

“Práctica 5”

**Objetivo:**

El alumno elaborará pseudocódigos que representen soluciones algorítmicas empleando la sintaxis y semántica adecuadas.

**Desarrollo :**

* Comenzando la práctica se dio un breve resumen sobre el pseudocódigo diciéndonos que es una forma de escribir los pasos para resolver un problema, usando palabras sencillas en lugar de un lenguaje de programación. Después de entender y planear cómo solucionar el problema, el pseudocódigo ayuda a organizar las ideas antes de convertirlas en código real.
* Después continuamos con sus reglas :

1. Alcance del programa: Todo pseudocódigo está limitado por las etiquetas de

INICIO y FIN. Dentro de estas etiquetas se deben escribir todas las instrucciones

del algoritmo.

2. Palabras reservadas con mayúsculas: Todas las palabras propias del pseudocódigo

deben de ser escritas en mayúsculas.

3. Sangría o tabulación: El pseudocódigo debe tener diversas alineaciones para que

el código sea más fácil de entender y depurar.

4. Lectura / escritura: Para indicar lectura de datos se utiliza la etiqueta LEER. Para

indicar escritura de datos se utiliza la etiqueta ESCRIBIR.

5. Declaración de variables: la declaración de variables la definen un identificador

(nombre), seguido de dos puntos, seguido del tipo de dato

6. Operadores aritméticos: Se tiene la posibilidad de utilizar operadores aritméticos y

lógicos

7. Notación de camello. Para nombrar variables y nombres de funciones se debe hacer

uso de la notación de camello.

* Despues las alabras reservadas :

INICIO / FIN: Indican el comienzo y el final del algoritmo.

SI / ENTONCES / SINO: Se usan para tomar decisiones, equivalentes a las estructuras condicionales.

MIENTRAS / FIN MIENTRAS: Para crear bucles que se repiten mientras una condición sea verdadera.

PARA / FIN PARA: Se usa para definir bucles con un número determinado de repeticiones.

HACER: Indica las acciones que se deben ejecutar dentro de bucles o condiciones.

LEER: Para la entrada de datos.

ESCRIBIR: Para mostrar o imprimir resultados.

REPETIR / HASTA QUE: Define un bucle que se repite hasta que se cumpla una condición.

Despues se hizo un recuadro con las diferentes estructuras y como son represetadas en el pseudocódigo:

| **Estructura** | **Diagrama** | **Seundocodigo** |
| --- | --- | --- |
| **Secuencial** | Paralelograma  pasted-image.pngEs un conjuntos de acciones en que se ejecutan sucesivamente sin poder regresar o sobrepasar alguna accion | Escribir/Imprimir  Leer |
| **Condicional**   * Simple | Es cuando le dices a un programa que haga algo solo si se cumple una condición, sii la condición no se cumple, el programa simplemente sigue adelante sin hacer nada especial. | Si (enológica) entonces  Instrucciones  Finsi |
| **Condicional**   * Compuesta | Es una forma de tomar decisiones en un programa donde se tienen varias condiciones, si la primera condición no se cumple, el programa verifica otra condición, y así sucesivamente, hasta que encuentra una que sea verdadera o termina | Si expresión logica  ENTONES  (Acciones ENTCES)  DELOCONTRARIO  (aciones DE LO CONTRARIO)  FIN DEL SI |
| **Condicional**   * Multiple | Es una forma de tomar decisiones en un programa, nos permite elegir entre varias opciones posibles según diferentes condiciones | SELECCIONAR (variable) En CASO valor1->  (Acciones)  CASO valor2->  (Acciones)  CASO valor3->  (Acciones)  DEFECTO ->  (Acciones)  FIN SELECCIONAR |
| **terativa**   * Mientras | Es un especie de bucle que repite una acción varias veces mientras se cumpla una condición | MIENTRAS condicion  ENTONCES  [Acciones]  FIN MIENTRAS |
| **terativa**   * Hacer mientras | Es un tipo de bucle que repite un conjunto de instrucciones mientras una condición sea verdadera | REPETIR  [Instrucciones]  MIENTRAS condicion |
| **terativa**   * Para | Es un tipo de bucle que repite un conjunto de instrucciones un número determinado de veces | ES DONDE PODEMOS INCORPORAR A LOS ANTERIORES |

* Despues se hicieron actividades de clase realizando Estructura de ejemplo, Ejercicio solicitando un valor al usuario, ejercicio para completar datos.

