

Exemplos Práticos

Exemplo 1

```
define data local
1 empregos
    2 nome (a20)
    2 idade (i2)
    2 cargo (a20)
    2 salario (n7.2)
end-define

/* Atribuicoes. move ou :=
move 'Joca' to nome
idade := 33
cargo := 'Digitador'
salario := 2000

/* Saída de dados
write 'Nome: ' nome
write 'Idade: ' idade
write 'Cargo: ' cargo
write 'Salario: ' salario

END
```

Natural Web I/O Output

Page	1	23-05-11 15:25:11
Nome: Joca		
Idade:	33	
Cargo:	Digitador	
Salario:	2000.00	

Exemplo 2

```
+ * >Natural Source Header 000000[...]  
- define data local  
- 1 empregos  
    2 nome (a20)  
    2 idade (i2)  
    2 salario (n7.2) /* ex: 1000000.22  
    2 novosalario (n7.2)  
    2 salariodata (d) init <*datx> /* <*datx> obtem a data atual  
    2 salariodataAlpha (a10)  
end-define  
move 'Joca' to nome  
idade := 33  
salario := 1000  
  
- /* compute => executa operações aritmeticas e  
  /* permite a utilização de funções do Natural  
  /* rounded => permite que o resultado seja arredondado  
compute rounded novosalario := salario * 1.15000086  
  
add 7 to salariodata /* add => efetua a operação de soma  
  
- /* move edited => converte a variavel para outros tipos  
  /* o move edited permite usar uma máscara de edição p/ o formato requerido  
move edited salariodata (em=YYYY-MM-DD) to salariodataAlpha  
  
write 'Nome: ' nome  
write 'Idade: ' idade 'Salario: ' salario  
write 'Novo Salario: ' novosalario 'a partir de ' salariodataAlpha  
END
```

Natural Web I/O Output

```
Page      1                                     23-05-11 15:55:01  
  
Nome: Joca  
Idade:      33 Salario:      1000.00  
Novo Salario:      1150.00 a partir de 2023-05-18
```

Exemplo 3

```
* >Natural Source Header 000000...
define data local
1 empregos
  2 nome (a20)
  2 idade (i2)
  2 salario (n7.2) /* ex: 1000000.22
  2 novosalario (n7.2)
  2 salariodata (d) init <*datx> /* <*datx> obtem a data atual
  2 salariodataAlpha (a10)
end-define

input nome idade salario /* input => entrada de dados

/* compute => executa operações aritmeticas e
/* permite a utilização de funções do Natural
/* rounded => permite que o resultado seja arredondado
compute rounded novosalario := salario * 1.15000086

add 7 to salariodata /* add => efetua a operação de soma

/* move edited => converte a variavel para outros tipos
/* o move edited permite usar uma máscara de edição p/ o formato requerido
move edited salariodata (em=YYYY-MM-DD) to salariodataAlpha

write 'Nome: ' nome
write 'Idade: ' idade 'Salario: ' salario
write 'Novo Salario: ' novosalario 'a partir de ' salariodataAlpha
END
```

Natural Web I/O Output

NOME IDADE SALARIO

Natural Web I/O Output

Page 1 23-05-11 16:01:59

Nome: Joca
Idade: 33 Salario: 1000.00
Novo Salario: 1150.00 a partir de 2023-05-18

Exemplo 4

```
+ * >Natural Source Header 000000[...]  
- define data local  
- 1 empregos  
    2 nome (a20)  
    2 idade (i2)  
    2 salario (n7.2) /* ex: 1000000.22  
    2 novosalario (n7.2)  
end-define  
  
input nome idade salario /* input => entrada de dados  
  
/* if-else => se idade < que 40, terá aumento de 15%  
/* senao terá aumento de 10%  
- if idade < 40  
    compute rounded novosalario := salario * 1.15  
- else  
    compute rounded novosalario := salario * 1.10  
end-if  
  
write 'Nome: ' nome 'Idade: ' idade  
write 'Salario: ' salario  
write 'Novo Salario: ' novosalario  
  
END
```

Natural Web I/O Output

NOME IDADE SALARIO

Natural Web I/O Output

Page 1 23-05-11 16:14:53

Nome: joca Idade: 33
Salario: 1000.00
Novo Salario: 1150.00

Natural Web I/O Output

NOME IDADE SALARIO

Natural Web I/O Output

Page 1 23-05-11 16:16:27

Nome: joca Idade: 41
Salario: 1000.00
Novo Salario: 1100.00

Exemplo 5

```
* >Natural Source Header 000000
define data local
1 empregos
    2 nome (a20)
    2 idade (i2)
    2 salario (n7.2) /* ex: 1000000.22
    2 novosalario (n7.2)
end-define

input nome idade salario /* input => entrada de dados

/* decide on se assemelha ao switch
/* com o none somos obrigados a informar o que
/* deve acontecer caso nenhum valor bata
decide on first value idade
    value 25, 30 /* valores especificos
        compute rounded novosalario := salario * 1.15
    value 31:40 /* intervalos
        compute rounded novosalario := salario * 1.10
    none
        novosalario := salario
end-decide

write 'Nome: ' nome 'Idade: ' idade
write 'Salario: ' salario
write 'Novo Salario: ' novosalario

END
```

Natural Web I/O Output

```
Page      1                                     23-05-11 16:28:10

Nome: joca                                Idade:      25
Salario:      1000.00
Novo Salario:      1150.00
```

Natural Web I/O Output

```
Page      1                                     23-05-11 16:30:08

Nome: joca                                Idade:      33
Salario:      1000.00
Novo Salario:      1100.00
```

Natural Web I/O Output

Page 1 23-05-11 16:31:02

Nome: joca Idade: 42
Salario: 1000.00
Novo Salario: 1000.00

Exemplo 6

```
+ * >Natural Source Header 000000[...]  
- define data local  
- 1 empregos  
    2 nome (a20)  
    2 idade (i2)  
    2 salario (n7.2) /* ex: 1000000.22  
    2 novosalario (n7.2)  
end-define  
input nome idade salario /* input => entrada de dados  
  
/* decide for serve para expressoes booleanas  
- decide for first condition  
- when idade > 30 and salario >= 1000  
    compute rounded novosalario := salario * 1.15  
- when idade < 30 and salario >= 1000  
    compute rounded novosalario := salario * 1.10  
    when none  
        novosalario := salario  
end-decide  
  
write 'Nome: ' nome 'Idade: ' idade  
write 'Salario: ' salario  
write 'Novo Salario: ' novosalario  
END
```

Natural Web I/O Output

Page 1 23-05-11 16:47:38

Nome: joca Idade: 33
Salario: 1000.00
Novo Salario: 1150.00

Natural Web I/O Output

```
Page      1                                     23-05-11  16:49:38

Nome:  joca                                Idade:      29
Salario:      1000.00
Novo Salario:      1100.00
```

Exemplo 7

```

❏ * >Natural Source Header 000000❏
❏ define data local
❏ 1 empregos
    2 primeironome (a30)
    2 ultimnome (a30)
    2 nomecompleto (a) dynamic /* dynamic => nao é necessario informar o tamanho
end-define

primeironome := 'Joca'
ultimnome := 'Tigre'

❏ /* compress => compacta várias constantes/variáveis numa String
/* a string compactada é jogada dentro da variavel nomecompleto
compress 'Senhor' primeironome ultimnome into nomecompleto

print nomecompleto

END

```

Natural Web I/O Output	
Page	1
23-05-12 09:51:39	
Senhor Joca Tigre	

Exemplo 8

```
* >Natural Source Header 000000
define data local
1 empregos
  2 primeironome (a30)
  2 ultimonome (a30)
  2 nomecompleto (a) dynamic /* dynamic => n é necessario informar tamanho
end-define

primeironome := 'Joca'
ultimonome := 'Tigre'

/* compress => compacta várias constantes/variáveis numa String
/* a string compactada é jogada dentro da variavel nomecompleto
/* with delimiter => posso definir um delimitador ao invés do
/* espaço, como uma vírgula
compress 'Senhor' primeironome ultimonome into nomecompleto
with delimiter ','

print nomecompleto

END
```

Natural Web I/O Output

Page 1

23-05-12 09:58:28

Senhor,Joca,Tigre

Exemplo 9

```
+ * >Natural Source Header 000000[...]  
- define data local  
- 1 empregos  
    2 primeironome (a30)  
    2 ultimnome (a30)  
    2 nomecompleto (a) dynamic /* dynamic => n precisa informar tamanho  
    2 matriz (a30/1:3) /* matriz com 1 linha e 3 colunas  
end-define  
  
primeironome := 'Joca'  
ultimonome := 'Tigre'  
  
- /* compress => compacta várias constantes/variáveis numa String  
  /* a string compactada é jogada dentro da variavel nomecompleto  
  /* with delimiter => posso definir um delimitador ao invés do  
  /* espaço, como uma vírgula  
  compress 'Senhor' primeironome ultimnome into nomecompleto  
  with delimiter ','  
  
- /* separo nomecompleto nos elementos da matriz  
  /* o asterisco indica que não estamos lidando com um índice  
  /* específico da matriz, mas com a própria matriz  
  separate nomecompleto into matriz(*)  
  
display matriz(*)  
  
END
```

Natural Web I/O Output

```
Page      1                                     23-05-12 10:11:14  
  
          MATRIZ  
-----  
  
Senhor  
Joca  
Tigre
```

Exemplo 10

```
* >Natural Source Header 000000
define data local
1 empregos
    2 primeironome (a30)
    2 ultimnome (a30)
    2 nomecompleto (a) dynamic /* dynamic => n precisa informar tamanho
1 n (i2) /* número de ocorrências
1 ix (i2) /* posição da primeira ocorrência
end-define
primeironome := 'Joca'
ultimnome := 'Tigre'

/* compress => compacta várias constantes/variáveis numa String
/* a string compactada é jogada dentro da variavel nomecompleto
/* with delimiter => posso definir um delimitador ao invés do
/* espaço, como uma vírgula
compress 'Senhor' primeironome ultimnome into nomecompleto
with delimiter ','

/* examine => pesquisa um campo (alfanumérico ou array)
/* a variavel nomecompleto sera examinada
/* a virgula será pesquisada
/* giving number => obtém o nº de vezes que ocorreu a virgula
/* na variavel nomecompleto. Se a virgula n for encontrado
/* retornará 0 (zero)
/* position obtém a posicao da primeira virgula no indice
examine nomecompleto for ',' giving number n position ix

print nomecompleto
write 'Numero: ' n 'index ' ix

END
```

Natural Web I/O Output

Page 1

23-05-12 11:32:08

Senhor,Joca,Tigre

Numero: 2 index

7

Primeira virgula na posicao 7

Nº de vírgulas na string

Exemplo 11

```
* >Natural Source Header 000000
define data local
1 funcionarios
  2 primeironome (a30)
  2 ultimonome (a30)
  2 nomecompleto (a) dynamic
end-define
primeironome := 'Joca'
ultimonome := 'Tigre'

compress 'Senhor' primeironome ultimonome into nomecompleto with delimiter ','

print 'Antes: ' nomecompleto

/* transforma nomecompleto para maiusculas
examine nomecompleto translate into upper

print 'Depois: ' nomecompleto

END
```

Natural Web I/O Output

Page	1	23-05-12 11:40:09
Antes: Senhor,Joca,Tigre		
Depois: SENHOR,JOCA,TIGRE		

Exemplo 12

```
⊕ * >Natural Source Header 000000...
- define data local
  1 i (i2) /* variavel de controle. Iterador
  1 root (n1.16) /* ex: 1.1234567898765432
  end-define

- /* valor inicial = 1
  /* valor final = 5
  /* sera executado 5 voltas
  /* em cada volta a raiz do nº i será executada
  /* e armazenada na variavel root
- for i := 1 to 5
  root := sqrt(i)
  write i root
end-for

END
```

Natural Web I/O Output

Page	1	23-05-12 11:49:42
1	1.0000000000000000	
2	1.4142135623730951	
3	1.7320508075688772	
4	2.0000000000000000	
5	2.2360679774997898	

Exemplo 13

```
+ * >Natural Source Header 000000[...]  
- define data local  
  1 stop-loop (1) /* 1 é logical/boolean  
  end-define  
  
- /* esse repeat é parecido com um while  
  /* o stop-loop é iniciado com false  
  /* enquanto stop-loop for false o write será executado  
  /* no final do repeat o stop-loop recebe true  
  /* entao o write é mostrado apenas uma vez  
- repeat  
  while stop-loop = false  
  write 'Estou no loop'  
  move true to stop-loop  
end-repeat  
  
END
```

Natural Web I/O Output

Page 1

23-05-12 11:59:55

Estou no loop

Exemplo 14

```
+ * >Natural Source Header 000000[...]  
- define data local  
  1 stop-loop (1) /* 1 é logical/boolean  
  end-define  
  
- /* esse repeat é parecido com um do while  
  /* o stop-loop é iniciado com false  
  /* apos o write o stop-loop recebe true  
  /* until => condicao no final, se stop-loop  
  /* for true o laço é encerrado  
- repeat  
  write 'Estou no loop'  
  move true to stop-loop  
  until stop-loop = true  
end-repeat  
  
END
```

Natural Web I/O Output

Page 1

23-05-12 13:09:39

Estou no loop

Exemplo 15

```
+ * >Natural Source Header 000000[...]  
- define data local  
  1 stop-loop (1) /* 1 é logical/boolean  
  end-define  
  
- /* o stop-loop é iniciado com false  
  /* apos o write o stop-loop recebe true  
  /* no if temos um escape bottom que serve  
  /* para encerrar o loop no caso de  
  /* stop-loop ser true  
- repeat  
  write 'Estou no loop'  
  move true to stop-loop  
-   if stop-loop = true  
    escape bottom  
  end-if  
end-repeat  
  
END
```

