

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE



Nombre: Criollo Solano Jahir

NRC: 29583

Fecha: 06/12/2025

Tema: Ejercicios de Cap. 3



1. EJERCICIOS

Ejercicio 1 (Número Perfecto)

Un número perfecto es un número natural que es igual a la suma de sus divisores propios positivos, sin incluirse el mismo. Por ejemplo, el 6 es un número perfecto, porque sus divisores propios son 1, 2 y 3; y $6 = 1+2+3$. El siguiente perfecto es $28 = 1+2+4+7+14$.

Se pide:

1. Programe la función **esperfecto**, que recibe como argumento un número natural y devuelve un valor 1 si el número es perfecto, y 0 en caso contrario.
2. Desarrolle la función principal del programa que calcule los 4 primeros números perfectos y los muestre por pantalla.

Objeto	Nombre	Valor	Tipo
C. Perfectos	num	Variable	Entero
C. Iterador	i	Variable	Entero
Núm. a evaluar	n	Variable	Entero
Suma de divisores	res	Variable	Entero
Indicador de salida	salida	Variable	Entero
Función verificación	esperfecto	Función	Entero
Valor inicial iterador	1	Constante	Entero
Valor limite perfectos	4	Constante	Entero
Valor Verdadero	1	Constante	Entero
Valor Falso	0	Constante	Entero

PSEUDOCÓDIGO

```
Funcion salida <- esperarfecto(n)
  Definir i, res, salida Como Entero;
  res <- 0;
  salida <- 0;

  Para i <- 1 Hasta n-1 Hacer;
    Si n % i = 0 Entonces
      res <- res + i;
    FinSi;
  FinPara;

  Si res = n Entonces
    salida <- 1;
  FinSi;
FinFuncion
```

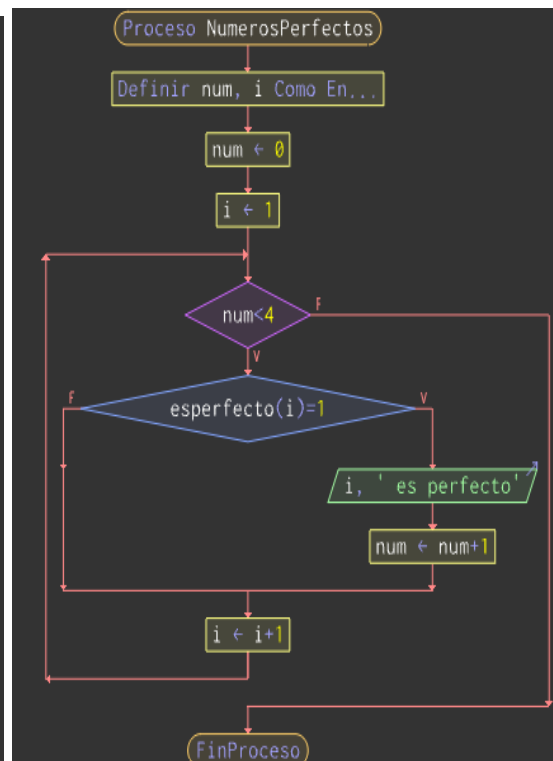
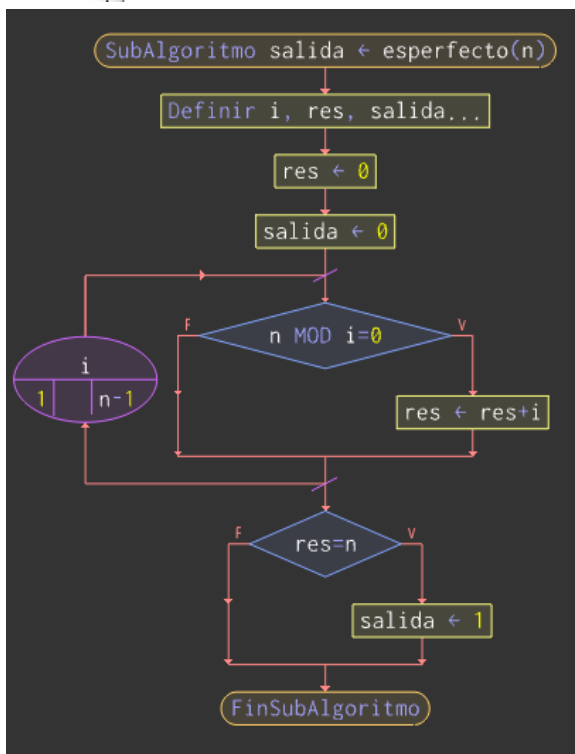
Proceso NumerosPerfectos

```
Definir num, i Como Entero;
num <- 0;
i <- 1;
```

```
Mientras num < 4 Hacer;
  Si esperfecto(i) = 1 Entonces
    Escribir i, " es perfecto";
    num <- num + 1;
  FinSi;
  i <- i + 1;
FinMientras
```

