

En este documento encontrarás una serie de recomendaciones para la resolución de ejercicios donde se debe aplicar el concepto de modularización.

### Tema: Modularización – PSeInt

#### Aspectos a tener en cuenta para pensar soluciones que incorporen la utilización de funciones o procedimientos

- 1- Analizar el enunciado y aplicar descomposición dividiendo el problema en subtareas. **Dividir un problema en subtareas permite comenzar a resolver el problema implementando un subalgoritmo por cada subtarea.**
- 2- Analizar cada subalgoritmo identificado y decidir si su solución debe implementarse como función o procedimiento.
- 3- Analizar para cada función y procedimientos planteados el pasaje de parámetros. ¿Se deben definir parámetros? ¿Qué parámetros deben definirse? ¿De qué tipo serán?

#### Enunciado:

Diseñar un algoritmo que permita ingresar y almacenar adecuadamente en un único arreglo, la edad y el nivel educativo alcanzado (primario, secundario, terciario, universitario) de 50 personas. Luego, se debe informar cuántas personas mayores a 25 han alcanzado el nivel terciario y cuál es la edad promedio de las personas que han alcanzado el nivel universitario.

Se pide:

- a) Analizar el enunciado, identificar las tareas principales, decidir qué **variables** se van a necesitar para guardar los datos , resultados, etc. Graficar el/los arreglos a definir.
- b) Realizar la versión 1 del algoritmo principal que dé solución al problema planteado, tenga en cuenta lo analizado en el punto a)
- c) Escribir el algoritmo completo que resuelva el enunciado, definiendo los subalgoritmos (procedimientos o funciones) necesarios para resolver cada tarea planteada en la versión 1.
- d) Realizar el diagrama de flujo del algoritmo codificado.

Resolución:

- a) **Analizar el enunciado, identificar las tareas principales, decidir qué variables se van a necesitar para guardar los datos , resultados, etc. Graficar el/los arreglos a definir.**

El enunciado plantea que se va a trabajar con datos de Personas: la edad y el nivel educativo alcanzado (Primario, Secundario, Terciario y Universitario)

Se puede usar **un arreglo** para guardar la edad de las personas y **otro arreglo** para guardar el nivel educativo alcanzado. Ambos arreglos pueden ser de tipo entero con dimensión 50.

Para el dato de nivel educativo alcanzado se puede codificar, 1=Primario, 2=Secundario, 3=Terciario y 4=Universitario

#### Arreglo 1 EDAD:

27	35	60	45	29	36					
1	2	3	4	5	6	7	...			50

#### Arreglo 2 NivelEduc:

1	1	3	4	2	2	1				
1	2	3	4	5	6	7	...			50

Por ejemplo, según lo almacenado en los arreglos: la persona 1 tiene 27 y ha alcanzado el nivel primario. ¿Cuáles son los datos de la persona 3?

También se necesitan al menos 2 variables para guardar los resultados ¿cuáles serían? ¿Para qué resultados? ¿De qué tipo deben ser?

**b) Realizar la versión 1 del algoritmo principal que dé solución al problema planteado, tenga en cuenta lo analizado en el punto a)**

Para realizar la versión 1 ahora vamos a pensar cada tarea identificando cuáles subalgoritmos vamos a usar, es decir cuáles subalgoritmos y si estos deben ser Procedimientos o Funciones. (buscar explicación en el manual Lenguaje de Diseño en PSeInt Subalgoritmos)

Entonces ¿Cómo escribimos ahora la versión 1?

**Versión 1 del algoritmo principal**

T1: Definir una variable arreglo llamada **Edad** de tipo entero y dimensión 50 para guardar la edad de las personas. Definir una variable arreglo llamada **NivelEduc** de tipo entero y dimensión 50 para guardar el nivel educativo de las personas. Definir las variables *res1* y *res2*.

T2: **Invocar** al **procedimiento IngresoDatos** para ingresar los datos. Los parámetros necesarios son: *Edad*, *NivelEduc* y el número 50 que es la dimensión de los arreglos para



T3: **Invocar** a la **función MayTerc** para calcular la cantidad de personas mayores de 25 que alcanzaron el nivel Terciario. Los parámetros necesarios son *Edad*, *NivelEduc* y 50. La variable de retorno será *r1*.