Natural Web I/O Output

Page 1

23-05-12 13:17:04

Estou no loop

Exemplo 16

Neste exemplo criaremos 3 arquivos:

1º arquivo principal

2º arquivo com as variáveis necessárias (parameter data area)

3º subprograma com a logica

Arquivo com as variáveis (parameter data area)

```
TESTE.NSP *VAR.NSA LOG.NSN Natural I/O
DEFINE DATA PARAMETER
/* >Natural Source Header 000000
1 nome (a) dynamic
1 saudacao (a) dynamic
END-DEFINE
```

Subprograma com a logica

```
TESTE.NSP *VAR.NSA LOG.NSN Natural I/O
/* >Natural Source Header 000000
define data parameter
using var
end-define

compress 'Saudacoes ' nome 'O tempo é agora ' *timx into saudacao
END
```

Programa principal

```
TESTE.NSP VAR.NSA LOG.NSN Natural I/O
/* >Natural Source Header 000000
define data local
using var
end-define

move 'Joca Tigre' to nome

CALLNAT "LOG" nome saudacao

print saudacao

END
```

Natural Web I/O Output

Page 1

23-05-12 13:53:28

Saudacoes Joca Tigre O tempo é agora 13:53:28

Exemplo 17

```
+ * >Natural Source Header 000000[.]
- define data local
- 1 vw VIEW OF VEICULOS
    2 PLACA (A7)
    2 MARCA (N6.0)
    2 COR (N2.0)
  end-define

- /* read => inicia um loop de processamento p/
  /* leitura sequencial de um arquivo ADABAS
  /* display => imprimir dados do programa em
  /* formato de colunas
  /* o display automaticamente inclui cabeçalhos
  /* com o mesmo nome da variável caso não seja
  /* informado um cabeçalho diferente
- read vw
    display vw
  end-read

END
```

Natural Web I/O Output

Page 1

23-05-12 14:17:47

VW

PLACA	MARCA	COR
-----	-----	----
AA 0001	103502	4
AA 0009	104901	9
AM 9594	117099	15
BC 1032	7699	1
AW 0732	7699	11
AW 0772	7699	1
AX 0957	10699	11
AM 0425	4899	15
AM 4028	117902	14
AA 9664	201103	2
AA 0044	117903	4
AA 0046	103102	3
AA 0050	117799	14
AX 0859	4899	15
AM 0074	104199	15
AM 8890	117902	4

MORE

Exemplo 18

```
+ * >Natural Source Header 000000...  
- define data local  
- 1 vw VIEW OF VEICULOS  
    2 PLACA (A7)  
    2 MARCA (N6.0)  
    2 COR (N2.0)  
  end-define  
  
- /* by => busca a placa a partir da AA 0001  
  /* placa é um descritor  
- read vw by placa = 'AA 0001'  
    display vw  
  end-read  
  
END
```

Natural Web I/O Output

Page 1

23-05-12 14:24:37

VW

PLACA	MARCA	COR
AA 0001	103502	4
AA 0009	104901	9
AA 0025	200699	4
AA 0044	117903	4
AA 0046	103102	3
AA 0050	117799	14
AA 0056	104399	6
AA 0057	117902	2
AA 0058	117902	9
AA 0066	104499	2
AA 0068	116799	4
AA 0078	104706	4
AA 0090	103299	2
AA 0101	104799	10
AA 0106	300599	2
AA 0109	117902	5

MORE

Exemplo 19

```
+ * >Natural Source Header 000000[.]
- define data local
- 1 vw VIEW OF VEICULOS
    2 PLACA (A7)
    2 MARCA (N6.0)
    2 COR (N2.0)
  end-define

- /* by => busca a placa a partir da AA 0001
  /* valor final = AA 0050
  /* a placa é um descritor
- read vw by placa = 'AA 0001' ending at 'AA 0050'
    display vw
  end-read

END
```

Natural Web I/O Output

Page 1 23-05-12 14:29:18

VW

PLACA	MARCA	COR
-----	-----	----
AA 0001	103502	4
AA 0009	104901	9
AA 0025	200699	4
AA 0044	117903	4
AA 0046	103102	3
AA 0050	117799	14

Exemplo 20

```
* >Natural Source Header 000000[...]  
define data local  
1 vw VIEW OF VEICULOS  
  2 PLACA (A7)  
  2 MARCA (N6.0)  
  2 COR (N2.0)  
end-define  
  
/* find seleciona um conjunto de registros e inicia um  
/* loop de processamento, onde os registros selecionados  
/* são lidos um a um  
/* no find os resultados são por padrão sempre  
/* ordenados pela isn  
find vw with placa >= 'AA 0001' and placa <= 'AA 0060'  
  display vw *isn /* isn é chave unica de um registro  
end-find  
  
END
```

Natural Web I/O Output

Page 1

VW			ISN
PLACA	MARCA	COR	
AA 0001	103502	4	1
AA 0009	104901	9	2
AA 0044	117903	4	30
AA 0046	103102	3	31
AA 0050	117799	14	32
AA 0056	104399	6	46
AA 0057	117902	2	47
AA 0058	117902	9	48
AA 0025	200699	4	26495

Ordenados pela ISN

Exemplo 21

```
+ * >Natural Source Header 000000...
- define data local
- 1 vw VIEW OF VEICULOS
    2 PLACA (A7)
    2 MARCA (N6.0)
    2 COR (N2.0)
  end-define

- find vw with placa >= 'AA 0001' and placa <= 'AA 0030'
    display vw *isn
  end-find

  skip 1 /* skip pula linhas
  display '-----'

- /* get obtém um unico registro atraves do isn
  /* get nao precisa de end
  get vw 2
    display vw *isn

END
```

Natural Web I/O Output

Page 1

VW			ISN
PLACA	MARCA	COR	

AA 0001	103502	4	1
AA 0009	104901	9	2
AA 0025	200699	4	26495

AA 0009	104901	9	2

Find

Get

Exemplo 22

```
* >Natural Source Header 000000
define data local
1 vw VIEW OF VEICULOS
  2 PLACA (A7)
end-define

/* histogram so funciona com a variavel do descritor
/* histogram mostra o nº de ocorrencias de cada
/* descritor ou indice
histogram vw placa
  display placa *number
end-histogram

END
```

Natural Web I/O Output

Page 1

PLACA	NMBR
-----	-----

AA 0001	1
AA 0009	1
AA 0025	1
AA 0044	1
AA 0046	1
AA 0050	1
AA 0056	1
AA 0057	1
AA 0058	1
AA 0066	1
AA 0068	1
AA 0078	1
AA 0090	1
AA 0101	1
AA 0106	1
AA 0109	1
AA 0118	1
AA 0127	1

MORE

Exemplo 23

```
* >Natural Source Header 000000
define data local
1 vw VIEW OF VEICULOS
  2 BAIRRO (A10)
  2 PLACA (A7)
  2 NOME (A30)
end-define

read vw by placa = 'AA 0001' ending at 'AA 0030'
  display placa nome bairro
end-read

display '-----'

/* dados do novo registro
placa := 'AA 0002'
nome := 'Joca Tigre'
bairro := 'Legal'

/* store => adiciona um novo registro
/* end transaction => atualiza os registros
/* dos comandos STORE, UPDATE e DELETE
store vw
end transaction

read vw by placa = 'AA 0001' ending at 'AA 0030'
  display placa nome bairro
end-read

END
```

Natural Web I/O Output

Page 1

PLACA	NOME	BAIRRO

AA 0001	CATIA SARAIVA	FLORES
AA 0002	Joca Tigre	Legal
AA 0009		CACHOEIRIN
AA 0025	ROSANGELA HITOTUZI	CENTRO

AA 0001	CATIA SARAIVA	FLORES
AA 0002	Joca Tigre	Legal
AA 0002	Joca Tigre	Legal
AA 0009		CACHOEIRIN
AA 0025	ROSANGELA HITOTUZI	CENTRO

Exemplo 24

```
+ * >Natural Source Header 000000[.]
- define data local
- 1 vw VIEW OF VEICULOS
    2 BAIRRO (A10)
    2 PLACA (A7)
    2 NOME (A30)
  end-define

- /* update atualiza registro
  /* update funciona apenas dentro
  /* do find ou do read
- find vw with placa = 'AA 0002'
    display placa nome *isn
    nome := 'Joca Jr'
    update
  end-find

- /* end transaction => atualiza os registros
  /* dos comandos STORE, UPDATE e DELETE
  end transaction

  display '-----'

- find vw with placa = 'AA 0002'
    display placa nome *isn
  end-find

END
```

Natural Web I/O Output

Page 1

PLACA	NOME	ISN

AA 0002 Joca Tigre		10
AA 0002 Joca Tigre		11
AA 0002 Joca Tigre		12
AA 0002 Joca Tigre		13

AA 0002 Joca Jr		10
AA 0002 Joca Jr		11
AA 0002 Joca Jr		12
AA 0002 Joca Jr		13

Exemplo 25

```
* >Natural Source Header 000000[...]  
define data local  
1 vw VIEW OF VEICULOS  
  2 BAIRRO (A10)  
  2 PLACA (A7)  
  2 NOME (A30)  
end-define  
  
find vw with placa = 'AA 0002'  
  display placa nome *isn  
end-find  
  
display '-----'  
  
find vw with placa = 'AA 0002'  
  delete  
end-find  
  
find vw with placa = 'AA 0002'  
  display placa nome *isn  
end-find  
  
end transaction  
  
END
```

Natural Web I/O Output

Page 1 23-05-12 16:17:11

PLACA	NOME	ISN

AA 0002 Joca Jr		10
AA 0002 Joca Jr		11
AA 0002 Joca Jr		12
AA 0002 Joca Jr		13



Nao trouxe nada pois
os dados foram
deletados

Exemplo 26

```
+ * >Natural Source Header 000000[.]
- define data local
  1 nome (a30)
  1 idade (i2)
end-define

- /* as barras apos o input significam a
  /* quantidade de linhas a serem puladas
  /* 10T => espaço da margem a esquerda
  input ///// 10T 'NOME: ' nome

  write 'Nome: ' nome

  END
```

Natural Web I/O Output

NOME:

10T - tabulação

Quantidade de Barras

Exemplo 27

```
+ * >Natural Source Header 000000[...]  
- define data local  
  1 nome (a30)  
  1 idade (i2)  
end-define  
  
- /* O i apos a string o deixa em negrito  
  /* A barra pula uma linha  
input 'NOME: ' (i) nome / 'IDADE: ' idade  
  
write 'NOME: ' nome 'IDADE: ' idade  
  
END
```

Natural Web I/O Output

NOME:

IDADE:

Exemplo 28

```
+ * >Natural Source Header 000000[...]  
- define data local  
  1 nome (a30)  
  1 idade (i2)  
end-define  
  
- /* '.' é um caracter de preenchimento  
  /* definido pelo usuario  
input (ad='.') 'NOME: ' nome  
  
write 'NOME: ' nome  
  
END
```


Natural Web I/O Output

NOME:

Exemplo 29

```
+ * >Natural Source Header 000000[.]  
- define data local  
  1 nome (a30)  
  1 idade (i2)  
  end-define  
  
- /* mark 2 => inicia o programa com o  
  /* cursor no segundo input (idade)  
  input mark 2 /// nome idade  
  
  write 'NOME: ' nome 'IDADE: ' idade  
  
  END
```

Natural Web I/O Output

NOME IDADE 

Exemplo 30

```
* >Natural Source Header 000000[...]  
define data local  
1 nome (a30)  
1 idade (i2)  
end-define  
  
input with text 'Digite seu nome' 'NOME: ' nome  
  
write 'NOME: ' nome  
  
END
```

Natural Web I/O Output

NOME:

Digite seu nome

Exemplo 31

alarm => faz um som

```
+ * >Natural Source Header 000000[...]  
- define data local  
  1 usuario (a10)  
  1 senha (a10)  
end-define  
  
input (ad=m) with text 'Preencher os campos' //  
  10T 'Usuario: ' (i) usuario /  
  10T 'Senha: ' (i) senha (ad=n) /  
  
- if usuario = ' '  
  reinput full 'Informe o usuario.' mark *usuario alarm  
end-if  
  
- if senha = ' '  
  reinput full 'Informe a senha.' mark *senha alarm  
end-if  
  
END
```

Natural Web I/O Output

Usuario:

Senha:

Preencher os campos

Natural Web I/O Output

Usuario:

Senha:

Informe a senha.

Exemplo 32

```
+ * >Natural Source Header 000000
- define data local
- 1 vw VIEW OF VEICULOS
    2 PLACA (A7)
    2 NOME (A30)
  end-define

- read vw
    display notitle vw
  end-read

END
```

Natural Web I/O Output

VW			
PLACA	NOME		

AA 0001	CATIA SARAIVA		
AA 0009			
AM 9594	JOSE LUDGERO	SOUZA	F2T215440
BC 1032	JOSE LUDGERO	SOUZA	
AW 0732	JOSE LUDGERO	SOUZA	
AW 0772	JOSE LUDGERO	SOUZA	
AX 0957	JOSE LUDGERO	SOUZA	
AM 0425	JANIE G DOMINGUES		F2T212511
AM 4028	JANIE G DOMINGUES		F2T213044
AA 9664	JANIE G DOMINGUES		F2T281935

Natural Web I/O Output

VW	
PLACA	NOME

AA 0001	CATIA SARAIVA
AA 0009	
AM 9594	JOSE LUDGERO SOUZA F2T215440
BC 1032	JOSE LUDGERO SOUZA

Sem o "notitle" a pagina, a data e a hora sao visiveis.

