



**UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO**

---



**FACULTAD DE INGENIERÍA**

**INGENIERÍA EN COMPUTACIÓN**

**UNIDAD DE APRENDIZAJE:  
PROGRAMACIÓN ESTRUCTURADA**

**TEMAS:  
ESTRUCTURAS DE CONTROL, ARREGLOS,  
MÓDULOS Y REGISTROS**

**PROBLEMARIO  
DISEÑO DE PSEUDOCÓDIGOS**

**ELABORADO POR:**

**M. EN A. SILVIA EDITH ALBARRÁN TRUJILLO**

**SEPTIEMBRE 2016**



# ÍNDICE

<b>▣ Presentación.....</b>	<b>8</b>
<b>☞ Guía de uso.....</b>	<b>10</b>
<b>✍ Serie de Ejercicios.....</b>	<b>17</b>
Ejercicio 1.....	17
Ejercicio 2.....	18
Ejercicio 3.....	19
Ejercicio 4.....	20
Ejercicio 5.....	21
Ejercicio 6.....	22
Ejercicio 7.....	23
Ejercicio 8.....	24
Ejercicio 9.....	25
Ejercicio 10.....	26
Ejercicio 11.....	27

Ejercicio 12.....	28
Ejercicio 13.....	29
Ejercicio 14.....	30
Ejercicio 15.....	31
Ejercicio 16.....	32
Ejercicio 17.....	33
Ejercicio 18.....	34
Ejercicio 19.....	35
Ejercicio 20.....	36
Ejercicio 21.....	37
Ejercicio 22.....	38
Ejercicio 23.....	39
Ejercicio 24.....	40
Ejercicio 25.....	41
Ejercicio 26.....	42
Ejercicio 27.....	43
Ejercicio 28.....	44
Ejercicio 29.....	45
Ejercicio 30.....	46
Ejercicio 31.....	47

Ejercicio 32.....	<b>48</b>
Ejercicio 33.....	<b>49</b>
Ejercicio 34.....	<b>50</b>
Ejercicio 35.....	<b>50</b>
Ejercicio 36.....	<b>52</b>
Ejercicio 37.....	<b>53</b>
Ejercicio 38.....	<b>54</b>
Ejercicio 39.....	<b>55</b>
Ejercicio 40.....	<b>56</b>
Ejercicio 41.....	<b>57</b>
Ejercicio 42.....	<b>58</b>
Ejercicio 43.....	<b>59</b>
Ejercicio 44.....	<b>60</b>
Ejercicio 45.....	<b>61</b>
Ejercicio 46.....	<b>62</b>
Ejercicio 47.....	<b>63</b>
Ejercicio 48.....	<b>64</b>
Ejercicio 49.....	<b>65</b>
Ejercicio 50.....	<b>66</b>
Ejercicio 51.....	<b>67</b>

Ejercicio 52.....	<b>68</b>
Ejercicio 53.....	<b>69</b>
Ejercicio 54.....	<b>70</b>
Ejercicio 55.....	<b>71</b>
Ejercicio 56.....	<b>72</b>
Ejercicio 57.....	<b>73</b>
Ejercicio 58.....	<b>74</b>
Ejercicio 59.....	<b>75</b>
Ejercicio 60.....	<b>76</b>
Ejercicio 61.....	<b>77</b>
Ejercicio 62.....	<b>78</b>
Ejercicio 63.....	<b>79</b>
Ejercicio 64.....	<b>80</b>
Ejercicio 65.....	<b>81</b>
Ejercicio 66.....	<b>82</b>
Ejercicio 67.....	<b>83</b>
Ejercicio 68.....	<b>84</b>
Ejercicio 69.....	<b>85</b>
Ejercicio 70.....	<b>86</b>
Ejercicio 71.....	<b>87</b>

Ejercicio 72.....	88
Ejercicio 73.....	89
Ejercicio 74.....	90
Ejercicio 75.....	91
Ejercicio 76.....	92
Ejercicio 77.....	93
Ejercicio 78.....	94
Ejercicio 79.....	95
Ejercicio 80.....	96
Ejercicio 81.....	97
Ejercicio 82.....	98
Ejercicio 83.....	99
Ejercicio 84.....	100

**☞ Ejercicios Resueltos.....**

Ejercicio 85.....	101
Ejercicio 86.....	103
Ejercicio 87.....	105
Ejercicio 88.....	107
Ejercicio 89.....	109

Ejercicio 90.....	111
Ejercicio 91.....	113
Ejercicio 92.....	115
Ejercicio 93.....	117
Ejercicio 94.....	119
Ejercicio 95.....	120
Ejercicio 96.....	121
 <b>Bibliografía</b> .....	122



## PRESENTACIÓN

El programa de Programación Estructurada está conformado en la actualidad de 5 unidades de competencia, dentro de las cuales se aborda el tema de Metodología de la Programación, dentro de esta metodología se debe realizar la definición de variables, algoritmos, diseño de pseudocódigos, diseño de diagramas de flujo de datos y pruebas de escritorio para problemas de programación, bajo los cuales se orienta principalmente este problemario.

Esta serie de ejercicios está enfocada definición de variables, algoritmos, diseño de pseudocódigos, diseño de diagramas de flujo de datos y pruebas de escritorio de cada uno de los problemas que se presentan, la definición de variables es una parte importante para los alumnos que inician con la programación ya que les permite diferenciar de las variables de entrada, proceso y salida. Las variables de entrada son aquellas que requieren de la intervención del usuario del programa, es decir, requieren de un dispositivo de entrada, principalmente el teclado; las variables de proceso son aquellas que son requeridas por el programa pero transparentes al usuario (no se despliegan en ningún momento) y las variables de salida son aquellas que serán desplegadas en algún dispositivo de salida, principalmente la pantalla. Los algoritmos son la solución de problemas de manera redactada (en español común), el pseudocódigo es el planteamiento del problema en español estructurado que es una serie de instrucciones más parecida a los lenguajes de programación, el diagrama de flujo de datos es la representación en un diagrama de las instrucciones y la prueba de escritorio se refiere a la comprobación de que las instrucciones planteadas en el pseudocódigo y el diagrama de flujo sean las correctas. El **objetivo de este problemario** es desarrollar en los alumnos la habilidad de comprender los problemas expresados en texto, los datos que pueden ser requeridos antes de iniciar el diseño de los programas, el planteamiento los pasos para resolver el problema, el planteamiento de instrucciones, diagrama de flujo y su verificación.

Este problemario está orientado principalmente a los alumnos que cursan la unidad de aprendizaje de Programación Estructurada (Unidad de Aprendizaje de la Licenciatura de Ingeniería en Computación), sin embargo, también puede servir de apoyo a los alumnos que en otras licenciaturas (Civil, Mecánica, Computación, Electrónica e Ingeniería en Sistemas Energéticos Sustentables) cursan la unidad de aprendizaje de Programación Básica.

Cabe mencionar que se omite toda teoría debido a que es sólo una herramienta para apoyo. La teoría se presenta en el libro de texto de Programación Estructurada<sup>1</sup>.

