

ELABORADO POR:

KEVIN VELASQUEZ 00018616

EMERSON NOLASCO 00215316

KEVIN LOPEZ 00251716

KARLA CRUZ 00010216

ILIANA IRAHETA 00013014

Manual de ayuda

Técnica

<b>Descripción del diseño del lenguaje y de sus posibles aplicaciones.</b>	<b>2</b>
Tipos de datos	2
Instrucciones	2
<b>Palabras reservadas</b>	<b>4</b>
Built - in Functions	4
<b>Expresiones regulares</b>	<b>5</b>
Operadores y caracteres especiales	5
<b>Expresiones regulares</b>	<b>6</b>
Expresión regular para los identificadores	6
Expresiones regulares para números enteros y números reales	6
<b>Gramatica</b>	<b>7</b>
<b>Tokens utilizados</b>	<b>8</b>
<b>Diagramas de sintaxis</b>	<b>9</b>
Bloque	9
Introducción	9
Declaración	10
Asignación	10
Declaración y asignación	10
Condición	11

## **Descripción del diseño del lenguaje y de sus posibles aplicaciones.**

El diseño del lenguaje se basa en la programación por bloques, a través de una presentación visual del paradigma y metodología de la programación, permitiendo centrarse en la lógica de la programación y en el flujo de la ejecución dejando a un lado la sintaxis propia de los lenguajes tradicionales de programación.

### **Tipos de datos**

Tipo	Alias	Descripción	Bits	Rango de valores
Char	chary	Caracteres unicode	16	['\u0000','\uFF FF']
String	stringy	Cadenas de caracteres	Variable	Lo que permite la memoria
Boolean	booly	Valores logicos	8	true, false
Int	inty	Enteros normales sin signo	32	[0, 4,294,967,295]
Float	floaty	Decimal simple	4	- 3,4·1038 a 3,4·1038

### **Instrucciones**

If - else	inty gamma; gamma = 121; if ((gamma % 2) == 0) output "the number is even"; else output "the number is odd"; endif
forsy	inty gamma; forsy (gamma = 1; gamma < 11; ++gamma);

	output "%d ", gamma; endforsy
whiley	inty gamma = 10; whiley( gamma < 20 ) output "value of a: %d\n", gamma; gamma++; endwhiley
dowhiley	inty gamma=0; do output "Value of variable gamma is: %d\n", gamma; gamma++; Enddo whiley (gamaa<=3); end
Inny - outty	inty testInteger; output "Enter an integer: "; inny "%d", &testInteger; output "Number = %d",testInteger;

## Palabras reservadas

### Built - in Functions

absy (num)	Devuelve el valor absoluto de un entero. El valor absoluto de un número siempre es positivo.
maxy (num1, num2)	Es una función que devuelve el mayor de a y b. Pueden ser cualquier valor numérico de números enteros.
miny (num1, num2)	Es una función que devuelve el menor de a y b. Pueden ser cualquier valor numérico de números enteros.
sqryt (num)	Esta función se usa para encontrar la raíz cuadrada del argumento.
lengthy (str)	Calcula la longitud de una cadena dada. La función toma una cadena como argumento y devuelve su longitud.
substry (str, i, j)	Devuelve la subcadena de la cadena de origen en la posición m y terminando en la posición n-1
powy (num, i)	Se utiliza para encontrar la potencia del número dado.
charyat (str, index)	Devuelve el carácter en el índice especificado en una cadena
sorty (arr)	Función capaz de ordenar cualquier arreglo.
randy (num, num)	Es una función que devuelve un número entre dos enteros a y b.
oddy(num)	Comprueba si el argumento es un número impar.