FinProceso

```
_titulo>* X
Funci
D
r
s
P
*** Ejecución Iniciada. ***
6 es perfecto
28 es perfecto
496 es perfecto
```



GDB Online: <https://onlinegdb.com/Qrcsz-arl>

Ejercicio 2 (Números primos)

Realice un programa que resuelva adecuadamente los siguientes apartados:

1. Programe la función **esprimo**, que recibe como argumento un número entero, y devuelve un valor 1 si el número es primo y 0 en caso contrario.
2. Almacene un vector todos los números primos comprendidos entre 2 números introducidos por teclado y luego imprima dicho vector.

Objeto	Nombre	Valor	Tipo
Número inicial del intervalo	N1	Variable	Entero
Número final del intervalo	N2	Variable	Entero
Contador del intervalo	I	Variable	Entero
Índice del arreglo	K	Variable	Entero
Arreglo de números primos	Primos	Variable	Entero
Número evaluado	N	Variable	Entero
Indicador de primalidad	Primo	Variable	Entero
Función verificación primo	Esprimo	Función	Entero
Tamaño del arreglo	100	Constante	Entero
Límite inferior de primalidad	2	Constante	Entero
Valor primo verdadero	1	Constante	Entero
Valor primo falso	0	Constante	Entero

PSEUDOCÓDIGO

Funcion primo <- esprimo(n)

Definir i, primo Como Entero;

Si $n < 2$ Entonces

 primo <- 0;

SiNo;

 primo <- 1;

 Para i <- 2 Hasta n-1 Hacer;

 Si $n \% i = 0$ Entonces

 primo <- 0;

 FinSi;

 FinPara;

FinSi;

FinFuncion

Proceso PrimosEntreDosNumeros

Definir i, k, N1, N2 Como Entero;

Definir primos Como Entero;

Dimension primos[100];

k <- 1; // ?? CLAVE: índices comienzan en 1

Escribir "Introduzca el primer numero:";

Leer N1;

Escribir "Introduzca el segundo numero:";

Leer N2;

Escribir "";

Escribir "Los numeros primos entre ", N1, " y ", N2, " son:";

Para i <- N1 Hasta N2 Hacer;

Si esprimo(i) = 1 Entonces

 primos[k] <- i;

 k <- k + 1;

FinSi;

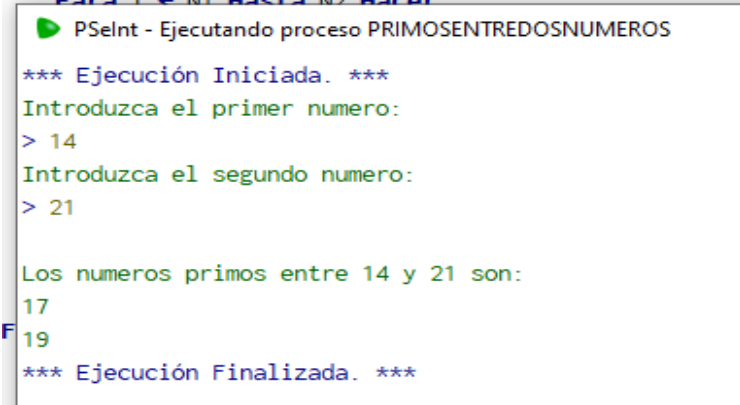
FinPara;

Para i <- 1 Hasta k-1 Hacer;

 Escribir primos[i], " ";

