|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Profa. Karina Garcia Morales |
| *Asignatura:* | Fundamentos de la programación. |
| *Grupo:* | 22 |
| *No. de práctica(s):* | 5 |
| *Integrante(s):* |  |
| *No. de lista o brigada:* | 14 |
| *Semestre:* | Primero |
| *Fecha de entrega:* |  |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**OBJETIVO:**

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

**CONCEPTOS:**

Reglas de generación de un seudocódigo: No hay una forma estándar de escribir un programa en pseudocódigo, ya que depende en gran medida del criterio del programador. No obstante, es aconsejable respetar lo más rigurosamente posible las siguientes normas de carácter general.

1. Todo programa escrito en pseudocódigo comienza con la palabra «Inicio» y termina con la palabra «Fin».  
2. Cada instrucción se debe escribir en una línea.  
3. Para su descripción se utiliza una serie de palabras reservadas, tales como: «Inicio», «Fin», «Si», «Entonces», «Si no», «Fin\_si», «Mientras», «Fin\_Mientras», Seguir», «Fin\_Seguir», que tienen un significado específico.  
4. Debe escribirse indentado (tabulando) para mostrar claramente las dependencias de unas instrucciones del programa con respecto a otras.  
5. Cada una de las estructuras que definen un programa tendrá un punto de comienzo y uno de fin, especificados por las palabras reservadas que definen la estructura.  
6. Se escribirá en minúsculas, excepto aquellos nombres que elija el programador (como NUM, MAYOR, MENOR, etc…) que irán en mayúsculas.

Palabras reservadas:

|  |  |
| --- | --- |
| **PALABRA** | **FUNCIÓN** |
| **Algoritmo** | Se utiliza en la cabecera de un [algoritmo](https://www.abrirllave.com/pseudocodigo/estructura-de-un-algoritmo.php), precediendo al nombre ([identificador](https://www.abrirllave.com/pseudocodigo/identificadores.php)) que se le asigne. |
| **Cadena** | [Tipo de dato cadena](https://www.abrirllave.com/pseudocodigo/datos-de-tipo-cadena.php). |
| **Caracter** | [Tipo de dato carácter](https://www.abrirllave.com/pseudocodigo/datos-de-tipo-caracter.php). |
| **Como** | Se utiliza para indicar el tipo de dato de una [variable](https://www.abrirllave.com/pseudocodigo/variables.php) en su definición. |
| **Con Paso** | En una instrucción repetitiva para ([**Para**](https://www.abrirllave.com/pseudocodigo/repetitiva-para.php)), se utiliza para indicar el ***<valor\_incremento>*** que se sumará, en cada iteración del bucle, a la variable inicializada. |
| **De Otro Modo** | Caso por defecto en una instrucción de control alternativa múltiple (**[Segun](https://www.abrirllave.com/pseudocodigo/alternativa-multiple.php)**). |
| **Definir** | Sirve para declarar [variables](https://www.abrirllave.com/pseudocodigo/variables.php). |
| **Dimension** | Sirve para declarar [arrays](https://www.abrirllave.com/pseudocodigo/arrays.php). |
| **Entero** | [Tipo de dato entero](https://www.abrirllave.com/pseudocodigo/datos-de-tipo-entero.php). |
| **Entonces** | Se utiliza en instrucciones de control alternativas simples ([**Si**](https://www.abrirllave.com/pseudocodigo/alternativa-simple.php)) y dobles ([**Si SiNo**](https://www.abrirllave.com/pseudocodigo/alternativa-doble.php)) precediendo al bloque de instrucciones que se ejecutará en el caso de que la condición de la instrucción de control sea verdadera. |
| **Escribir** | Se emplea para escribir una [instrucción de salida](https://www.abrirllave.com/pseudocodigo/salida.php). |
| **Falso** | Posible valor de un tipo de dato **[Logico](https://www.abrirllave.com/pseudocodigo/datos-de-tipo-logico.php)**. |
| **FinAlgoritmo** | Indica la finalización de un [algoritmo](https://www.abrirllave.com/pseudocodigo/estructura-de-un-algoritmo.php). |
| **FinFuncion** | Indica la finalización de una [función](https://www.abrirllave.com/pseudocodigo/funciones.php). |
| **FinMientras** | Indica la finalización de un bucle [**Mientras**](https://www.abrirllave.com/pseudocodigo/repetitiva-mientras.php). |
| **FinPara** | Indica la finalización de un bucle [**Para**](https://www.abrirllave.com/pseudocodigo/repetitiva-para.php). |
| **FinSegun** | Indica la finalización de una instrucción de control alternativa múltiple (**[Segun](https://www.abrirllave.com/pseudocodigo/alternativa-multiple.php)**). |
| **FinSi** | Indica la finalización de instrucciones de control alternativas simples ([**Si**](https://www.abrirllave.com/pseudocodigo/alternativa-simple.php)) y dobles ([**Si SiNo**](https://www.abrirllave.com/pseudocodigo/alternativa-doble.php)). |
| **FinSubProceso** | Indica la finalización de un [procedimiento](https://www.abrirllave.com/pseudocodigo/procedimientos.php). |
| **Funcion** | Se utiliza en la cabecera de una [función](https://www.abrirllave.com/pseudocodigo/funciones.php), precediendo al nombre ([identificador](https://www.abrirllave.com/pseudocodigo/identificadores.php)) que se le asigne. |
| **Hacer** | Se emplea en la sintaxis de la instrucción de control alternativa múltiple (**[Segun](https://www.abrirllave.com/pseudocodigo/alternativa-multiple.php)**), y también, en la sintaxis de las instrucciones de control repetitivas [**Mientras**](https://www.abrirllave.com/pseudocodigo/repetitiva-mientras.php) y [**Para**](https://www.abrirllave.com/pseudocodigo/repetitiva-para.php). |
| **Hasta** | En una instrucción repetitiva [**Para**](https://www.abrirllave.com/pseudocodigo/repetitiva-para.php), se utiliza para indicar el ***<valor\_final>*** que deberá tener la variable inicializada, para que el bucle finalice. |
| **Hasta Que** | En una instrucción repetitiva [**Repetir**](https://www.abrirllave.com/pseudocodigo/repetitiva-repetir.php), se escribe precediendo a la condición que deberá ser verdadera para que el bucle finalice. |
| **Leer** | Se emplea para escribir una [instrucción de entrada](https://www.abrirllave.com/pseudocodigo/entrada.php). |
| **Logico** | [Tipo de dato lógico](https://www.abrirllave.com/pseudocodigo/datos-de-tipo-logico.php). |
| **Mientras** | Se usa para escribir una instrucción de control repetitiva [**Mientras**](https://www.abrirllave.com/pseudocodigo/repetitiva-mientras.php). |
| **Para** | Se usa para escribir una instrucción de control repetitiva [**Para**](https://www.abrirllave.com/pseudocodigo/repetitiva-para.php). |
| **Por Referencia** | En la cabecera de un subalgoritmo ([procedimiento](https://www.abrirllave.com/pseudocodigo/procedimientos.php) o [función](https://www.abrirllave.com/pseudocodigo/funciones.php)), se utiliza para indicar que un parámetro se pasa por referencia. |
| **Por Valor** | En la cabecera de un subalgoritmo ([procedimiento](https://www.abrirllave.com/pseudocodigo/procedimientos.php) o [función](https://www.abrirllave.com/pseudocodigo/funciones.php)), se utiliza para indicar que un parámetro se pasa por valor. |
| **Real** | [Tipo de dato real](https://www.abrirllave.com/pseudocodigo/datos-de-tipo-real.php). |
| **Segun** | Se usa para escribir una instrucción de control [alternativa múltiple](https://www.abrirllave.com/pseudocodigo/alternativa-multiple.php). |
| **Si** | Se usa para escribir instrucciones de control alternativas [simples](https://www.abrirllave.com/pseudocodigo/alternativa-simple.php) y [dobles](https://www.abrirllave.com/pseudocodigo/alternativa-doble.php). |
| **Sin Saltar** | En una instrucción de salida ([**Escribir**](https://www.abrirllave.com/pseudocodigo/salida.php)), se emplea para indicar que, después de su ejecución, el cursor no debe "saltar" a la siguiente línea. |
| **SiNo** | Se usa en instrucción de control [alternativa doble](https://www.abrirllave.com/pseudocodigo/alternativa-doble.php) precediendo al bloque de instrucciones que se ejecutará en el caso de que la condición sea falsa. |
| **SubProceso** | Se utiliza en la cabecera de un [procedimiento](https://www.abrirllave.com/pseudocodigo/procedimientos.php), precediendo al nombre ([identificador](https://www.abrirllave.com/pseudocodigo/identificadores.php)) que se le asigne. |
| **Verdadero** | Posible valor de un tipo de dato **[Logico](https://www.abrirllave.com/pseudocodigo/datos-de-tipo-logico.php)**. |

Estructuras:

|  |  |  |
| --- | --- | --- |
| **Estructura** | **Diagrama** | **Seudocódigo** |
| Secuencial |  | Escribir/imprimir  Leer |
| Condicional |  |  |
| * Simple | Diagrama  Descripción generada automáticamente | SI (exp lógica) entonces  Instrucción  FIN SI |
| * Compuesta | Diagrama  Descripción generada automáticamente | SI <expresión lógica> ENTONCES  Bloque 1  SINO  Bloque 2  FIN SI |
| * Múltiple |  | SELECCIONAR (variable) EN  CASO valor1 -> [Instrucción]  CASO valor2 -> [Instrucción]  CASO valor3 -> [Instrucción]  DEFECTO -> [Instrucción]  FIN SELECCIONAR |
| Iterativa |  |  |
| * Mientras | no  si | MIENTRAS que <expresión lógica>  Instrucciones  FIN DEL MIENTRAS |
| * Hacer mientras | no  si | Hacer  Instrucciones  Mientras <expresión lógica> |
| * Para |  | PARA (valor inicial, condición, incremento/decremento)  Instrucciones  FIN PARA |

