

Manual de Usuario del Lenguaje de Programación

1. Introducción

Este lenguaje de programación está diseñado para ser intuitivo y fácil de usar, con sintaxis en español que facilita su comprensión para programadores que apenas empezaran.

2. Estructura Básica de un Programa

Un programa en este lenguaje se estructura de la siguiente manera:

```
programa {  
    // Declaraciones y sentencias van aquí  
}
```

3. Tipos de Datos

El lenguaje soporta cuatro tipos de datos principales:

- `numero`: Para números enteros
- `decimal`: Para números con punto decimal
- `texto`: Para cadenas de caracteres
- `logico`: Para valores booleanos (verdadero/falso)

Ejemplos de Declaración de Variables:

```
crea numero edad;  
crea decimal altura = 1.75;  
crea texto nombre = "Juan";  
crea logico esEstudiante = verdadero;
```

4. Operadores

Operadores Aritméticos:

- `+` (suma)
- `-` (resta)
- `×` (multiplicación)

- \div (división)
- $^$ (potencia)
- $\sqrt{}$ (raíz)

Operadores de Comparación:

- $=$ (igual a)
- \neq (diferente de)
- $>$ (mayor que)
- $<$ (menor que)
- \geq (mayor o igual que)
- \leq (menor o igual que)

Operadores Lógicos:

- \wedge (AND)
- \vee (OR)
- \neg (NOT)

5. Estructuras de Control

Condicionales

Estructura `si` (if):

```
si (condicion) {
    // sentencias
}
```

Estructura `si-sino` (if-else):

```
si (condicion) {
    // sentencias si es verdadero
} sino {
    // sentencias si es falso
}
```

Bucles

Bucle **mientras** (while):

```
mientras (condicion) {  
    // sentencias  
}
```

Bucle **para** (for):

```
para (crea numero i = 0; i < 10; i = i + 1) {  
    // sentencias  
}
```

6. Funciones

Declaración de Función:

```
funcion nombreFuncion(numero param1, texto param2) {  
    // cuerpo de la función  
    retorna valorDeRetorno;  
}
```

Llamada a Función:

```
resultado = nombreFuncion(argumento1, argumento2);
```

7. Entrada y Salida

Entrada de Datos:

```
pregunta "Ingrese su edad" guarda numero en edad;
```

Salida de Datos:

```
muestra "Hola, mundo!";  
muestra edad;
```

8. Arreglos y Matrices

Declaración de Arreglo:

```
crea arreglo numero miArreglo = [1, 2, 3, 4, 5];
```

Declaración de Matriz:

```
crea matriz numero miMatriz = [  
    [1, 2, 3],  
    [4, 5, 6],  
    [7, 8, 9]  
];
```

Acceso a Elementos:

```
elemento = miArreglo[2];  
elemento = miMatriz[1][1];
```

9. Comentarios

Los comentarios de una línea se escriben usando //:

```
// Este es un comentario
```

10. Ejemplos Completos

Ejemplo 1: Suma de dos números

```
programa {  
    crea numero a = 5;  
    crea numero b = 3;  
    crea numero suma;  
  
    suma = a + b;  
    muestra suma;  
}
```

Ejemplo 2: Función con condicional

```
funcion esPositivo(numero x) {  
    si (x > 0) {  
        regresa verdadero;  
    } sino {  
        regresa falso;  
    }  
}  
  
programa {  
    crea numero num = 10;  
    crea logico resultado = esPositivo(num);  
    muestra resultado;
```

11. Consejos Finales

- Presta atención a la sintaxis, especialmente a los puntos y comas (;)
- Los nombres de variables son sensibles a mayúsculas y minúsculas
- Usa nombres descriptivos para variables y funciones
- Comenta tu código para mayor claridad

¡Feliz programación!

Documento Elaborado por:

Canche Ciau Rusell Emmanuel

Alvarado Chuc José Miguel