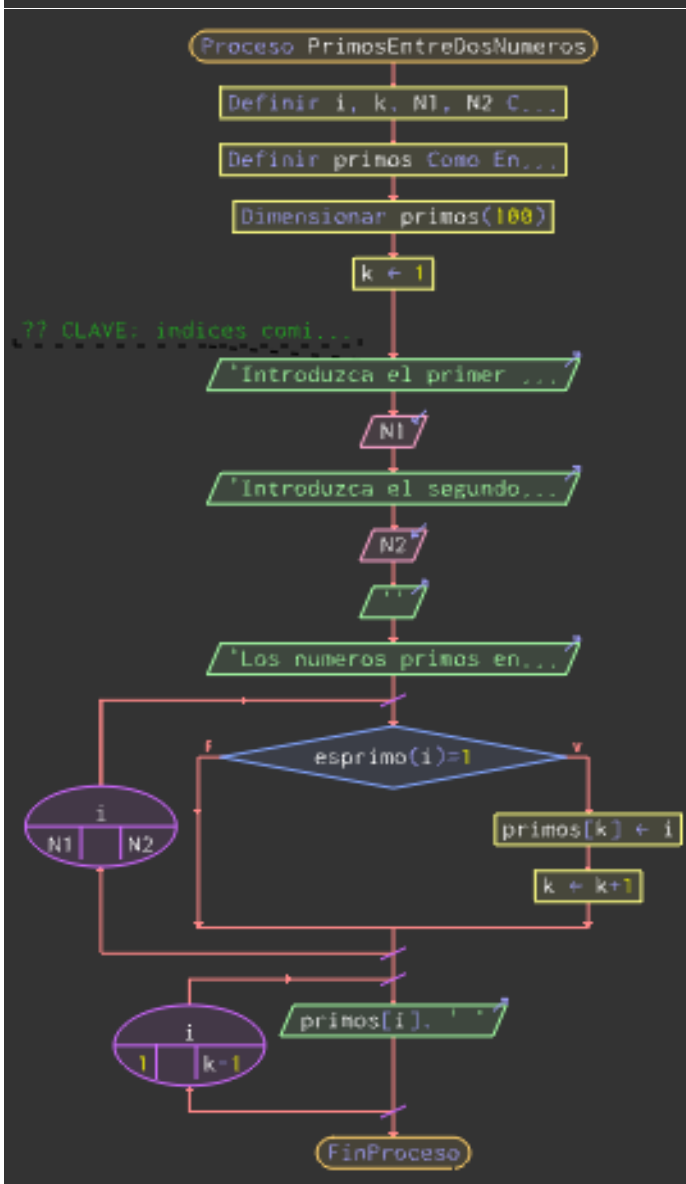
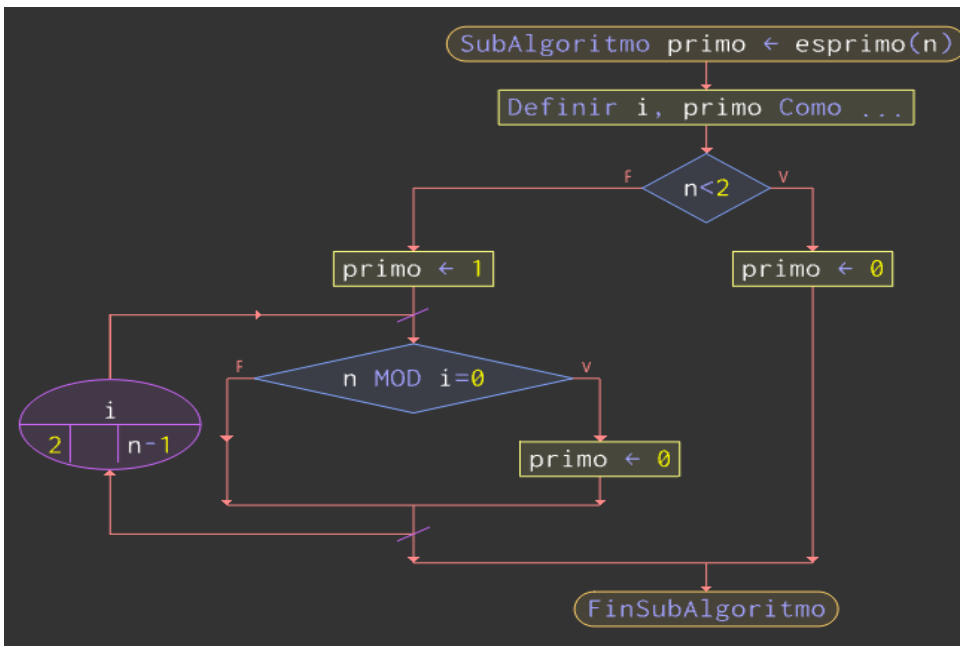
FinPara;

FinProceso



```
PSeInt - Ejecutando proceso PRIMOSENTREDOSNUMEROS
*** Ejecución Iniciada. ***
Introduzca el primer numero:
> 14
Introduzca el segundo numero:
> 21

Los numeros primos entre 14 y 21 son:
17
19
*** Ejecución Finalizada. ***
```



Ejercicio 3 (Factorial)

Desarrolle un programa que lea dos números enteros por teclado y determine si el primero de ellos es divisible por el Segundo, Se mostrará en pantalla el resultado. Utilice el operador modulo % que devuelve el resto de la división.

