

Tema: Resolución de Problemas – PSeInt

Etapas de la Resolución de Problemas:

Vamos a considerar las etapas de resolución de problemas teniendo en cuenta la existencia de una computadora:



1. La **primera etapa** involucra comprender el problema, es decir, definir el problema, para esto es necesario:
 - **identificar** y **listar** los datos necesarios para resolver el problema (**datos de entrada**).
 - **Identificar** y **especificar** los resultados que se deben alcanzar en el estado final (**resultados o salidas**).
2. En la **segunda etapa** se emplea un método o procedimiento para hallar la solución al problema, es decir, encontrar un algoritmo que dé solución. Para esto proponemos:
 - **Aplicar** descomposición para expresar el problema en subproblemas o tareas.
 - **Aplicar** la técnica de refinamiento sucesivo, considerando la sintaxis de PSeInt como conjunto de acciones primitivas.
3. La **tercera etapa**, llamada codificación, consiste en expresar el algoritmo de manera tal que pueda ser interpretado por un procesador. En este caso:
 - **Transcribir** el algoritmo diseñado en la etapa anterior a un programa en PSeInt y verificar la utilización correcta del lenguaje.
4. En la cuarta etapa se realiza la ejecución del programa diseñado, en esta etapa es importante:
 - **Ejecutar** el programa.
 - **Comparar** ejecuciones con **distintos conjuntos** de datos.
 - **Verificar** que se han implementado todos los **controles** correspondientes
 - **Detectar** casos donde el programa **no da solución** al problema.
 - **Formular** mejoras, en caso de ser necesario.

Conceptos a trabajar:

- Variables (Tipos de Datos)
- Operaciones y expresiones
- Entrada y salida
- Estructuras de control

Evaluación, en cada ejercicio se debe:

- aplicar descomposición correctamente.
- aplicar, al menos, dos niveles de desagregación
- escribir la versión final respetado la sintaxis de PSeInt

Enunciados

1- Marcela está interesada en diseñar un juego que le permita calcular la edad de un usuario pidiéndole un sólo dato. ¿Qué solución le plantearía?

Descomposición y desagregación

Versión 1

T1: Declarar objetos necesarios.

T2: **Pedir** al usuario ingrese su año de nacimiento

T2: **Calcular** su edad en función del año actual

T3: **Mostrar** el resultado obtenido en T2

Versión 2

T1.1: Declarar la variable entera **A_Nac** para almacenar el año de nacimiento

T1.2: Declarar la variable entera **Res** para almacenar la edad

T1.3: Pedirle al usuario que ingrese el año de nacimiento y guardarlo en la variable **A_Nac** corroborando que sea un año válido

T2.1: Restar el año actual menos el valor guardado en **A_Nac** y guardar el resultado en **Res**

T3.1: Mostrar el cartel " Tu edad es: "

T3.2: Mostrar el valor guardado en **Res**

Observe la sintaxis y la traducción realizada entre la versión 2 y la versión final.

Versión 3

ALGORITMO adivinoTuEdad

Definir A_Nac Como Entero

Definir Res Como Entero

Escribir "Ingrese su año de nacimiento"

Leer A_Nac

Res <- 2023 - A_Nac

Escribir "Tu edad es: "

Escribir Res

Declaración de variables para almacenar datos necesarios

Salida y Entrada. Interacción con el usuario

Proceso

Salida. Interacción con el usuario. Muestra de Resultados.

FinAlgoritmo

Nota: La solución está planteada como un conjunto de acciones secuenciales. Es decir que la estructura de control usada en esta solución es Secuencial, por lo tanto se ejecuta una acción luego de la otra.

Esta solución funciona para todos los casos??? Prueba el algoritmo con fechas de años válidos y fechas inválidas.

2- Diseñe un programa que informe por pantalla si el número ingresado por el usuario es par o impar.

