

Materia: Resolución de Problemas y Algoritmos.

Carreras: Ingeniería en Computación y Profesorado en Ciencias de la Computación.

Práctico Nº 5: P5elnt. Estructura de Datos: Arreglos

Tabla con los días para trabajar en este práctico.

Semana	Fecha	Teoría	Práctico	Parcialito
	Viernes	24/4/26	Arreglos	
8	Lunes	27/4/26		
	Martes	28/4/26	P5 Arreglos	
	Miércoles	29/4/26		
	Jueves	30/4/26	P5 Arreglos	
	Viernes	1/5/26	Feriado	
9	Lunes	4/5/26		
	Martes	5/5/26	P5 Arreglos	
	Miércoles	6/5/26		
	Jueves	7/5/26	P5 Arreglos	
	Viernes	8/5/26		P5 Arreglos

Ejercicio 1: Se tiene un algoritmo que calcula el importe total a pagar por la compra de 3 productos, a partir de los costos ingresados por el usuario. Para ello, utiliza un arreglo para almacenar los costos, valida que cada valor ingresado sea positivo y luego realiza el cálculo del total.

Se pide:

a) Analizar el algoritmo dado e identificar: *las variables utilizadas, las estructuras de control presentes y los recorridos del arreglo y qué se realiza en cada uno.*

b) Realizar el **diagrama de flujo** del algoritmo.

c) Completar la **tabla de ejecución** para los valores de entrada: **12300, 20500 y 11500.**

```
1 Algoritmo TotalCompra
2   Definir costos Como Real
3   Dimension costos[3]
4   Definir total Como Real
5   Definir i Como Entero
6   total ← 0.0
7   Para i ← 1 Hasta 3 con paso 1 Hacer
8       Escribir "Ingrese el costo del producto:"
9       Leer costos[i]
10      Mientras costos[i] < 0 Hacer
11          Escribir "Ingrese UN VALOR POSITIVO para el costo del producto:"
12          Leer costos[i]
13      FinMientras
14   FinPara
15   Para i ← 1 Hasta 3 con paso 1 Hacer
16       total ← total + costos[i]
17   FinPara
18   Escribir "El total a pagar es: ", total
19 FinAlgoritmo
```

Acción (línea)	i	costos			total	Pantalla
		[1]	[2]	[3]		
6					0.0	
$i \leftarrow 1$	1					
$(i \leq 3) ? V$						
8						Ingrese el costo...
9		12300				
$(costos[i] < 0) ? F$						
$i \leftarrow i + 1$	2					
$(i \leq 3) ? V$						
8						Ingrese el costo...
9			20500			
$(costos[i] < 0) ? F$						
$i \leftarrow i + 1$	3					
$(i \leq 3) ? V$						
8						Ingrese el costo...
9				11500		
$(costos[i] < 0) ? F$						
$i \leftarrow i + 1$	4					
$(i \leq 3) ? F$						

Ejercicio 2: Se debe diseñar un algoritmo que permita: ingresar un conjunto de 100 números enteros y determinar cuántos de ellos son positivos y cuántos son negativos. Se dispone de la versión 1 para resolver el problema y de un ejemplo gráfico del antes y después de ejecutar la solución.

Versión 1:

T1: Definir un arreglo *aNum* de 100 posiciones de tipo entero. Definir las variables *npos* y *nneg* de tipo entero y la variable índice *i* de tipo entero.

T2: Inicializar las variables *npos* y *nneg* en 0.

T3: Recorrer el arreglo desde la primera hasta la última posición, ingresando los 100 números enteros.

T4: Recorrer el arreglo desde la primera hasta la última posición y si el número es mayor o igual que 0, incrementar en uno la variable *npos*. En caso contrario, incrementar en uno la variable *nneg*.

T5: Mostrar la cantidad de números positivos y negativos calculadas en T4.

Ejemplo gráfico:



Se pide:

- Escribir el algoritmo en PSeInt que implemente la versión 1 dada.
- Realizar el diagrama de flujo del algoritmo logrado.

Ejercicio 3: Diseñar un algoritmo que permita realizar las siguientes acciones con una secuencia de caracteres:

- Ingresar 10 caracteres y almacenarlos en un arreglo.
- Reemplazar todas las apariciones de un carácter por otro, según lo que indique el usuario.
- Mostrar el arreglo actualizado después de realizar el reemplazo.

Se pide:

- Leer y analizar la versión 1** para comprender la solución propuesta:

T1: Definir un arreglo de 10 posiciones de tipo carácter para almacenar la secuencia.

T2: Definir dos variables de tipo carácter, *car1* y *car2*, para el reemplazo y la variable índice *i* de tipo entero.