Objeto	Nombre	Valor	Tipo
Tamaño del vector	N	Constante	Entero
Índice de recorrido	I	Variable	Entero
Índice de multiplicación	J	Variable	Entero
Vector de números	Vec	Variable	Entero
Vector de factoriales	Fact	Variable	Real
Número a calcular factorial	Num	Variable	Entero
Resultado de la factorial	Resultado	Variable	Real
Función cálculo factorial	Calc_fact	Función	Real
Valor inicial de la factorial	1	Constante	Entero
Límite inferior del ciclo	2	Constante	Entero
Primer índice del vector	1	Constante	Entero
Condición número no negativo	0	Constante	Entero

PSEUDOCÓDIGO

Funcion resultado <- calc_fact(num)

Definir j Como Entero;

Definir resultado Como Real;

resultado <- 1;

Si num >= 0 Entonces

Para j <- 2 Hasta num Hacer;

resultado <- resultado * j;

FinPara;

FinSi;

FinFuncion

Proceso FactorialesVector

Definir i Como Entero;

Definir N Como Entero;

Definir vec Como Entero;

Definir fact Como Real;

N <- 15;

Dimension vec[N];


Dimension fact[N];

```
vec[1] <- 0;  
vec[2] <- 1;  
vec[3] <- 2;  
vec[4] <- 3;  
vec[5] <- 4;  
vec[6] <- 5;  
vec[7] <- 6;  
vec[8] <- 7;  
vec[9] <- 8;  
vec[10] <- 9;  
vec[11] <- 10;  
vec[12] <- 11;  
vec[13] <- 12;  
vec[14] <- 13;  
vec[15] <- 14;
```