Es posible declarar más de una variable de un mismo tipo de dato utilizando arreglos, indicando la cantidad de variables que se requieren, su sintaxis es la siguiente:

juego: REG

Nombre: CADENA

Consola:CADENA

Precio: ENTERP

FIN REG

Juego1 Rea juego -> variable llamada usuario de tipo registro

Juego1.nombre := Batman Arkham Knight

Juego1.consola := Xbox

Juego1.precio := 449

Juego2 Rea juego -> variable llamada usuario de tipo registro

Juego2.nombre := Batman Arkham Knight

Juego2.consola := Xbox

Juego2.precio := 449

Juego3 Rea juego -> variable llamada usuario de tipo registro

Juego3.nombre := Batman Arkham Knight

Juego3.consola := Xbox

Juego3.precio := 449

INICIO

valorInicial, valorFinal: ENTERO

valorInicial:=0

ESCRIBIR “Indica el valor de valorFinal”

LEER “Valor\_Final”

MIENTRAS valorInicial < valorFinal

ESCRIBIR valorInicial

valorInicial

FIN MIENTRAS

**Tarea:**

1.- Desarrolla un algoritmo en pseudocódigo que permita calcular el promedio de las notas de un estudiante. El algoritmo debe permitir al usuario ingresar las notas de 5 asignaturas y calcular el promedio. Si el promedio es mayor o igual a 6, se mostrará un letrero de aprobado con la calificación obtenida, en caso contrario le mostrará un letrero de Reprobado, intentarlo de nuevo.

1. Iniciar
2. Sumar las notas de 5 asignaturas
3. Calcular el promedio
4. Si el promedio >= 6

* Mostrar "Aprobado con calificación: [promedio]"

1. Sino

* Mostrar "Reprobado, intentarlo de nuevo"

1. Fin

2.-Genera el pseudocódigo del siguiente ejercicio que trabajaste en la práctica 4.

Solicitar al usuario los centímetros cúbicos para el pago de agua y debe ser mayor a cero, en caso contrario se le debe volver a pedir el valor, los rangos son: De 1 a 500 centímetros cúbicos pagará 5.45 por centímetro cúbico , de 501 a 750 centímetros cúbicos pagará 7.49 por centímetro cúbico y mayor a 750 centímetros cúbicos pagará 10.3 por centímetro cúbico, además si excede los 200 pesos se le aplicará un descuento del %10.