T3: Recorrer el arreglo desde la primera hasta la última posición, ingresando los 10 caracteres.

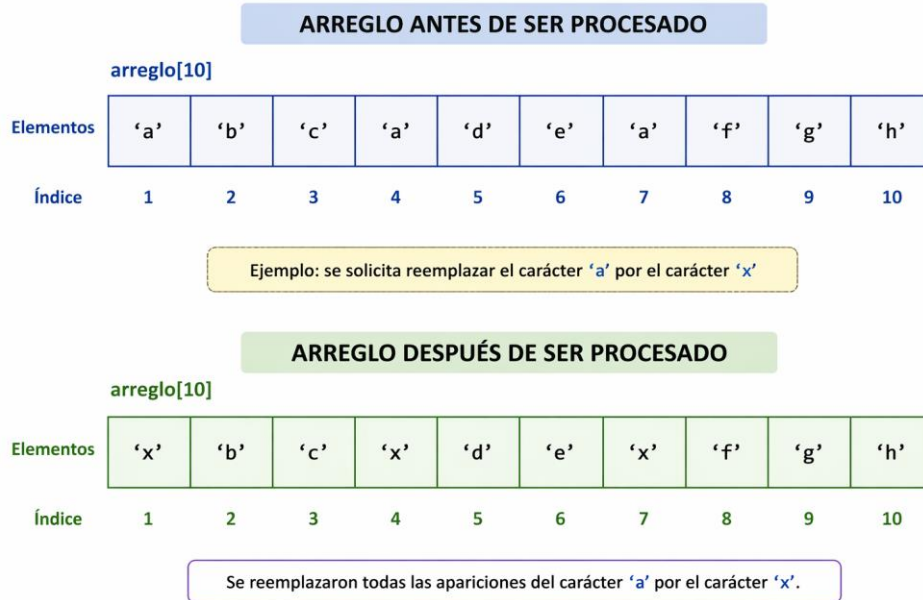
T4: Ingresar el carácter a reemplazar en *car1*

T5: Ingresar el carácter nuevo en *car2*

T6: Recorrer el arreglo desde la primera hasta la última posición, sustituyendo cada carácter del arreglo que sea igual a *car1*, por *car2*.

T7: Recorrer el arreglo desde la primera hasta la última posición mostrando su contenido.

b) Observar y analizar la situación graficada:



c) Escribir el algoritmo completo que implemente la versión 1 dada. Use el nombre "AlgoritmoUno" para este algoritmo.

d) Realizar el Diagrama de Flujo del "AlgoritmoUno".

Ejercicio 4: Diseñar un algoritmo que permita realizar las siguientes acciones con un conjunto de números reales:

- Ingresar valores numéricos reales correspondientes a las temperaturas diarias durante el año 2025.
- Calcular el promedio anual de temperatura.
- Mostrar el valor promedio calculado.

Se pide:

a) **Leer y analizar la versión 1** para comprender la solución propuesta, identificando: Qué datos se almacenan y qué proceso se realiza sobre el arreglo.

Versión 1:

T1: Definir un arreglo temperaturas de 365 posiciones de tipo real. Definir las variables *sum* y *prom* de tipo real y la variable índice *i* de tipo entero.

T2: Inicializar la variable *sum* en 0.

T3: Recorrer el arreglo desde la primera hasta la última posición, ingresando los 365 valores de temperatura.

T4: Recorrer el arreglo desde la primera hasta la última posición, sumando los valores almacenados en el arreglo, en la variable *sum*.

T5: Calcular el promedio dividiendo *sum* por la cantidad de elementos (365) y guardarlo en *prom.*
 T6: Mostrar el valor promedio obtenido en T5

- b) Representar gráficamente el arreglo indicando: Nombre del arreglo, Tipo de dato y dimensión.
- c) Escribir el algoritmo en PSeInt que implemente la versión 1. Utilizar el nombre *AlgoritmoTemperaturas*.
- d) Realizar el diagrama de flujo del algoritmo *AlgoritmoTemperaturas*.

Ejercicio 5: Diseñar un algoritmo que permita registrar la asistencia de un grupo de estudiantes a una clase. Para cada estudiante se almacenará si asistió o no (valor lógico: verdadero si asistió, falso si no asistió). El algoritmo debe permitir:

- Ingresar como máximo 12 valores de asistencia.
- Contar cuántos estudiantes asistieron.
- Contar cuántos estudiantes estuvieron ausentes.
- Calcular el porcentaje de estudiantes asistentes y ausentes.
- Mostrar los resultados obtenidos.

