```
newinstance "myCpt", "myCpt" ;Cria nova instância do componente myCpt
activate "myCpt".exec() ;Chama a operação exec do componente "myCpt"

;Operatores matemáticos
*       ;Multiplication
/       ;Division
%       ;Modulo
+       ;Addition
-       ;Subtraction

clear/e "CUSTOMER.SALES" ;Limpa os registros da entidade CUSTOMER.SALES

;Exemplo de condicional if simples
if (myVar1 > myVar2) ;Abertura do bloco if e aplicação da condição
 ;Seu código aqui
elseif ($1 > "M") ;Abertura do bloco elseif e aplicação da condição
 ;Seu código aqui
else ;Abertura do bloco else
 ;Seu código aqui
endif ;Fechamento do bloco if/else

;Exemplo de condicional switch/case ou selectcase simples
selectcase vVariable ;Abertura do bloco selectcase e definição da variável a ser
avaliada
 case "" ;Caso em que a variável é vazia
    ;Seu código aqui
 case "ABC" ;Caso em que a variável é igual a "ABC"
    ;Seu código aqui
 case "abc" ;a string ;Caso em que a variável é igual a "abc"
    ;Seu código aqui
 case 123 ;Caso em que a variável é igual a 123
    ;Seu código aqui
 elsecase ;Caso em que a variável não se encaixa em nenhum dos casos anteriores
    ;Seu código aqui
endselectcase ;Fechamento do bloco selectcase

;Tipos de dados disponíveis para variáveis e parâmetros de entrada e saída
boolean          ;T or F, 1 or 0
date             ;Integer representing a date in the format ccyymmdd
datetime         ;Integer representing a date and time in the format
ccyymmddhhnnss  Combined date and Time.
float            ;Number with a floating decimal point  Floating decimal point
to allow high precision data and calculations.
image            ;Binary data for an image
lineardate       ;Integer from 0 through 3652425        Linear Date represents a
number of days.
lineardatetime   ;Number from 0 through 10000   Linear Date and Time represents
a number of days. Partial days (hours, minutes, seconds) can be expressed as a
fraction of a day.
lineartime       ;Integer from 0 through 24     Linear Time represents a number
of hours.
numeric          ;Number to a maximum of 9 decimal places, including +, -, and
decimal point .
raw              ;Binary data
```

```
string           ;Any characters in the UTF-8 character set, or an empty string.
time             ;Integer representing a date in the format hhnnss
xmlstream        ;Well-formed XML data
entity           ;All occurrences of the specified entity       entity and
occurrence are
occurrence       ;Current occurrence of the specified entity
handle           ;Reference to a component instance
struct           ;Reference to a Struct used for complex, dynamic data

;Declare um ProcScript entry, um módulo de script que pode ser invocado dentro
do mesmo componente

entry doSomething ;Inicializa o script local no coponente
 ;Código da operação
end ;Fecha o script local no coponente

;Pode ser chamado no próprio componente como uma função
vResult = doSomething()

;Pode ser chamado no prório componente usando o call
call doSomething()



;Exemplo de aplicação de um loop for
variables ; Abertura de variáveis
 numeric vCounter ; Definição de variável contador
 numeric vLoops ; Definição de variável contadora de loops
endvariables ; Fechamento de variáveis

vLoops = 0 ;Insntancia da variável contadora de loops
for vCounter = 100 to 0 step -2 ; Abertura do loop for com defínção de contador
das repetições
 vLoops += 1 ; Incrementa a variável contadora de loops
 putmess "Counter: %%vCounter, Loop count: %%vLoops " ; Imprime o valor do
contador e o número de loops
 if (vLoops >= 6) ; Abertura de condicional if
    ; Se o número de loops for maior ou igual a 6, o loop é interrompido
    ; e uma mensagem é impressa na tela.
    putmess "Loop processing stopped"
 break ; Interrompe o loop
  ; Se o número de loops for menor que 6, o loop continua.