En la última parte se presentan algunos ejercicios resueltos, que sirvan de apoyo para explicar la forma de cómo resolver diferentes casos que se pueden presentar en cada uno de los ejercicios. Por motivos de la naturaleza del problemario no se anexa un apartado de la solución de los problemas. Sin embargo, los ejercicios resueltos presentan una descripción clara para comprender el uso de este problemario.

Al final del problemario se anexa una bibliografía con la finalidad de que el alumno pueda, si así lo requiere, consultar algún material para su apoyo en la interpretación y definición de variables.

Los conocimientos previos para el uso de este material son: Estructuras de control, estructuras de datos, instrucciones primitivas elementales, arreglos unidimensionales, arreglos bidimensionales, modularidad y registros.

---

<sup>1</sup> Albarrán Trujillo S.E, Salgado Gallegos M, **Programación Estructurada**, UAEM, 2008



## GUÍA DE USO

La presente guía de uso pretende orientar al alumno en el cómo resolver cada uno de los problemas que se presentan en este compendio de problemas describiendo las partes que componen cada ejercicio y ejemplificando un programa.

El cuadro que deberá ser completado por los alumnos es el siguiente (ver Figura 1):

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>	<b>PARTE 3</b>	<b>PARTE 4</b>
E					
P		<b>PARTE 1</b>			
S					
<b>Algoritmo</b>					
<b>PARTE 2</b>					
<b>Prueba de Escritorio</b>					
<b>PARTE 5</b>					

Figura 1. Formato para solución de problemas.

### DESCRIPCIÓN DEL FORMATO

## PARTE 1. Definición de Variables:

La definición de variables se compone por las partes:

- Nombre
- Tipo
- Descripción
- Renglones con letras E, P y S

Las cuales se muestran y describen a continuación (ver Figura 2):

	NOMBRE	TIPO	DESCRIPCIÓN
E			
P			
S			

Figura 2. Apartado para Definición de Variables

Donde:

*Nombre:* En esta columna se deben diferencias entre variables de entrada proceso y salida. Se debe escribir el nombre que se le dará a la variable, el cual no deberá contener caracteres especiales (#,\$,\*,etc), ni espacios.

*Tipo:* En esta columna se especifica el tipo de datos que puede tomar la variable (entero-E, real-R, carácter-C, lógico-L, cadena-S)

*Descripción:* Se debe escribir lo que representa la variables, por ejemplo si representa un área, edad, nombre, etc.

*E, P y S:* Representan las variables de entrada-E, proceso-P y salida-S.

## PARTE 2. Algoritmo

El algoritmo es una descripción en español (común) de la solución del problema, marcando los pasos a seguir (ver figura 3).

<b>Algoritmo</b>

Figura 3. Apartado para Algoritmo

## PARTE 3. Pseudocódigo

Es una serie de instrucciones en español estructurado entre las que se encuentran:

<b>Tipo de Instrucciones</b>	<b>Instrucción a utilizar</b>
Inicio y fin del programa	Inicio Termina
Instrucción de entrada	Leer
Instrucción de salida	Escribe
Operadores aritméticos	+, -, *, /, mod (cálculo de residuo), div (división entera), sqrt (raíz cuadrada), ** (cálculo de potencias)
Operadores lógicos	Y O No
Operadores relacionales	<, >, <=, >=, =
Asignación	←
Concatenación	+

Tabla 1. Instrucciones Elementales

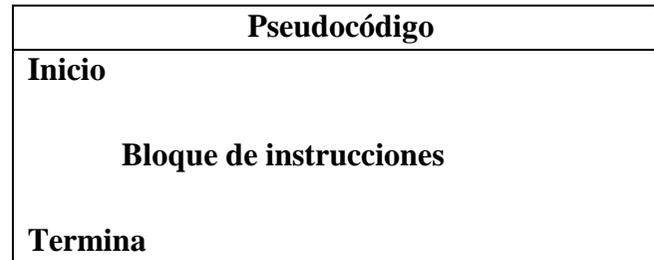
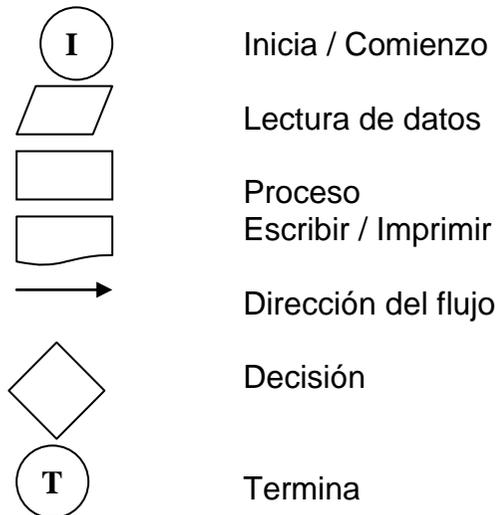


Figura 4. Apartado para Pseudocódigo

#### PARTE 4. Diagrama de Flujo de Datos

Se utiliza la siguiente nomenclatura:



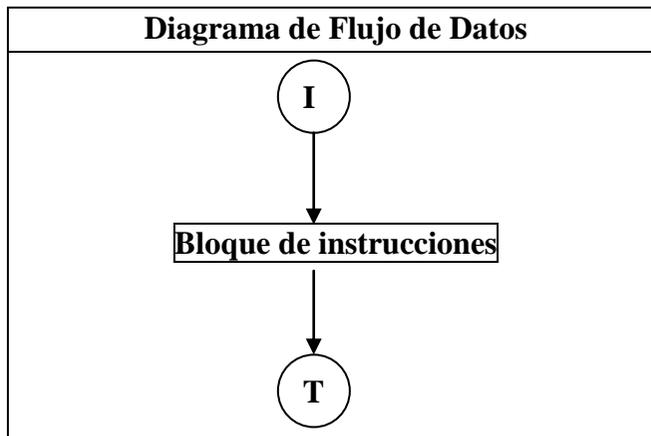


Figura 5. Apartado para Diagrama de Flujo de Datos

### **PARTE 5. Prueba de Escritorio**

En este apartado se deben listar las variables definidas y dar calores para comprobar las instrucciones (ver Figura 6).

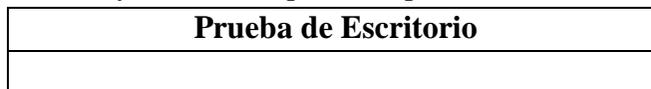


Figura 6. Apartado para Prueba de Escritorio

Para mayor entendimiento se presenta a continuación un ejercicio resuelto.

