



# PORTAFOLIO

## ALGORITMO 1



CICLO I-2013

## GENERALIDADES DE LA MATERIA

### INTRODUCCION

El curso, ofrece los conceptos y términos técnicos de la lógica algorítmica, con el objetivo de su utilización en el ámbito de la automatización y desarrollo de soluciones a diversos problemas, en la práctica, se trabaja con métodos y técnicas para elaborar algoritmos, se utiliza el recurso del Internet y finalmente se aprende el uso del programa PSEint.

Este curso de Algoritmo 1 ha sido diseñado para ser impartido sobre la base de clases no-presenciales, utilizando la plataforma de Moodle a través del portal de EVA dentro del sitio web de la Universidad. Esperamos gradualmente incorporar de forma más eficiente e integral el concepto de "Clases Virtuales" a la metodología empleada en la formación profesional

### OBJETIVOS

- Desarrollar un pensamiento lógico, analítico, inductivo y deductivo a partir del estudio y elaboración de algoritmos de complejidad moderada y su aplicación en un lenguaje de programación buscando su validación.
- Introducir en el uso de una metodología para resolver problemas calculables de moderada complejidad a través de modelos, herramientas y técnicas de programación formal.
- Introducir en la utilización de un lenguaje de programación en sus estructuras básicas.

### COMPETENCIAS

- Desarrollar la capacidad de análisis
- Modelar el problema, generando alternativas de solución
- Validar y refutar alternativas de solución con razonamiento lógico
- Desarrollar habilidades lógicas aplicadas a la solución de problemas
- Desarrollar la capacidad de análisis lógico y pensamiento ordenado, para los problemas matemáticos a resolver
- Desarrollar la capacidad creativa, enfrentando y resolviendo problemas en forma independiente del lenguaje utilizado



## Diagrama de flujo

Los diagramas de flujo son descripciones gráficas de algoritmos; usan símbolos conectados con flechas para indicar la secuencia de instrucciones.

Los diagramas de flujo son usados para representar algoritmos pequeños, ya que abarcan mucho espacio y su construcción es laboriosa. Por su facilidad de lectura son usados como introducción a los algoritmos, descripción de un lenguaje y descripción de procesos

### PSEUDOCODIGO

El pseudocódigo (*falso lenguaje*, el prefijo *pseudo* significa *falso*) es una descripción de alto nivel de un algoritmo que emplea una mezcla de lenguaje natural con algunas convenciones sintácticas propias de lenguajes de programación, como asignaciones, ciclos y condicionales, aunque no está regido por ningún estándar.

### ESTRUCTURAS SELECTIVAS

#### ESTRUCTURA SELECTIVA SIMPLE:

Se identifican porque están compuestos únicamente de una condición. La estructura si - entonces evalúa la condición y en tal caso:

Si la condición es verdadera, entonces ejecuta la acción Si (o acciones si son varias).

Si la condición es falsa, entonces no se hace nada.

Representación en Flujograma:

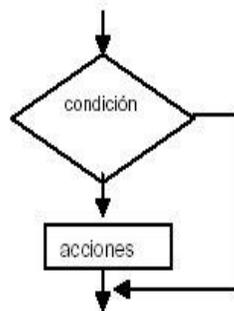
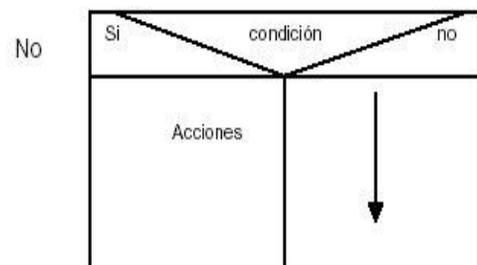


Diagrama N-S



ESTRUCTURA SELECTITA SIMPLE

# EJEMPLO

Construir un algoritmo, que dado como dato la calificación de un alumno en un examen, escriba "Aprobado" en caso que la calificación sea mayor que 8

# PSEUDOCODIGO

```

Inicio
Leer (cal)
Si cal >8 entonces
Escribir "APROBADO"
Fin Si
Fin
    
```

FLUJOGRAMA

Flujograma:

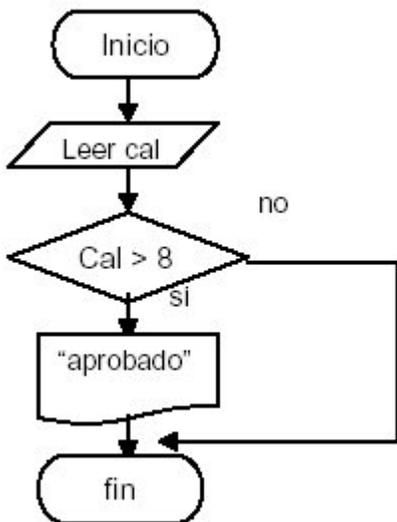
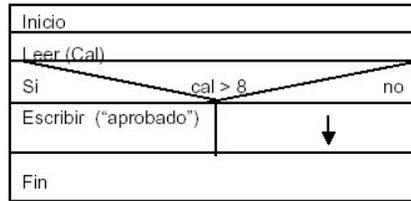


DIAGRAMA N-S

Diagrama N-S



**ESTRUCTURA SELECTIVA DOBLE:**

Son estructuras lógicas que permiten controlar la ejecución de varias acciones y se utilizan cuando se tienen dos opciones de acción, por la naturaleza de estas se debe ejecutar una o la otra, pero no ambas a la vez, es decir, son mutuamente excluyentes. Entonces, si una condición C es verdadera, se ejecuta la acción S1 y si es falsa, se ejecuta la acción S2.

ESTRUCTURA SELECTITA DOBLE

# EJEMPLO

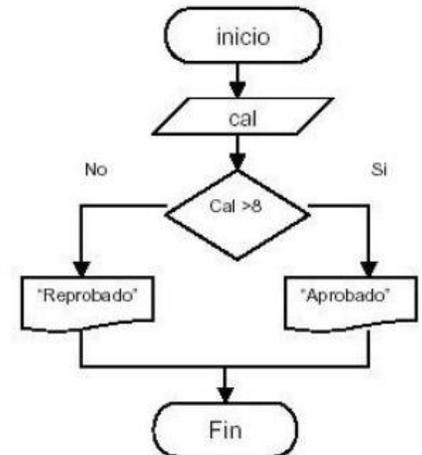
Construir un algoritmo, que dado como dato la calificación de un alumno en un examen, escriba "Aprobado" en caso que la calificación sea mayor que 8 y reprobado en caso contrario

# PSEUDOCODIGO

```

Inicio
Leer (cal)
Si cal >8 entonces
    Escribir "APROBADO"
Sino
    Escribir "REPROBADO"
Fin Si
Fin
    
```

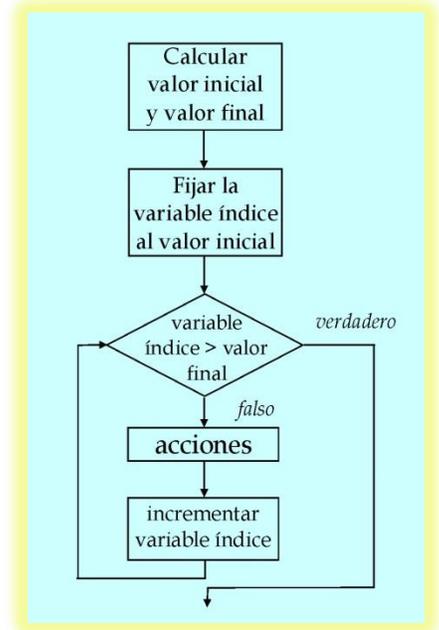
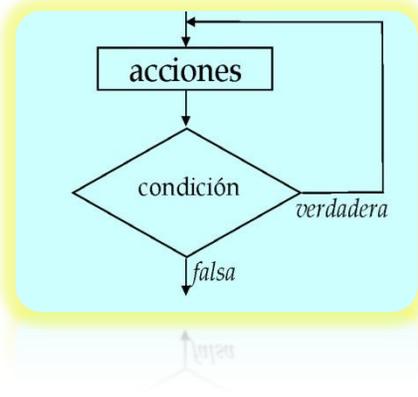
FLUJOGRAMA



ESTRUCTURAS REPETITIVAS

ESTRUCTURAS REPETITIVAS

Para crear una estructura repetitiva, también llamada (bucle, lazo o ciclo), necesitamos: 1) Una condición de conteo o condición lógica 2) Un contador, el cual se incrementa o decremento por cada iteración 3) Un conjunto de pasos del algoritmo los cuales se necesitan repetir



ESTRUCTURA REPETITIVA MIENTRAS

La estructura repetitiva Mientras (While) es aquella en la que el número de repeticiones del bucle no se conoce por anticipado, y el cuerpo del bucle se repite mientras se cumple una determinada condición.