## Expresiones regulares

### Operadores y caracteres especiales

+	Suma
-	Resta
*	Multiplicacion
/	Division
^	Exponencial
(	Paréntesis abierto
)	Parentesis cerrado
,	Continuación de instrucción
;	Fin de instrucción
=	asignacion
&&	Operador Y lógico
/	Operador O lógico
!	Operador negación lógica
&	Operador AND entre bits
^	Operador OR exclusivo entre bits
/	Operador OR inclusivo entre bits
~	Operador complemento a uno de bits

&=	Asignación compuesta
^=	Asignación compuesta
=	Asignación compuesta
!=	Operador relacional de desigualdad

## Expresiones regulares

### Expresión regular para los identificadores

Identifier\_tk {char\_tk} ({char\_tk} | {digit\_tk} | "\_")\*

Char\_tk [ a - zA - ZñÑ ]

Digit\_tk [ 0 - 9 ]

### Expresiones regulares para números enteros y números reales

Digitos digit\_tk [ 0 - 9 ]

Enteros int\_tk { digit\_tk }+

Decimal simple (Real) float\_tk { digit\_tk } + "." { digit\_tk }+

### Un ejemplo de programa para escribir “Hola mundo” en el lenguaje

```
Begin
    output "Hola mundo";
End
```

### Cualquier otra observación (o limitación –o aclaración) que considere necesario

- Para la extensión de los archivos de los programas realizados en este lenguajes es .bly
- Ejemplo de la construcción de un diagrama.

## **Gramatica**

condición -----> expresión comparación expresión

comparación -----> < | > | <= | >= | == | !=

expresión -----> expresión + término | expresión - término | término

término -----> término \* factor | término / factor | factor

factor -----> identificador | número | (expresión) | string\_literal

instrucciones-----> instrucción | instrucciones |

Statement -----> asignación | if statement | while statements | for statement

asignación-----> identificador = expresión

identificador -----> letra | identificador letra | identificador dígito

Letra -----> a | b | c | d | e | f | g | h | i | j | k | l | m | n | ñ | o | p | q | r | s | t | u | v | w | x | y | z |  
A | B | C | D | E | F | G | H | I | J | K | L | M | N | Ñ | O | P | Q | R | S | T | U | V | W | X | Y | Z

dígito -----> 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

número -----> entero | entero . entero

entero -----> dígito | entero dígito

cadenas -----> comillas carácter comillas | comillas caracteres comillas

caracter -----> letra | dígito | caracteres especiales |

caracteres -----> carácter | caracteres

comillas -----> “

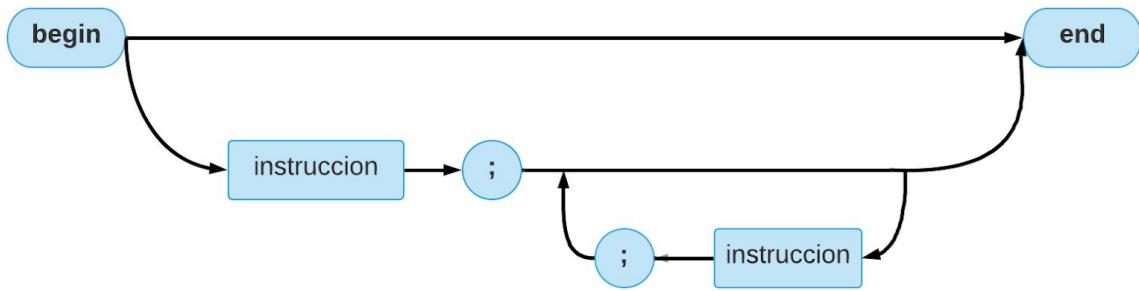
## Tokens utilizados

Los tokens definidos en el código fuente del proyecto son los siguientes:

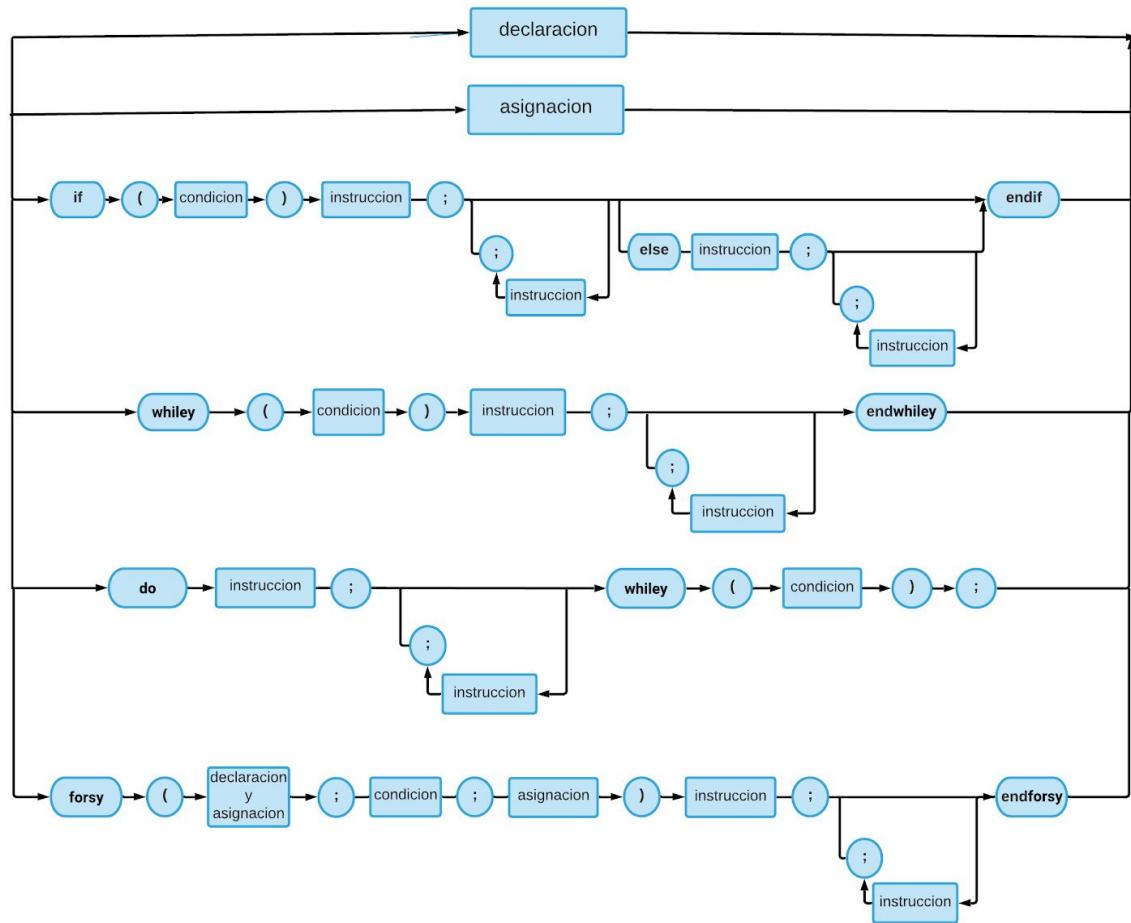
- CharToken
- IntToken
- FloatToken
- StringTokenizer
- PlusToken
- MinusToken
- StarToken
- SlashToken
- HatToken
- PercentToken
- AssignmentToken
- AndToken
- OrToken
- EqualsToken
- NotEqualsToken
- LessToken
- LessEqualsToken
- GreaterToken
- GreaterEqualsToken
- OpenParenthesisToken
- CloseParenthesisToken
- CommaToken
- IdentifierToken
- EndOfInstructionToken
- EndOfFileToken
- BadToken
- BangToken