Exemplo 33

```
* >Natural Source Header 000000
define data local
1 vw VIEW OF VEICULOS
  2 PLACA (A7)
  2 NOME (A30)
end-define

/* sf => espacamento entre as colunas (de 1 a 30)
/* hc => posicao cabecalho das colunas (center, left, right)
/* uc => caracter p/ sublinhar cabecalho (default: "-")
/* fc => caracter preenchimento cabecalho (default: branco)

read vw
  display (sf=10 hc=c uc=_ fc=+) vw
end-read

END
```

Natural Web I/O Output

Page 1 23-05-15 13:38:51

VW		
+PLACA+	+++++++NOME+++++++	
-----	-----	
AA 0001	CATIA SARAIVA	
AA 0009		
AM 9594	JOSE LUDGERO SOUZA	F2T215440
BC 1032	JOSE LUDGERO SOUZA	
AW 0732	JOSE LUDGERO SOUZA	
AW 0772	JOSE LUDGERO SOUZA	
AX 0957	JOSE LUDGERO SOUZA	
AM 0425	JANIE G DOMINGUES	F2T212511
AM 4028	JANIE G DOMINGUES	F2T213044
AA 9664	JANIE G DOMINGUES	F2T281935

Exemplo 34

```
+ * >Natural Source Header 000000[.]
- define data local
- 1 vw VIEW OF VEICULOS
    2 PLACA (A7)
    2 NOME (A30)
  end-define

/* podemos renomear as colunas
- read vw
    display 'PLACA' placa 'NOME' nome
  end-read

END
```

Natural Web I/O Output

Page 1 23-05-15 13:46:52

PLACA NOME

AA 0001 CATIA SARAIVA
AA 0009
AM 9594 JOSE LUDGERO SOUZA F2T215440
BC 1032 JOSE LUDGERO SOUZA
AW 0732 JOSE LUDGERO SOUZA
AW 0772 JOSE LUDGERO SOUZA
AX 0957 JOSE LUDGERO SOUZA

Exemplo 35

```
* >Natural Source Header 000000[...]  
define data local  
1 vw VIEW OF VEICULOS  
  2 BAIRRO (A10)  
  2 PLACA (A7)  
  2 MARCA (N6.0)  
  2 NOME (A30)  
end-define  
  
/* AL => tamanho maximo p/ exibicao alfanumerico  
/* NL => tamanho maximo p/ exibicao numerico  
/* SF => espacamento entre colunas  
read vw  
  display (sf=5) placa nome(al=10) marca(nl=3)  
end-read  
  
END
```

Natural Web I/O Output

Page 1

23-05-15 14:01:02

PLACA	NOME	MARCA
-----	-----	-----
AA 0001	CATIA SARA	502
AA 0009		901
AM 9594	JOSE LUDGE	099
BC 1032	JOSE LUDGE	699
AW 0732	JOSE LUDGE	699
AW 0772	JOSE LUDGE	699
AX 0957	JOSE LUDGE	699

Exemplo 36

```
+ * >Natural Source Header 000000[.]
- define data local
- 1 vw VIEW OF VEICULOS
    2 BAIRRO (A10)
    2 PLACA (A7)
    2 MARCA (N6.0)
    2 NOME (A30)
end-define

/* 10x => espacamento a esquerda
- read vw
    write 10x 'PLACA: ' placa 'NOME: ' nome 'BAIRRO: ' bairro
end-read

END
```

Natural Web I/O Output

Page	1	23-05-15 14:24:49
FLORES	PLACA: AA 0001 NOME: CATIA SARAIVA	BAIRRO:
CACHOEIRIN	PLACA: AA 0009 NOME:	BAIRRO:
CENTRO	PLACA: AM 9594 NOME: JOSE LUDGERO SOUZA F2T215440	BAIRRO:
ALVORADA 2	PLACA: BC 1032 NOME: JOSE LUDGERO SOUZA	BAIRRO:
HILEIA 1	PLACA: AW 0732 NOME: JOSE LUDGERO SOUZA	BAIRRO:
ALVORADA 2	PLACA: AW 0772 NOME: JOSE LUDGERO SOUZA	BAIRRO:

Exemplo 37

```
+ * >Natural Source Header 000000[...]  
- define data local  
- 1 vw VIEW OF VEICULOS  
    2 BAIRRO (A10)  
    2 PLACA (A7)  
    2 MARCA (N6.0)  
    2 NOME (A30)  
end-define  
  
- /* write title left => define um titulo  
  /* no canto esquerdo  
  write title left 'NOME E PLACA'  
  
  skip 1 /* gera linha em branco  
  
- /* write trailer left => linhas impressas  
  /* no rodapé de um relatorio  
  write trailer left 'Página' *page-number  
  
  skip 1 /* gera linha em branco  
  
- read vw  
    display placa nome  
end-read  
  
END
```

Natural Web I/O Output

NOME E PLACA

PLACA

NOME

AA 0001 CATIA SARAIVA

AA 0009

AM 9594 JOSE LUDGERO SOUZA F2T215440

BC 1032 JOSE LUDGERO SOUZA

AW 0732 JOSE LUDGERO SOUZA

AW 0772 JOSE LUDGERO SOUZA

AA 0044

AA 0046

AA 0050

AX 0859 JANIE G DOMINGUES F2T217342

AM 0074 JANIE G DOMINGUES F2T217320

AM 8890 JOSE LUDGERO SOUZA F2T216030

AA 0056

AA 0057

Página

1

Exemplo 38

```
* >Natural Source Header 000000[.]
define data local
1 a (n2)
1 b (n3.2) /* 3 antes da virgula e 2 depois
1 c (n2)
1 data (d)
end-define

/* T0 => os valores sao adicionados a
/* variavel A
add +5 -2 -1 to a
write a

/* GIVING => o resultado da soma dos valores
/* é armazenado na variavel B
add 0.231 3.6 giving b
write b

/* ROUNDED => o resultado é arredondado
/* ajustando-se ao tamanho de C
add rounded 2.9 3.8 giving c
write c

/* Atribui a data atual para a variavel data
/* em seguida adiciona 7
move *datx to data
add 7 to data
write data

END
```

Natural Web I/O Output

Page	1	23-05-15 15:48:36
2		
3.83		
7		
23-05-22		

Exemplo 39

```
* >Natural Source Header 000000[...]  
define data local  
1 a (n2.2) init <50>  
1 b (p2.2)  
1 c (p2.1) init <2.4>  
end-define  
  
/* A - 6 = 44  
subtract 6 from a  
write 'A: ' a  
  
/* 11 - 6 = 5  
subtract 6 from 11 giving a  
write 'A: ' a  
  
/* a(5) - (3 + 4 = 7) => -2  
/* -2 é colocado dentro de b  
subtract 3 4 from a giving b  
write 'A:' a 'B: ' b  
  
/* a(5) - (-7) => 12  
/* 12 é colocado dentro do b  
subtract -3 -4 from a giving b  
write 'A: ' a 'B: ' b  
  
/* 2.4 - 2.06 = 0.34 => 0.3  
subtract rounded 2.06 from c  
write 'C: ' c  
  
END
```

Natural Web I/O Output

Page 1

23-05-15 16:25:04

A: 44.00

A: 5.00

A: 5.00 B: -2.00

A: 5.00 B: 12.00

C: 0.3

Exemplo 40

```
* >Natural Source Header 000000[.]
define data local
1 a (p2.2) init <2>
1 b (p2.2)
1 c (i2)
end-define

/* a(2) * 3 = 6
/* 6 é armazenado em a
multiply a by 3
write 'A: ' a

/* 3 * a(6) = 18
/* 18 é armazenado em b
multiply 3 by a giving b
write 'B: ' b

/* 3 * 3.6 = 10.8
/* Arredonda de acordo com o campo receptor
/* como "c" aceita apenas inteiro
/* o valor fica em c fica 11
multiply rounded 3 by 3.6 giving c
write 'C: ' c

END
```

Natural Web I/O Output

Page	1	23-05-16 10:32:53
A:	6.00	
B:	18.00	
C:	11	

Exemplo 41

```
* >Natural Source Header 000000[.]
/* 0 # é somente para diferenciar a
/* variavel dos outros comandos
define data local
1 #A (n5) init <20>
1 #B (n5.2)
1 #C (n3.2)
1 #D (n1)
1 #E (n1) init <3>
1 #F (n1)
end-define

/* A(20) / 5 = 4 | 4 é armazenado em A
divide 5 into #A
write 'A: ' #A

/* A(4) / 5 = 0.80 | 0.80 é armazenado em B
divide 5 into #A giving #B
write 'B: ' #B

/* 3.1 / 3 = 1.03 | 1.03 é armazenado em C
divide 3 into 3.1 giving #C
write 'C: ' #C

/* 3.1 / 3 = 1.03 | 1.03 é armazenado em D
/* porem, como D tem o tipo n1
/* ele aceita apenas 1 algarismo
/* entao pega apenas o 1
divide 3 into 3.1 giving #D
write 'D: ' #D

/* E(3) / 2 = 1.5 | 1 é armazenado em E
/* pois E tem o tipo n1 que aceita apenas
/* 1 algarismo
/* remainder => resto da divisao é armazenado
/* em F
divide 2 into #E remainder #F
write 'E: ' #E 'F: ' #F
END
```

Natural Web I/O Output

Page 1

23-05-16 11:00:26

A: 4
B: 0.80
C: 1.03
D: 1
E: 1 F: 1

Exemplo 42

```
* >Natural Source Header 000000...
/* 0 # é somente para diferenciar a
/* variavel dos outros comandos
define data local
1 #A (n3)
1 #B (n3)
1 #C (n3.4)
1 #D (n3.4) init <31.3567>
end-define

/* compute => executa operacoes aritmeticas
/* e permite funcoes do Natural

/* o resultado da expressao é 7
/* 7 é armazenado em A
compute #A = 3 * 2 + 4 / 2 - 1
write 'A: ' #A

/* 3.4 * 2.7 = 9.18
/* 9 é armazenado em B
compute rounded #B = 3.4 * 2.7
write 'B: ' #B

/* o comando compute é opcional desde
/* que tenha o := (ponto igual)
/* sqrt é a funcao de raiz quadrada
/* raiz quadrada de 31.3567 é 5.5997
#C := sqrt(#D)
write 'C: ' #C

END
```

Natural Web I/O Output

Page 1

23-05-16 11:19:23

A: 7

B: 9

C: 5.5997

Exemplo 43

```
* >Natural Source Header 000000[.]
define data local
1 #NOME (a20)
1 #SOBRENOME (a20)
1 #NOMECOMPLETO (a20)
1 #DIA (i2)
1 #MES (i2)
1 #ANO (i4)
1 #DATA (a20)
end-define

#NOME := 'Joca'
#SOBRENOME := 'Tigre'
#DIA := 11
#MES := 05
#ANO := 2023

/* compress => combinar conteudo p/ um campo

/* combina nome e sobrenome p/ nomecompleto
compress #NOME #SOBRENOME into #NOMECOMPLETO
write 'Nome Completo: ' #NOMECOMPLETO

/* leaving no => nao deixa espaço em branco
/* entre os dados
compress #DIA #MES #ANO into #DATA leaving no
write 'Data: ' #DATA

/* with delimiter => especifica um caracter p/
/* servir de separador entre os campos
compress #DIA #MES #ANO into #DATA with delimiter '/'
write 'Data: ' #DATA

END
```

Natural Web I/O Output

Page 1 23-05-16 11:46:59

Nome Completo: Joca Tigre
Data: 1152023
Data: 11/5/2023

Exemplo 44

```
* >Natural Source Header 000000[...]  
define data local  
1 pessoas1  
  2 primeironome1 (a20) init <'Jose'>  
  2 idade1 (n2) init <33>  
1 pessoas2  
  2 primeironome2 (a20)  
  2 idade2 (n2)  
end-define  
  
/* move by position => movimenta todos os campos  
/* de um estrutura p/ outra, podendo ter nomes  
/* diferentes  
/* Regras:  
/* - nº de campos nos 2 grupos precisa ser igual  
/* - níveis dos campos precisam ser iguais  
/* - se houver array, precisam ter as mesmas dimensoes  
move by position pessoas1 to pessoas2  
  
display pessoas2  
  
END
```

Natural Web I/O Output

Page 1 23-05-16 13:41:24

PESSOAS2

PRIMEIRONOME2	IDADE2
-----	-----
Jose	33

Exemplo 45

```
* >Natural Source Header 000000[...]  
define data local  
1 nome (a10)  
1 data (a10)  
end-define  
  
/* move all => preenche uma variavel string  
/* com o parametro especificado  
move all '*' to nome  
write 'Nome:' nome  
  
skip 1 /* pula 1 linha  
  
/* edited => movimenta um campo com uma mascara  
/* para outro campo  
move edited *datx(em=dd/mm/yyyy) to data  
write 'Data:' data  
  
END
```

Natural Web I/O Output

Page 1

23-05-16 13:53:48

Nome: *****

Data: 16/05/2023

Exemplo 46

```
* >Natural Source Header 000000[...]  
define data local  
1 nome1 (a10) init <'Joca Tigre'>  
1 nome2 (a10)  
end-define  
  
/* substring => move a porcao de um campo  
/* p/ outro campo  
/* move substring(varivael1, i, q) to variavel2  
/* i => posicao inicial | q => quantidade bytes  
move substring(nome1,6,5) to nome2  
write nome2  
  
END
```