Escribir un programa que calcule en área de un triángulo dadas la base y la altura.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>		
E	<i>E</i>	<i>Be</i> <i>A</i>	<i>E</i> <i>E</i>		
P	<i>P</i>	<b>PARTE 1</b>			
S	<i>S</i>	<i>area</i>	<i>R</i>		
<b>Algoritmo</b>				<b>PARTE 3</b>	<b>PARTE 4</b>
1. Pedir base y altura del triángulo 2. Obtener la base y altura 3. Calcular el área 4. Mostrar el resultado					
<b>Prueba de Escritorio</b>				Inicio	
A	5			Escibe ("Teclea la base y altura")	
B	8			Leer A, B	
Area	20			$area=(A*B)/2$	
				Escibe ("El área del triángulo es:", area)	
				Termina	
<b>PARTE 5</b>					

**Donde por columnas en la Parte 1:**

- Los nombres de las variables son:  
 B: ubicada en el renglón de E (variables de entrada)

- A: ubicada en el renglón de E (variable de entrada)
- area: ubicada en el renglón S (variable de salida)
- El tipo de variables corresponde a:
  - E la primera variable declarada (B) es de tipo entero.
  - E la segunda variable declarada (A) es de tipo entero.
  - R la tercera variable declarada (area) es de tipo real.
- Descripción:

En el renglón correspondiente a cada variable, se describe lo que representa en el problema, es decir, B, representa la base del triángulo, A representa la altura del triángulo y area representa el área del triángulo.

#### **PARTE 2:**

Se plantearon cuatro pasos utilizando español común.

#### **PARTE 3.**

Es el planteamiento de pseudocódigo y se utilizan las instrucciones mencionadas en la Tabla 1.

#### **PARTE 4.**

Es el planteamiento de diagrama de flujo de datos y se utilizan las figuras mencionadas antes mencionadas en la página 11

#### **PARTE 5.**

Se realiza un listado de variables definidas y se realiza un seguimiento de las instrucciones en el pseudocódigo y/o diagrama de flujo de datos.



## SERIE DE EJERCICIOS

Realiza el análisis correspondiente y completa la tabla que se presenta después de cada caso.

1. Escribir un programa para calcular el cuadrado y el cubo de un número entero.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

2. Escribir un programa para calcular el perímetro de una circunferencia y el área del círculo cuyo radio se lee por el teclado.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

3. Escribir un programa que lea dos números enteros y calcule el residuo y el cociente entero de uno entre otro.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

4. Escribir un programa que calcule el área y perímetro de un pentágono dados el valor de un lado y la apotema.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

5. Escribir un programa que calcule la media de cinco valores. La media es la medida estadística más común. La media ( $\bar{y}$ ) de una muestra se define como la suma de los datos individuales ( $y_i$ ) dividido por el número de puntos ( $n$ ).

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

6. Realizar un programa que sume dos números enteros.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

7. Escribir un programa que calcule y escriba el cuadrado de 243.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

8. Realizar un programa que de calcule el perímetro y la superficie de un rectángulo dada la base y la altura.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

9. Realizar un programa que determine el área y volumen de un cilindro cuyas dimensiones radio y altura se leen desde el teclado.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

10. Escribir un programa que lea un nombre de una marca de automóviles seguida del nombre de su modelo e informe el modelo seguido del nombre.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

11. Diseñar un programa para determinar la hipotenusa de un triángulo rectángulo conocidas las longitudes de los catetos.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

12. Diseñar un programa que calcule el área de un triángulo en función de las longitudes de sus lados:  $\text{Área} = p((p-a)(p-b)(p-c))^{1/2}$ , donde  $p=(a+b+c)/2$  (semiperímetro).

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

13. Escribir un programa que lea un número y escriba su cuadrado.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

14. Realice la conversión de coordenadas polares a cartesianas

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

15. Realizar un programa que intercambie los valores de dos variables numéricas.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

16. Diseñar un programa que permita convertir pulgadas a yardas, y pies a pulgadas.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

17. Escribir un programa que imprima la conversión de kilómetros a millas. (1 milla marina = 1852 mts y 1 milla = 1609 mts).

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

18. Diseñar un programa que permita imprimir la conversión de centímetros a pulgadas (1 pulgada = 24.5 mm).

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

19. Escribir un programa tal que, dado el costo de un artículo vendido y la cantidad de dinero entregada por el cliente, calcule e imprima el cambio que debe entregársele al mismo.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

20. Diseñar un programa para resolver una ecuación de primer grado con una sola variable.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

21. Escriba un programa completo que calcule el volumen y el área de una esfera usando las fórmulas:

$$V = (4/3) \pi r^3$$

$$A = 4\pi r^2$$

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

22. Elaborar un programa para calcular la calificación final de un alumno, teniendo como datos 4 calificaciones parciales.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

23. Elabore un programa que lea números del 1 al 7 y despliegue "Domingo" si es 1, "Lunes" si es 2, "Martes" si es 3,...., "Sábado" si es 7.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

24. Elabore un programa similar al anterior pero para desplegar los meses.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

25. Escribir un programa que lea cuatro números y a continuación imprima el mayor de los cuatro.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

26. Realizar un programa que determine si un número es divisible entre 14.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

27. Realizar un programa que determine si un número es múltiplo de 6.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

28. Diseñar un programa para introducir un número y determinar si es par o impar.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

29. Realizar un programa tal que de dos números cualesquiera, encontrar la suma e indicar si es positiva, negativa o cero.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

30. Realizar un programa que determine si un número dado es divisible entre 2 y 5.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

31. Realizar un programa tal que dados dos números enteros determinar si un número es divisor de otro

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

32. Diseñar un programa que permita imprimir los enteros del 9 al 43.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

33. Diseñar un programa que permita imprimir los enteros impares del 7 al 51.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

34. Diseñar un programa que permita imprimir los enteros pares del 2 al 48.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

35. Diseñar un programa que imprima y sume la serie de números 3, 6, 9, ..., 99.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

36. Diseñar un programa que permita imprimir los enteros del 1 al 30, apareados con sus recíprocos

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

37. Diseñar un programa que permita imprimir una tabla de potencias del 2 que no exceda al 1000.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

38. Diseñar un programa que permita introducir un conjunto de 25 números. Determinar la cantidad de números positivos y negativos del conjunto.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

39. Diseñar un programa que permita imprimir la tabla de multiplicar hasta 12\*12

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

40. Diseñar un programa que permita introducir un entero positivo N. Encontrar la suma de los N enteros. Imprimir cada uno de los enteros y la suma.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

41. Elaborar un programa que dados N números imprima cuántos 0 (ceros) se teclearon.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

42. Diseñar un programa que permita introducir N enteros. Calcular e imprimir el producto de los números pares

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

43. Diseñar un programa que realice la división de dos números utilizando el método de restas sucesivas

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

44. Diseñar un programa que permita realizar la multiplicación de dos números utilizando el método de sumas sucesivas.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

45. Diseñar un programa que permita encontrar la suma de los cuadrados de los enteros del 1 al N. Es decir, su programa calculará:  $1^2+2^2+3^2+\dots+N^2$ .