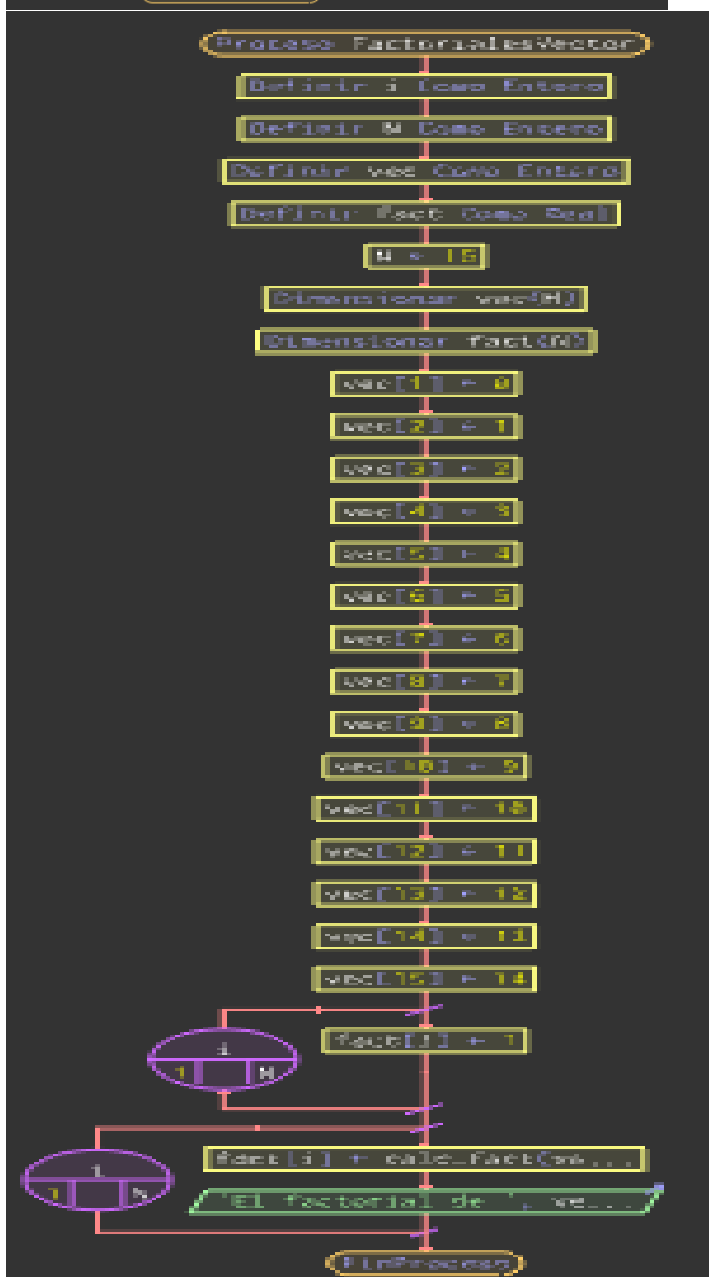
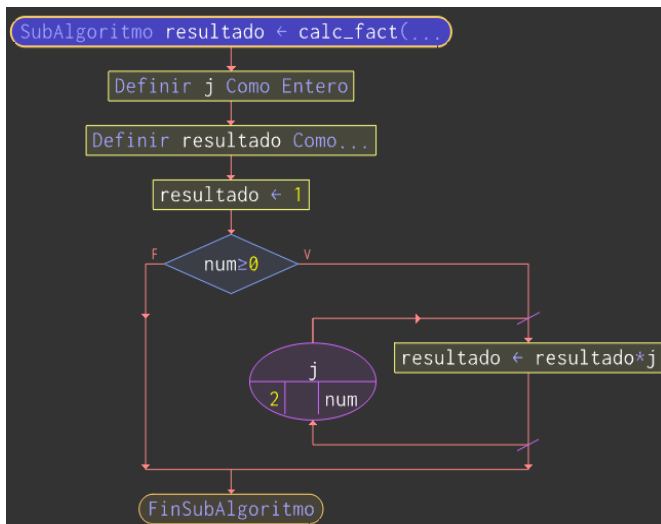
```
Para i <- 1 Hasta N Hacer;  
    fact[i] <- 1;  
FinPara;
```

```
Para i <- 1 Hasta N Hacer;  
    fact[i] <- calc_fact(vec[i]);  
    Escribir "El factorial de ", vec[i], " es ", fact[i];  
FinPara;
```

```
FinProceso
```

 PSeInt - Ejecutando proceso FACTORIALESVECTOR

```
El factorial de 1 es 1  
El factorial de 2 es 2  
El factorial de 3 es 6  
El factorial de 4 es 24  
El factorial de 5 es 120  
El factorial de 6 es 720  
El factorial de 7 es 5040  
El factorial de 8 es 40320  
El factorial de 9 es 362880  
El factorial de 10 es 3628800  
El factorial de 11 es 39916800  
El factorial de 12 es 479001600  
El factorial de 13 es 6227020800
```



Ejercicio 4 (Factorial Recursivo)

Programa la función **calc_fact**, que recibe como argumento un número entero, y devuelva el valor de la factorial. A continuación, use esta función en un programa que, dado un vector de 15 números enteros **vec**, calcule un vector **fact** con sus factoriales y lo muestre por pantalla.

Objeto	Nombre	Valor	Tipo
Tamaño del vector	N	Constante	Entero
Índice de recorrido	I	Variable	Entero
Índice de multiplicación	J	Variable	Entero
Vector de números	Vec	Variable	Entero
Vector de factoriales	Fact	Variable	Real
Número a calcular factorial	Num	Variable	Entero
Resultado de la factorial	Resultado	Variable	Real
Función cálculo factorial	Calc_fact	Función	Real
Valor inicial de la factorial recursivo	num > 1	Condición	Lógica
Valor inicial de la factorial	1	Constante	Entero
Paso recursivo	num - 1	Expresión	Entero
Índice inicial del vector	0	Constante	Entero
Límite superior del ciclo	N	Constante	Entero

PSEUDOCÓDIGO

Funcion resultado <- calc_fact(num)

Definir resultado Como Real;

Si num > 1 Entonces

 resultado <- num * calc_fact(num - 1);

SiNo;

 resultado <- 1;

FinSi;

FinFuncion

Proceso FactorialRecursivoVector

Definir i Como Entero;

Definir N Como Entero;

Definir vec Como Entero;

Definir fact Como Real;