Natural Web I/O Output

Page 1

23-05-16 14:10:20

Tigre

Exemplo 47

```
Ⓢ * >Natural Source Header 000000[.]
Ⓢ define data local
1 nome (a10)
1 idade (n3) init <231>
1 numero (n3)
1 logico (l)
1 data (a10)
end-define

nome := 'Joca'
numero := 34
logico := true
move edited *datx(em=dd/mm/yyyy) to data

write 'Nome:' nome / 'Idade:' idade / 'Numero:' numero /
      'Logico:' logico / 'Data:' data

skip 1

Ⓢ /* reset estabelece valores nulos de acordo com
/* o tipo da variavel ou dos valores especificados
/* inicialmente
reset nome idade numero logico data
write 'Nome:' nome / 'Idade:' idade / 'Numero:' numero /
      'Logico:' logico / 'Data:' data

skip 1

Ⓢ /* reset initial restabelece o valor original
/* definido no define data pelo init
reset initial idade
write 'Idade inicial:' idade

END
```

Natural Web I/O Output

Page 1

23-05-16 14:54:51

Nome: Joca

Idade: 231

Numero: 34

Logico: X

Data: 16/05/2023

Nome:

Idade: 0

Numero: 0

Logico:

Data:

Idade inicial: 231

Exemplo 48

```
> * >Natural Source Header 000000[...]  
define data local  
1 a (a10)  
1 b (a10)  
1 c (a10)  
1 n1 (a10)  
1 n2 (a10)  
1 array (a3/3)  
end-define  
  
/* separate é usado p/ separar o conteudo de um campo  
/* p/ 2 ou mais campos (alfanumericos, array ou grupos)  
separate 'a,b,c' into a b c  
write 'A:' a 'B:' b 'C:' c  
  
skip 1  
  
/* delimiter faz a separacao a partir do caracter  
/* indicado  
/* se omitido, o delimitador é qualquer caracter  
/* menor que a letra "a" minuscula  
separate '123.4,234.1' into n1 n2 with delimiter ','  
write 'n1:' n1 'n2:' n2  
  
skip 1  
  
separate '1 2 3' into array(*)  
write 'Array com 3 posicoes:' array(*)  
  
END
```

Page 1

A: a B: b C: c

n1: 123.4 n2: 234.1

Array com 3 posicoes: 1 2 3

Exemplo 49

```
> * >Natural Source Header 000000[.]
define data local
1 letras1 (a10) init <'aaabbbcccd'>
1 letras2 (a10) init <'aabbccdde'>
1 letras3 (a10) init <'aaabbbccc'>
1 letras4 (a10) init <'aaabbbccc'>
1 array1 (a2/4) init (1) <'aa'> (2) <'bb'> (3) <'cc'> (4) <'dd'>
1 i (i1)
1 j (i1)
1 z (i1)
end-define

/* examine pesquisa um campo (alfa ou array) por uma opcao
/* e altera, deleta ou conta

/* giving position obtem a partir de qual byte o 'bbb' inicia
/* se o 'bbb' nao for encontrado, retorna 0
examine letras1 for 'bbb' giving position i
write i

/* giving index obtem o indice em que o 'dd' ocorre
/* caso nao seja encontrado, retorna 0
examine array1(*) for 'dd' giving index j
write j

/* giving number obtem a quantidade de vezes que
/* ocorreu o 'bb', se n for encontrado retorna 0
examine array1(*) for 'bb' giving number z
write z

/* replace troca o 'cc' pelo 'xy'
examine letras2 for 'cc' replace 'XY'
write letras2

/* replace first troca apenas a primeira
/* ocorrencia
examine letras3 for 'b' replace first 'X'
write letras3

/* delete apaga a ocorrencia especificada
/* se first for especificado, apaga apenas
/* a primeira ocorrencia
examine letras4 for 'bbb' delete
write letras4

END
```

Page 1

23-05-16 16:15:22

4
4
1
aabbXYdde
aaaXbbccc
aaaccc

Exemplo 50

```
* >Natural Source Header 000000[.]
define data local
1 pessoa (a20)
1 salario (n5)
end-define

input pessoa salario

if salario >= 4000
    write notitle 'Salario do'pessoa'é maior/igual que 4000.'
else
    write notitle 'Salario do'pessoa'é menor que 4000.'
end-if

END
```

Natural Web I/O Output

Salario do Joca é menor que 4000.

Exemplo 51

```
* >Natural Source Header 000000[.]
define data local
1 valor (n3)
end-define

input valor

/* decide é semelhante ao switch
/* decide on => usado p/ executar uma ou mais
/* ações, dependendo de um valor
/* os valores podem ser especificos ou intervalos
/* none é executado se nenhum valor for encontrado
/* o none é obrigatorio
/* ignore é usado quando n se deseja executar nada
decide on first value of valor
    value 1, 2, 3, 4
        write 'Numeros de 1 a 4'
    value 5:10
        write 'Numeros de 5 a 10'
    none
        ignore
end-decide

END
```

Natural Web I/O Output

Page 1

23-05-17 10:33:14

Numeros de 5 a 10

Exemplo 52

```
+ * >Natural Source Header 000000[...]  
- define data local  
  1 valor (n3)  
  end-define  
  
  input valor  
  
- /* o decide for first condition executa uma ou mais  
  /* ações, dependendo de multiplas condicoes  
- decide for first condition  
-   when valor >= 1 and valor <= 10  
    write 'Valor de 1 a 10'  
-   when valor > 10 and valor < 20  
    write 'Valor maior que 10 e menor que 20'  
    when none  
      ignore  
  end-decide  
  
END
```

Natural Web I/O Output

Page 1

23-05-17 11:06:51

Valor maior que 10 e menor que 20

Exemplo 53

```
+ * >Natural Source Header 000000[.]
- define data local
  1 valor (n3)
  end-define

  input valor

- /* every => executa todas as instrucoes
  /* que satisfaçam as condicoes
  /* se o valor for 5, a primeira e a
  /* segunda condição serao executadas
- decide on every value of valor
-   value 1:6
      write 'Valor de 1 a 6'
-   value 4:10
      write 'Valor de 7 a 10'
-   none
      write 'Valores acima de 10'
  end-decide

  END
```

Natural Web I/O Output

Page 1

23-05-17 11:38:21

Valor de 1 a 6

Valor de 7 a 10

Exemplo 54

```
+ * >Natural Source Header 000000...
- define data local
  1 valor (n3)
  end-define

  input valor

- /* as instrucoes de dentro do any sao executadas
  /* se alguma das condicoes forem verdadeiras
  /* nesse exemplo a condicao 1:6 é executada
  /* juntamente com o any
- decide on first value of valor
-   value 1:6
      write 'Valor de 1 a 6'
-   value 4:10
      write 'Valor de 7 a 10'
-   any value
      write 'any'
-   none
      write 'Valores acima de 10'
  end-decide

  END
```

Natural Web I/O Output

Page 1

23-05-17 13:06:52

Valor de 1 a 6

any

Exemplo 55

```
+ * >Natural Source Header 000000[.]
- define data local
  1 valor (n3)
  end-define

  input valor

- /* as instrucoes de dentro do any sao executadas
  /* se alguma das condicoes forem verdadeiras
  /* nesse exemplo a condicao 1:6 e 4:10 são executadas
  /* juntamente com o any, devido ao comando every
- decide on every value of valor
-   value 1:6
      write 'Valor de 1 a 6'
-   value 4:10
      write 'Valor de 7 a 10'
-   any value
      write 'any'
-   none
      write 'Valores acima de 10'
  end-decide

  END
```

Natural Web I/O Output

Page 1

23-05-17 13:12:17

Valor de 1 a 6

Valor de 7 a 10

any

Exemplo 56

```
+ * >Natural Source Header 000000[...]  
- define data local  
  1 valor (n3)  
  end-define  
  
  input valor  
  
  write 'Valor digitado:' valor  
  
  skip 1  
  
- /* all => as instrucoes de dentro do all  
  /* sao executadas somente se todas as  
  /* condicoes forem verdadeiras  
  /* é obrigado o uso do every  
- decide on every value of valor  
-   value 1:6  
    write 'Valor de 1 a 6'  
-   value 4:10  
    write 'Valor de 7 a 10'  
-   value 6:8  
    write 'Valor de 5 a 8'  
-   all value  
    write 'all'  
-   none  
    write 'Valores acima de 10'  
  end-decide  
  
  END
```

Natural Web I/O Output

Page 1

23-05-17 13:22:35

Valor digitado: 6

Valor de 1 a 6

Valor de 7 a 10

Valor de 5 a 8

all

Exemplo 57

```
+ * >Natural Source Header 000000...  
- define data local  
  1 valor1 (n3) init <3>  
  1 valor2 (n3) init <5>  
end-define  
  
write 'Valor:' valor1  
skip 1 /* gera linha em branco  
write 'Valor:' valor2  
  
END
```

Natural Web I/O Output

Page	1
Valor:	3
Valor:	5

Exemplo 58

```
+ * >Natural Source Header 000000[...]  
- define data local  
  1 i (i2)  
  1 numpar (n2)  
  end-define  
  
- /* i => iterador, valor alterado a cada loop  
  /* 2 => valor inicial  
  /* 10 => valor final  
  /* 2 => incremento ou decremento  
- for i 2 10 2  
    write 'Número par:' i  
  end-for  
  
END
```

Natural Web I/O Output

Page	1	23-05-17 13:41:51
Número par:	2	
Número par:	4	
Número par:	6	
Número par:	8	
Número par:	10	

Exemplo 59

```
* >Natural Source Header 000000[...]  
define data local  
  1 condicao1 (1)  
  1 condicao2 (1) init <false>  
end-define  
  
/* repeat until se assemelha ao do while  
/* a condicao é posicionada no inferior do laço  
/* o write é executado até que condicao1 seja true  
/* aqui o write é executado apenas 1 vez  
repeat  
  write 'Estou no loop do repeat until.'  
  condicao1 := true  
  until condicao1 = true  
end-repeat  
  
skip 1  
  
/* a condicao é posicionada no inicio do laço  
/* como condicao2 inicia como false, o write é  
/* executado. Em seguida o condicao2 recebe true  
/* o write é executado apenas uma vez  
repeat  
  while condicao2 = false  
  write 'Estou no loop do repeat while.'  
  condicao2 := true  
end-repeat  
  
END
```

Natural Web I/O Output

Page 1

23-05-17 14:10:29

Estou no loop do repeat until.

Estou no loop do repeat while.

Exemplo 60

```
* >Natural Source Header 000000[.]
define data local
1 condicao1 (1)
1 condicao2 (1) init <false>
end-define

/* se while ou until forem omitidos podemos encerrar
/* o loop com escape bottom ou stop
repeat
  write 'Estou no loop do repeat usando escape bottom e o stop.'
  condicao1 := true
  if condicao1 = true
    /* escape bottom
    stop
  end-if
end-repeat

END
```

Natural Web I/O Output

Page 1

23-05-17 14:20:13

Estou no loop do repeat usando escape bottom e o stop.

Exemplo 61

```
* >Natural Source Header 000000[.]
define data local
1 vw VIEW OF VEICULOS
  2 PROPRIETARIO (A60)
  2 BAIRRO (A10)
  2 PLACA (A7)
end-define

/* Accept => usado p/ aceitar registros de acordo
/* com condicoes logicas em loops
/* sf => espacamento entre colunas
/* al => tamanho maximo p/ exibicao alfanumerico
read vw by placa = 'AA 0001' ending at 'AA 0100'
  accept if bairro = 'CENTRO'
  display proprietario(al=20) bairro placa
end-read

END
```

Natural Web I/O Output

Page 1

23-05-17 15:27:06

PROPRIETARIO	BAIRRO	PLACA
KWAN TANIGUCHI	CENTRO	AA 0025
JONAN FERREIRA DA SI	CENTRO	AA 0057
EMILIO TAIROVICHE	CENTRO	AA 0058
RAFAEL BARCELAR DE S	CENTRO	AA 0090

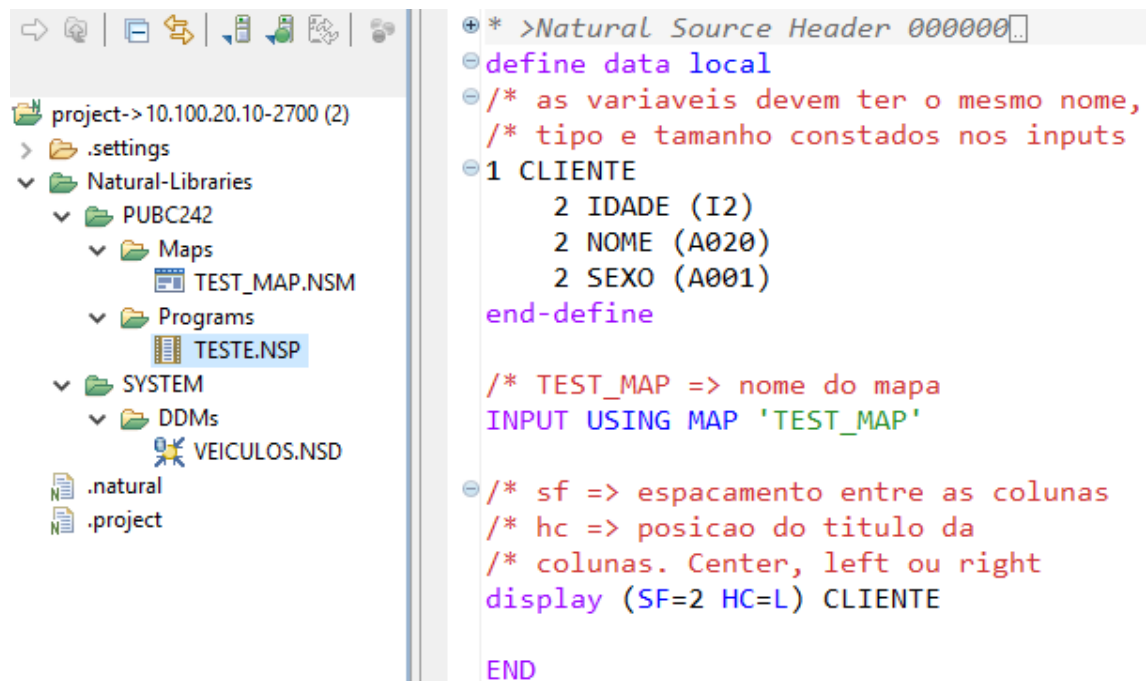
Exemplo 62

```
+ * >Natural Source Header 000000[...]  
- define data local  
  1 i (i2) init <1>  
  end-define  
  
- /* escape top => retorna p/ o inicio do loop  
  /* escape bottom => encerra o loop. Continuará  
  /* com o primeiro comando apos o loop  
- repeat  
-   if i = 5  
     escape bottom  
  end-if  
  i := i + 1  
  write i  
  escape top  
end-repeat  
  
write 'Fim do loop.'  
  
END
```

Natural Web I/O Output

Page	1	23-05-17 15:51:01
2		
3		
4		
5		
Fim do loop.		