1. Iniciar
2. Pedir al usuario los centímetros cúbicos
3. Mientras el valor <= 0

* Pedir nuevamente los centímetros cúbicos

1. Si centímetros cúbicos <= 500

* Calcular pago = centímetros cúbicos \* 5.45

1. Sino si centímetros cúbicos <= 750

* Calcular pago = centímetros cúbicos \* 7.49

1. Sino

* Calcular pago = centímetros cúbicos \* 10.3

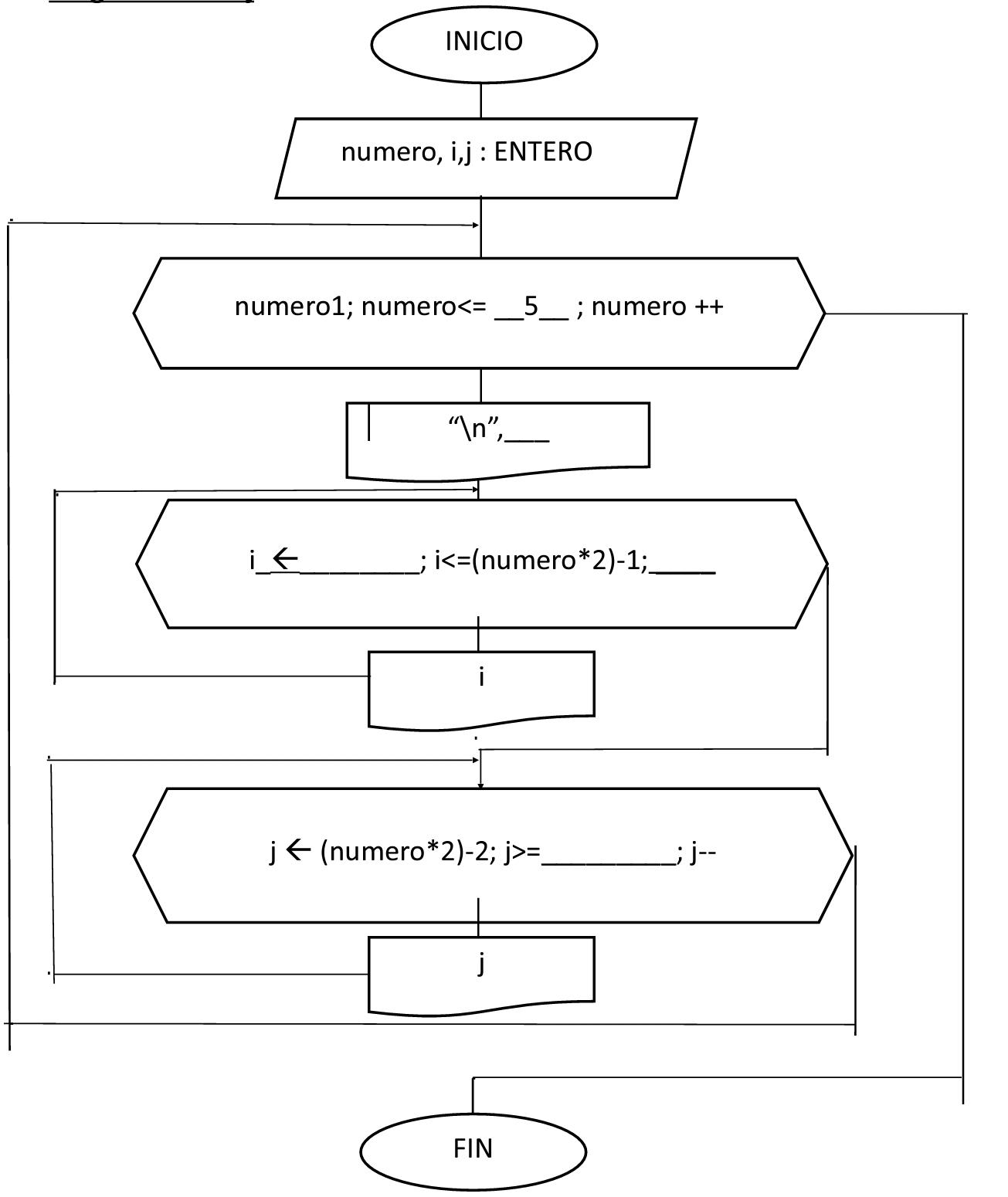
1. Si pago > 200

* Aplicar descuento del 10% al pago

1. Mostrar el monto final a pagar
2. Fin

3.- Pseudocódigo que muestre el ejercicio de la hoja utilizando la estructura MIENTRAS en lugar de PARA.

Análisis: Se va a imprimir patrones de números usando bucles. Generando secuencias numéricas organizadas en varias líneas, donde cada línea contiene diferentes números calculados según las iteraciones de los bucles.

Diafragma de flujo:

Pseudocódigo:

INICIO numero,i,j: ENTERO Para numero\_\_\_\_=1\_\_\_\_ Hasta 5 Con incremento numero++ Hacer Escribir \_\_\_\*\n\*\_\_,numero; Para i=numero+1 Hasta \_\_\_\_5\_\_\_\_\_ Con incremento i++ Hacer Escribir \_\_\_i\_\_\_ Fin Para Para j=(numero\*2)-2 Hasta numero Con decremento j-- Hacer Escribir \_\_\_\_\_ j \_\_\_\_\_ Fin Para Fin Para FIN

4.- Realiza tus pruebas de escritorio

(Prueba de escritorio del ejercicio 1)

| **Entrada** | **Proceso** | **Salida** |
| --- | --- | --- |
| 5,6,7,8,9 | 5+6+7+8+9=35  35÷5= 7 | Aprobado |
| 6,5,4,7,5 | 6+5+4+7+5=27  27÷5=5.4 | Reprobado |
| 6,8,9,10,9 | 6+8+9+10+9=42  42÷5=8.4 | Aprobado |

(Prueba de escritorio del ejercicio 2)

| **Entrada** | **Proceso** | **Salida** |
| --- | --- | --- |
| 400 | 400×5.45=218 pesos  218×0.10=21.8 pesos  218−21.8=196.2 pesos | Total a pagar: 196.2 pesos |
| 600 | 600×7.49=449.4 pesos  449.4×0.10=44.94 pesos  449.4−44.94=404.46 pesos | Total a pagar: 404.46 pesos |
| 750 | 750×7.49=561.75 pesos  561.75×0.10=56.18 pesos  561.75−56.18=505.57 pesos | Total a pagar: 505.57 pesos |

**Conclusión :**

En conclusión el pseudocódigo es una forma sencilla de planificar cómo resolver un problema. Nos ayuda a expresar ideas de manera clara, sin ser tan complicado como las reglas de un lenguaje de programación. Usarlo hace que sea más fácil darse a entender y enseñar las soluciones lo que nos ayuda a programar de forma más eficiente.

**Bibliografía:**

(En este caso no fue necesario hacer investigación, por lo que no fue necesario hacer uso de búsqueda)