N <- 15;

Dimension vec[N];

Dimension fact[N];

vec[1] <- 0;

```
vec[2] <- 2;  
vec[3] <- 3;  
vec[4] <- 4;  
vec[5] <- 5;  
vec[6] <- 6;  
vec[7] <- 7;  
vec[8] <- 8;  
vec[9] <- 9;  
vec[10] <- 10;  
vec[11] <- 11;  
vec[12] <- 12;  
vec[13] <- 13;  
vec[14] <- 14;  
vec[15] <- 15;
```

```
Para i <- 1 Hasta N Hacer;  
  fact[i] <- calc_fact(vec[i]);  
  Escribir "EL factorial de ", vec[i], " es ", fact[i];  
FinPara;
```

```
Escribir "";  
Escribir "Presione ENTER para finalizar...";  
Esperar Tecla;
```

FinProceso

PS/Mini - Ejecutando proceso FACTORIAL RECURSIVO VECTOR

```
EL factorial de 6 es 720  
EL factorial de 7 es 5040  
EL factorial de 8 es 40320  
EL factorial de 9 es 362880  
EL factorial de 10 es 3628800  
EL factorial de 11 es 39916800  
EL factorial de 12 es 479001600  
EL factorial de 13 es 6227020800  
EL factorial de 14 es 87178291200  
EL factorial de 15 es 1307674368000
```

Presione ENTER para finalizar...

*** Ejecución Finalizada. ***

☐ No cerrar esta ventana ☐ Siempre visible

Reiniciar

SubAlgoritmo resultado \leftarrow calc_fact(...)

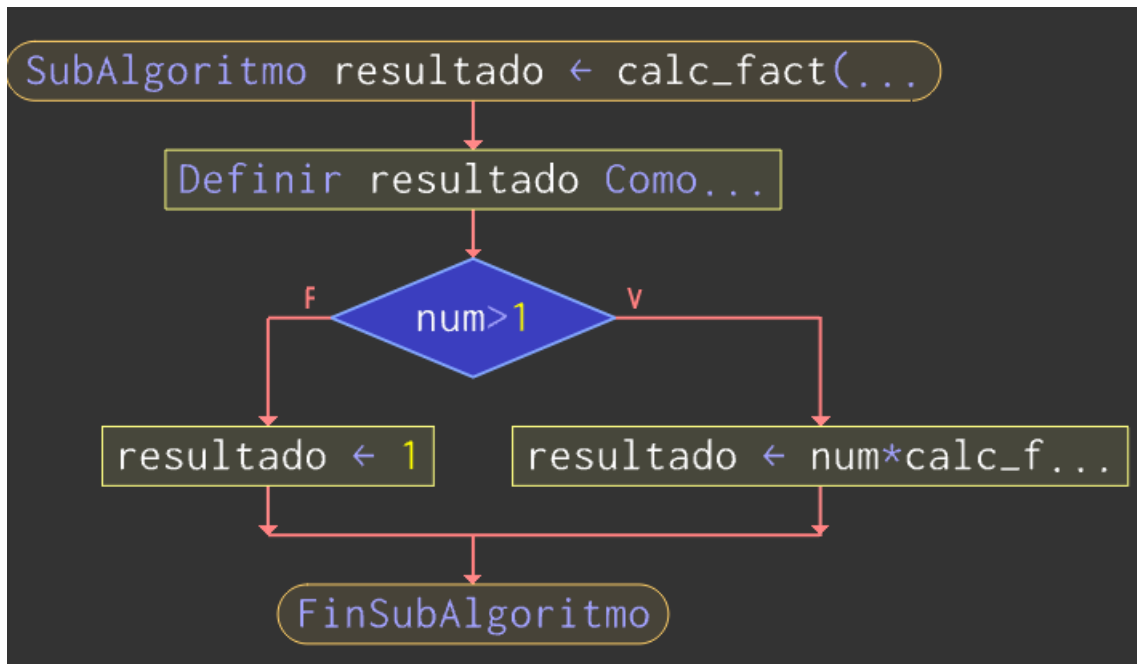
Definir resultado Como...