Exemplo 63

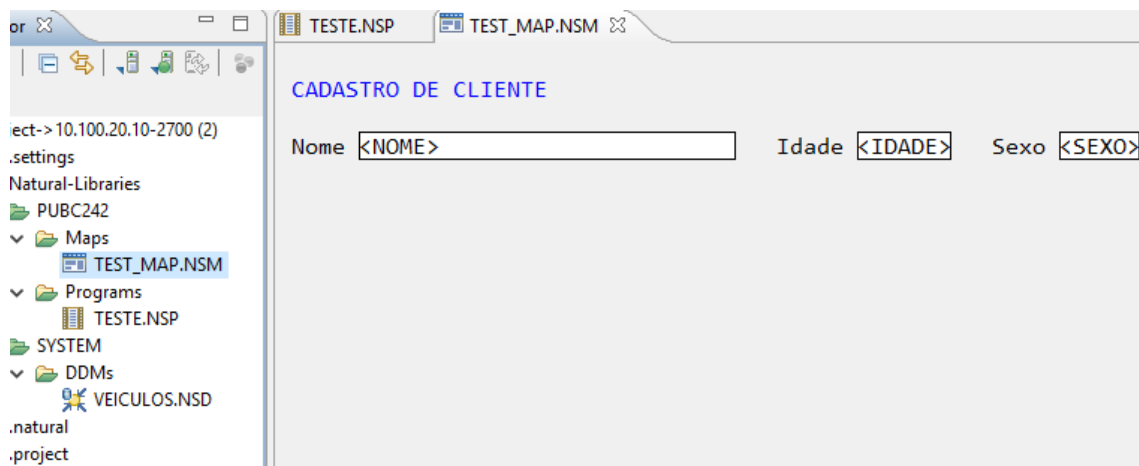


```
* >Natural Source Header 000000...
define data local
/* as variaveis devem ter o mesmo nome,
/* tipo e tamanho constados nos inputs
1 CLIENTE
    2 IDADE (I2)
    2 NOME (A020)
    2 SEXO (A001)
end-define

/* TEST_MAP => nome do mapa
INPUT USING MAP 'TEST_MAP'

/* sf => espacamento entre as colunas
/* hc => posicao do titulo da
/* colunas. Center, left ou right
display (SF=2 HC=L) CLIENTE

END
```



CADASTRO DE CLIENTE

Nome Idade Sexo

Exemplo 64

```
* >Natural Source Header 000000...
DEFINE DATA LOCAL
  1 IDADE (I2)
  1 NOME (A020)
  1 SEXO (A001)
END-DEFINE

/* janela1 é o nome

/* size 10 * 100 => tamanho da janela linhas X colunas
/* size auto => tamanho da janela é determinado automaticamente

/* base => posicao da abertura da janela
/* base pode ser 5 / 12 ou TOP/BOTTOM LEFT/RIGHT

/* framed on => janela terá moldura

/* position on => se a janela a ser exibida for maior que a
/* tela fisica, será mostrado na ultima linha uma informacao
/* que existe mais janela a ser exibida

/* set window => torna a janela ativa
/* set window off => desabilita a janela

DEFINE WINDOW JANELA1
size auto
base top right
title 'Cadastro de Cliente'
framed on position off

SET WINDOW 'JANELA1'
INPUT USING MAP 'test_map'
SET WINDOW OFF

END
```


Natural Web I/O Output

Cadastro de Cliente

CADASTRO DE CLIENTE

Nome Idade Sexo

Exemplo 65

Programa

Navigator

- project-> 10.100.20.10-2700 (2)
 - .settings
 - Natural-Libraries
 - PUBC242
 - Programs
 - TESTE.NSP
 - Subprograms
 - TEST2.NSN
 - SYSTEM
 - DDMs
 - VEICULOS.NSD
 - .natural
 - .project

```

+ * >Natural Source Header 000000
- DEFINE DATA LOCAL
  1 #NUM1 (N2) INIT<25>
  1 #NUM2 (N2) INIT<35>
  1 #SOMA (N5)
  END-DEFINE

/* CALLNAT chama o subprograma
/* passando os parametros
CALLNAT 'TEST2' #NUM1 #NUM2 #SOMA

WRITE '=' #NUM1 / '=' #NUM2 // '=' #SOMA

END

```

Subprograma

The screenshot shows the Natural IDE interface. On the left is the Navigator pane showing a project structure: project->10.100.20.10-2700 (2) with subfolders .settings, Natural-Libraries (containing PUBC242, Programs with TESTE.NSP, and Subprograms with TEST2.NSN), and SYSTEM (containing DDMs with VEICULOS.NSD). On the right is the editor pane showing the content of TEST2.NSN. The code is as follows:

```
* >Natural Source Header 000000
DEFINE DATA PARAMETER
1 #A (N2)
1 #B (N2)
1 #S (N5)
END-DEFINE

#S := #A + #B

END
```

Natural Web I/O Output

Page 1 23-05-19 11:17:59

#NUM1: 25
#NUM2: 35
#SOMA: 60

Exemplo 66

Programa

The screenshot shows the Natural IDE interface. On the left is the Navigator pane showing a project structure: project->10.100.20.10-2700 (2) with subfolders .settings, Natural-Libraries (containing PUBC242, Programs with TESTE.NSP, and Subroutines with SOMA.NSS), and SYSTEM (containing DDMs with VEICULOS.NSD). On the right is the editor pane showing the content of TESTE.NSP. The code is as follows:




```
* >Natural Source Header 000000
DEFINE DATA LOCAL
1 #NUM1 (N2) INIT<25>
1 #NUM2 (N2) INIT<35>
1 #SOMA (N5)
END-DEFINE









/* PERFORM chama a sub-rotina
/* passando os parametros
PERFORM SOMA #NUM1 #NUM2 #SOMA

WRITE '=' #NUM1 / '=' #NUM2 // '=' #SOMA

END
```

Sub-rotina


Navigator   

project-> 10.100.20.10-2700 (2)

- > .settings
- ▼ Natural-Libraries
 - ▼ PUBC242
 - ▼ Programs
 - TESTE.NSP
 - ▼ Subroutines
 - SOMA.NSS
 - ▼ SYSTEM
 - ▼ DDMs
 - VEICULOS.NSD

.natural
.project

TESTE.NSP SOMA.NSS  Natural I/O

```
* >Natural Source Header 000000[...]  
- DEFINE DATA PARAMETER  
  1 #NUM1 (N2)  
  1 #NUM2 (N2)  
  1 #SOMA (N5)  
  END-DEFINE  
- DEFINE SUBROUTINE SOMA  
  #SOMA := #NUM1 + #NUM2  
  END-SUBROUTINE  
  
END
```

Natural Web I/O Output

Page 1 23-05-19 11:36:22

#NUM1: 25
#NUM2: 35
#SOMA: 60

Exemplo 67

```
TESTE.NSP  CP_CD1.NSC  OI.NSP  Natural I/O
+ * >Natural Source Header 000000
- DEFINE DATA LOCAL
  1 #NUM1 (N2)
  1 #NUM2 (N2)
  1 #SOMA (N5)
  END-DEFINE

  /* executa aqui a copycode CP_CD1
  INCLUDE CP_CD1

  /* M => mode: modifiable
  /* D => apresentacao: default
  /* L => alinhamento: left
  /* T => tratamento caracter: transforma em maiuscula
  /* ZP => determina se os valores de campo de 0
  /* devem ser exibidos ou nao
  INPUT (AD=MDLT ZP=OFF)
    'ENTRE COM UM NÚMERO...:' #NUM1 /
    'ENTRE COM OUTRO NÚMERO:' #NUM2 //
    '-' (27) / /* Aqui o traço (-) é imprido 27X
    'PF3 - SAI'

  WRITE '=' #NUM1 / '=' #NUM2

  END
```

```
TESTE.NSP  CP_CD1.NSC  OI.NSP  Natural I/O
* >Natural Source Header 000000
* :Mode S
* :CP
* <Natural Source Header
/* SET KEY ALL => Faz com que todas as chaves
/* sejam sensíveis ao programa. Todas as atribuições
/* de função a qualquer tecla são substituídas
SET KEY ALL

/* Ao teclar f3 o programa OI é executado
SET KEY PF3 = 'OI'
```

TESTE.NSP CP_CD1.NSC OI.NSP Natural I/O

+ * >Natural Source Header 000000

- /* A instrução TERMINATE é usada para encerrar uma sessão
/* Natural. Uma instrução TERMINATE pode ser colocada em
/* qualquer lugar dentro de um programa Natural. Quando
/* uma instrução TERMINATE é executada, nenhum processamento
/* de fim de página ou loop final será executado.
TERMINATE
END

TESTE.NSP CP_CD1.NSC OI.NSP Natural I/O

Natural Web I/O Output

ENTRE COM UM NÚMERO...:
ENTRE COM OUTRO NÚMERO:

PF3 - SAI

TESTE.NSP CP_CD1.NSC OI.NSP Natural I/O

Natural Web I/O Output

Page 1 23-05-19 13:38:46

#NUM1: 22
#NUM2: 4

Exemplo 68

```
+ * >Natural Source Header 000000..
- DEFINE DATA LOCAL
- /* nesse exemplo os arrays possuem o mesmo
  /* tamanho (3,2). 3 linhas e 2 colunas
  /* o valor fica dentro dos caracteres <>
  1 #A (N2/3,2) INIT
    (1,1) <1> (1,2) <1>
    (2,1) <2> (2,2) <0>
    (3,1) <3> (3,2) <2>
  1 #B (N2/3,2) INIT
    (1,1) <3> (1,2) <6>
    (2,1) <4> (2,2) <8>
    (3,1) <8> (3,2) <3>
  END-DEFINE

- /* A barra significa pular uma linha
  /* '-'(22) => o traço será repetido 22x
  WRITE 'VALORES INICIAIS DE #A' / '-'(22)
  WRITE '=' #A(1,1) '=' #A(1,2) /
    '=' #A(2,1) '=' #A(2,2) /
    '=' #A(3,1) '=' #A(3,2)

  WRITE / 'VALORES INICIAIS DE #B' / '-'(22)
  WRITE '=' #B(1,1) '=' #B(1,2) /
    '=' #B(2,1) '=' #B(2,2) /
    '=' #B(3,1) '=' #B(3,2)

  /* SOMA DO ARRAY "A" COM O "B"
  ADD #A(*,*) TO #B(*,*)

  WRITE / 'VALORES FINAIS DE #B' / '-'(20)
  WRITE '=' #B(1,1) '=' #B(1,2) /
    '=' #B(2,1) '=' #B(2,2) /
    '=' #B(3,1) '=' #B(3,2)

  END
```

Natural Web I/O Output

Page 1

23-05-22 10:20:35

VALORES INICIAIS DE #A

#A: 1 #A: 1

#A: 2 #A: 0

#A: 3 #A: 2

VALORES INICIAIS DE #B

#B: 3 #B: 6

#B: 4 #B: 8

#B: 8 #B: 3

VALORES FINAIS DE #B

#B: 4 #B: 7

#B: 6 #B: 8

#B: 11 #B: 5

MORE

Exemplo 69

```
* >Natural Source Header 000000...
DEFINE DATA LOCAL
1 #A (N2/3,2) INIT
    (1,1) <1> (1,2) <1>
    (2,1) <2> (2,2) <0>
    (3,1) <3> (3,2) <2>
1 #B (N2/3,2) INIT
    (1,1) <3> (1,2) <6>
    (2,1) <4> (2,2) <8>
    (3,1) <8> (3,2) <3>
1 #C (N2) INIT <6>
END-DEFINE

WRITE 'VALORES INICIAL DE #C' / '-'(21)
WRITE '=' #C /

WRITE 'VALORES INICIAIS DE #A' / '-'(22)
WRITE '=' #A(1,1) '=' #A(1,2) /
    '=' #A(2,1) '=' #A(2,2) /
    '=' #A(3,1) '=' #A(3,2)

WRITE / 'VALORES INICIAIS DE #B' / '-'(22)
WRITE '=' #B(1,1) '=' #B(1,2) /
    '=' #B(2,1) '=' #B(2,2) /
    '=' #B(3,1) '=' #B(3,2)

/* SOMA 6 APENAS A PRIMEIRA LINHA DE B
ADD #C TO #B(1,*)

WRITE / 'VALORES FINAIS DE #B' / '-'(20)
WRITE '=' #B(1,1) '=' #B(1,2) /
    '=' #B(2,1) '=' #B(2,2) /
    '=' #B(3,1) '=' #B(3,2)

END
```