 else ; Abertura do bloco else
    putmess "Loop processing continues" ; Imprime mensagem de continuação do
loop
 endif ; Fechamento do bloco if
endfor ; Fechamento do loop for

; Exemplo de aplicação de um loop for com entidade
variables ; Abertura de variáveis
 numeric vLoops ; Definição de variável contadora de loops
 string vFullName ; Definição de variável para armazenar o nome completo
endvariables ; Fechamento de variáveis
```

```
retrieve/e "PERSON.ORG" ; Recupera registros da entidade "PERSON.ORG" para
processamento
vLoops = 0 ; Instancia a variável contadora de loops

forentity "PERSON.ORG" ; Abertura do loop for com entidade
 vLoops += 1 ;Incrementa a variável contadora de loops
 vFullName = FULLNAME.PERSON.ORG ; Armazena o nome completo na variável
vFullName
 if (vFullName = "Donald Duck") ; Abertura de condicional if
    ; Se o nome completo for "Donald Duck", o loop é interrompido
    ; e uma mensagem é impressa na tela.
    putmess "Loop processing stopped on Name: %%vFullName Loop count: %%vLoops"
;Imprime mensagem de parada do loop
    break ; Interrompe o loop
 endif ; Fechamento do bloco if
 putmess "Processing %%vFullName, Loop count: %%vLoops " ; Imprime o nome
completo e o número de loops
endfor ; Fechamento do loop for com entidade

; Exemplo de aplicação de um loop forlist
variables ; Abertura de variáveis
 string vList ; Definição de variável para armazenar a lista
 string vItem ; Definição de variável para armazenar o item atual da lista
 numeric vIndex ; Definição de variável para armazenar o índice do item atual
endvariables ; Fechamento de variáveis

vList = "Athens;Rome;Syracuse;Pompey;Sparta" ; Instancia a variável vList com
uma lista de cidades

forlist vItem, vIndex in vList ; Abertura do loop forlist
 if (vItem = "Pompey") ; Abertura de condicional if
    ; Se o item atual for "Pompey", o loop é interrompido
    ; e uma mensagem é impressa na tela.
    putmess "Loop processing stopped on Item number: %%vIndex, Value: %%vItem"
;Imprime mensagem de parada do loop
 break ; Interrompe o loop
 endif ; ; Fechamento do bloco if
    putmess "Processing Item number: %%vIndex, Value: %%vItem" ; Imprime o
número do item e o valor atual
endfor ; Fechamento do loop forlist

; Exemplo de aplicação de um loop forlist com id
variables ; Abertura de variáveis
 string vList ; Definição de variável para armazenar a lista
 string vItemId ; Definição de variável para armazenar o id do item atual
 string vItemValue ; Definição de variável para armazenar o valor do item atual
 string vIndex ; Definição de variável para armazenar o índice do item atual
endvariables ; Fechamento de variáveis
vList = "A=Athens;R=Rome;Sy=Syracuse;P=Pompey;Sp=Sparta" ; Instancia a variável
vList com uma lista de cidades e seus ids
forlist/id vItemId, vItemValue, vIndex in vList ; Abertura do loop forlist com
id
    if (vItemId = "P") ; Abertura de condicional if
        putmess "Loop processing stopped on Item number: %%vIndex, Id:
```

```
%%vItemId, Value: %%vItemValue" ;Imprime mensagem de parada do loop
        break ; Interrompe o loop
    endif ; ; Fechamento do bloco if
    putmess "Processing Item number: %%vIndex, Id: %%vItemId, Value:
%%vItemValue" ; Imprime o número do item, o id e o valor atual
endfor ; Fechamento do loop forlist com id


;Exemplo de loop while
while (FIELD.ENT != vValue) & ($status >= 0)) ;Abertura do loop while e
aplicação da condição
 ;Seu código aqui
 ;Se a condição for verdadeira, o loop continua
 ;Caso contrário, o loop é encerrado
endwhile ;Fechamento do loop while

creocc "CUSTOMER",  -1 ;Cria uma ocorrencia vazia na entidade especificada

remocc "CUSTOMER", 0  ;Remove a ocorrencia especificada na entidade CUSTOMER

sort "CUSTOMER", "FLD2:d ;FLD1:a" ;Ordena os registros da entidade ENTITY pelos
campos FLD2 e FLD1, sendo o primeiro em ordem decrescente e o segundo em ordem
crescente


Sort Option                     Valid Values
   Meaning
Order                       ascending | A descending | D
Sort order
Unique                      unique | U
Keep only unique items; discard duplicates. Using Sort Options.


store/e "DEPT" ;Armazena os registros da entidade DEPT

Qualifier           Description
/e                  Stores the modified occurrences starting with Entity. All
child component entities of the stored entity are also stored.