Mientras (condicion) Hacer

Grupo de instrucciones

Fin Mientras

EJEMPLO

Desarrollar un algoritmo que escriba los primeros 100 números enteros

PSEUDOCÓDIGO

```

inicio
cont<-1

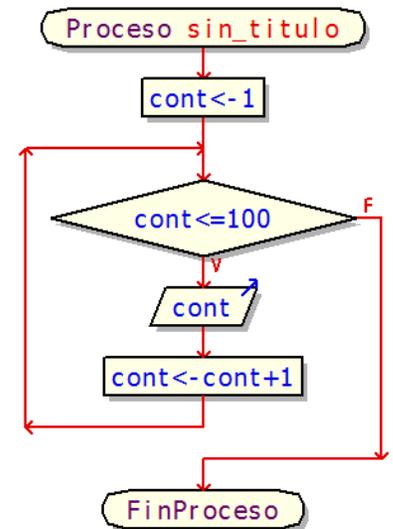
Mientras cont<=100 Hacer

    Imprimir cont

    cont=cont+1

Fin Mientras
Fin
    
```

FLUJOGRAMA



**ESTRUCTURA REPETITIVA DESDE-HASTA**

La estructura repetitiva Para-Hasta (For) es aquella en la que el número de repeticiones del bucle se conoce por anticipado, y el cuerpo del bucle se repite desde un valor inicial, hasta un valor final previamente fijado.

Para  $i=n$  hasta  $j=n$  Hacer

Grupo de instrucciones

FinDesde

# EJEMPLO

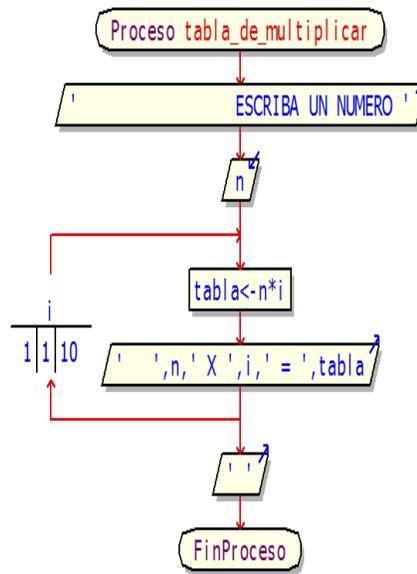
Desarrollar un algoritmo que imprima cualquier tabla de multiplicar

## PSEUDOCÓDIGO

```

Proceso tabla de multiplicar
Escribir "¿QUE TABLA DE MULTIPLICAR DESEA GENERAR?"
Escribir "ESCRIBA UN NUMERO"
Leer N;
para i=1 hasta 10 con paso 1
hacer
    tabla= n*i
    imprimir " ",n," X ",i, " = ",tabla
FinPara
Fin Proceso
    
```

**FLUJOGRAMA**



**ESTRUCTURA REPETIR HASTA**

La estructura repetitiva Repetir-Hasta (For) es la misma que la función Mientras, la diferencia es que la estructura mientras comprueba la condición al inicio y repetir lo hace al final; por eso la estructura repetir se ejecuta por lo menos una vez

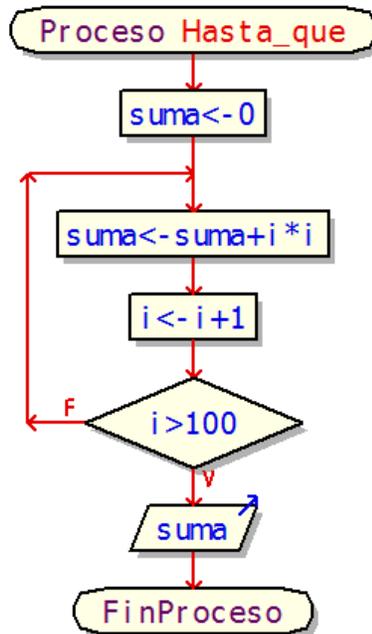
# EJEMPLO

Desarrollar un algoritmo que imprima la suma de los cuadrados de los primeros 100 números enteros

## PSEUDOCÓDIGO

```

Proceso prod_num
suma<-0
Repetir
    suma<-suma+i*i
    i<-i+1
Hasta Que i>100
Escribir suma
FinProceso
    
```



## ARREGLOS

Un arreglo es una estructura de datos, o un espacio de memoria que permite almacenar una colección de elementos, todos del mismo tipo. Es decir es una secuencia continua de celdas (espacios en memoria) o casillas de cada una de las cuales se puede guardar un elemento de la colección.