Natural Web I/O Output

Page 1

23-05-22 10:35:12

VALORES INICIAL DE #C

#C: 6

VALORES INICIAIS DE #A

#A: 1 #A: 1

#A: 2 #A: 0

#A: 3 #A: 2

VALORES INICIAIS DE #B

#B: 3 #B: 6

#B: 4 #B: 8

#B: 8 #B: 3

VALORES FINAIS DE #B

#B: 9 #B: 12

#B: 4 #B: 8

#B: 8 #B: 3

MORE

Exemplo 70

```
* >Natural Source Header 000000...
DEFINE DATA LOCAL
1 #A (N2/3,2) INIT
    (1,1) <1> (1,2) <1>
    (2,1) <2> (2,2) <0>
    (3,1) <3> (3,2) <2>
END-DEFINE

WRITE 'VALORES INICIAIS DE #A' / '-'(22)
WRITE '=' #A(1,1) '=' #A(1,2) /
      '=' #A(2,1) '=' #A(2,2) /
      '=' #A(3,1) '=' #A(3,2)

/* SOMA 2 A SEGUNDA COLUNA DE "A"
ADD 2 TO #A(*,2)

WRITE / 'VALORES FINAIS DE #A' / '-'(20)
WRITE '=' #A(1,1) '=' #A(1,2) /
      '=' #A(2,1) '=' #A(2,2) /
      '=' #A(3,1) '=' #A(3,2)
END
```

Natural Web I/O Output

Page 1

23-05-22 11:23:07

VALORES INICIAIS DE #A

#A: 1 #A: 1
#A: 2 #A: 0
#A: 3 #A: 2

VALORES FINAIS DE #A

#A: 1 #A: 3
#A: 2 #A: 2
#A: 3 #A: 4

Exemplo 71

```
* >Natural Source Header 000000[.]
DEFINE DATA LOCAL
1 #A (A15) INIT <'TX YZU'>
1 #B (A10) INIT <'XP01'>
1 #C (A10) INIT <'XP01'>
1 #CONT1 (N2)
1 #CONT2 (N2)
1 #CONT3 (N2)
END-DEFINE

/* EXAMINE => PESQUISA UMA STRING EM UM CAMPO (ALFA OU ARRAY)
/* GIVING NUMBER => OBTEM A QUANTIDADE DE VEZES QUE OCORREU
/* ESPAÇO EM BRANCO NA VARIÁVEL #A, SE NÃO FOR ENCONTRADO
/* RETORNA 0
EXAMINE #A FOR ' ' GIVING NUMBER #CONT1
WRITE 'QUANTIDADE DE ESPAÇO EM BRANCO SEM FULL:' #CONT1

/* EXAMINE FULL => PROCURA NA VARIÁVEL TODA DE ACORDO
/* COM SEU TAMANHO O ESPAÇO EM BRANCO
EXAMINE FULL #B FOR ' ' GIVING NUMBER #CONT2
WRITE 'QUANTIDADE DE ESPAÇO EM BRANCO COM FULL:' #CONT2

/* RETORNA 0 POIS NÃO ENCONTROU ESPAÇO EM BRANCO
EXAMINE #C FOR ' ' GIVING NUMBER #CONT3
WRITE 'QUANTIDADE DE ESPAÇO EM BRANCO SEM FULL:' #CONT3

END
```

Natural Web I/O Output

Page 1

23-05-23 10:13:05

QUANTIDADE DE ESPAÇO EM BRANCO SEM FULL: 1

QUANTIDADE DE ESPAÇO EM BRANCO COM FULL: 6

QUANTIDADE DE ESPAÇO EM BRANCO SEM FULL: 0

Exemplo 72

```
* >Natural Source Header 000000...
DEFINE DATA LOCAL
1 #A (A10) INIT<'XYZ'>
1 #B (A10) INIT<'XYZ'>
1 #C (A10) INIT<'XXYYZ'>
1 #D (A10) INIT<'XYZZK'>
END-DEFINE

/* EXAMINE DELETE => PESQUISA O VALOR NA VARIÁVEL
/* LOGO EM SEGUIDA O DELETA
WRITE '=' #A
EXAMINE #A FOR 'Y' DELETE
WRITE '#A DEPOIS DO EXAMINE / DELETE: ' #A

SKIP 1

/* EXAMINE REPLACE => PESQUISA O VALOR NA VARIÁVEL
/* LOGO EM SEGUIDA O SUBSTITUI PELO VALOR ESPECIFICADO
/* APOS O REPLACE
WRITE '=' #B
EXAMINE #B FOR 'Y' REPLACE '3'
WRITE '#B DEPOIS DO EXAMINE / REPLACE: ' #B

SKIP 1

/* REPLACE FIRST => APENAS A PRIMEIRA OCORRENCIA
/* ENCONTRADA SERÁ EXCLUÍDA OU SUBSTITUÍDA
WRITE '=' #C
EXAMINE #C FOR 'Y' REPLACE FIRST '*'
WRITE '#C DEPOIS DO EXAMINE / REPLACE: ' #C

END
```

Natural Web I/O Output

Page	1	23-05-23 11:04:20
------	---	-------------------


```
#A: XYZ
#A DEPOIS DO EXAMINE / DELETE: XZ

#B: XYZ
#B DEPOIS DO EXAMINE / REPLACE: X3Z

#C: XXYYZ
#C DEPOIS DO EXAMINE / REPLACE: XX*YZ
```

Exemplo 73

```
+ * >Natural Source Header 000000[.]
- DEFINE DATA LOCAL
  1 #A (A15) INIT<'XYZXYZXYZXYZXYZ'>
  1 #CONT (N2)
  END-DEFINE

  WRITE '=' #A

  SKIP 1

  /* GIVING NUMBER => fornece o nº de ocorrencias do valor
  /* pesquisado
  /* Aqui o Z será trocado por *
  EXAMINE #A FOR 'Z' REPLACE WITH '*' GIVING NUMBER #CONT

  WRITE '#A DEPOIS DO EXAMINE / REPLACE: ' #A //
    'NUMERO DE OCORRÊNCIA DE *: ' #CONT

  END
```

Natural Web I/O Output

Page	1	23-05-23 11:17:32
#A: XYZXYZXYZXYZXYZ		
#A DEPOIS DO EXAMINE / REPLACE: XY*XY*XY*XY*XY*		
NUMERO DE OCORRÊNCIA DE *: 5		

Exemplo 74

```
+ * >Natural Source Header 000000[.]
- DEFINE DATA LOCAL
1 #A (A20) INIT<'XYZXYZXYZXYZXYZXYZXY'>
1 #CONT (N2)
END-DEFINE

WRITE '=' #A

SKIP 1

/* GIVING POSITION => fornece a posicao da primeira
/* ocorrencia do valor pesquisado
/* Aqui o valor YZ será deletado da variavel A
/* e a primeira posicao será armazenada na variavel
/* #CONT
EXAMINE #A FOR 'YZ' DELETE GIVING POSITION #CONT

WRITE '#A DEPOIS DO EXAMINE / DELETE: ' #A //
      'POSIÇÃO DO PRIMEIRO YZ ANTES DO DELETE:' #CONT

END
```

Natural Web I/O Output

Page	1	23-05-23 11:28:58
------	---	-------------------


```
#A: XYZXYZXYZXYZXYZXYZXY

#A DEPOIS DO EXAMINE / DELETE: XXXXXXXXY

POSIÇÃO DO PRIMEIRO YZ ANTES DO DELETE: 2
```

Exemplo 75

```
DEFINE DATA LOCAL
1 #A (A10) INIT<'XYZX'>
1 #CONT (N2)
END-DEFINE

WRITE '=' #A

SKIP 1

/* GIVING LENGTH => fornece o tamanho do campo pesquisado
/* depois que todas as operacoes de REPLACE/DELETE forem
/* feitas
/* Aqui o "k*k" substituirá o "Z" e apos isso seu novo tamanho
/* será armazenado na variavel #CONT
EXAMINE #A FOR 'Z' REPLACE 'k*k' GIVING LENGTH #CONT

WRITE '#A DEPOIS DO EXAMINE / REPLACE: ' #A //
      'O NUMERO DE CARACTERES APOS O REPLACE: ' #CONT

END
```

Natural Web I/O Output

Page	1	23-05-23 11:52:41
------	---	-------------------

#A: XYZX

#A DEPOIS DO EXAMINE / REPLACE: XYk*kX

O NUMERO DE CARACTERES APOS O REPLACE: 6

Exemplo 76

```
* >Natural Source Header 000000
DEFINE DATA LOCAL
/* #TAB (A1/5,3,3) => matriz tridimensional com
/* 5 linhas, 3 colunas e 3 tabelas
1 #TAB (A1/5,3,3) INIT (1,2,3) <'A'> (5,1,2) <'B'>
1 #LINHA (N2)
1 #COLUNA (N2)
1 #TABELA (N2)
END-DEFINE

/* GIVING INDEX => localiza um valor em um ARRAY e
/* fornece o índice da primeira ocorrência
/* EXAMINE #TAB(*,*,*) => busca em todas as linhas, colunas
/* e tabelas
EXAMINE #TAB(*,*,*) FOR 'A' GIVING INDEX #LINHA #COLUNA #TABELA

WRITE 'A OCORRÊNCIA DO VALOR "A" é' //
      'LINHA.:' #LINHA /
      'COLUNA:' #COLUNA /
      'PLANO.:' #TABELA /

EXAMINE #TAB(*,*,*) FOR 'B' GIVING INDEX #LINHA #COLUNA #TABELA

WRITE 'A OCORRÊNCIA DO VALOR "B" é' //
      'LINHA.:' #LINHA /
      'COLUNA:' #COLUNA /
      'PLANO.:' #TABELA /

END
```

Natural Web I/O Output

Page 1

23-05-24 10:34:47

A OCORRÊNCIA DO VALOR 'A' é

LINHA.: 1

COLUNA: 2

PLANO.: 3

A OCORRÊNCIA DO VALOR 'B' é

LINHA.: 5

COLUNA: 1

PLANO.: 2

Exemplo 77

```
* >Natural Source Header 000000
DEFINE DATA LOCAL
1 #MARCA (A10/5) INIT <'FIAT','CITROEN','VOLVO','AUDI','FORD'>
1 #VEICULO (A15/5) INIT <'PALIO','PICASSO','REVOLUTION','A6','KA'>

1 #MARCA_AUX (A10)
1 #INDICE (N1)
END-DEFINE

#MARCA_AUX := 'AUDI'

/* AQUI É PESQUISADO O VALOR "AUDI" QUE ESTA NA VARIABEL
/* #MARCA_AUX
/* A PESQUISA É FEITA DENTRO DO VETOR #MARCA
/* O INDICE DO VALOR PESQUISADO É ARMAZENADO NA VARIABEL
/* #INDICE
EXAMINE #MARCA(*) #MARCA_AUX GIVING INDEX #INDICE

DISPLAY #MARCA(*) #VEICULO(*) #MARCA_AUX #INDICE #MARCA(#INDICE)
#VEICULO(#INDICE)

END
```

Natural Web I/O Output

#MARCA	#VEICULO	#MARCA_AUX	#INDICE	#MARCA	#VEICULO
FIAT	PALIO	AUDI	4	AUDI	A6
CITROEN	PICASSO				
VOLVO	REVOLUTION				
AUDI	A6				
FORD	KA				

Exemplo 78

```
+ * >Natural Source Header 000000[.]
- DEFINE DATA LOCAL
1 #D (A26) INIT <'TESTECOMANDOEXAMINENOCURSO'>
1 #NUMBER (N2)
1 #POSICAO (N2)
END-DEFINE

- /* EXAMINE SUBSTRING => permite que uma porcao do campo
/* seja examinada
/* SUBSTRING(#D,13,7) => primeiro a variavel, seguido da
/* primeira posicao a ser examinada e depois o tamanho
/* do campo a ser pesquisado
/* Neste exemplo a pesquisa será feita em EXAMINE
/* O Nº DE OCORRENCIAS É ARMAZENADO NA VARIÁVEL #NUMBER
/* A POSICAO DO VALOR A SER PESQUISADO É ARMAZENADO NA
/* VARIÁVEL #POSICAO
EXAMINE SUBSTRING(#D,13,7) 'MINE' GIVING #NUMBER POSITION #POSICAO

WRITE #D / '=' #NUMBER / '=' #POSICAO

END
```

Natural Web I/O Output

Page	1	23-05-24 11:29:24
TESTECOMANDOEXAMINENOCURSO		
#NUMBER:	1	
#POSICAO:	4	

Exemplo 79

```
* >Natural Source Header 000000..
DEFINE DATA LOCAL
1 #NOME (A15/8) INIT <'LUCIANO','LUIZA','ANA','LUCIA',
                     'PERDIGAO','ROSA','MONICA','LUCAS'>
1 #NUMBER (N2)
END-DEFINE

/* A PESQUISA SERÁ FEITA NO VETOR #NOME
/* EM CADA VALOR A PESQUISA INICIARÁ NA POSICAO 1
/* O TAMANHO A SER PESQUISADO SERÁ DE 3 BYTES, OU
/* SEJA, APENAS NAS 3 PRIMEIRAS LETRAS DOS NOMES
/* O Nº DE OCORRENCIAS DO VALOR PESQUISADO SERÁ
/* ARMAZENADO NA VARIÁVEL #NUMBER
EXAMINE SUBSTRING(#NOME(*),1,3) 'LUC' GIVING #NUMBER

WRITE #NOME(*) // '=' #NUMBER|

END
```