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

46. Diseñar un programa que permita calcular la suma de las raíces cuadradas de los números impares que hay entre 1 y 1000.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

47. Escribir un programa para calcular el factorial,  $n!$ , de un entero positivo  $n$  cualquiera. El programa debe leer el valor de  $n$  e imprimir  $n!$  después de calcularlo.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

48. Diseñar un programa que permita leer un número N y calcular la suma de la siguiente serie:  
 $11 + 22 + 33 + \dots + NN$

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

49. Diseñar un programa que permita encontrar el mayor número entre N números no nulos. Su programa calculará N, contando el número de valores no nulos que precede a un cero final.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

50. Diseñar un programa que permita leer una serie de números distintos de cero ( el último número de la serie es -99) obtener el número mayor. Como resultado se debe visualizar el número mayor y un mensaje de indicación de número negativo, caso de que se haya leído un número negativo.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

51. .Escribir un programa que calcule y visualice el más grande, el más pequeño y la media de N números. El valor de N se solicitará al principio del programa y los números serán introducidos por el usuario.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

52. Escribe un programa que con un arreglo unidimensional, tamaño de 5, asignándole los valores numéricos desde teclado y mostrarlos.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

53. Escribir un programa con un arreglo de n posiciones y llenarlo con nombres de personas.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

54. Escribir un programa con dos arreglos unidimensionales, uno que almacene 2 nombres y otro que almacene 3 números.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

55. Escribe un programa que sume los elementos de dos vectores y guarde el resultado en otro vector.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

56. Escribe un programa que llene un vector con números enteros (números positivos y negativos) y muestre la cantidad de números positivos y la cantidad de ceros que hay en dicho arreglo.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

57. Escribe un programa que llene dos arreglos con números proporcionados desde teclado, sumar las posiciones pares de cada vector.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

58. Escribe un programa que genere una matriz de 3x3 y almacene números impares.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

59. Escribir un programa que genere una matriz de n filas y n columnas, con números desde teclado.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

60. Escribir un programa que pida al usuario 12 datos, los guarde en una matriz y luego muestre su media y moda.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

61. Escribir un programa que calcule el determinante de una matriz de 2x2, a partir de datos que introduzca el usuario.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

62. Crea un programa que llene una matriz con 15 datos, luego pregunte al usuario qué dato desea buscar, avise si ese dato no aparece, y que en caso contrario diga cuántas veces se ha encontrado.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

63. Considere que usted es el dueño de una tienda de herramientas y necesita mantener un inventario que le permita llevar un control mínimo de cuáles herramientas tiene, cuántas tiene y el costo de cada una.  
 Escriba un programa que: a) permita listar todas sus herramientas b) permita introducir nuevas herramientas. C) permita actualizar la información de las herramientas.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

64. Generar un programa que cuente con un menú principal que permita seleccionar las opciones: agregar, consultar, modificar y eliminar auto. Para cualquier entrada de autos mínimo se almacenará: placa, tipo, marca, modelo, clasificación, estado, reparaciones.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

65. Escribir un programa que mediante un menú y considerando un registro de tipo neumático que contenga clave, tamaño, distribuidor y precio. Donde el distribuidor tenga a su vez clave, nombre, dirección. a) Definir el registro b) Definir una variable tipo arreglo que contenga 10 neumáticos c) Capturar los 10 neumáticos d) Buscar neumático por clave e) Buscar neumático más caro y desplegar información f) Modificar información

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

66. Escribir un programa mediante modularización que calcule el área, perímetro y diámetro de un círculo de radio R.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

67. Elabore un programa modular para calcular la suma, el promedio, el máximo y el mínimo de un vector de 100 elementos.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

68. Definir la función FACTORIAL de forma recursiva e iterativa sabiendo que el factorial de N es 1 si N es 0 y en otro caso será N veces el factorial de N-1.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

69. Definir una función que pase de grados centígrados a grados Fahrenheit, sabiendo que:  $F = (C + 40) \times 1.8 - 40$

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

70. Escribir un programa que rellene un arreglo con los 100 primeros números enteros y los muestre en pantalla en orden ascendente.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

71. Escribir un programa que rellene un vector con los números pares comprendidos entre 1 y 100 y los muestre en pantalla en orden ascendente

<b>Definición de Variables</b>				<b>Pseudocódigo</b>	<b>Diagrama de Flujo de Datos</b>
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

72. Escribir un programa que lea 10 números por teclado, los almacene en un vector y muestre la suma, resta, multiplicación y división de todos.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

73. Escribir un programa que lea 10 números por teclado, 5 para un vector y 5 para otro vector distinto. Mostrar los 10 números en pantalla mediante un solo vector.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

74. Escribir un programa que rellene un vector con los números impares comprendidos entre 1 y 100 y los muestre en pantalla en orden ascendente

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

75. Escribe un programa que muestre los primeros 100 números de izquierda a derecha usando un arreglo de dos dimensiones.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

76. Escribir un programa que rellene un arreglo de dos dimensiones con números pares y después que pida una posición X,Y y mostrar el número correspondiente.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

77. Escribir un programa que rellene una matriz de 3x3 y muestre su traspuesta (la traspuesta se consigue intercambiando filas por columnas y viceversa).

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

78. Escribir un programa que capture en una matriz la producción de 10 plantas por treinta días; muestre y determine la planta con más ventas en un mes.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

79. Escribir un programa utilizando registros que gestione los datos de stock de una tienda de comestibles, la información a recoger será: nombre del producto, precio, cantidad en stock. La tienda dispone de 10 productos distintos. El programa debe ser capaz de:
- Dar de alta un producto nuevo.
  - Buscar un producto por su nombre.
  - Modificar el stock y precio de un producto dado

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

80. Escribir un programa utilizando registros que gestiona las notas de una clase de 20 alumnos de los cuales sabemos el nombre y la nota. El programa debe ser capaz de:
- Buscar un alumno.
  - Modificar su nota.
  - Realizar la media de todas las notas.
  - Realizar la media de las notas menores de 5.
  - Mostrar el alumno que mejores notas ha sacado.
  - Mostrar el alumno que peores notas ha sacado.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

81. Escribir un programa utilizando registros que permita guardar datos de "imágenes" De cada imagen se debe guardar: nombre (texto), ancho en píxeles (por ejemplo 2000), alto en píxeles (por ejemplo, 3000), tamaño en Kb (por ejemplo 145,6). El programa debe ser capaz de almacenar hasta 700 imágenes (deberá avisar cuando su capacidad esté llena). Debe permitir las opciones: añadir una ficha nueva, ver todas las fichas (número y nombre de cada imagen), buscar la ficha que tenga un cierto nombre.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

82. Escribir un programa modular que calcule el área y perímetro de un círculo. Una función deberá recibir el radio y regresar el valor del área, otra función deberá recibir el diámetro y regresar el valor del perímetro. El programa principal deberá pedir el valor del radio.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

83. Escribir un programa modular que calcule el costo de una serie de productos. Una función deberá leer la cantidad de producto x comprado y su precio, calcular el costo y regresarlo. El programa principal deberá permitir que el usuario dé tantos productos como quiera. Se deberá mostrar el costo final de todos los productos deseados por el usuario.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					

