1
  IMPRIMA "RELATORIO DE CLIENTES      PAGINA: " ,PAG
  PULAPAGINA
  PAG=2
  IMPLETRA 14
  IMPCOR VERMELHO
  IMPRIMA "RELATORIO DE CLIENTES      PAGINA: " ,PAG
  IMPFIM
FIMROTINA

```

8. Instruções gráficas

O Inter-S possui um conjunto de instruções gráficas que dão suporte ao sistema cartesiano para a construção de aplicações matemáticas.

<u>INSTRUÇÃO</u>	<u>DESCRIÇÃO</u>
------------------	------------------

Escala N	Determina a escala da área gráfica, onde N é o valor máximo.
Gráfico	Habilita a área gráfica assumindo a escala definida pela instrução Escala.
FechaGrafico	Desabilita (fecha) a área gráfica.
LimpaGrafico.....	Processa a limpeza da área gráfica.
CorPonto S	Define a cor dos elementos gráficos, onde S é o nome da cor.
TamanhoPonto N.....	Define o tamanho do ponto, onde N é o tamanho (de 1 a 4).
Ponto X,Y.....	Plota um ponto gráfico na coordenada X,Y.
Circulo X,Y,R	Desenha um círculo na área gráfica, onde X,Y é o centro e R é o raio.
Elipse X,Y,rx,ry	Desenha uma elipse, onde X,Y é o centro, rx é o raio x e ry é o raio y.
Linha X1,Y1/X2,Y2....	Traça uma linha entre os pontos X1,Y1 e X2,Y2.

PROGRAMA EXEMPLO:

```

ROTINA
  DECLARE X NUMERICO
  ESCALA 100
  GRAFICO
  CORPONTO VERMELHO

```

```
PARA X = -10 ATE 80 FAÇA
  PONTO X,50
FIMPARA
LINHA -25,35 / 35,-55
CIRCULO 25,10,15
PAUSA
FECHAGRAFICO
FIMROTNA
```

9. Variáveis

Variáveis são áreas de memória onde armazenamos valores que podem ser alterados durante a execução do programa.

Para utilizar uma variável devemos primeiramente dar um nome à ela e informar qual o tipo de dado que ela receberá, isto é feito através da instrução DECLARE.

Uma variável pode ser numérica (que armazena números) ou caractere (que armazena textos).

Nomes de variáveis devem obedecer as seguintes regras:

- Podem ser formados por letras e números
- Não podem começar com números
- Não podem conter espaços
- Não podem conter letras acentuadas ou cedilha
- Não podem ser palavras reservadas (instruções)

EXEMPLOS DE NOMES VÁLIDOS:

VALOR , A1 , N10 , SALARIO , W

EXEMPLOS DE NOMES INVÁLIDOS:

9VALOR , RECEBA , W K , SALÁRIO

Roberto Vichinsky
Março/2011