# Universidad Tecnológica Equinoccial Tecnología Superior en Desarrollo de Software

# Programación I

**Nombr**e: Mikaela Zurita

**Paralelo:** “B”

# Conceptos básicos de Pseudocódigo: Diseño de algoritmos de forma estructurada.

## ¿Qué es el pseudocódigo?

El pseudocódigo es una forma de escribir algoritmos utilizando una estructura similar a los lenguajes de programación, pero sin la sintaxis específica de uno en particular. Se usa para representar la lógica de un programa de manera clara y comprensible antes de su implementación en código real.

# *Características del Pseudocódigo:*

* Usa lenguaje natural con estructura lógica.
* No depende de un lenguaje de programación específico.
* Facilita la comprensión del algoritmo.
* Sirve como guía para la implementación en código real.

## PSeint

Es una herramienta que permite escribir y ejecutar pseudocódigos, facilitando la enseñanza de programación.

* Usa una sintaxis clara y sencilla.
* Soporta estructuras de control, funciones y procedimientos.
* Permite ejecutar y depurar algoritmos.

## Estructura básica de un algoritmo en PSeint

# Un algoritmo en PSeInt sigue una estructura ordenada y lógica, lo que facilita su comprensión y ejecución.

# A continuación, se describe su estructura general:

# Inicio y Fin del Algoritmo:

# Todo algoritmo en PSeInt debe comenzar con la palabra clave Algoritmo seguida del nombre del algoritmo, y finalizar con FinAlgoritmo. Todo algoritmo en PSeInt debe comenzar con la palabra clave Algoritmo seguida del nombre del algoritmo, y finalizar con FinAlgoritmo.

# 

# Ejemplo: Algoritmo que muestra un mensaje en pantalla

# 

# Lo que muestra en la terminal es lo siguiente:

# 

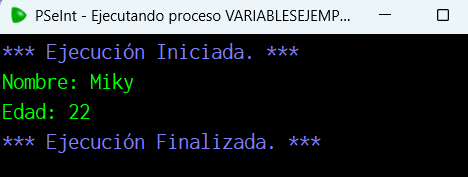
# Declaración de variables

# Las variables deben declararse antes de ser utilizadas. Se usa la palabra clave Definir, seguida del nombre de la variable y su tipo de dato.

# Ejemplo: Algoritmo que declaran variables y les asigna valores

# 

**Lo que muestra en la terminal es lo siguiente:**

****

## Operaciones básicas

Los operadores permiten realizar cálculos y comparaciones en un algoritmo. Se clasifican en:

* **Operadores Aritméticos**

Se usan para realizar cálculos matemáticos:

+ (Suma)

- (Resta)

\* (Multiplicación)

/ (División)

% (Módulo, devuelve el residuo de una división)

## Ejemplo: Operación suma con variables declaradas

## 

## Lo que muestra en la terminal es lo siguiente:

## 

# *Estructuras de control en Pseudocódigo*

# Las estructuras de control permiten definir el flujo del algoritmo, es decir, cómo se ejecutan las instrucciones. En estas se utilizan los siguientes operadores:

# Operadores Relacionales

# Permiten comparar valores y devuelven Verdadero o Falso

**=** (Igual)

**<>** (Diferente)

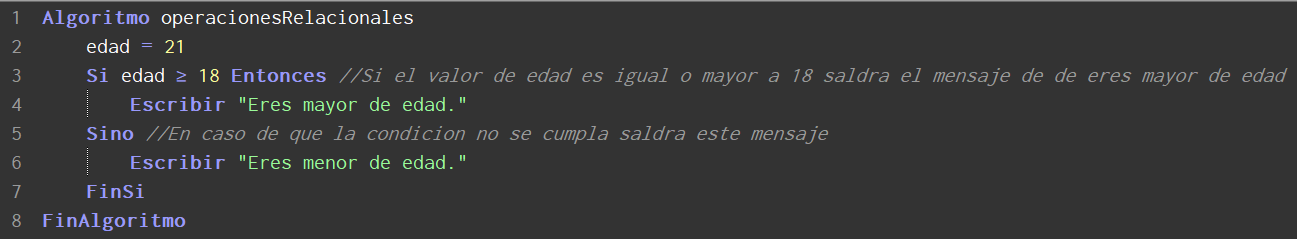
**>** (Mayor que)

**<** (Menor que)

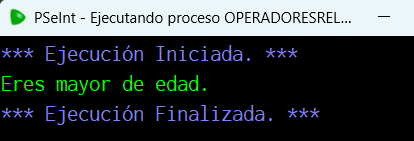
**>=** (Mayor o igual)