/complete           Builds any incomplete hitlists by issuing the appropriate
DBMS calls, prior to starting the actual store process. This allows the user to
continue working through the hitlists after the function has been processed.
store/complete may affect performance when operating on extensive sets of data.
/truncate           Truncates all hitlists (both inner and outer) after the
store statement has been executed. This is the default behavior.




;Definição de operation
operation do_it ;Abertura de operation e seu nome
 ; Seu código aqui
end ;Fechamento da operation

; Operation Exemplo
```

```
operation DISCOUNT ;Abertura de operation
params ;Abertura de parêmtros
  string CUSTID : IN ;Parâmetro de entrada
  numeric AMOUNT : INOUT ;Parâmetro de entrada e saída
  numeric PERCENTAGE : OUT ;Parâmetro de saída
endparams ;Fechamento de parâmetros

; Código da operação
PERCENTAGE = 0
if ( CUSTID == "ufbv" ) PERCENTAGE = 20
if ( CUSTID == "acme" ) PERCENTAGE = 15
AMOUNT = AMOUNT * ( 100 - PERCENTAGE) / 100

end ; Fechamento da operação

; Operation Exemplo
operation FACTORIAL ;Abertura de operation
params  ;Abertura de parêmtros
  numeric N : IN ;Parâmetro de entrada
  numeric F : OUT ;Parâmetro de saída
endparams ;Fechamento de parâmetros
variables  ;Abertura de variáveis
  numeric W ;Definição de variável
endvariables ;Fechamento de variáveis

if ( N > 1 ) ;Exemplo de abertura de condicional if
  W = N - 1
  activate "calculator".FACTORIAL (W, F) ;Chamada de operação
  F = N * F
else ;Exemplo abertua de bloco else para if
  if ( N = 1 ) ;Exemplo de abertura de condicional if aninhada
    F = 1
  else ;Exemplo abertua de bloco else para if aninhado
    F = 0
  endif ;Exemplo de fechamento de bloco if aninhado
endif  ;Exemplo de fechamento de bloco if
end ; ;Fechamento da operação

----------------------------------------

;Operadores para uniface
Indirection                    @                                    Field
Indirection
De-reference               ->{ }                          Identifier Struct
Dereference and Operation Activation Struct Index
Extraction                   [ ]                            Extraction
Indirect de-reference      -> "SubstitutionString"       Dereference with
string substiution

;Operadores de comparação para uniface
<     Less than
<=    Less than or equal to
!=    Not equal to
=     Equal to
```

```
==    Equal to
>=    Greater than or equal to
>     Greater than

;Operadores lógicos para uniface
!     Logical NOT
&     Logical AND
|     Logical OR

;Consulta registros de uma entidade com o comando retrieve
retrieve/e "CUSTOMER.SALES" ;Recupera registros da entidade CUSTOMER.SALES para
processamento

;Opções de recuperação de registros

/e                  Activates the read trigger of the current entity ($entname)
or of Entity.
/x              Retrieves an additional occurrence of the specified entity
without discarding the hitlist. For more information, see retrieve/x.
/a                  Retrieves multiple additional occurrences of the specified
entity without discarding the hitlist. For more information, see retrieve/a.
/o                  Attempts to locate an occurrence of an entity using the
current primary key value. For more information, see retrieve/o.
/reconnect          Reconnects data loaded from an XML stream with the
occurrences in a database or component. For more information, see
retrieve/reconnect.

;Exemplo de utilização do return para retorno de códigos

trigger quit ;Aplicação em trigger quit, abertua de trigger quit

if ($status < 0) ;Abertura de condicional if, considerando que $status retornou
um erro
    message "Store error number %%$status%%%" ;Imprime mensagem de erro
    return (-1) ;Retorna código de erro -1
endif ;Fechamento do bloco if

return (0) ;Retorno sem erro
end ; quit

;Definição do que é o $status para retornos em unice
$status returns a condition code that indicates the result of a runtime action,
such as an I/O request.
    In general:
        A negative value in $status indicates an error.
        0 indicates a successful operation.
        A positive value indicates a warning or information.