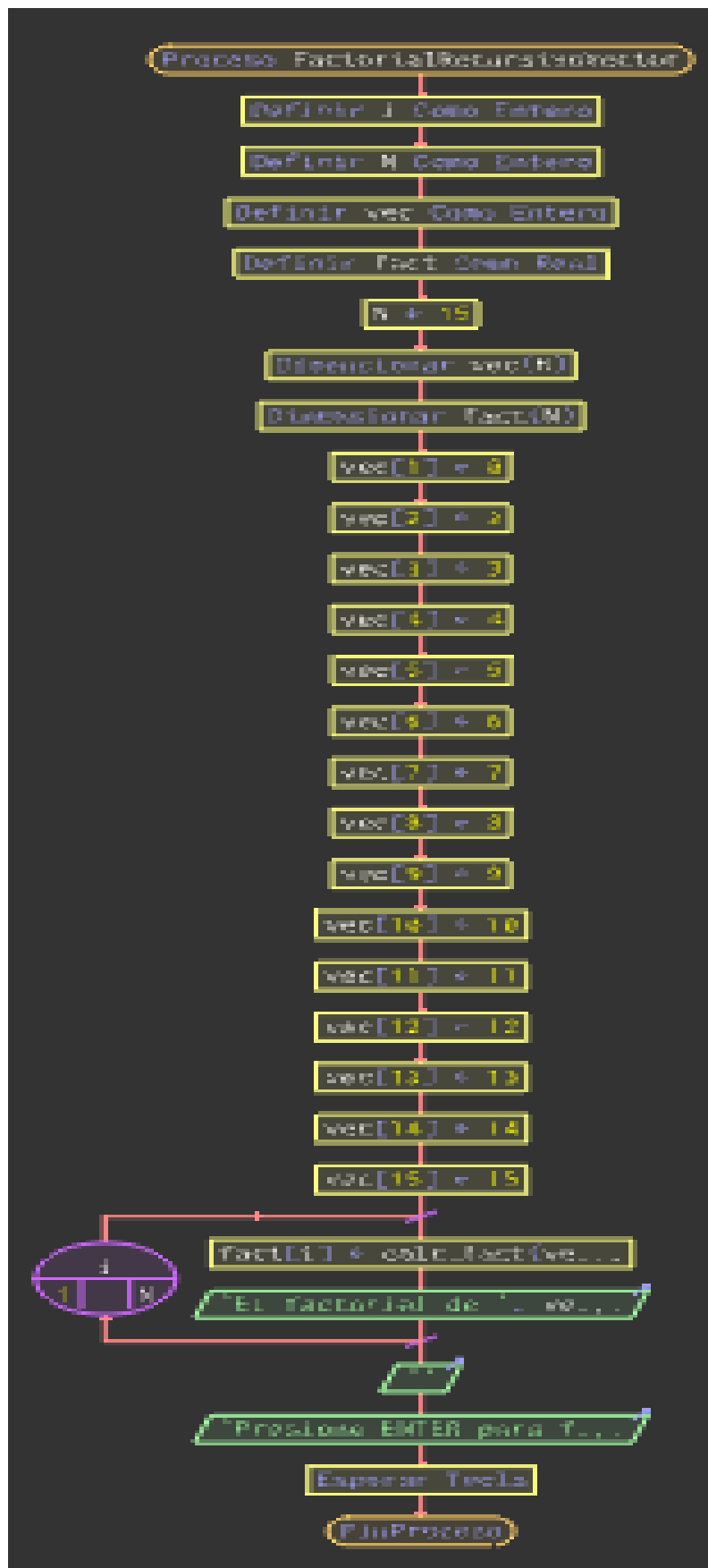
F $\text{num} > 1$ V

resultado \leftarrow 1

resultado \leftarrow num*calc_f...

FinSubAlgoritmo





GDB Online: <https://onlinegdb.com/Ly4M0qYFMO>

2. CONCLUSIONES

1. La función `calc_fact`, que toma un número entero como argumento y devuelve su factorial correctamente, se ha podido implementar con éxito, respetando así los principios de modularidad y reutilización del código.
2. Almacenar y procesar los valores originales junto a sus factoriales de manera estructurada fue posible gracias al uso de vectores, lo que hizo más sencillo el manejo de datos y la visualización de los resultados.

3. RECOMENDACIONES

1. Para futuras actividades, se recomienda utilizar el tipo de dato **“double”** en lugar de **“float”** para el cálculo de factoriales, con el fin de evitar pérdidas de precisión al trabajar con números grandes.

4. REFERENCIAS

Joyanes Aguilar, L. (2013). *Fundamentos de programación: algoritmos, estructuras de datos y objetos*.