84. Escribir un programa modular que permita mediante un menú calcular área de las siguientes figuras: rombo, trapezoide, triángulo y trapecio. Cada figura deberá recibir como parámetros los valores necesarios para calcular el área y regresar el valor de la misma área.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>		
E					
P					
S					
<b>Algoritmo</b>					
<b>Prueba de Escritorio</b>					



## EJERCICIOS RESUELTOS

85. Escribir un programa que sume los números de 1 al 1000 no terminados en 5.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
E				<p>Inicio</p> <p>Para i=1 hasta i&lt;=1000</p> <p>  Si(i mod 5 =0 y i mod 10=1)entonces</p> <p>    S=s+i</p> <p>  Fin Si</p> <p>Fin Para</p> <p>Escribe("la suma es:", s)</p> <p>Termina</p>	<pre> graph TD     I((I)) --&gt; I1[i=1]     I1 --&gt; D1{i &lt;= 1000}     D1 -- No --&gt; O1((T))     D1 -- Yes --&gt; D2{i mod 5 = 0 y i mod 10 = 1}     D2 -- Yes --&gt; R1[s = s + i]     R1 --&gt; D2     D2 -- No --&gt; O2[/La suma es, s/]     O2 --&gt; O1           </pre>
P	i	E	Contador de los 1000 números		
S	s	E	Suma de números no terminados en 5		
<b>Algoritmo</b>					
<p>Inicio</p> <p>Verificar si contador es divisor de 5</p> <p>Verificar que contador no sea divisor de 10</p> <p>Si se cumplen las condiciones entonces</p> <p>  Realizar suma</p> <p>Sino</p> <p>  No hacer nada</p> <p>Mostrar resultados</p>					
<b>Prueba de Escritorio</b>					
Se realiza la prueba con 10 números.					

Se definen dos variables:

- No hay de entrada:
- Una variable de proceso
  - i. I de tipo entero que es el contador de los 1000 números
- Una variable de salida
  - i. s de tipo entero que representala suma de los número no terminado en cinco

86. Elabore un programa que calcule el sueldo de un empleado; si el número de horas trabajadas es mayor que 40, el excedente de 40 se paga al doble de la cuota por hora. En caso de no ser mayor que 40 se paga la cuota normal. El usuario debe determinar cuál es la cuota para la hora normal.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>	<p>Inicio</p> <p>Escribe("Dame el número de horas trabajadas")</p> <p>Lee(Horast)</p> <p>Escribe("Dame Cuota por hora")</p> <p>Lee(cuota)</p> <p>Si(Horast&gt;40)entonces</p> <p>    Horase=Horast-40</p> <p>    Sueldo=40*cuota+Horase*2*cuota</p> <p>Otro</p> <p>    Sueldo=Horast*cuota</p> <p>FinSi</p> <p>Escribe ("El sueldo es: ". Sueldo)</p> <p>Termina</p>	<pre> graph TD     I((I)) --&gt; A[Dame el número de horas trabajadas]     A --&gt; B[/Horast/]     B --&gt; C[Dame cuota por hora]     C --&gt; D[/cuota/]     D --&gt; E{Horast&gt;40}     E -- No --&gt; F[Sueldo=Horast*cuota]     E -- Yes --&gt; G[Horase=Horast-40]     G --&gt; H[Sueldo=40*cuota+Horase*2*cuota]     F --&gt; I1[El sueldo es :. Sueldo]     H --&gt; I1     I1 --&gt; T((T))         </pre>
E	Horast cuota	<b>R</b> <b>R</b>	Horas trabajadas Cuota por hora		
P	- Horase	<b>R-</b>	-Horas extra		
S	sueldo	<b>R</b>	Sueldo del empleado		
<b>Algoritmo</b>					
<p>Inicia</p> <p>    Pedir el número de horas trabajadas y cuota por hora</p> <p>    Leer número de horas trabajadas y cuota por hora.</p> <p>    Si es mayor que 40</p> <p>        Obtener diferencia de horas trabajadas</p> <p>        Calcular pago de excedente</p> <p>    Sino</p> <p>        Calcular cuota normal</p> <p>Mostrar resultados</p> <p>Termina</p>					
<b>Prueba de Escritorio</b>					
<p>Horast=50</p> <p>cuota=20</p> <p>C1 = 1</p> <p>Horase=10</p> <p>Sueldo=40*20+10*2*20=1200</p>					

☞ Se definen tres variables:

- *Dos de entrada:*
  - i. *Horast, de tipo real y representa las horas trabajadas*
  - ii. *Cuota de tipo real y representa la cuota por hora*
- *No hay variables de proceso*
- *Una variable de salida*
  - i. *Sueldo que representa el sueldo del empleado*

87. Dado el peso, la altura y el sexo de N personas que pertenecen a un estado de la República; obtenga el promedio del peso y la altura de esta población. Imprimir ambos promedios.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>		
E	Peso	R	Peso		
	Altura	R	Altura		
	Sexo	C	Sexo		
	N	E	Número de personas		
	Suma P	R	Acumulador de pesos		
P	SumaA	R	Acumulador de Alturas		
	promedioP	R	Promedio del peso		
S	promedioA	R	Promedio de altura		
	<b>Algoritmo</b>				
<p>Inicia</p> <p>Pedir número de personas.</p> <p>Leer número de personas</p> <p>Para contador menor número de personas</p> <p>Pedir peso, altura, sexo</p> <p>Leer peso, altura, sexo</p> <p>Incrementar contador</p> <p>Incrementar suma de Pesos</p> <p>Incrementar suma de Altura</p> <p>Obtener promedio</p> <p>Mostrar promedios</p> <p>Termina</p>				<p>Inicia</p> <p>Escribe("Dame el número de personas")</p> <p>Lee(N)</p> <p>Para i=1 hasta i=N</p> <p>Escribe("Dame peso, altura, sexo(M/F)")</p> <p>Lee(Peso, Altura, Sexo)</p> <p>sumaP = sumaP + Peso</p> <p>sumaA = sumaA + Altura</p> <p>FinPara</p> <p>promedioP=sumaP/N</p> <p>promedioA=sumaA/N</p> <p>Escribe("El promedio de pesos es:", promedioP)</p> <p>Escribe("El promedio de altura es:", promedioA)</p> <p>Termina</p>	

<p style="text-align: center;"><b>Prueba de Escritorio</b></p> <p>N=4  I=1,2,3,4  Peso=10,15,12,19  Altura=100,110, 90, 112  Sexo=f, f, m, f  sumaP=56  sumaA=412  promedioP=14  promedioA=103</p>		
--	--	--

