|  |
| --- |
| **Práctica 5.1**  **Lenguaje: Codext**  **Definiciones generales y reglas**  **Segunda revisión** |

****

**Definición de instrucciones del lenguaje de generación de documentos.**

|  |
| --- |
| Palabra reservada 1: DOC\_START |
| Indica el inicio del documento, para comenzar a escribir texto. |
| SINTAXIS |
| DOC\_START |
| PARÁMETROS |
| No aplica |
| EJEMPLOS |
| DOC\_START |

|  |
| --- |
| Palabra reservada 2: DOC\_END |
| Indica el fin del documento, para terminar con la escritura de texto. |
| SINTAXIS |
| DOC\_END |
| PARÁMETROS |
| No aplica |
| EJEMPLOS |
| DOC\_END |

|  |
| --- |
| Palabras reservadas 3 y 4: -> , <- |
| Indican el inicio y fin, respectivamente, de un bloque de instrucciones. |
| SINTAXIS |
| Expresión  ->  Instrucciones  <- |
| PARÁMETROS |
| Expresión -> Palabra reservada, expresión o instrucción que permita abrir un bloque de código.  Instrucciones -> Conjunto de instrucciones. |
| EJEMPLOS |
| DECIDE(LOGIC(!,4#,3#))  ->  NEXT\_LINE(5#))  <- |

|  |
| --- |
| Palabra reservada 5: TITLE |
| Muestra en pantalla un título del tamaño de fuente especificado. |
| SINTAXIS |
| TITLE(tamaño, texto) |
| PARÁMETROS |
| tamaño -> Constante numérica, identificador texto -> Cadena, identificador |
| EJEMPLOS |
| TITLE(12#, <“teoría de autómatas”>) |

|  |
| --- |
| Palabra reservada 6: PAGE\_JUMP |
| Provoca que se dé un salto a una nueva página para la generación del documento. |
| SINTAXIS |
| PAGE\_JUMP |
| PARÁMETROS |
| No aplica. |
| EJEMPLOS |
| No aplica. |

|  |
| --- |
| Palabra reservada 7: NEXT\_LINE |
| Provoca que se dé un salto de línea para continuar con la generación del documento. También se puede especificar si se quiere saltar más de una sola línea |
| SINTAXIS |
| NEXT\_LINE  NEXT\_LINE(cantidad) |
| PARÁMETROS |
| cantidad -> Constante numérica, identificador |
| EJEMPLOS |
| NEXT\_LINE  NEXT\_LINE(3#) |

|  |
| --- |
| Palabra reservada 8: ARITH |
| Calcula el resultado de una expresión aritmética cuyo resultado puede asignarse a una variable. |
| SINTAXIS |
| ARITH(operador, operando 1, operando 2, …) |
| PARÁMETROS |
| operador -> Operador  operando 1, operando 2, … -> Constante numérica, Identificador, palabra reservada 8 |
| EJEMPLOS |
| ARITH(+,2#,2#) |

|  |
| --- |
| Palabra reservada 9, 10: TABLE, TABLE\_ROW |
| Permite generar una tabla con 1 o más filas. |
| SINTAXIS |
| TABLE (cantidad)  ->  TABLE\_ROW(contenido1, contenido2, …)  TABLE\_ROW(contenido1, contenido2, …)  .  .  .  <- |
| PARÁMETROS |
| cantidad -> Constante numérica entera que determina la cantidad de columnas de la tabla contenido1, contenido2, … -> Constante numérica, cadena, identificador, la cantidad de estos se define por el argumento de la palabra TABLE. |
| EJEMPLOS |
| TABLE  ->  TABLE\_ROW(<“Hola”>, <“Mundo”>)  TABLE\_ROW(<“Lenguajes”>, <“Automatas”>)  TABLE\_ROW(1#, 2#, 3#)  .  .  <- |

|  |
| --- |
| Palabras reservadas 11 y 12: DECIDE , DECFALSE |
| Estructura de selección, con una condición, una opción verdadera y una falsa. |
| SINTAXIS |
| DECIDE(expresión\_lógica)  ->  opcion1  <-  DECIDE(expresión\_lógica)  ->  opcion1  <-  DECFALSE  ->  opcion2  <- |
| PARÁMETROS |
| expresión\_lógica -> Expresión evaluada utilizando la palabra reservada 13 (LOGIC).  opcion1, opción2 -> Bloques de instrucciones para ejecutar en cada caso. |
| EJEMPLOS |
| DECIDE(LOGIC(!,4#,3#))  ->  NEXT\_LINE(5#)  <-  DECIDE(LOGIC(!,4#,3#))  ->  NEXT\_LINE(5#)  <- DECFALSE  ->  PAGE\_JUMP  <- |