**<=** (Menor o igual)

**Ejemplo: Utiliza operadores relacionales para verificar si eres mayor de edad**

****

**Lo que muestra en la terminal será lo siguiente:**

****

* **Operadores Lógicos**

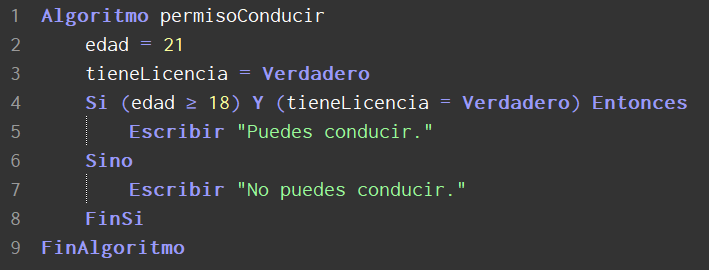
Se utilizan para evaluar múltiples condiciones:

**Y (AND)** = Devuelve Verdadero si ambas condiciones son verdaderas.

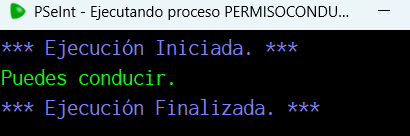
**O (OR)** = Devuelve Verdadero si al menos una condición es verdadera.

**NO (NOT)** = Invierte el valor lógico.

**Ejemplo: Verificación de permiso para conducir**

****

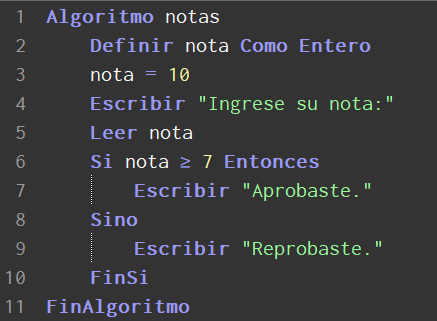
**Lo que muestra en la terminal será lo siguiente:**

****

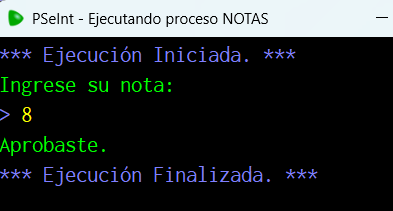
# *Estructuras de control: Condicionales y Bucles*

* **Estructuras Condicionales ( Si – Entonces – Sino )**

Permiten ejecutar un bloque de código solo si se cumple una condición.

**Ejemplo: Comprobación de notas para ver si aprobó o reprobó**

**Lo que muestra en la terminal es lo siguiente:**

****

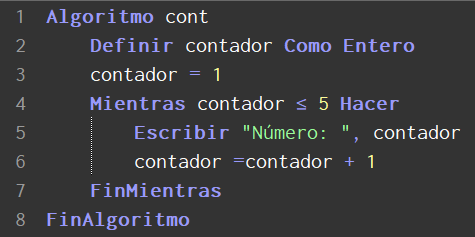
* **Estructuras Repetitivas (Bucles)**

Los bucles permiten repetir una acción varias veces.

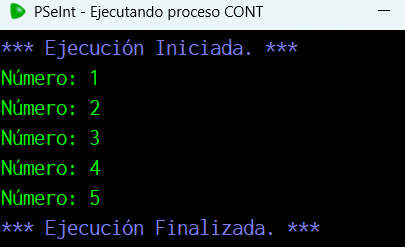
1. **Bucle – Mientras que (While)**

Ejecuta un conjunto de instrucciones **mientras que** una condición sea verdadera.

**Ejemplo: Contadores de número**



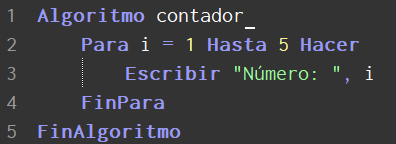
**Lo que muestra en la terminal es lo siguiente:**



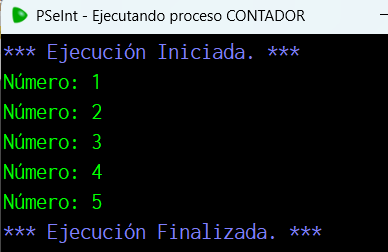
1. **Bucle – Para (For)**

Se usa cuando conocemos la cantidad exacta de repeticiones.