*Se definen ocho variables:*

- *Cuatro de entrada:*
  - i. *Peso de tipo real representa el peso de cada persona y se lee de teclado*
  - ii. *Altura de tipo real que representa la altura de la persona y se lee de teclado*
  - iii. *Sexo de tipo carácter que representa el sexo de la persona y se lee de teclado*
  - iv. *N de tipo entero que representa el número de personas y se lee de teclado*
- *Dos de proceso*
  - i. *P de tipo real que es un acumulador de pesos*
  - ii. *A de tipo real que es un acumulador de alturas*
- *Dos variables de salida*
  - i. *PromedioA de tipo real que es el promedio de alturas*
  - ii. *PomedioP de tipo real que es el promedio de pesos.*

88. Construir un programa tal que dado como dato Y, calcule el resultado de la siguiente función:

$$\begin{aligned}
 X &= 3Y + 36 && \text{si } 0 < Y \leq 11 \\
 X &= Y^2 - 10 && \text{si } 11 < Y \leq 33 \\
 X &= Y^3 + Y^2 - 1 && \text{si } 33 < Y \leq 64 \\
 X &= 0 && \text{para cualquier otro valor de Y}
 \end{aligned}$$

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>	Inicio Pedir y Leer y Si (0 < y ≤ 11) entonces $X = 3 * y + 36$ Otro Si (11 < y ≤ 33) entonces $X = y^2 - 10$ Otro Si (33 < y ≤ 64) entonces $X = y^3 + y^2 - 1$ FinSi FinSi FinSi Termina	
E	y	R	Dato dado por usuario		
P	-	-	-		
S	X	R	Resultado de la función.		
<b>Algoritmo</b>					
Inicia Pedir y Leer y Verificar condición y evaluar función Mostrar resultados Termina					

<b>Prueba de Escritorio</b> Y=12 X=4 C1=0 C2=1		
--	--	--

☞ Se definen dos variables:

- Una de entrada:
  - i. y de tipo real representa el datos que se lee de teclados
- No hay variables de proceso
- Una variable de salida
  - i. X de tipo real que representa el valor que tomará la función.

89. Dados tres datos enteros positivos que representan las longitudes de los lados de un probable triángulo. Hacer un programa que determine si los datos corresponden a un triángulo. En caso de que sí correspondan, escribir si el triángulo es equilátero, isósceles o escaleno. Calcular además su área. Considerar que es triángulo si se cumple que la suma de los dos lados menores es mayor que la del lado mayor. Tome en cuenta además que el área de un triángulo se calcula:  $(S(S-A)(S-B)(S-C))/2$ . S es la mitad de la suma de los lados, es decir,  $S=(A+B+C)/2$ .

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos		
	NOMBRE	TIPO	DESCRIPCIÓN				
E	A	E	Lado del triángulo	<p>Inicia  Escribe("Dame longitud de lados del triángulo")  Leer A, B, C  Si (A = B = C) entonces  Escribe("Triángulo equilátero")  Otro  Si (A &lt;&gt; B &lt;&gt; C) entonces  Escribe("Triángulo escaleno")  Otro  Escribe("Triángulo isósceles")  FinSi  FinSi  <math>S=(A+B+C)/2</math>  <math>area=(S(S-A)(S-B)(S-C))/2</math>  Escribe ("Área: ", area)  Termina</p>	<pre> graph TD     I((I)) --&gt; P1[Dame longitud de lados del triángulo]     P1 --&gt; O1[/A,B,C/]     O1 --&gt; D1{A=B=C}     D1 -- Yes --&gt; P2[/Triángulo equilátero/]     D1 -- No --&gt; D2{A&lt;B&lt;C}     D2 -- Yes --&gt; P3[/Triángulo escaleno/]     D2 -- No --&gt; P4[/ángulo isósceles/]     P2 --&gt; P5[S=(A+B+C)/2]     P3 --&gt; P5     P4 --&gt; P5     P5 --&gt; P6[area=(S(S-A)(S-B)(S-C))/2]     P6 --&gt; O2[/Area: , area/]     O2 --&gt; T((T))     </pre>		
	P	S	R			Semiperímetro	
	S	area	R			área del triángulo	
<b>Algoritmo</b>							
<p>Inicio  Pedir 3 lados  Leer A, B, C  Verificar condiciones de lados  Mostrar tipo de triángulo  Obtener suma de la mitad de los lados  Obtener área  Mostrar área  Termina</p>							

<p style="text-align: center;"><b>Prueba de Escritorio</b></p> <p>A=10  B=12  C=12  C1=0  C2=1  S=17  Area=1487.5</p>		
---	--	--

☞ Se definen cinco variables:

- Tres de entrada:
  - i. A de tipo entero representa el primer lado del triángulo
  - ii. B de tipo entero representa el segundo lado del triángulo
  - iii. C de tipo entero representa el tercer lado del triángulo
- Una variable de proceso
  - i. S de tipo real que representa el semiperímetro del triángulo
- Una variables de salida
  - i. area de tipo real que representa el área del riángulo

90. Pedro invierte \$100 a una tasa del 8% compuesta anual. Jorge invierte \$150 a una tasa de 5% compuesta anualmente. ¿Después de cuántos años la cuenta de Pedro rebasará la de Jorge? Realizar un programa que calcule los años.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
E				Inicia CP=100 CJ=150 Mientras (cp < pj) hacer CP=CP+ CP*8/100 CJ=CJ+CJ*5/100 i=i+1 FinMientras Escribir ("Años:", i); Termina	<pre>                     graph TD                         I((I)) --&gt; CP[CP=100]                         CP --&gt; CJ[CJ=150]                         CJ --&gt; D{cp &lt; cj}                         D -- Yes --&gt; CP2[CP=CP+CP*8/100]                         CP2 --&gt; CJ2[CJ=CJ+CJ*5/100]                         CJ2 --&gt; Iplus[I=i+1]                         Iplus --&gt; D                         D -- No --&gt; Out[/Años, i/]                         Out --&gt; T((T))                     </pre>
P	CP CJ	R R	Cantidad ahorrada por Pedro Cantidad ahorrada por Jorge		
S	i	E	Contador de años		
<b>Algoritmo</b>					
Obtener Cantidad con interés de Pedro Obtener Cantidad con interés de Jorge Incrementar i Verificar si cantidad de Pedro es menor a cantidad de Jorge Si es menor repetir pasos anteriores.					