Natural Web I/O Output

Page	1			23-05-24 11:43:15
LUCIANO	LUIZA	ANA	LUCIA	PERDIGAO
ROSA	MONICA	LUCAS		
#NUMBER:	3			

Exemplo 80

```
+ * >Natural Source Header 000000[.]
- DEFINE DATA LOCAL
  1 #TAB (A10/4) INIT<'CARRO','CARROÇA','GARRAFA','CAVALO'>
  1 #I (N3)
  1 #N (N3)
  END-DEFINE

- /* PATTERN => pesquisa um campo excluindo as posicoes
  /* selecionadas

- /* NOS EXEMPLOS ABAIXO O Nº DE OCORRENCIAS DO VALOR
  /* PESQUISADO SERÁ ARMAZENADO NA VARIÁVEL #N

- /* ponto(.) interrogacao(?) e underline(_) representam
  /* uma posicao a ser ignorada
  WRITE 'PESQUISANDO COM APENAS UM CAMPO ENTRE O "A" E O "O"'
- FOR #I 1 4
    EXAMINE #TAB(#I) PATTERN 'A.O' GIVING NUMBER #N
    WRITE #TAB(#I) '-' #N
  END-FOR

  SKIP 1

- /* asterisco(*) e percentual(%) representam uma posicao
  /* ou mais a serem ignoradas
  WRITE 'PESQUISANDO COM UM OU MAIS CAMPOS ENTRE O "A" E O "O"'
- FOR #I 1 4
    EXAMINE #TAB(#I) PATTERN 'A*O' GIVING NUMBER #N
    WRITE #TAB(#I) '-' #N
  END-FOR

  END
```

Natural Web I/O Output

Page 1

23-05-24 15:24:02

PESQUISANDO COM APENAS UM CAMPO ENTRE O 'A' E O 'O'

CARRO	-	0
CARROÇA	-	0
GARRAFA	-	0
CAVALO	-	1

PESQUISANDO COM UM OU MAIS CAMPOS ENTRE O 'A' E O 'O'

CARRO	-	1
CARROÇA	-	1
GARRAFA	-	0
CAVALO	-	1

Exemplo 81

```
+ * >Natural Source Header 000000[...]  
- DEFINE DATA LOCAL  
  1 #T (A40) INIT<' -A-BC A B C .A. .B. .C. -A- -B-AB'>  
  1 #P1 (N3)  
  1 #P2 (N3)  
  1 #P3 (N3)  
  END-DEFINE  
  
- /* AQUI O "A" É PESQUISADO NA VARIÁVEL #T E O  
  /* Nº DE OCORRENCIAS É ARMAZENADO NA VARIÁVEL  
  /* #P1  
  EXAMINE #T FOR 'A' GIVING NUMBER #P1  
  
- /* WITH DELIMITERS => USADO P/ UM VALOR QUE ESTÁ  
  /* ENTRE ESPAÇO EM BRANCO OU QUALQUER OUTRO QUE NAO  
  /* SEJA LETRA OU NUMERO  
  EXAMINE #T FOR 'A' WITH DELIMITERS GIVING NUMBER #P2  
  
- /* AQUI SERÁ PESQUISADO O VALOR "A" QUE SE ENCONTRA  
  /* ENTRE TRAÇOS(-)  
  EXAMINE #T FOR 'A' WITH DELIMITERS '-' GIVING NUMBER #P3  
  
  WRITE '=' #P1 / '=' #P2 / '=' #P3  
  
  END
```

Natural Web I/O Output

Page 1

23-05-24 16:00:03

#P1: 5

#P2: 4

#P3: 2

Exemplo 82

```
+ * >Natural Source Header 000000
- DEFINE DATA LOCAL
1 #T (A40) INIT<' -A-BC A B C .A. .B. .C. -A- -B-AB'>
1 #Q (A40) INIT<' -* -@# * @ # .* .@. .#. -* -@-*@'>
1 #T1 (A2/3) INIT<'A*', 'B@', 'C#'>
1 #NOME (A10) INIT<'JOCA TIGRE'>
END-DEFINE

- /* TRANSLATE => TRANSFORMA CARACTERES DE UM CAMPO
/* P/ MAIUSCULO, P/ MINUSCULO OU P/ OUTRO CARACTER
/* ESPECIFICADO

- /* AQUI A VARIABEL #T SERÁ EXAMINADA OS VALORES SERAO
/* TRANSFORMADOS DE ACORDO COM O VETOR #T1
/* DE ACORDO COM O VETOR #T1, O "A" SERÁ TROCADO
/* POR ASTERISCO(*), O "B" POR ARROBA(@) E
/* ASSIM SUCESSIVAMENTE
WRITE '#T ANTES DO 1º EXAMINE: ' #T
EXAMINE #T TRANSLATE USING #T1(*)
WRITE '#T DEPOIS DO 1º EXAMINE: ' #T

SKIP 1

- /* AQUI A VARIABEL #Q SOFRERÁ A TRANSFORMACAO
/* INVERTENDO, ONDE TIVER (*) (@) (#) SERÁ A, B E C
WRITE '#Q ANTES DO 2º EXAMINE: ' #Q
EXAMINE #Q TRANSLATE USING INVERTED #T1(*)
WRITE '#Q DEPOIS DO 2º EXAMINE: ' #Q

SKIP 1

- /* TRANSLATE INTO LOWER CASE TRANSFORMA A STRING
/* P/ MINUSCULO
WRITE '#NOME ANTES DO 3º EXAMINE: ' #NOME
EXAMINE #NOME TRANSLATE INTO LOWER CASE
WRITE '#NOME DEPOIS DO 3º EXAMINE: ' #NOME

END
```

Natural Web I/O Output

```
Page      1                                     23-05-25  10:44:20

#T ANTES DO 1º EXAMINE: -A-BC A B C .A. .B. .C. -A- -B-AB
#T DEPOIS DO 1º EXAMINE: -*-@# * @ # .*. .@. .#. -* -@-*@

#Q ANTES DO 2º EXAMINE: -*-@# * @ # .*. .@. .#. -* -@-*@
#Q DEPOIS DO 2º EXAMINE: -A-BC A B C .A. .B. .C. -A- -B-AB

#NOME ANTES DO 3º EXAMINE: JOCA TIGRE
#NOME DEPOIS DO 3º EXAMINE: joca tigre
```

Exemplo 83

```
+ * >Natural Source Header 000000[ ]
- DEFINE DATA LOCAL
1 #A (A30) INIT<'XXXXXXXXXXXXXXXX.YYYY.ZZZZZZZZ'>
1 #A1 (A15)
1 #A2 (A15)
1 #A3 (A15)
END-DEFINE

- /* SEPARATE => SEPARA O CONTEUDO DE UM CAMPO
/* ALFANUMERICO EM 2 OU MAIS CAMPOS ALFANUMERICOS
/* OU OCORRENCIAS DE UM ARRAY ALFANUMERICO

WRITE '=' #A

SKIP 1

- /* COLOCA TODOS OS "X" NA VARIABEL #A1
/* TODOS OS "Y" NA VARIABEL #A2
/* TODOS OS "Z" NA VARIABEL #A3
SEPARATE #A INTO #A1 #A2 #A3

WRITE '=' #A1 / '=' #A2 / '=' #A3

END
```

Natural Web I/O Output

Page 1

23-05-25 11:08:03

#A: XXXXXXXXXXXXXXXX.YYYY.ZZZZZZZZZ

#A1: XXXXXXXXXXXXXXXX

#A2: YYYY

#A3: ZZZZZZZZZ

Exemplo 84

```
+ * >Natural Source Header 000000...
- DEFINE DATA LOCAL
1 #A (A32) INIT<'1111111111 222'>
1 #A1 (A10) INIT<'JOCA'>
1 #A2 (A10) INIT<'JOCA'>
1 #A3 (A10) INIT<'JOCA'>
END-DEFINE

- /* NESSE EXEMPLO TEMOS 3 VARIÁVEIS QUE RECEBERÃO
/* OS VALORES DA VARIÁVEL #A, PORÉM NA VARIÁVEL
/* TEMOS APENAS 2 GRUPOS DE VALORES (1 e 2)
/* O RESULTADO DISSO SERÁ QUE O VALOR DA 3ª VARIÁVEL
/* SERÁ ANULADA PELO NATURAL

WRITE 'ANTES DO SEPARATE'
WRITE '=' #A / '=' #A1 / '=' #A2 / '=' #A3

SEPARATE #A INTO #A1 #A2 #A3

SKIP 1

WRITE 'APÓS O SEPARATE'
WRITE '=' #A1 / '=' #A2 / '=' #A3

END
```


Natural Web I/O Output

```
Page      1                                     23-05-25  11:20:41

ANTES DO SEPARATE
#A: 1111111111 222
#A1: JOCA
#A2: JOCA
#A3: JOCA

APÓS O SEPARATE
#A1: 1111111111
#A2: 222
#A3:
```

Exemplo 85

```
* >Natural Source Header 000000...
DEFINE DATA LOCAL
1 #A (A20) INIT<'111..22222'>
1 #A1 (A20)
1 #A2 (A20)
1 #A3 (A20)
END-DEFINE

/* AQUI TEMOS 2 DELIMITADORES CONSECUTIVOS
/* OS DELIMITADORES SE TORNAM ESPAÇOS EM BRANCO

SEPARATE #A INTO #A1 #A2 #A3

WRITE '=' #A / '=' #A1 / '=' #A2 / '=' #A3

END
```

Natural Web I/O Output

```
Page      1                                     23-05-25  11:35:18

#A: 111..22222
#A1: 111
#A2:
#A3: 22222
```

Exemplo 86

```
+ * >Natural Source Header 000000[.]
- DEFINE DATA LOCAL
  1 #A (A20) INIT<'JOCA TIGRE'>
  1 #ARRAY (A10/2)
  END-DEFINE

- /* SEPARA O CONTEUDO DA VARIÁVEL #A PARA
  /* DENTRO DO ARRAY
  SEPARATE #A INTO #ARRAY(*)

  WRITE '=' #A / '=' #ARRAY(*)

  END
```

Natural Web I/O Output

Page	1	23-05-25 11:40:39
#A:	JOCA TIGRE	
#ARRAY:	JOCA	TIGRE

Exemplo 87

```
+ * >Natural Source Header 000000[.]
- DEFINE DATA LOCAL
  1 #A (A32) INIT<'111111111@222222222@CCCCCCCCC'>
  1 #A1 (A10)
  1 #A2 (A10)
  1 #A3 (A10)
  END-DEFINE

- /* SEPARATE WITH DELIMITERS => separa o conteudo da
  /* variavel especificando um delimitador
  SEPARATE #A INTO #A1 #A2 #A3 WITH DELIMITERS '@'

  WRITE '=' #A // '=' #A1 / '=' #A2 / '=' #A3

  END
```

Natural Web I/O Output

Page 1

23-05-25 11:47:25

#A: 1111111111@2222222222@CCCCCCCCC

#A1: 1111111111

#A2: 2222222222

#A3: CCCCCCCCC

Exemplo 88

```
* >Natural Source Header 000000[.]
DEFINE DATA LOCAL
1 #A (A32) INIT<'1111111111.22233.44555'>
1 #A1 (A10) INIT<'LUCIANO'>
1 #A2 (A10) INIT<'LUCIANO'>
1 #A3 (A10) INIT<'LUCIANO'>
1 #A4 (A10) INIT<'LUCIANO'>
1 #CONT (N3)
END-DEFINE

/* GIVING NUMBER fornece o nº de ocorrências
/* de campos diferentes de brancos
WRITE 'ANTES DO SEPARATE'
WRITE '=' #A / '=' #A1 / '=' #A2 / '=' #A3 / '=' #A4
SEPARATE #A INTO #A1 #A2 #A3 #A4 GIVING NUMBER #CONT

SKIP 1

WRITE 'APOS O SEPARATE'
WRITE '=' #A1 / '=' #A2 / '=' #A3 / '=' #A4 //
      'NÚMEROS DE CAMPOS RESULTANTES:' #CONT

END
```

Natural Web I/O Output

Page 1

23-05-25 15:28:37

ANTES DO SEPARATE

#A: 1111111111.22233.44555

#A1: LUCIANO

#A2: LUCIANO

#A3: LUCIANO

#A4: LUCIANO

APOS O SEPARATE

#A1: 1111111111

#A2: 22233

#A3: 44555

#A4:

NÚMEROS DE CAMPOS RESULTANTES: 3

Exemplo 89

```
* >Natural Source Header 000000[ ]
- DEFINE DATA LOCAL
1 #A (A30) INIT<'111. 1111. 22233'>
1 #A1 (A20)
1 #A2 (A20)
1 #A3 (A20)
END-DEFINE

- /* AQUI NO WRITE TEREMOS OS GRUPOS DE VALORES
/* MOSTRADOS ENTRE OS PONTOS(.)
/* NA VARIABEL #A1 TEREMOS '111'
/* NA VARIABEL #A2 TEREMOS ' 1111'
/* NA VARIABEL #A3 TEREMOS ' 22233'
/* NOTE QUE TEMOS ESPAÇOS EM BRANCO
SEPARATE #A INTO #A1 #A2 #A3 WITH DELIMITERS ' .'
WRITE 'ALINHAMENTO SEM LEFT JUSTIFIED:' /
#A1 '#A1' / #A2 '#A2' / #A3 '#A3' //

/* LEFT JUSTIFIED => remove brancos a esquerda do campo
SEPARATE #A LEFT JUSTIFIED INTO #A1 #A2 #A3 WITH DELIMITERS ' .'
WRITE 'ALINHAMENTO COM LEFT JUSTIFIED:' /
#A1 '#A1' / #A2 '#A2' / #A3 '#A3' //

END
```