;Manipulação de strings em uniface, exemplo de extração de valores de uma string
NAME = "HOLLERITH" ; Definição de variável string
$1 = NAME[4,8]          ;Extract positions 4 through 8  LERIT
$1 = NAME[3:3]          ;Extract positions 3 through 5  LLE
$1 = NAME[$10:4]        ;Extract positions 2 through 5  OLLE
$1 = NAME[3]            ;Extract positions 3 though 9  LLERITH length NAME  Get
```

```
length of NAME  9
$1 = NAME[$result]      ;Extract last character of NAME  H scan NAME, "LL?"  Get
position of string matching ‘LL?’  3
$10 = $result+2         ;Get position of character following ‘LL’  5
$1 = NAME[$10:1]        ;Extract character following ‘LL’  E


;Verificando o tamanho de uma string
strlength = $length(str1) ; strlength = 9


;Removendo caracteres de uma string a sua esquerda
vString1="xxxxUniface"
vString2 = $ltrim(vString1,"x")                          ;vString2 now contains
"Uniface"
vString = $ltrim("xxxxUnifacexxxx",$regex("x"))          ; vString now is
"Unifacexxxx"
vString = $ltrim("xxxx",$regex("x"))                     ; vString now is ""
vString = $ltrim("1111234567890111uniface",$regex("\d"))  ; vString now is
"uniface"

;Removendo caracteres de uma string a sua direita
vInput="UNIFACExxxx"
vOutput = $rtrim(vInput,"x")                               ; vOutput now
contains "UNIFACE"
vString = $rtrim("xxxxUnifacexxxx",$regex("x"))            ; vString now is
"xxxxUniface"
vString = $rtrim("xxxx",$regex("x"))                       ; vString now is ""
vString = $rtrim("uniface1111234567890111",$regex("\d"))   ; vString now is
"uniface"

;Alterando caracteres com replace
vReplaced = $replace("a should be uppercase", 1, "a", "A", -1)         ; Result:
vReplaced = "A should be uppercAse"
vReplaced = $replace("a should be replaced by B", 1, "a", "B")         ; Result:
vReplaced = "B should be replaced by B"
vReplaced = $replace("a should be uppercase",1,$regex("a"),"A",-1)     ; Result:
vReplaced = "A should be uppercAse"
vReplaced = $replace("F should be replaced",1,$regex("f","i"),"a",-1)  ; Result:
vReplaced = " a should be replaced


;Exemplo de struct em uniface
vStruct = $newstruct ;Cria uma nova struct
vStruct->name = "Uniface" ;Adiciona membro na struct
vStruct->age = 30 ;Adiciona membro na struct
vStruct->address = $newstruct ;Cria nova struct dentro da struct
vStruct->address->street = "Rua 1" ;Adiciona membro na struct
vStruct->address->city = "São Paulo" ;Adiciona membro na struct

vName = vStruct->name ;Acessa membro da struct

vName = vStruct->address{2} ;Acessa membro da struct pelo index
```

```
vName = vStruct->* ;Acessa todos os membros da struct

vStruct->$collsize ;Retorna quantidade de membros da struct

vStruct->$dbgString ;Retorna uma string com os membros da struct e seus valores

;Verifica se duas variáveis do tipo struct referenciam a mesma struct física.
if ($equalStructRefs(vStructA, vStructB) != 1) ;Abre bloco condicional
    ;Seu código
endif ;Fecha bloco condicional

vIndex = vStruct->$index ;Retorna o índice da struct
vStruct->$index = 1 ;Atribui valor ao índice da struct

vLeaf = vStruct->$isLeaf ;Verifica se o nó da struct é o ultimo nó da árvore

vScarlar = vStruct->$isScalar ;Verifica se o nó da struct é um nó escalar

jsonToStruct vStruct , JsonSource ;Converte um json para uma struct

xmlToStruct vStruct, XmlDocument ;Converte um xml para uma struct

VMembers = vStruct->$MemberCount ;Retorna a quantidade de membros da struct

vNameMember = vStruct->$name ;Retorna o nome do membro da struct

structToJson JsonTarget, vStruct ;Converte uma struct para um json

structToXml XmlTarget, vStruct ;Converte uma struct para um xml

Struct->$tags ;Obtem ou informa anotações para uma struct

;Tag            Values                                Comments
;jsonClass          object | array                          Type of JSON
construct
;jsonDataType   boolean | string | number | null        Data type of the Struct
member
```