## Diagramas de sintaxis

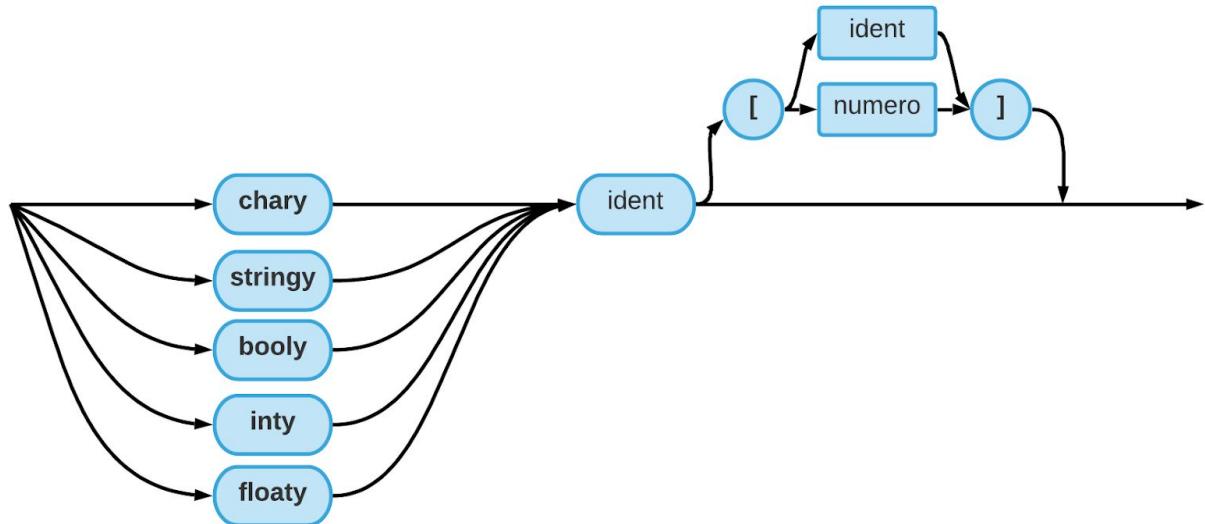
### Bloque



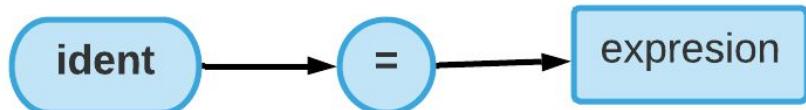
### Introducción



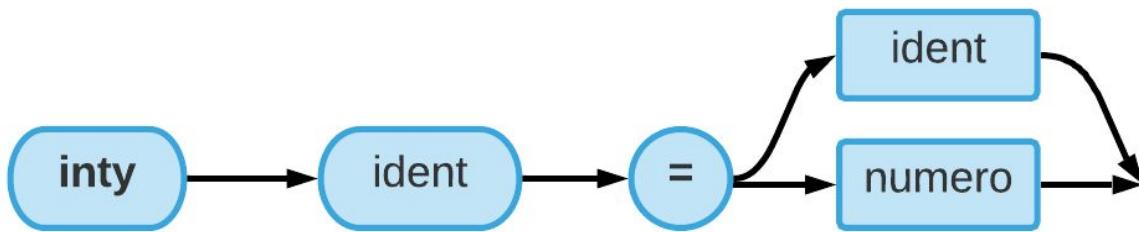