Prueba de Escritorio		
CP=100,	108,	116.64, 125.97, 136.04, 146.92,
		158.67, 171.36, 185.06, 199.86, 215.84, 233.10,
		251.74, 271.87, 293.62, 317.10
CJ=150,	157.5,	165.37, 173.63, 182.31, 191.42,
		200.99, 211.03, 221.58, 232.65, 244.28, 256.49,
		269.31, 282.77, 296.90, 311.74
I=0,	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	

☞ Se definen dos variables:

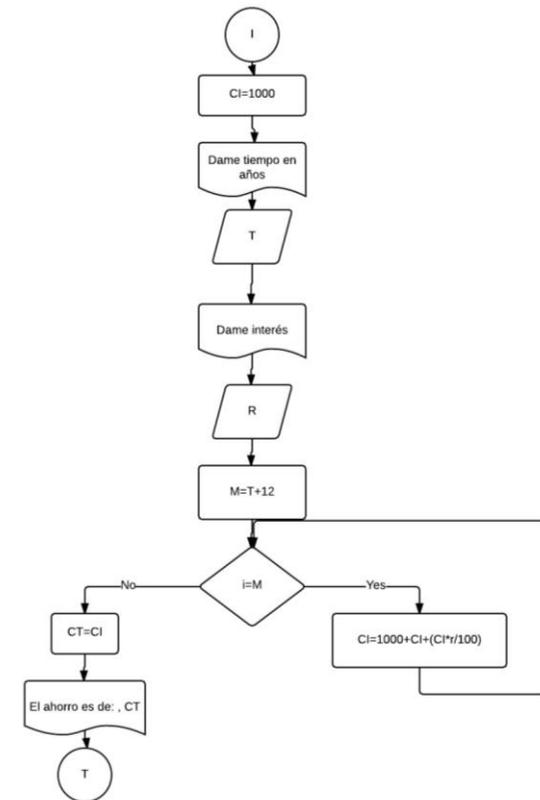
- No hay variables de entrada:
- Dos variables de proceso
  - i. CP de tipo real que representa la cantidad de dinero ahorrada por Pedro
  - ii. CJ de tipo real que representa la cantidad de dinero ahorrada por Jorge
- Una variable de salida
  - i. I de tipo entero que representa el año de ahorro.

91. Realizar un programa que calcule la cantidad que tendría ahorrada después de T años, si se depositan mil pesos mensualmente a una tasa de interés mensual del R% capitalizable cada mes, es decir, que al capital se le agregan los intereses.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>		
E	T CI R	E R R	Años Cantidad Inicial Interés	Inicia CI=1000 Escribe("Dame tiempo en años") Lee(T) Escribe("Dame interés") Lee( R) M=T*12; Para i=1 hasta i=M CI= 1000+CI+(CI*R/100) FinPara CT=CI Escribe("El ahorro es de: ", CT) Termina	
P	i M	E E	Contador Meses		
S	CT	R	Cantidad Total		
<b>Algoritmo</b>					
Inicia Inicializa primer pago en 1000 Pedir años y tasa de interés Leer años y tasa de interés Obtener número de meses Para i menor al número de años Obtener interés y ahorro Mostrar resultado Termina					

### Prueba de Escritorio

$T=1$   
 $CI=1000, 2100, 3121, 4152.21, 5567.43, 7124.17,$   
 $8836.58, 10720.23, 12792.25, 15071.47, 16222.18,$   
 $18844.39, 21728.82$   
 $R=10$   
 $i=1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12$   
 $M=12$   
 $CT=21728.82$



☞ Se definen cinco variables:

- Tres de entrada:
  - $T$  de tipo entero representa los años que se hará el ahorro.
  - $CI$  de tipo real que representa la cantidad inicial de ahorro.
  - $R$  de tipo real que representa la tasa de interés del ahorro.
- Una variable de proceso
  - $I$  de tipo entero que es el contador de los años.
- Una variables de salida
  - $CT$  de tipo real que representa la cantidad total de ahorro después de  $T$  años.

92. Escribir un programa que lea un número real positivo y determine y despliegue el número de dígitos a la izquierda del punto decimal (sugerencia: divida repetidamente el número por 10 hasta que sea menor que 1).

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>	Inicia Escribe("Dame el número") Lee(x) Mientras(x>1) $x=x/10$ $i=i+1$ FinMientras Escribe("El número de dígitos a las izquierda es: ", i) Termina	<pre>                     graph TD                         Start((I)) --&gt; Input[ Dame el número ]                         Input --&gt; Output[ x ]                         Output --&gt; Decision{ x &gt; 1 }                         Decision -- Yes --&gt; Process1[ x = x / 10 ]                         Process1 --&gt; Process2[ i = i + 1 ]                         Process2 --&gt; Decision                         Decision -- No --&gt; Output2[ "El número de dígitos a las izquierda es:", i ]                         Output2 --&gt; End((T))                     </pre>
E	x	E	Númeoro que indica el usuario		
P					
S	I	E	Contador de dígitos a la izquierda del punto decimal		
<b>Algoritmo</b>					
Pedir número. Leer numero Verificar que numero sea mayor a 1 $\text{Dividir el numero entre } 10$ Incrementar el contador Repetir paso 3 hasta que sea menor Mostrar resultados					
<b>Prueba de Escritorio</b>					
$x=25980, 2598,259, 25, 2,$ $I=1,2,3,4,5$					

Se definen cinco variables:

- *Una entrada:*
  - i. *T de tipo entero representa los años que se hará el ahorro.*
- *No hay variables de proceso*
- *Una variables de salida*
  - i. *I de tipo entero l que representa Contador de dígitos a la izquierda del punto decimal*