Natural Web I/O Output

Page 1

23-05-25 15:41:56

ALINHAMENTO SEM LEFT JUSTIFIED:

```
111          #A1
1111         #A2
22233        #A3
```

ALINHAMENTO COM LEFT JUSTIFIED:

```
111          #A1
1111         #A2
22233        #A3
```

Exemplo 90

```
+ * >Natural Source Header 000000
- DEFINE DATA LOCAL
1 #A (A20) INIT<'11111.22222.33333'>
1 #A1 (A5)
1 #A2 (A5)
1 #A3 (A5)
END-DEFINE

- /* SEPARATE SUBSTRING => é realizado um separate em
/* parte de um campo
/* Depois da variavel, especifica-se a posicao inicial
/* e entao o tamanho da porção a se pesquisar
/* Nesse exemplo será pesquisado na porção '11.222'
SEPARATE SUBSTRING(#A,4,6) INTO #A1 #A2 #A3

WRITE '=' #A // '=' #A1 / '=' #A2 / '=' #A3

END
```

Natural Web I/O Output

Page 1

23-05-25 15:54:42

#A: 11111.22222.33333

#A1: 11

#A2: 222

#A3:

Exemplo 91

```
* >Natural Source Header 000000[.]
DEFINE DATA LOCAL
1 #A (A20) INIT<'11111.22222.33333'>
1 #A1 (A5)
1 #A2 (A5)
END-DEFINE

/* SEPARATE IGNORE => se nao houver campos
/* suficientes, o Natural irá ignorar
/* O restante do valor da variavel #A('33333')
/* nao será armazenada em nenhuma variavel
SEPARATE #A INTO #A1 #A2 IGNORE

WRITE '=' #A // '=' #A1 / '=' #A2

END
```

Natural Web I/O Output

Page	1	23-05-25 16:09:43
#A: 11111.22222.33333		
#A1: 11111		
#A2: 22222		

Exemplo 92

```
* >Natural Source Header 000000[.]
DEFINE DATA LOCAL
1 #A (A20) INIT<'RRRRR.OOOOO.AAAAA'>
1 #A1 (A5)
1 #A2 (A5)
1 #R (A5)
END-DEFINE

/* SEPARATE REMAINDER => a porção que nao for
/* armazenada nas variaveis será carregada na
/* variavel #R apos o remainder
SEPARATE #A INTO #A1 #A2 REMAINDER #R

WRITE '=' #A // '=' #A1 / '=' #A2 / '=' #R

END
```

Natural Web I/O Output

```
Page      1                                     23-05-25  16:20:30

#A: RRRRR.00000.AAAAA

#A1: RRRRR
#A2: 00000
#R: AAAAA
```

Exemplo 93

```
* >Natural Source Header 000000...
DEFINE DATA LOCAL
1 #A (A20) INIT<'RRRRR.00000.AAAAA'>
1 #A1 (A5)
1 #A2 (A5)
1 #A3 (A5)
1 #A4 (A5)
1 #A5 (A5)
END-DEFINE

/* Normalmente os caracteres delimitadores não são
/* carregados nos campos receptores do SEPARATE
/* Com a opção WITH RETAINED DELIMITERS, o delimitador
/* será também carregado no campo receptor
SEPARATE #A INTO #A1 #A2 #A3 #A4 #A5 IGNORE WITH RETAINED DELIMITERS

WRITE '=' #A // '=' #A1 / '=' #A2 / '=' #A3 / '=' #A4 / '=' #A5

END
```

Natural Web I/O Output

```
Page      1                                     23-06-06  14:14:05

#A: RRRRR.00000.AAAAA

#A1: RRRRR
#A2: .
#A3: 00000
#A4: .
#A5: AAAAA
```

Exemplo 94

```
* >Natural Source Header 000000[...]  
- DEFINE DATA LOCAL  
1 #A (A40) INIT<'Curso de Natural e Adabas - Dia'>  
1 #B (N2) INIT<10>  
1 #C (A50)  
END-DEFINE  
  
- /* A instrução COMPRESS serve para concatenar o conteúdo  
/* de dois ou mais campos em um outro campo. Os operandos  
/* a serem combinados podem ser de qualquer formato, porém  
/* o campo resultante deve ser alfanumérico. Pode-se  
/* utilizar as variáveis de sistema, como *NUMBER ou *COUNTER,  
/* diretamente no comando COMPRESS  
  
COMPRESS #A #B TO #C LEAVING NO  
  
WRITE '=' #A / '=' #B // '=' #C  
  
END
```

Natural Web I/O Output

```
Page      1                                     23-06-06 14:30:29  
  
#A: Curso de Natural e Adabas - Dia  
#B: 10  
  
#C: Curso de Natural e Adabas - Dia10
```


Exemplo 95

```
+ * >Natural Source Header 000000[...]  
- DEFINE DATA LOCAL  
1 #NOME (A8) INIT<'Joca'>  
1 #SOBRENOME (A8) INIT<'Tigre'>  
1 #NOME_COMPLETO (A30)  
END-DEFINE  
  
/* PM=I => inverte os dados  
  
COMPRESS #NOME(PM=I) #SOBRENOME INTO #NOME_COMPLETO  
  
WRITE '=' #NOME / '=' #SOBRENOME // '=' #NOME_COMPLETO  
  
END
```

Natural Web I/O Output

```
Page      1                                     23-06-06  14:52:03  
  
#NOME: Joca  
#SOBRENOME: Tigre  
  
#NOME_COMPLETO: acoJ Tigre
```

Exemplo 96

```
* >Natural Source Header 000000
DEFINE DATA LOCAL
  1 #DIA (N2) INIT<10>
  1 #MES (N2) INIT<02>
  1 #ANO (N4) INIT<2004>
  1 #DATA (A10)
END-DEFINE

/* DELIMITER(S) permite especificar um caracter delimitador
/* entre cada valor no campo resultante
COMPRESS #DIA #MES #ANO INTO #DATA WITH DELIMITERS '/'
WRITE '=' #DIA / '=' #MES / '=' #ANO / '=' #DATA

skip 1

/* Se não for especificada a opção DELIMITERS, é colocado um
/* espaço em branco entre os valores
COMPRESS #DIA #MES #ANO INTO #DATA
WRITE '=' #DIA / '=' #MES / '=' #ANO / '=' #DATA

skip 1

/* Se for especificado DELIMITERS sem caracter algum, será
/* assumido o caracter do Natural, cujo valor default é ","
COMPRESS #DIA #MES #ANO INTO #DATA WITH DELIMITERS
WRITE '=' #DIA / '=' #MES / '=' #ANO / '=' #DATA

skip 1

/* Se um dos operandos do comando COMPRESS for numérico e
/* tiver seu valor zerado, o zero aparecerá no campo resultante
#DIA := 0
COMPRESS #DIA #MES #ANO INTO #DATA WITH DELIMITERS
WRITE '=' #DIA / '=' #MES / '=' #ANO / '=' #DATA

END
```

Natural Web I/O Output

```
Page      1                                     23-06-06  15:16:18

#DIA:  10
#MES:   2
#ANO:  2004
#DATA: 10/2/2004

#DIA:  10
#MES:   2
#ANO:  2004
#DATA: 10 2 2004

#DIA:  10
#MES:   2
#ANO:  2004
#DATA: 10,2,2004

#DIA:   0
#MES:   2
#ANO:  2004
#DATA: 0,2,2004
```

Exemplo 97

```
* >Natural Source Header 000000...
DEFINE DATA LOCAL
  1 #DIA (N2) INIT<10>
  1 #MES (N2) INIT<02>
  1 #ANO (N4) INIT<2004>
  1 #DATA (A10)
END-DEFINE

/* LEAVING NO SPACE serve para suprimir o espaço
/* em branco entre dois ou mais campos concatenados

COMPRESS #DIA #MES #ANO INTO #DATA
WRITE '=' #DIA / '=' #MES / '=' #ANO / '=' #DATA

skip 1

COMPRESS #DIA #MES #ANO INTO #DATA LEAVING NO
WRITE '=' #DIA / '=' #MES / '=' #ANO / '=' #DATA

END
```

Natural Web I/O Output

```
Page      1                                     23-06-06  15:26:26

#DIA:  10
#MES:   2
#ANO:  2004
#DATA: 10 2 2004

#DIA:  10
#MES:   2
#ANO:  2004
#DATA: 1022004
```

Exemplo 98

```
+ * >Natural Source Header 000000[.]
- DEFINE DATA LOCAL
1 #NUM1 (N3.2) INIT<-123.45>
1 #NUM2 (N2.2) INIT<-9.1>
1 #NUM3 (A12)
END-DEFINE

- /* NUMERIC inclui os caracteres de sinal e
/* decimal na concatenação de valores

COMPRESS #NUM1 #NUM2 INTO #NUM3 WITH DELIMITER ' '
WRITE 'COMPRESS SEM NUMERIC:' #NUM3

SKIP 1

COMPRESS NUMERIC #NUM1 #NUM2 INTO #NUM3 WITH DELIMITER ' '
WRITE 'COMPRESS COM NUMERIC:' #NUM3

END
```

Natural Web I/O Output

```
Page      1                                     23-06-07  10:19:14

COMPRESS SEM NUMERIC: 12345 910

COMPRESS COM NUMERIC: -123.45 -9.1
```

Exemplo 99

```
+ * >Natural Source Header 000000[...]  
- DEFINE DATA LOCAL  
  1 #S (A79)  
  END-DEFINE  
  
  MOVE 'Frase concatenada no' - ' programa Natural' TO #S  
  
  WRITE #S  
  
  END
```

Natural Web I/O Output

Page 1

23-06-07 10:23:51

Frase concatenada no programa Natural

Exemplo 100

```
+ * >Natural Source Header 000000[...]  
  WRITE 'VARIAVEIS DATE'  
  
  SKIP 1  
  
  WRITE '*DATD - DD.MM.YY - ' *DATD  
  WRITE '*DATE - DD/MM/YY - ' *DATE  
  WRITE '*DATI - YY-MM-DD - ' *DATI  
  WRITE '*DATJ - YY??? - ' *DATJ  
  WRITE '*DATU - MM/DD/YY - ' *DATU  
  WRITE '*DATG - DDnomeYYYY - ' *DATG  
  WRITE '*DATN - YYYYMMDD - ' *DATN  
  WRITE '*DATX - YY-MM-DD - ' *DATX  
  WRITE '*DAT4D - DD.MM.YYYY - ' *DAT4D  
  WRITE '*DAT4E - DD/MM/YYYY - ' *DAT4E  
  WRITE '*DAT4I - YYYY-MM-DD - ' *DAT4I  
  WRITE '*DAT4J - YYYY??? - ' *DAT4J  
  WRITE '*DAT4U - MM/DD/YYYY - ' *DAT4U  
  
  END
```

Natural Web I/O Output

Page 1

23-06-07 13:12:25

VARIAVEIS DATE

```
*DATD - DD.MM.YY - 07.06.23
*DATE - DD/MM/YY - 07/06/23
*DATI - YY-MM-DD - 23-06-07
*DATJ - YY??? - 23158
*DATU - MM/DD/YY - 06/07/23
*DATG - DDnomeYYYY - 07June 2023
*DASN - YYYYMMDD - 20230607
*DATX - YY-MM-DD - 23-06-07
*DAT4D - DD.MM.YYYY - 07.06.2023
*DAT4E - DD/MM/YYYY - 07/06/2023
*DAT4I - YYYY-MM-DD - 2023-06-07
*DAT4J - YYYY??? - 2023158
*DAT4U - MM/DD/YYYY - 06/07/2023
```

Exemplo 101

```
* >Natural Source Header 000000
WRITE 'VARIAVEIS TIME'

SKIP 1

/* Formato HH:MM:SS:T
WRITE '*TIME - ' *TIME

/* Valor do relógio interno da máquina
WRITE '*TIMESTAMP - ' *TIMESTAMP

/* Contém o nº de segundos restantes da transação
/* corrente em execução (somente c/ Natural Security)
WRITE '*TIME-OUT - ' *TIME-OUT

/* Formato HHMMSSST
WRITE '*TIMN - ' *TIMN

/* Formato interno
WRITE '*TIMX - ' *TIMX

/* *TIMD => Tempo decorrido durante um
/* comando SETTIME (HHMMSSST)

END
```

Natural Web I/O Output

Page 1

23-06-07 13:27:13

VARIAVEIS TIME

```
*TIME      - 13:27:13.8
*TIMESTAMP - DD69DBB10BA10000
*TIME-OUT  - 0
*TIMN      - 1327138
*TIMX      - 13:27:13
```

Exemplo 102

```
+ * >Natural Source Header 000000[.]
- DEFINE DATA LOCAL
  1 #DATA (D) INIT<*DATX>
  1 #HORA (T) INIT<*TIMX>
  END-DEFINE

- /* NUMEROS PODEM SER ADICIONADOS OU SUBTRAIDOS DE
  /* VARIAVEIS DE FORMATO (D)
  /* OS Nº ADICIONADOS EM (D) SAO DIAS

  WRITE 'DATA DE HOJE:' *DATX(EM=DD/MM/YYYY)
  ADD 5 TO #DATA
  WRITE 'DATA DE HOJE + 5 DIAS:' #DATA(EM=DD/MM/YYYY)

  SKIP 1

- /* OS Nº ADICIONADOS EM VARIAVEIS DE FORMATO (T)
  /* SAO DECIMOS DE SEGUNDO
  /* DECIMO DE SEGUNDO: EQUIVALE A 0,1 SEGUNDO
  /* 600 DECIMOS DE SEGUNDOS É IGUAL A 60 SEGUNDOS (1 MIN)

  WRITE 'HORA ATUAL:' *TIMX
  ADD 600 TO #HORA
  WRITE 'HORA ATUAL + 60 SEGUNDOS:' #HORA

  END
```

Natural Web I/O Output

Page 1

23-06-07 13:46:40

DATA DE HOJE: 07/06/2023

DATA DE HOJE + 5 DIAS: 12/06/2023

HORA ATUAL: 13:46:40

HORA ATUAL + 60 SEGUNDOS: 13:47:40

Exemplo 103

asdasd