Los lenguajes de programación permiten al programador declarar arreglos de cualquier tamaño. En Pseudocódigo, un arreglo se declara usando el formato siguiente:

<Nombre>: Arreglo [N] de tipo

Ejemplo:

Letras: arreglo [15] de carácter

## PSEUDOCÓDIGO

Inicio

Nombres<-[1...5] de enteros

Para i<-1 Hasta 5 Hacer

Escribir "Digite un nombre"

Leer nombres [i]

FinPara

Para i<-1 hasta 5 Hacer

Escribir " El nombre de la posicion ", i, "es", nombres [i]

FinPara

Fin



## ARREGLOS UNIDIMENSIONALES

## EJEMPLO

Solicitar 5 nombres y luego escribirlos en pantalla

## EJECUCION COMPLETA:

```

*** Ejecucion Iniciada. ***
Digite un nombre
 analisis
Digite un nombre
 diseño
Digite un nombre
 desarrollo
Digite un nombre
 www.monje
Digite un nombre
 oneble.com
El nombre de la posicion 1es: analisis
El nombre de la posicion 2es: diseño
El nombre de la posicion 3es: desarrollo
El nombre de la posicion 4es: www.monje
El nombre de la posicion 5es: oneble.com
*** Ejecucion Finalizada. ***

```

## CONCLUSIONES

El algoritmo es un método para resolver un problema mediante una serie de pasos definidos, precisos y finitos. Diríamos que un algoritmo es una serie de instrucciones que colocadas en un orden lógico conducen a la solución de un problema. También se puede decir que un algoritmo es la fase preliminar al escribir un programa en cualquier lenguaje de programación, por lo que la forma de escribirlo puede ser muy personal según el programador que lo esté diseñando, pero en general se debe hacer en una forma tan clara que cualquier otro programador pueda coger dicho algoritmo, entenderlo fácilmente y pueda traducir a su lenguaje de programación favorito.

El programador de computadoras es antes que nada una persona que resuelve problemas, por lo que para llegar a ser un programador eficaz se necesita aprender a resolver problemas de un modo lógico, riguroso y sistemático.

En la vida cotidiana se emplean algoritmos frecuentemente para resolver problemas.

Algunos ejemplos son los manuales de usuario, que muestran algoritmos para usar un aparato, o las instrucciones que recibe un trabajador por parte de su patrón.

En términos generales un algoritmo debe presentar las siguientes características:

- **Realizable:** debe terminar después de una cantidad finita de pasos, un algoritmo es inaplicable cuando el proceso resulta infinito o no arroja ningún resultado.
- **Comprensible:** debe existir un procedimiento claro que determine el proceso de ejecución para que al momento de ser aplicado ya sea por un ser humano o una máquina sepa que, En términos generales un algoritmo debe presentar las siguientes características:
  - **Realizable:** debe terminar después de una cantidad finita de pasos, un algoritmo es inaplicable cuando el proceso resulta infinito o no arroja ningún resultado.
  - **Comprensible:** debe existir un procedimiento claro que determine el proceso de ejecución para que al momento de ser aplicado ya sea por un ser humano o una máquina sepa que hacer.

## ANEXOS

### ALGUNOS PROBLEMAS RESUELTOS DURANTE LA MATERIA.

1. Programa que dada 3 opciones de producto, calcula la cantidad de personas que los consumen

Proceso ejc5p3

Escribir ' ';

Escribir ' BIENVENIDO ';

Escribir ' ';

acumA<-0;

acumB<-0;

acumAYB<-0;

ning<-0;

DEFINIR Resp COMO CHARACTER;

Para l<-1 Hasta 30 Con Paso 1 Hacer

    Escribir ' ';

    Escribir ' ';

    Escribir ' ¿QUE PRODUCTO CONSUME EL CLIENTE? ',l;

    Leer Resp;

    Segun Resp Hacer

        'A','a':

            acumA<-acumA+1;

        'B','b':

            acumB<-acumB+1;

        'AYB','ayb':

            acumAYB<-acumAYB+1;

        'NINGUNO','ninguno':

            ning<-ning+1;

        De Otro Modo:

            Escribir ' DATO NO VALIDO, INTENTELE DE NUEVO ';

            error<-error+1;

    FinSegun

FinPara

Escribir ' DE LOS 30 ENTREVISTADOS PREFIEREN EL PRODUCTO ';

Escribir ' ';

Escribir ' A :',acumA;

Escribir ' B :',acumB;

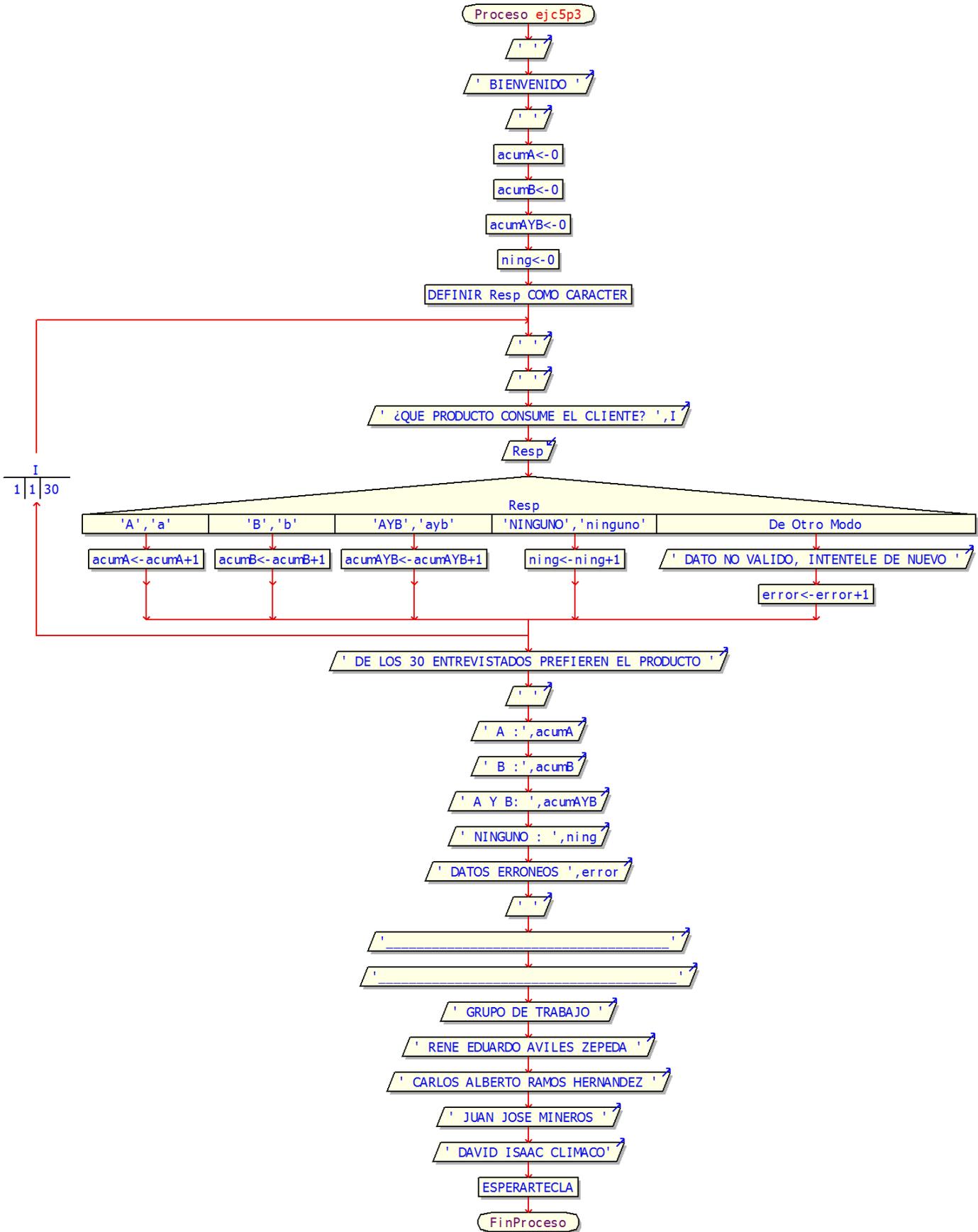
Escribir ' A Y B :',acumAYB;

Escribir ' NINGUNO :',ning;