## Declaración



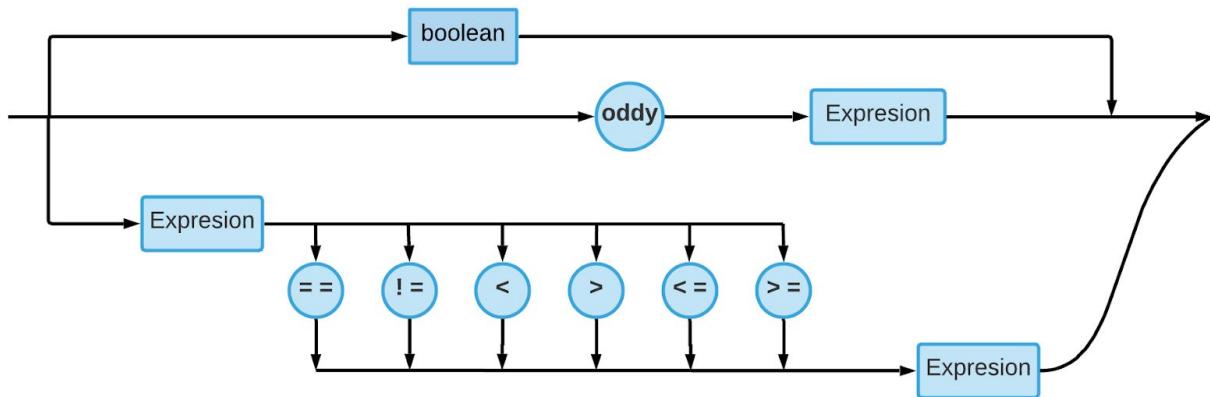
## Asignación



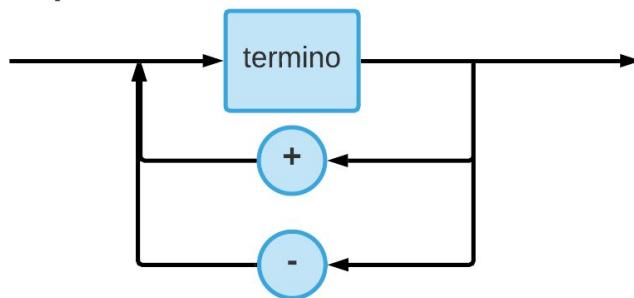
## Declaración y asignación



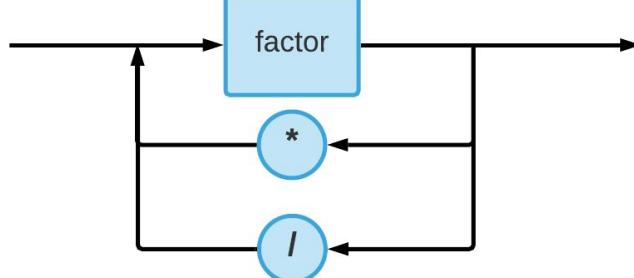
## Condición



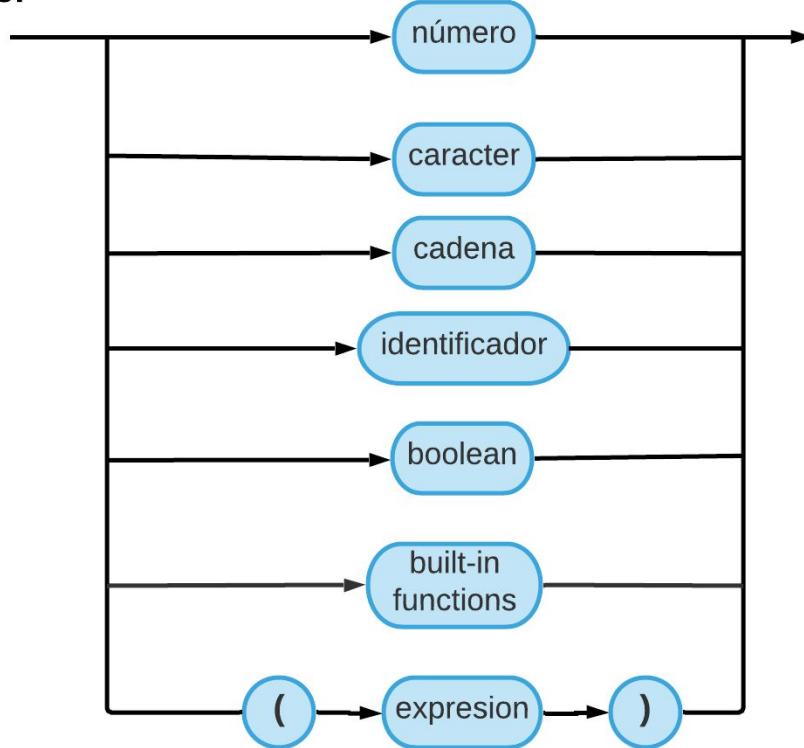