**ACTIVIDAD:**

registro de videojuegos

juego:REG

nombre: CADENA

consola: CADENA

precio: ENTERO

FIN REG

Juego1:REG juego -> variable llamada usuario de tipo registro

Juego1.nombre := “Devil May Cry 5”

Juego1.consola := “Ps/Xbox/pc”

juego1.precio := 600

Juego2:REG juego -> variable llamada usuario de tipo registro

Juego2.nombre := “Marvel’s Spider-man 2”

Juego2.consola := “Ps5”

Juego2.precio := 1,500

Juego3:REG juego -> variable llamada usuario de tipo registro

Juego3.nombre := “HALO REACH”

Juego3.consola := “Xbox 360”

Juego3.precio := 250

INICIO

valorInicial,valorFinal:ENTERO

valorInicial:=0

ESCRIBIR ingrese valorFinal

LEER valorFinal

MIENTRAS

valorInicial < valorFinal

ESCRIBIR valorInicial valorInicial := valorInicial + 1

FIN MIENTRAS

FIN

prueba de escritorio

|  |  |  |  |
| --- | --- | --- | --- |
| instruccion | valorInicial | valorFinal | salida |
| valorInicial := 0 | 00 |  |  |
| valorFinal := 5 |  | 2 |  |
| valorInicial < valorFinal |  |  |  |
|  |  |  | 0 |
| valorInicial := valorInicial +1 | 1 |  |  |
| valorInicial < valorFinal |  |  |  |
|  |  |  | 1 |
| valorInicial := valorInicial +1 | 2 |  |  |
| valorInicial < valorFinal |  |  |  |
|  |  |  | 2 |

Diagrama

Descripción generada automáticamente

**ACTIVIDAD DE 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.

INICIO

// Declaración de variables

DEFINIR nota1, nota2, nota3, nota4, nota5 COMO REAL

DEFINIR promedio COMO REAL

// Entrada de las 5 notas

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente ESCRIBIR "Ingrese la nota de la primera asignatura: "

LEER nota1

ESCRIBIR "Ingrese la nota de la segunda asignatura: "

LEER nota2

ESCRIBIR "Ingrese la nota de la tercera asignatura: "

LEER nota3

ESCRIBIR "Ingrese la nota de la cuarta asignatura: "

LEER nota4

ESCRIBIR "Ingrese la nota de la quinta asignatura: "

LEER nota5

// Cálculo del promedio

promedio <- (nota1 + nota2 + nota3 + nota4 + nota5) / 5

// Evaluación del promedio

SI promedio >= 6 ENTONCES

ESCRIBIR "Aprobado con un promedio de: ", promedio

SINO

ESCRIBIR "Reprobado, intentarlo de nuevo. Promedio: ", promedio

FIN SI

Fin

1. 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.

Inicio

Definir var Como Entero

var <- cm^3

Repetir

Escribir ingresa, el, valor, de, var

Leer var

Hasta Que var>0

Si var>500 Entonces

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente Escribir var

Si var>750 Entonces

var <- 10.3\*var

SiNo

var <- 7.49\*var

FinSi

SiNo

Escribir var

var <- 5.45\*var

FinSi

Si var>200 Entonces

var <- var-var%10

SiNo

Escribir var

FinSi

Escribir var

Fin

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

Imagen que contiene Texto

Descripción generada automáticamenteInicio

Definir número, i, j Como ENTERO

número <- 1

MIENTRAS número <= 5 HACER

Escribir "\n", número;

// Inicializar i

i <- número + 1

MIENTRAS i <= (número \* 2 - 1) HACER

Escribir i;

i <- i + 1

FIN MIENTRAS

// Inicializar j

j <- (número \* 2 - 2)

MIENTRAS j >= número HACER

Escribir j;

j <- j - 1

FIN MIENTRAS

// Incrementar número para el próximo ciclo

número <- número + 1

FIN MIENTRAS

Fin

Conclusiones:

En esta practica tal y como lo dijo el objetivo escribí varios pseudocódigos que en cada problema me hice me enseñaron un poco a aplicar la sintaxis adecuada, mostrándome como utilizar los mismos conceptos en la construcción del diagrama de flujo pero ahora en forma de pseudocódigos.

Aunque algo que agregar es que todos los pseudocódigos de tarea los evalué en Pseint y tuve problemas con el de la hoja de tarea, me costo mucho poder ejecutarlo, al punto de pedirle ayuda a una IA a solucionarlo, eso sería todo, lo demás de la practica lo pude hacer con normalidad.

**BIBLIOGRAFÍA:**

* Huerta, A. (2016). REGLAS PARA ELABORAR UN PSEUDOCODIGO Y UN DIAGRAMA DE FLUJO. WordPress. recuperado el 11 de septiembre del 2024 de

<https://andreshuertawordpress.wordpress.com/2016/10/05/reglas-para-elaborar-un-pseudocodigo-y-un-diagrama-de-flujo/>

* <https://www.abrirllave.com/pseudocodigo/palabras-clave.php>
* las estructuras y su pseudocódigo en parte fueron sacadas de la presentación de la profesora que nos mostró en la clase.

**LIGA GITHUB:**