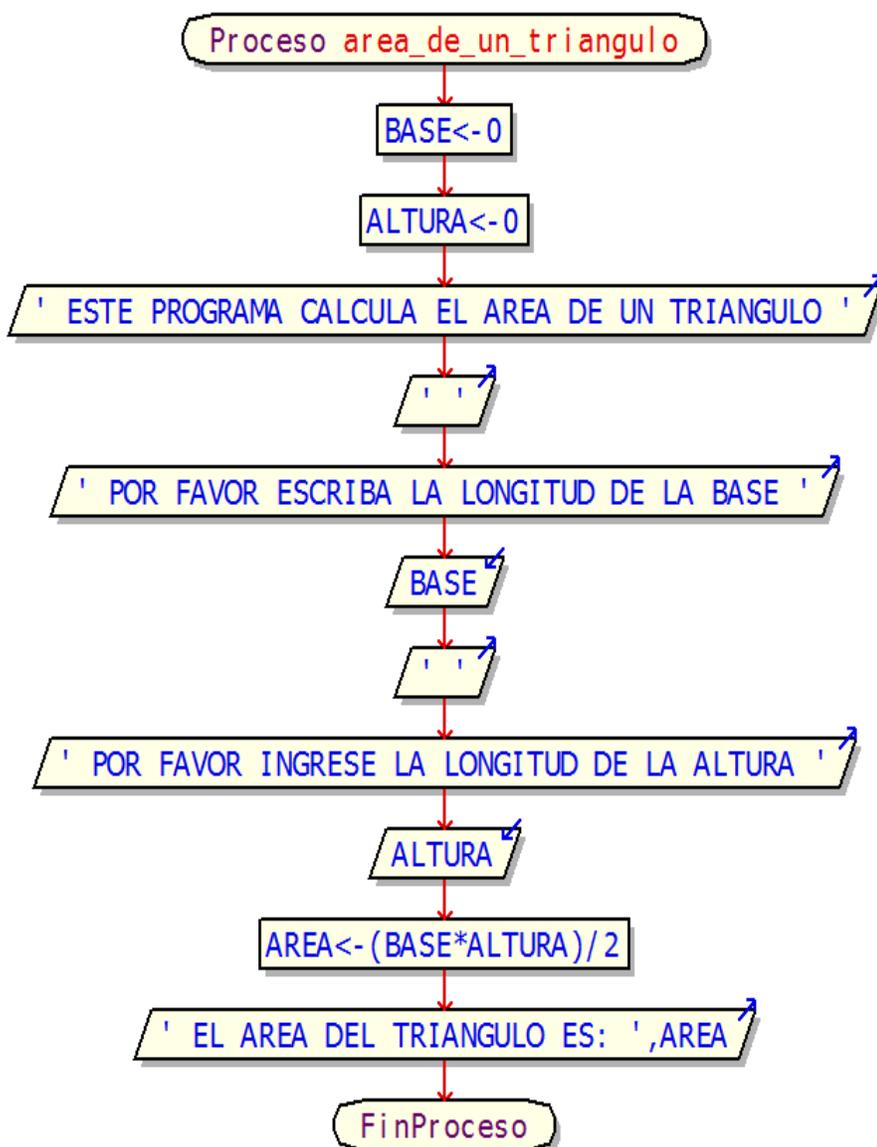
Escribir ' DATOS ERRONEOS ',error;

ESPERARTECLA;

FinProceso



## 2. Programa que calcula el área de un triángulo



Proceso area\_de\_un\_triángulo

BASE&lt;-0;

ALTURA&lt;-0;

Escribir 'ESTE PROGRAMA CALCULA EL AREA DE UN TRIANGULO';

Escribir '';

Escribir 'POR FAVOR ESCRIBA LA LONGITUD DE LA BASE';

Leer BASE;

Escribir '';

Escribir 'POR FAVOR INGRESE LA LONGITUD DE LA ALTURA';

Leer ALTURA;

AREA&lt;-(BASE\*ALTURA)/2;

Escribir 'EL AREA DEL TRIANGULO ES: ',AREA;

FinProceso

### 3. Programa que lee las calificaciones de N alumnos de N grupos con G Materias, con la intención de determinar el promedio de cada alumno y de cada grupo