Versión 1

T1: Declarar objetos necesarios

T2: Pedir el ingreso de un nro

T3: Determinar e informar si el nro ingresado es par o no

Versión 2

T1.1: Declarar la variable NroIngreso de tipo entero para guardar un nro

T2.1: Pedirle al usuario ingrese un número

T2.2: Guardar el número ingresado en la variable NroIngreso

Si el resto de dividir el valor de NroIngreso por 2 es igual a 0

T3.1.: Informar que el nro es par

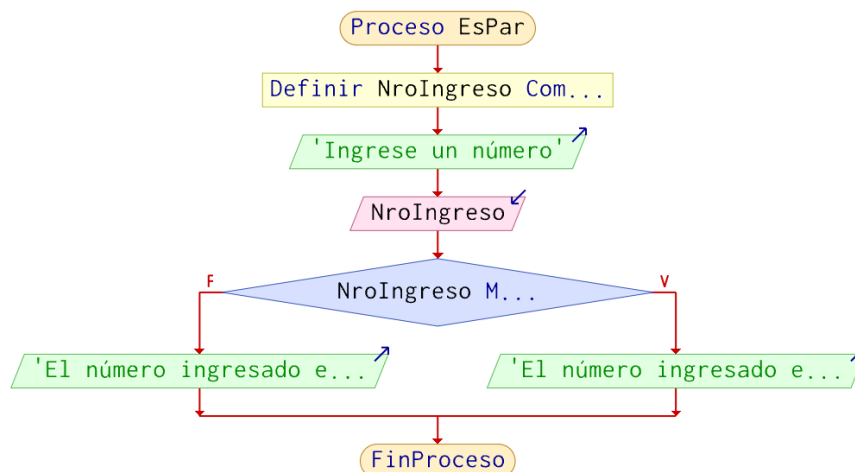
Sino

T3.2: Informar que el nro es impar

Versión Final

<pre>ALGORITMO EsPar Definir NroIngreso Como Entero Escribir "Ingrese un número" Leer NroIngreso Si NroIngreso % 2 = 0 Entonces Escribir "El número ingresado es PAR" SiNo Escribir "El número ingresado es IMPAR" FinSi FinAlgoritmo</pre>	<p>Esta solución contiene secuencia de acciones y además utiliza la estructura de control CONDICIONAL (Si)</p> <p>Observe qué estrategia utiliza (en la condición) para determinar si un nro es múltiplo de 2.</p>
---	---

Diagrama de Flujo en PSeInt:



Ejecución en Tabla de ejecución, considerando como número de ingreso el 123:

Acciones	variables	Pantalla
	NroIngreso	
Escribir "Ingrese un número"		Ingrese un número
Leer NroIngreso	123	
NroIngreso % 2 = 0 ? falso		
Escribir 'El número ingresado es IMPAR'		El número ingresado es IMPAR

- 3- Dados 30 caracteres ingresados por el usuario, mostrar por pantalla la cantidad de vocales cerradas (i y u) ingresadas. Se deben considerar tanto minúsculas como mayúsculas.

Versión 1

T1: Declarar objetos necesarios.

T2: ingresar 30 caracteres contando cuántos de ellos son vocales cerradas (i y u)

T3: mostrar el resultado obtenido en t2

Versión 2

T1.1: declarar una variable **Car** de tipo caracter para ingresar uno a uno los caracteres de tipo caracter

T1.2: declarar una variable **CantVocales** de tipo entero para ir contando la cantidad de vocales cerradas ingresadas

T1.3: declarar una variable **aux** de tipo entero para contar los caracteres ingresados, deben ser 30.

Mientras no se hayan ingresado 30 caracteres

T2.1: ingresar 1 caracter

Si el caracter ingresado en T2.1 es i o l o u o U

T2.2 contar uno más en la variable CantVocales

T3.1: Mostrar el cartel "La cantidad de vocales cerradas ingresadas fue:"

T3.2: Mostrar el valor de CantVocales

Versión Final

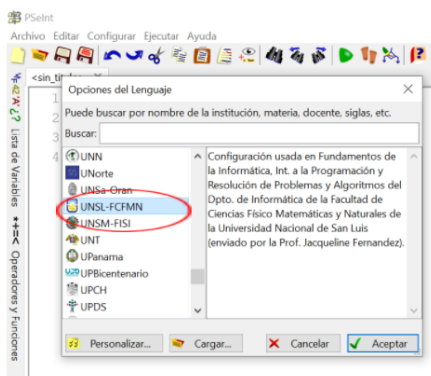
```
1 ALGORITMO vocalesCerradas
2   Definir Car Como Caracter
3   Definir CantVocales Como Entero
4   Definir aux Como Entero
5   CantVocales ← 0
6   aux ← 1
7   Mientras aux ≤ 30 Hacer
8     Escribir “Ingresar un caracter”
9     Leer Car
10    Si (Car = 'i' o Car = 'I' o Car = 'u' o Car = 'U') Entonces
11      CantVocales ← CantVocales + 1
12    FinSi
13    aux ← aux + 1
14  FinMientras
15  Escribir “La cantidad de vocales cerradas ingresadas fue:”, CantVocales
16 FinAlgoritmo
```

Notas:

- Observar que en las líneas 5 y 6 se realizan inicializaciones de variables, esto ocurre cuando se necesitan valores iniciales en variables utilizadas por ejemplo para contar, como en este caso.
- En la línea 10, observe la condición compuesta, en muchos casos se utilizará este tipo de estrategia.

Notas:

- 1- El perfil, en Pseint, con el que se evaluará la sintaxis de los programas es UNSL-FCFMyN. Es decir, la sintaxis planteada en el manual del tema.



[Lista de Reproducción](#) con videos cortos de los conceptos básicos utilizados en este práctico

