Universidad Autónoma de Baja California

Facultad de Ciencias Químicas e Ingeniería



**TRADUCTORES**

**Manual de usuario de**

**Inteprete de Pseudocódigo en Java**

**Docente:** Licea Sandoval Guillermo

**Alumno:** Gómez Cárdenas Emmanuel Alberto

**Matricula:** 01261509

Indice

[Introducción al intérprete de pseudocódigo 3](#_Toc123857203)

[Tipos de datos 3](#_Toc123857204)

[Instrucciones 3](#_Toc123857205)

[Operaciones 4](#_Toc123857206)

[Sugerencias 4](#_Toc123857207)

[Ejemplos 4](#_Toc123857208)

[Factorial 4](#_Toc123857209)

[Fibonacci 5](#_Toc123857210)

[Promedio calificaciones 6](#_Toc123857211)

[Mayor de dos numeros 6](#_Toc123857212)

[Ejecución del interprete 7](#_Toc123857213)

[Conclusión 7](#_Toc123857214)

# Introducción al intérprete de pseudocódigo

El intérprete de pseudocódigo es una herramienta que permite escribir y ejecutar programas utilizando un lenguaje de programación sencillo y fácil de entender. El pseudocódigo es un lenguaje que se utiliza para describir el comportamiento de un programa de manera clara y concisa, sin utilizar un lenguaje de programación específico.

# Tipos de datos

El intérprete de pseudocódigo admite dos tipos de datos: entero y flotante.

* **Entero:** Es un número entero sin decimales. Por ejemplo: 1, 2, 3, 4, etc.
* **Flotante:** Es un número con decimales. Por ejemplo: 1.5, 2.3, 3.14, etc.

# Instrucciones

El intérprete de pseudocódigo soporta las siguientes instrucciones:

* **inicio-programa:** Le indica al interprete el inicio del programa.
* **fin-programa:** Le indica al interprete el final del programa.
* **leer:** Permite ingresar datos desde el teclado.
  + La sintaxis es:
    - leer “mensaje para el usuario”, variable.
  + Por ejemplo:
    - leer “Ingrese su edad: ”, edad.
* **escribir:** Permite imprimir mensajes o variables en la pantalla.
  + La sintaxis es:
    - escribir “mensaje”.
    - escribir variable.
  + Por ejemplo:
    - escribir “Hola Mundo!”.
    - escribir edad.
* **mientras:** Permite ejecutar un bloque de código mientras se cumpla una condición
  + La sintaxis es:

mientras (condición)

bloque de código

fin-mientras

* **si:** Permite ejecutar un bloque de código u otro si se cumple una condición
  + La sintaxis es:

si (condición)

entonces

bloque de código

fin-entonces

sino

bloque de código sino

fin-sino

fin-si

# Operaciones

* **Matemáticas: +, -, \*, / y %**
* **Relacionales: <, >, ==, <=, >=, ¡=, así como su contraparte literal, mayorque, menorque, mayorigualque, menorigualque, distintoque, igualque.**
* **Asignación: =.**

# Sugerencias

* Asegúrate de utilizar la sintaxis y estructura del pseudocodigo correcta al escribir los programas. Si utilizas la sintaxis incorrecta, el intérprete puede dar un error o producir resultados inesperados.
* Cada instrucción y operación debe estar escrita de la manera correcta para que el intérprete pueda entenderla y ejecutarla de manera adecuada.
* Es recomendable comenzar con programas sencillos e ir aumentando la complejidad poco a poco, de esta manera te familiarizaras con la sintaxis y el funcionamiento de este interprete.
* Si estás trabajando con programas grandes o complejos, considera dividir el código en módulos o funciones más pequeñas. Esto te ayudará a hacer que el código sea más legible y fácil de entender, y también te permitirá reutilizar código de manera más eficiente.

# Ejemplos

### Factorial

Inicio-programa

entero numero

entero resultado

leer “Ingresar un numero: ”, numero

resultado = 1

mientras (numero > 0)

resultado = resultado \* numero

numero = numero - 1

fin mientras

escribir “El factorial es: ”, resultado

fin-programa

### Fibonacci

inicio-programa

entero numeros  
 entero n1 = 0  
 entero n2 = 1  
 entero count = 0  
 entero temp

leer "Ingresa numero tope: ", numeros

si (numeros <= 0)

entonces

escribir "Ingresar un numero positivo"

fin-entonces

fin-si

si (numeros == 1)

entonces

escribir "Secuencia Fibonacci hasta: ", numeros

escribir "",n1

fin-entonces

fin-si

si (numeros > 1)

entonces

escribir "Secuencia Fibonnaci hasta: ", numeros

mientras (count < numeros)

escribir "",n1

temp = n1 + n2

n1 = n2

n2 = temp

count = count + 1

fin-mientras

fin-entonces

fin-si

fin-programa

### Promedio calificaciones

inicio-programa

entero materias

entero calificacion

entero promedio

entero contador

leer "Ingresa numero tope: ", numeros

si (numeros <= 0)

entonces

escribir "Ingresar un numero positivo"

fin-entonces

fin-si

### Mayor de dos números

inicio-programa

entero num1  
 entero num2  
 leer "Ingresa primer numero: ", num1  
 leer "Ingresa segundo numero: ", num2

si (num1 > 0)   
 entonces  
 escribir "El mayor numero es: " , num1  
 fin-entonces

sino  
 si (num1 > num2)   
 entonces  
 escribir "El mayor numero es: " , num2  
 fin-entonces

sino  
 escribir "Los números son iguales"

fin-sino

fin-si

fin-sino

fin-si

fin-programa

# Ejecución del interprete

Para ejecutar el intérprete, debes ejecutar el archivo PseudoCompiler.java. Este archivo se encargará de iniciar el intérprete en una JVM y te pedirá que ingreses el nombre del archivo de programa pseudocódigo que deseas interpretar. Una vez que hayas ingresado el nombre del archivo, el intérprete procederá a ejecutarlo. Si el programa pseudocódigo es válido y cumple con la gramática del intérprete, este lo ejecutará y producirá el resultado esperado. Si, por el contrario, hay algún error en el programa pseudocódigo, el intérprete mostrará un mensaje de error y detendrá su ejecución. Es importante asegurarse de que el programa pseudocódigo esté escrito correctamente y siga la gramática del intérprete para evitar errores durante la ejecución.

# Conclusión

El proyecto final ha demostrado ser una herramienta valiosa para la interpretación de programas escritos en pseudocódigo. La implementación del intérprete ha sido exitosa y ha cumplido con los requisitos establecidos al inicio del proyecto.

A lo largo del desarrollo, se han realizado pruebas exhaustivas para asegurar la correcta funcionalidad del intérprete y se han corregido los errores encontrados. Además, se ha preparado un manual de usuario completo para facilitar su uso y se han proporcionado ejemplos y sugerencias para mejorar la experiencia del usuario.

En resumen, este proyecto final ha sido un éxito y esperamos que el intérprete de pseudocódigo sea de gran utilidad para los usuarios. Agradecemos a todos los que han participado en el desarrollo y esperamos seguir mejorando y expandiendo las funcionalidades del intérprete en el futuro.