Proceso ejercicio02\_parcial

```

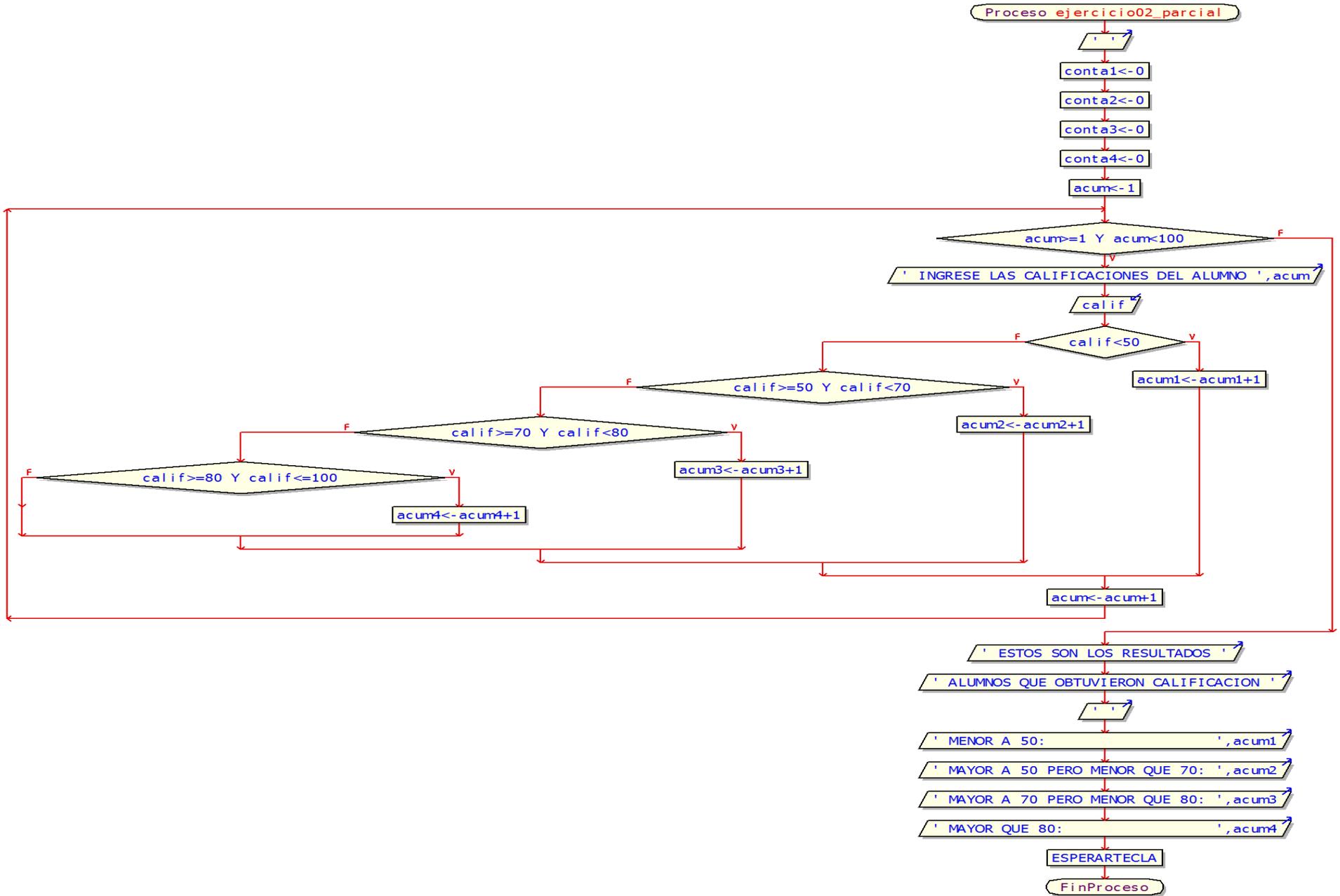
Imprimir " "
conta1<-0
conta2<-0
conta3<-0
conta4<-0
acum<-1

mientras acum>=1 y acum<100 Hacer

    Escribir " INGRESE LAS CALIFICACIONES DEL ALUMNO ", acum Sin Saltar
    leer calif;
        si calif < 50 Entonces
            acum1=acum1+1
        Sino
            si calif >=50 y calif <70
                acum2=acum2+1
            Sino
                si calif >=70 y calif <80
                    acum3=acum3+1
                Sino
                    si calif >=80 y calif <=100
                        acum4=acum4+1
                    FinSi
                FinSi
            FinSi
        FinSi
    acum=acum+1;
FinMientras
Escribir " ESTOS SON LOS RESULTADOS "
Escribir " ALUMNOS QUE OBTUVIERON CALIFICACION "
Escribir " ";
Escribir " MENOR A 50:          ",acum1;
Escribir " MAYOR A 50 PERO MENOR QUE 70: ",acum2;
Escribir " MAYOR A 70 PERO MENOR QUE 80: ",acum3;
Escribir " MAYOR QUE 80:          ",acum4;
Esperar Tecla