T4: **Invocar** a la **función PromEU** para calcular el promedio de edad de las personas que alcanzaron el nivel universitario. Los parámetros necesarios son: *Edad*, *NivelEduc* y 50, La variable de retorno será *r2*.

T5: Mostrar resultados que retornaron las funciones en T3 y T4.

**c) Escribir el algoritmo completo que resuelva el enunciado, definiendo los subalgoritmos (procedimientos o funciones) necesarios para resolver cada tarea planteada en la versión 1.**

//Procedimiento que ingresa los datos:

```

SubAlgoritmo IngresoDatos (Edad, NivelEduc, Ls)
  Definir i Como Entero
  //Ingreso de Datos
  Para I<- 1 Hasta Ls Con Paso 1 Hacer
    Escribir "Ingrese edad"
    Leer Edad[i]
    //***** Realizo control de Edad válida *****
    Mientras Edad[i] < 0 ∨ Edad[i] > 103 Hacer
      Escribir "Ingrese nuevamente edad"
      Leer Edad[I]
    FinMientras
    //***** Para poder ingresar el Nivel Educativo Alcanzado en el arreglo de enteros *****
    //***** se codificó cada uno de los Niveles con valores enteros *****
    Escribir "Ingrese Nivel Alcanzado (1-Primario; 2-Secundarios; 3-Terciario; 4-Universitario)"
    Leer NivelEduc[i]
    //***** Realizo control de Nivel Educativo Alvcanzado válido *****
    Mientras NivelEduc[i] < 1 ∨ NivelEduc[i] > 4 Hacer
      Escribir "Ingrese nuevamente Nivel Alcanzado (1-Primario; 2-Secundarios; 3-Terciario; 4-Universitario)"
      Leer NivelEduc[i]
    FinMientras
  FinPara
FinSubAlgoritmo

```

//función que calcule la cantidad de personas mayores de 25 que alcanzaron el nivel Terciario,

```

SubAlgoritmo res1 <- MayTerc(Edad, NivelEduc, Ls)
  Definir res1 Como Entero
  Definir i Como Entero
  res1 ← 0
  //**Recorro todo el arreglo saltando de 2 en 2 para analizar los dos datos de cada persona **
  //** Calcular la cantidad de Personas mayores de 25 con terciario**
  Para I<1 Hasta Ls con Paso 1 Hacer
    //***** Verifico Edad *****
    Si Edad[i] ≥ 25 Entonces
      //***** Verifico Nivel Educativo Alcanzado
      Si NivelEduc[i] = 3 Entonces
        res1 ← res1 + 1
      FinSi
    FinSi
  FinPara
FinSubalgoritmo

```

//función que calcule el promedio de edad de las personas que alcanzaron el nivel universitario,

```

SubAlgoritmo res2 <- PromEU(Edad, NivelEduc, Ls)
  Definir res2 como real
  Definir CantidadU, i Como Entero
  CantidadU ← 0
  res2 ← 0.0
  //**Calcular El promedio de edad de las personas con nivel Universitario**
  Para I<1 Hasta Ls Con Paso 1 Hacer
    //***** Verifico Nivel Educativo Alcanzado*****
    Si NivelEduc[i] = 4 Entonces
      //***** Acumulo la edad de las personas con Nivel Educativo Universitario *****
      res2 ← res2 + Edad[i]
      //***** Cuento la cantidad de personas con Nivel Educativo Alcanzado para luego calcular el promedio *****
      CantidadU ← CantidadU + 1
    FinSi
  FinPara
  Si CantidadU > 0 entonces
    res2 ← res2/ CantidadU
  FinSi
FinSubAlgoritmo

```

//Algoritmo principal (planteado en la versión 1)

**Algoritmo** EjerEjemplo**Definir** NivelEduc, Edad, Ls **Como Entero****Dimension** NivelEduc[50]**Dimension** Edad[50]**Definir** res1 **Como Entero****Definir** res2 **Como real**Ls  $\leftarrow$  50*//invocación al procedimiento que carga los arreglos*

IngresoDatos(Edad, NivelEduc, Ls)

