



TÍTULO: NOMENCLATOR

Leyenda:

Negrita se utiliza para las palabras clave del lenguaje

Cursiva se utiliza para los identificadores definidos por el usuario (nombres de variables, tipos, etc.)

Castellano se utiliza para describir instrucciones

Tipos básicos

integer
char
real
boolean
string

Declaración de tipos

type
typeName = declaración del tipo
end type

Declaración de constantes

const
constName: **type** = expresión;
end const

Declaración de variables

var
varName: tipo de la variable;
end var

Declaración de tuplas

typeName = **record**
fieldName: tipo del campo;
end record

Declaración de vectores

vectorName: **vector**[longitud] **of** tipo del campo;

Condicional

if expresión condicional **then**
 bloque de instrucciones
end if

o

if expresión condicional **then**
 bloque de instrucciones
else
 bloque de instrucciones
end if

Iteración

while expresión condicional **do**
 bloque de instrucciones
end while

o

for *varName* := expresión1 **to** expresión2 **do** [step
 expresión3]
 bloque de instrucciones
end for

Clase de parámetros

in
out
inout

Función

function *functionName* (*parName1*: tipo del parámetro, ...,
parNameN: tipo del parámetro): tipo del retorno
 bloque de instrucciones
end function



```
algorithm algorithmName
  bloque de instrucciones
end algorithm
```

Acción

```
action actionName (clase del parámetro parName1: tipo del
parámetro, clase del parámetro parName2: tipo del
parámetro,..., clase del parámetro parNameN: tipo del
parámetro)
  bloque de instrucciones
end action
```

Declaración de punteros

```
varName: pointer to tipo del campo
```

Operadores lógicos y constantes

```
and, or, not, TRUE, FALSE
```

Funciones y acciones de entrada/salida

```
function readInteger(): integer;
function readReal(): real;
function readChar(): char;
function readString(): string;
function readBoolean():boolean;
action writeInteger(in varName: integer);
action writeReal(in varName: real);
action writeChar(in varName: char);
action writeString(in varName: string);
```

Funciones de cambio de tipo

```
function integerToReal(varName: integer):real;
function realToInteger(varName: real):integer;
function charToCode(varName: char):integer;
function codeToChar(varName: integer):char;
```