Ejemplo de representación del arreglo antes y después de ejecutar el algoritmo:

ANTES DE EJECUTAR EL ALGORITMO

Arreglo: *asistencia* (tipo lógico) – dimensión: 12

Valores (lógicos)	?	?	?	?	?	?	?	?	?	?	?	
Índice	1	2	3	4	5	6	7	8	9	10	11	12

DESPUÉS DE EJECUTAR EL ALGORITMO

Arreglo: *asistencia* (tipo lógico) – dimensión: 12

Valores (lógicos)	V	F	V	V	F	V	F	V	V	F	F	V
Índice	1	2	3	4	5	6	7	8	9	10	11	12

RESULTADOS OBTENIDOS:

- Cantidad de asistentes (VERDADERO): 7
- Cantidad de ausentes (FALSO): 5

PORCENTAJES:

- Porcentaje de asistentes: 58,33 %
- Porcentaje de ausentes: 41,67 %

Se pide:

- a) **Escribir la versión 1** que permita dar solución al problema planteado.
- b) Escribir el algoritmo en PSeInt. Utilizar el nombre *AlgoritmoAsistencia*.
- c) Realizar el diagrama de flujo correspondiente al algoritmo.

Ejercicio 6: Diseñar un algoritmo que permita registrar y procesar información sobre un grupo de estudiantes. Para cada estudiante se debe almacenar:

- Número de registro (entero).
- Cantidad de materias que cursa (entero).
- Si posee o no beca de comedor (valor lógico: verdadero si posee beca, falso si no posee).

El total de estudiantes es 34.

El algoritmo debe permitir:

- Calcular el promedio de materias cursadas por los estudiantes.
- Calcular el porcentaje de estudiantes que poseen beca de comedor.
- Consultar los datos de un estudiante específico, ingresando su número de registro.

Imagen de ejemplo:

ANTES DE EJECUTAR EL ALGORITMO								
Arreglos vacíos (sin información cargada)								
registro (entero)	?	?	?	?	?	?	?	?
Índice	1	2	3	4	...	32	33	34
materias (entero)	?	?	?	?	?	?	?	?
Índice	1	2	3	4	...	32	33	34
beca (lógico)	?	?	?	?	?	?	?	?
Índice	1	2	3	4	...	32	33	34

Se pide:

- Observar la imagen presentada e identificar:
 - Qué información se almacena en cada arreglo.
 - Qué tipo de dato corresponde a cada uno.
 - Qué relación existe entre las posiciones de los arreglos.
- Representar gráficamente los arreglos utilizados indicando: Nombre de cada arreglo, tipo de dato y dimensión.
- Escribir la versión 1 que permita dar solución al problema planteado.
- Escribir el algoritmo completo en PSeInt que implemente la versión 1. Utilizar el nombre *AlgoritmoEstudiantes*.
- Realizar el diagrama de flujo correspondiente al algoritmo.

Ejercicio 7: Diseñar un algoritmo que permita registrar información sobre una recolección de rocas para prácticas de la materia Rocas 1.

Por cada roca recolectada se debe almacenar:

- Número de identificación de la roca.
- Tipo de roca (1: Ígnea, 2: Sedimentaria, 3: Metamórfica)

El total de rocas es como máximo 18.

El algoritmo debe permitir:

- Consultar cuántas rocas hay de un tipo específico, según lo indique el usuario.
- Determinar cuál es el tipo de roca con mayor cantidad de registros.
- Mostrar los resultados obtenidos.

Se pide:

- a) Representar gráficamente los arreglos utilizados indicando: *Nombre de cada arreglo, tipo de dato y dimensión.*
- b) Escribir la versión 1 que permita dar solución al problema planteado.
- c) Escribir el algoritmo PSeInt que implemente la versión 1 dada. Utilizar el nombre AlgoritmoRocas.
- d) Realizar el diagrama de flujo correspondiente al algoritmo.

Ejercicio 8: En la página 10 del manual se presenta el algoritmo “Ventas”, que utiliza un arreglo de 100 posiciones para almacenar las ventas de las sucursales, calcular el promedio de ventas y mostrar aquellas que superan dicho valor.

Se pide:

1. Modificar el algoritmo para que, además, para cada sucursal cuya venta sea mayor o igual al promedio, muestre:
 - La posición del arreglo, el valor de la venta y la diferencia entre esa venta y el promedio.
2. Escribir el algoritmo modificado en PSeInt con el nombre **AlgoritmoVentas**.



Licenciamiento: Esta publicación se distribuye bajo una Licencia Creative Commons DistribuciónNoComercial-CompartirIgual 4.0 Internacional.