

Asignatura: Metodología de la Programación

Profesores:

Ing. Gabriel Barrón Rodríguez

Ing. Apolinar Trejo Cuevas

Ing. Ricardo Muro Gómez

Ing. Daniel Sustaita Cruces

Unidad Temática:

III. Algoritmos, diagramas de flujo y pseudocódigos

Temas:

Reglas para elaborar diagramas de flujo y pseudocódigos

Contadores y Acumuladores

Estructuras de control de selección

Estructuras de control repetitivas.

Implementación de algoritmos en un lenguaje de programación

Grupo: Gds0611

Alumno(a):

Hernandez Luna Jesus Alejandro

I.- temas – Reglas para elaborar diagramas de flujo y pseudocódigos. Identificación de conceptos de unidad (saber) – 10 puntos,

1. 1 cada una. CUMPLIMIENTO TOTAL DE TAREAS (saber hacer) – 10 puntos,
2. 2 cada algoritmo. Instrucciones: 1. Lee cuidadosamente el enunciado del problema. 2. Identifica la salida esperada o resultado esperado.
3. Identifica las entradas suficientes para obtener la salida esperada.
4. Diseña la solución en pseudocódigo con la herramienta PseInt.
5. Designa el nombre del algoritmo de acuerdo a las reglas de identificadores, recuerda que el título indica de que trata el diseño de la solución del algoritmo.
6. Guarda la solución y hazlo corresponder con el nombre del algoritmo.
7. Documenta el código fuente: Descripción breve, autor y fecha de creación
8. Diseña la solución de acuerdo al problema planteado, documenta el uso de variables
9. Repite el paso 1 hasta que no haya problemas por resolver
10. Envía las soluciones de acuerdo a las indicaciones del facilitador(a).
11. Define los siguientes comandos con tus propias palabras en un documento electrónico con extensión .pdf o .docx que incluya todas las definiciones de cada tema proporcionados en el presente instrumento, con portada y que responda a las siguientes preguntas:

|  |  |  |
| --- | --- | --- |
| Comando | Descripción | Ejemplo |
| Proceso | Un serie de pasos concretos en forma de lista a seguir para ejecutar un algoritmo |  |
| Escribir | comando para mostrar una orden o acción de en nuestro código que puede obtener una respuesta |  |
| Leer | Un comando para guardar el resultado de una de nuestro código |  |
| Asignar | Comando que otorga un valor a una variable |  |
| Definir | Otorga un tipo de variable ya sea, tipo de carácter,entero,real y lógico. |  |
| Variable | Un carácter que puedes asignar que cambie de valor constante mente |  |
| Expresión | Combinación de valores y operaciones que al ser evaluados toman un valor |  |
| Identificador | Nombres que se proporcionan para las variables |  |
| Reglas de identificadores | * No llevar una letra mayúscula al principio * No debe llevar espacios o caracteres especiales * No puede llevar tildes * No debe coincidir con variables * Deben ser nombres significativos al programa |  |

|  |  |  |
| --- | --- | --- |
| comando | Descripción | ejemplo |
| Si-entonces | Es cuando una variable tiene ciertos requisitos podemos ponerlos para que realice una tarea especifica |  |
| Si-entonces-sino | Es la parte negativa al Si en caso de que no lo cumpla para realizar otra opción |  |
| Según | Permite tomar varias decisiones complejas en función del valor de una variable o expresión |  |
| Operadores lógicos | Son el “And”, “or”, “not” son operadores basados en tablas de verdad y ejecutan una sentencia |  |
| Operadores relaciones | Son los operadores =,<>,>=,<= que nos ayudan a evaluar expresiones del tipo lógico |  |

|  |  |  |
| --- | --- | --- |
| comando | Descripción | ejemplo |
| Para | Es un comando que nos ayuda a repetir hasta cierto punto una serie de instrucciones |  |
| Mientras | si es verdadera la condición ejecuta una serie de instrucciones |  |
| Repetir-hasta que | Repite una serie de instrucciones hasta que la opción sea verdadera |  |
| Contador | Variable de tipo entero que con el uso del programa va aumentando valor su constantemente |  |
| acumulador | Una variable que va sumándose así misma valores de otras variables |  |