**Nombre del compilador: Master**

**Nombre del lenguaje: Cobra**

**1.- inicio y fin**

|  |  |
| --- | --- |
| Def main{ | Inicio de código |
| }end | Fin de código |

**2.- Tipos de datos**

|  |  |
| --- | --- |
| int |  |
| decimal |  |
| String |  |
| boolen |  |

**Tipos de datos booleanos**

|  |  |
| --- | --- |
| **Tipo** | **valor** |
| **<Booleano>** | **Verdadero, falso** |

**Tipos de datos de cadena**

|  |  |
| --- | --- |
| **Tipo** | **Valor** |
| **<Cadena\_salida>** | **“dedede”** |

**3- concatenación**

|  |  |
| --- | --- |
| + |  |

**4.- estructuras condicionales**

|  |  |
| --- | --- |
| If |  |
| else |  |
| Else if |  |
| break |  |

**5.-ciclos**

|  |  |
| --- | --- |
| While =>condición; end |  |
| For=> instrucción; end |  |
| Do while |  |

**6.- entrada y salida**

|  |  |
| --- | --- |
| **print** | **Salida** |
| **input** |  |

**7.- Símbolos**

|  |  |
| --- | --- |
| **Tipo** | **valor** |
| **Indica el inicio de entrada de print** | **(** |
| **Indica la salida input** | **)** |
| **corchete arreglo inicio** | **[** |
| **Corchete arreglo final** | **]** |

**8.- Operadores de comparación**

|  |  |
| --- | --- |
| **Tip** | **Valor** |
| **Operadores de comparación** | **< , > , == , notEq , <= , >=** |

**9.- operadores lógicos**

|  |  |
| --- | --- |
| **Tipo** | **Valor** |
| **Operador lógico and** | **And** |
| **Operador lógico or** | **or** |
| **Operador lógico negación** | **not** |

**10.- operadores aritméticos**

|  |  |
| --- | --- |
| **Tipo** | **Valor** |
| **Sumador y resta** | **+ , -** |
| **Operador de incremento o decremento** | **++ , --** |
| **Operador multiplicador y divisor** | **\* , /, %** |
| **Operador potencia** | **^** |

**11.- Operadores de asignación**

|  |  |
| --- | --- |
| **Tipos** | **Valor** |
| **Operador de asignación** | **= , += , -= , \*= , /=** |

**12.- identificadores**

|  |  |
| --- | --- |
| **Tipo** | **valor** |
| **Identificadores** | **Área, p01, vVwee** |

**En blanco**

**Existe un token especial que nos brinda JavaCC, este es el token de tipo , el cuál cada vez que se lee, indica que ya no hay más valores dentro del código, por lo que lee espacios en blanco o vacíos, debido a que se consumieron todos los valores dentro del código pasado por el archivo de texto.**

**1.2 Tabla de errores**

**Los errores o palabras que no se encuentran dentro del lenguaje lingvo, se identifican dentro del lenguaje con el token , donde se mete todo lo que no esta incluido en la definición del lenguaje, es decir, en las tablas del punto anterior, como por el ejemplo el símbolo de arroba ‘@’, la tilde de la eñe ‘~’, entre otros.**

**Un punto importante a marcar, es que el lenguaje no acepta la vocales acentuadas (á,é,í,ó,ú) o la eñe (ñ) dentro de los identificadores, sin embargo, las cadenas de salida (todo aquello que vaya entre comillas dobles (“ ”)) y los char (todos los caractéres que se encuentren dentro de las comillas sencillas(‘ ’)) si aceptan las vocales acentuadas y la eñe, de otra forma se identificaran como errores léxico. Los errores léxicos son todos aquellos símbolos que el lenguaje no reconoce, o no pertenecen a él, como se habla en los párrafos anteriores, hay ciertos símbolos que el lenguaje no acepta.**

**Los errores sintácticos se marcan cuando las estructuras del programa no siguen las reglas de las gramáticas que se definieron en el código, tales como en vez de escribir ; (punto y coma) para terminar una oración, la terminan con un . (punto), o no se usa de manera correcta alguna de las plabras definidas en el lenguaje, entre otros errores que pueden ocasionar un error sintáctico.**

**Los errores se controlan mediante una pila, esta pila nos permite posicionarnos en el último método donde se encontró el error y seguir evaluando el código, así como darle una estructura a los paréntesis y llaves, es decir, toda aquella llave o paréntesis que se abra debe de cerrarse, por lo que de lo contrario todo el resto del código estaría erróneo en automático.**

|  |  |
| --- | --- |
| **Tipo** | **Valor** |
| **Errores** | **Ñ , Á @** |