**Ejemplo: Contadores de números**



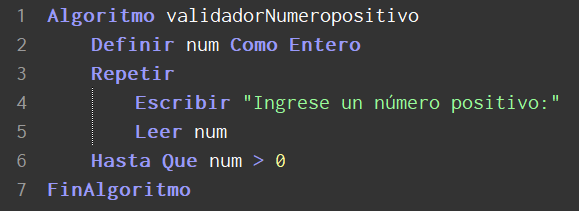
**Lo que muestra en la terminal es lo siguiente:**



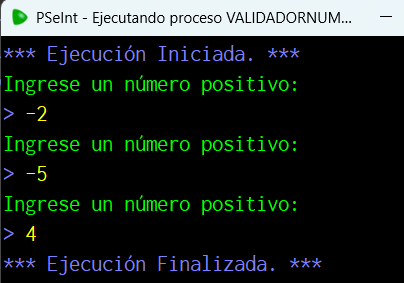
1. **Bucle – Repetir hasta que (Do While)**

Ejecuta las instrucciones al menos una vez, y luego verifica la condición.

**Ejemplo: Validación de numero positivos**

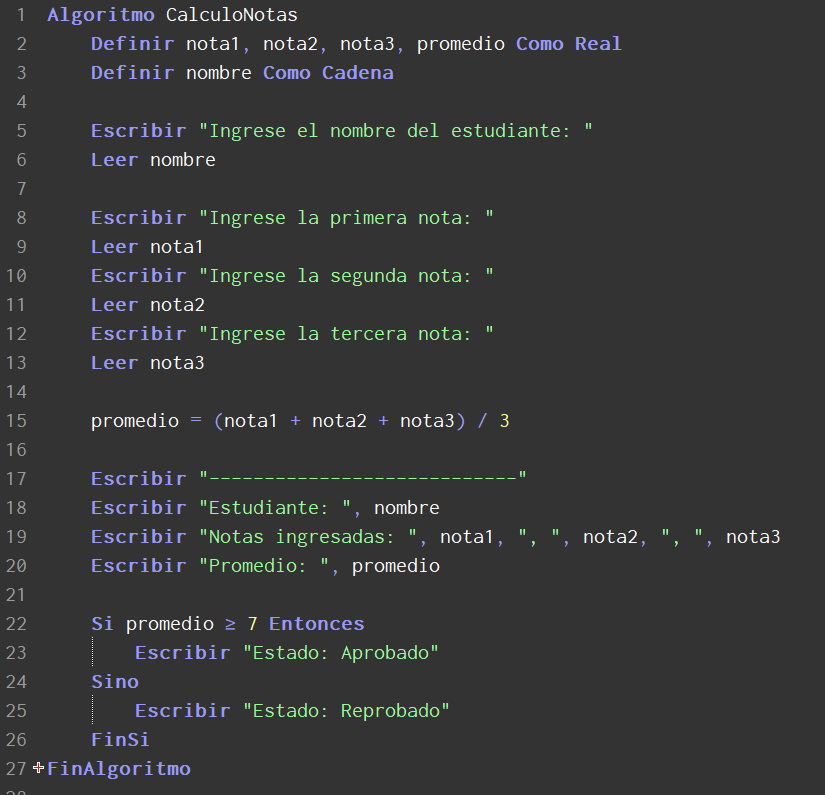
****

**Lo que muestra en la terminal es lo siguiente:**

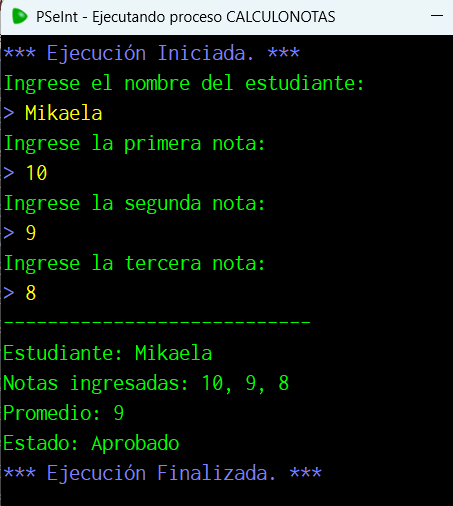
****

1. ***Ejemplo completo de algoritmo en PSeint***

* Es un programa que gestiona las calificaciones de un estudiante, permitiendo ingresar sus notas y calcular el promedio para determinar si aprobó o reprobó.



**Lo que muestra en la terminal es lo siguiente:**

****

1. ***Conclusión:***

El pseudocódigo es una herramienta importante para diseñar algoritmos de manera estructurada antes de programarlos. Permite organizar la lógica de un programa usando instrucciones claras y estructuradas. Con el uso de variables, operadores, estructuras de control y subprocesos, se pueden crear algoritmos eficientes y fáciles de entender.

La estructura básica de un algoritmo en **PSeInt** incluye:

* Inicio y fin *(Algoritmo - FinAlgoritmo*).
* Declaración de variables *(Definir).*
* Entrada y salida de datos *(Leer - Escribir).*
* Operaciones y cálculos. (+, -, \*, /)
* Estructuras de control *(Si - Mientras - Para - Repetir).*