## Expresion



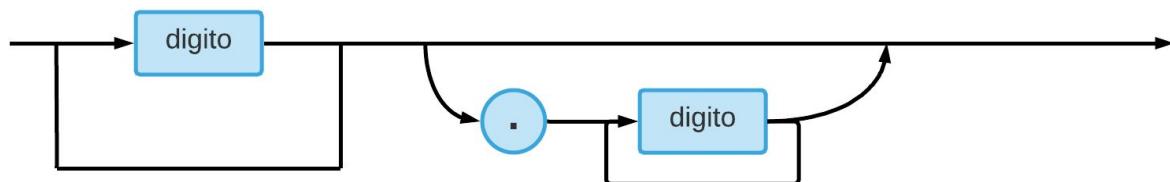
## Termino



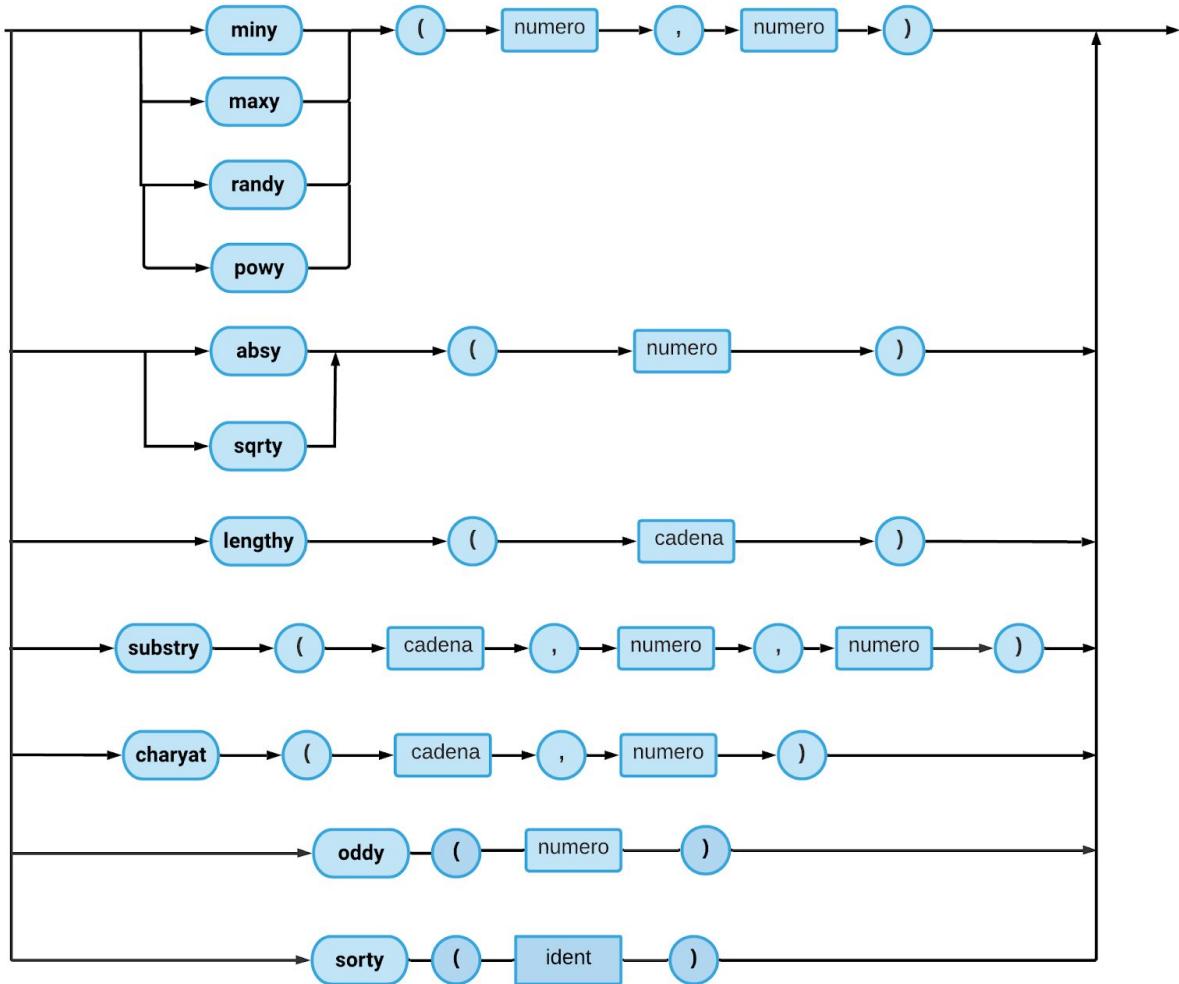
### Factor



### Numero



### Built-in functions



## Interfaz de usuario