*//invocación a la función que calcula la cantidad de Personas mayores de 25 con terciario*res1  $\leftarrow$  MayTerc(Edad, NivelEduc, Ls)*//invocación a la función que calcula El promedio de edad de las personas con nivel Universitario*res2  $\leftarrow$  PromEU(Edad, NivelEduc, Ls)*// Mostrar resultados***Escribir** "La cantidad de Personas mayores de 25 con terciario son ", res1**Escribir** "El promedio de edad de las personas con nivel Universitario es ",res2**FinAlgoritmo****d) Realizar el diagrama de flujo del algoritmo codificado.**

Sobre los diagramas de flujo, se debe realizar el diagrama de cada subalgoritmo, incluido el del Algoritmo principal, todos por separado, por ejemplo:

Diagrama de Flujo del Algoritmo Principal

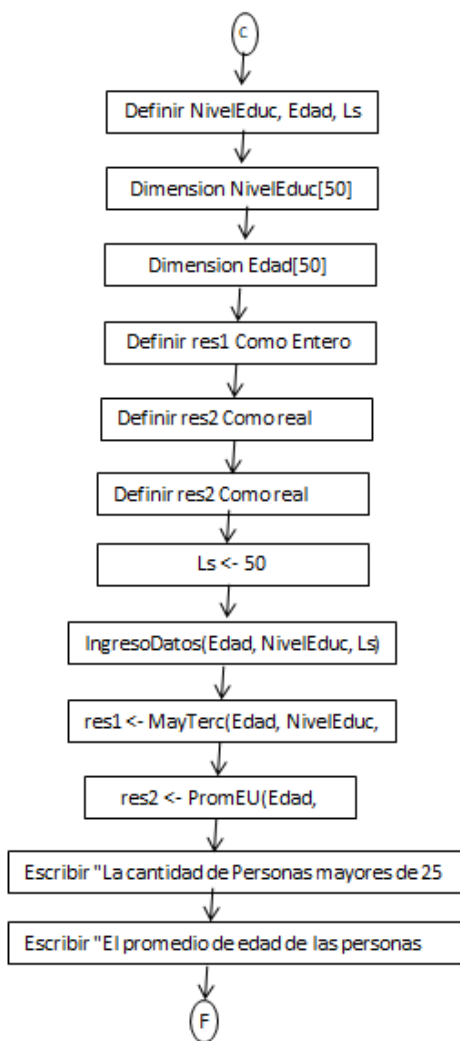
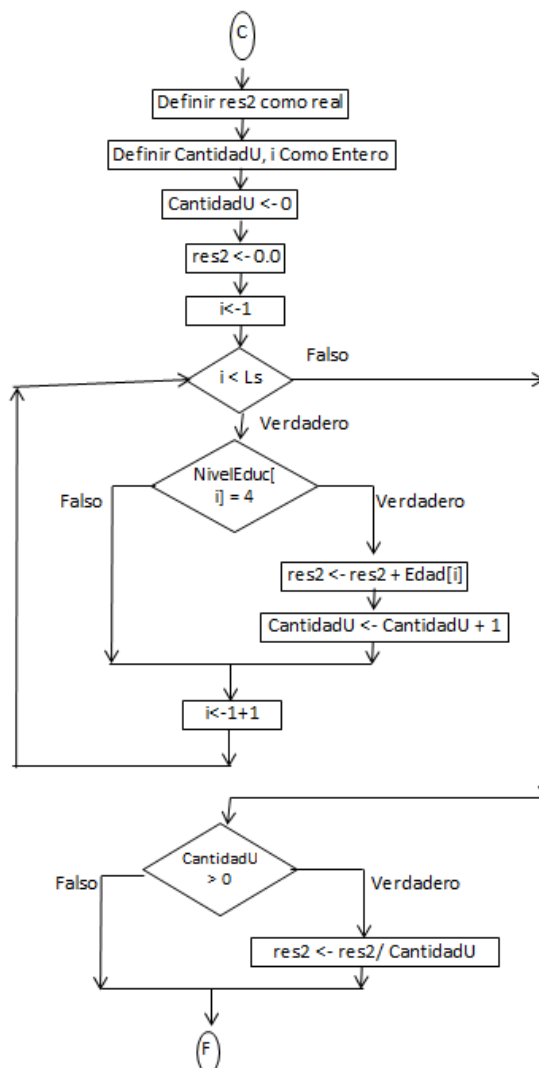


Diagrama de Flujo de la función PromEU



¿Cómo serían los diagramas de flujo de los subalgoritmos **IngresoDatos** y **MayTerc**? Realícelos.