```

FinProceso



#### 4. Programa que imprimi cualquier tabla de multiplicar

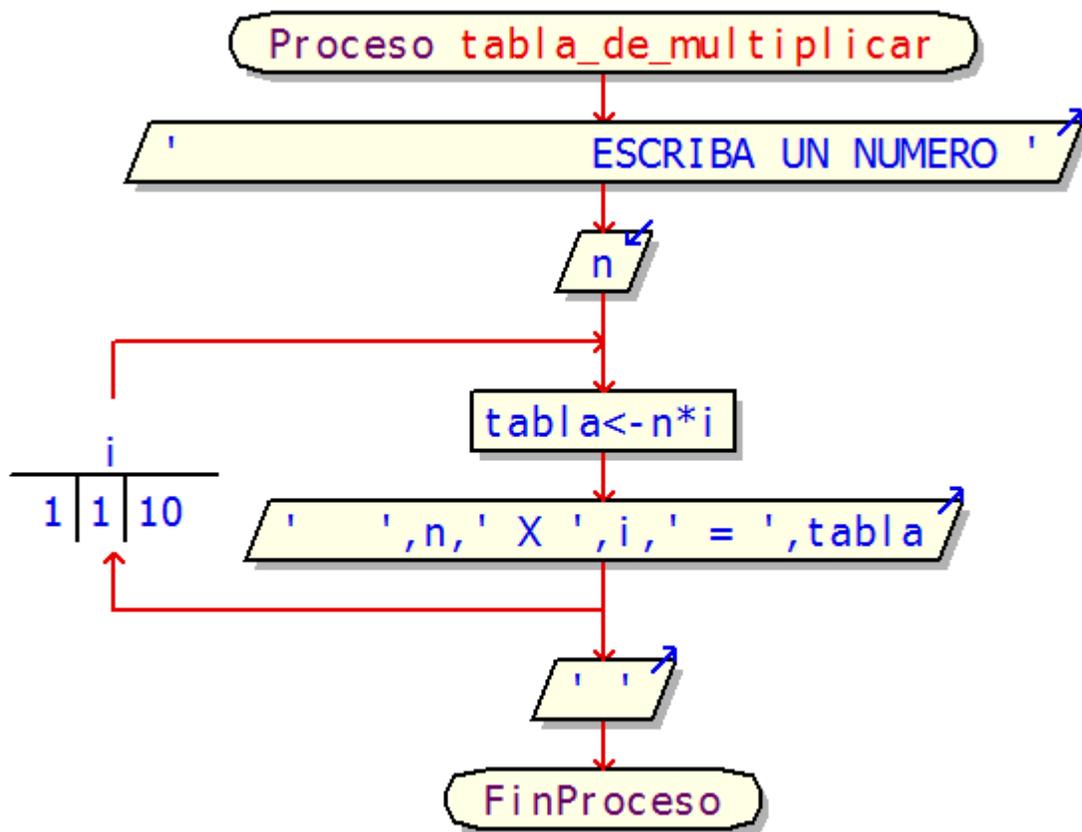
Proceso tabla\_de\_multiplicar

```

Escribir ' ESCRIBA UN NUMERO ';
Leer n;
Para i<-1 Hasta 10 Con Paso 1 Hacer
    tabla<-n*i;
    Escribir ' ',n,' X ',i,' = ',tabla;
FinPara
Escribir '';

```

FinProceso



**UNIVERSIDAD LUTERANA SALVADOREÑA  
FACULTAD DE CIENCIAS DEL HOMBRE Y LA NATURALEZA  
LICENCIATURA EN CIENCIAS DE LA COMPUTACION  
ALGORITMOS 1**



**TEMA: PORTAFOLIO ESTUDIANTIL**

**PRESENTADO A  
LIC. ANA LISSETTE GIRÓN**

**PRESENTADO POR:**

**DAVID ISAAC CLIMACO ORELLANA  
CARLOS ALBERTO RAMOS HERNANDEZ  
RENE EDUARDO AVILES ZEPEDA  
JUAN JOSE MINEROS**

**SAN SALVADOR, 29 DE MAYO 2013**