|  |
| --- |
| Palabra reservada 13: LOGIC |
| Resuelve una expresión lógica o relacional. |
| SINTAXIS |
| LOGIC(operador, operando 1, operando 2) |
| PARÁMETROS |
| operador -> Operador relacional, operador lógico  operando1, operando 2 -> palabra reservada 13, palabra reservada 16, identificador, constante numérica, cadena |
| EJEMPLOS |
| LOGIC(!, 4#, 3#)  LOGIC(=, <“Hola”>, <“hola”>) |

|  |
| --- |
| Palabras reservadas 14, 15 y 16: REPEAT , TIL, TIMES |
| Estructura cíclica con un bloque de código y una condición, así como un contador interno que lleva cuenta de las veces que se ha ejecutado. |
| SINTAXIS |
| REPEAT  ->  instrucciones  <-  TIL(condición) |
| PARÁMETROS |
| Instrucciones -> Bloque de instrucciones  condición -> Expresión lógica para control del ciclo, aquí se puede utilizar el contador interno TIMES |
| EJEMPLOS |
| REPEAT  ->  NEXT\_LINE  <-  TIL(LOGIC(=,TIMES,3#)) |

|  |
| --- |
| Palabra reservada 17: VAR |
| Define una variable. |
| SINTAXIS |
| VAR(nombre) |
| PARÁMETROS |
| nombre -> Cadena |
| EJEMPLOS |
| VAR(<”titulo”>) |

|  |
| --- |
| Palabra reservada 18: VALUE |
| Asigna valor a una variable. |
| SINTAXIS |
| VALUE(iden, valor) |
| PARÁMETROS |
| iden -> Identificador  valor -> Cadena, constante numérica, identificador. |
| EJEMPLOS |
| VALUE(\_Titulo,<“Mi titulo”>) |

|  |
| --- |
| Palabra reservada 19: VIEW |
| Muestra en pantalla el valor de una variable, constante numérica o cadena. |
| SINTAXIS |
| VIEW(vista) |
| PARÁMETROS |
| vista -> Cadena, constante numérica, identificador. |
| EJEMPLOS |
| VIEW(<“hola”>)  VIEW(1#)  VIEW(\_iden01) |

**Definiciones general y reglas del lenguaje**

**Palabras reservadas**

* Inician con el nombre definido para la palabra reservada.
* Si lleva parámetros, le debe seguir un paréntesis de apertura, después los parámetros separados por coma, después un paréntesis de cierre.
* Si permite abrir un bloque de código, este se abre con ->, después van las instrucciones del bloque, después se cierra con <-
* Terminan con un espacio en blanco.

**Cadenas**

* Inician con <”
* Puede contener números, letras, caracteres especiales, operadores.
* Terminan con “>

**Constantes numéricas**

* Pueden o no comenzar con un signo, es decir, el operador + o el operador -.
* Comienzan con un número.
* Le pueden seguir más números
* Puede contener un punto.
* Después del punto, deben uno o seguir más números.
* Puede o no contener notación científica, la cual consta de la letra ‘E’, un signo opcional, y uno o más números.
* Termina con un carácter numeral (#).

**Operadores**

* **Aritméticos**
  + Suma (+)
  + Resta (-)
  + Multiplicación (\*)
  + División (/)
  + Potencia (\*\*)
  + Modulo (%)
* **Relacionales**
  + Mayor que (>)
  + Menor que (<)
  + Igual (=)
  + Desigual (<>)
  + Mayor o igual que (>=)
  + Menor o igual que (<=)
* **Lógicos**
  + AND
  + OR
  + NOT

**Caracteres especiales**

* Guion bajo
* Guion intermedio
* Numeral
* Punto
* Coma
* Punto y coma
* Paréntesis de apertura y cierre
* Operadores
* Espacio en blanco
* Salto de línea

**Comentarios**

* Inician con </
* Puede contener números, letras, caracteres especiales, operadores.
* Terminan con />

**Identificador**

* Inicia con guion bajo.
* Debe continuar con al menos una letra, número, guion intermedio o guion bajo.
* Le pueden seguir más letras, números, guiones intermedios y/o guiones bajos.
* Termina con un espacio en blanco.