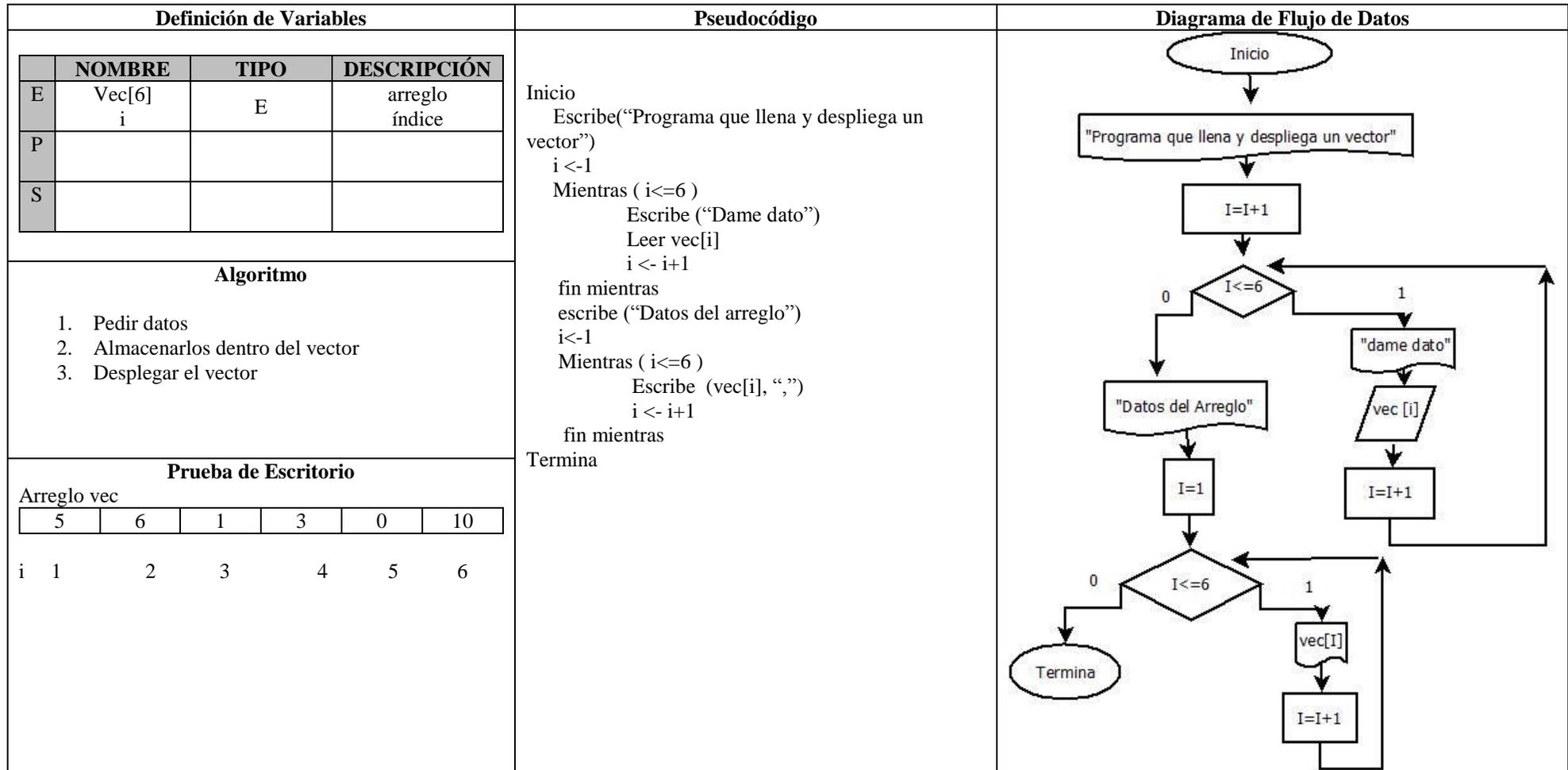
93. Escriba un programa interactivo que permita al usuario intentar hasta cuatro veces el teclear la capital de Sudáfrica. La siguiente es una corrida típica:  
 ¿Cuál es la capital de Sudáfrica ? El Cairo Mal, intente de nuevo. Marruecos Mal, intente de nuevo. Johannesburgo Correcto, lo logró en el intento 3.  
 Si el usuario no teclea la capital en cuatro intentos, su programa deberá indicarle la respuesta correcta.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	NOMBRE	TIPO	DESCRIPCIÓN		
E	capital	cadena	Capital introducida por el usuario	Inicia Mientras( $c < 4$ y $b = 0$ ) Escribe("Dame la capital de Sudáfrica") Lee(capital) Si ( $capital = "Johannesburgo"$ ) Escribe("Correcto, lo logró en el intento ", c) $b = 1$ Otro Escribe(capital, "intente de nuevo") FinSi $c = c + 1$ FinMientras Si ( $c = 4$ ) entonces Escribe("La respuesta correcta es: Johannesburgo") FinSi Termina	
P					
S	c	E	Contador de intentos		
<b>Algoritmo</b>					
Mientras c menor que 4 Pedir capital Leer capital Si la capital es correcta indicar Sino incrementa contador Si c es igual a 4 Muestra la respuesta correcta					
<b>Prueba de Escritorio</b>					
C=1,2, 3 capital="DF", Madrid", "Johanesburgo" C1=1, 1, 1, 0 C2=0, 0, 1 B=0, 1 C3=0					

Se definen cinco variables:

- *Una de entrada:*
  - i. *capital de tipo cadena* representa la capital que introduce el usuario.
- *No hay variables de proceso*
- *Una variables de salida*
  - i. *CT de tipo real* que representa la cantidad total de ahorro después de  $T$  años.

94. Programa que llena y despliega un vector.



95. Escribir un programa que en la diagonal principal se llene por unos.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
E	A [100,100]	E	Arreglo Bidimensional	<p>Inicio                      Escribe (“Programa matriz diagonal principal unos”)                      Repite                      Escribe (“Ingresa dimensión de la matriz cuadrada”)                      Leer n                      Hasta( <math>n \geq 2</math> y <math>n &lt; 100</math>)                      Para r=1 hasta n(inc)                      Para c=1 hasta n(inc)  <math>A[r,c] &lt; 0</math>                      Fin para                      Fin para                      Para r=1 hasta n(inc)  <math>A[r,r] = 1</math>                      Fin para                      Termina</p>	
P	r,c	E	índices		
S					
<b>Algoritmo</b>					
Pedir dimensión de la matriz Leer dimensión Utilizar bloque de instrucciones para llenar matriz con ceros (dos ciclos anidados) Utilizar un ciclo para asignar unos a la matriz principal					
<b>Prueba de Escritorio</b>					
Arreglo A					
	1	0	0		
	0	1	0		
	0	0	1		
r	1,2,3,4				
c	1,2,3,4,1,2,3,4,1,2,3,4,				

96. Escribir un programa con modularización que lea un número par e imprimir sus divisores.

Definición de Variables				Pseudocódigo	Diagrama de Flujo de Datos
	<b>NOMBRE</b>	<b>TIPO</b>	<b>DESCRIPCIÓN</b>		
E	i	E	índice	Principal() Inicio	<pre> graph TD     A[Principal()] --&gt; B[Leer_Par()]         </pre>
	Leer_Par()	E	Modulo Par	Escribir('Ingrese un número par')	
	N	E	Número Par	Leer_Par()	
P				Para i=1 hasta N (inc)	
				Si (N mod i=0) entonces	
				Escribir(i,'Es divisor')	
S				Fin si	
				Fin para	
				Termina	
				Leer_Par() Inicio	
				Leer(N)	
				Mientras (N mod 2 <> 0)	
				Escribir('El número debe ser par')	
				Leer N	
				Fin mientras	
				Termina	
<b>Prueba de Escritorio</b>					
Variables Globales					
Par	i	N			
3	1	3			
8	2				
	3				



## BIBLIOGRAFÍA

1. Albarrán Trujillo S.E, Salgado Gallegos M, **Programación Estructurada**, UAEM, 2010. ISBN. 978-607-422-127-5.
2. Cairó, Osvaldo, **Metodología de Programación**, Tomos I y II, Computec, Colombia, 2006. ISBN. 978-9701511008.
3. Cairó, Osvaldo. **Fundamentos de Programación. Piensa en C**, Pearson Educación, 2006. ISBN. 978-9702608103..
4. Casale Juan Carlos. **Introducción a la programación: Ingreso al mundo de la programación (Colección Introducción a la programación nº 3)**. Ed. Creative Andina Corp; 2012. ASIN: B00IP0159C
5. Gorka Urrutia Landa. **Curso de programación en C para principiantes: Aprende a programar en C desde cero**. Kindle Edition. 2012, 3a edición. ASIN: B006J9SXVE
6. Joyanes, Aguilar Luis, **Programación en C**, McGraw Hill, México, 2010. ISBN-10: 6071512123. ISBN-13: 978-6071512123.
7. Joyanes, Aguilar Luis, **Algoritmos, Programacion y Estructura de Datos**, McGraw-Hill Interamericana, Febrero 2006. ISBN 978-8448145149
8. Joyanes, Aguilar Luis, **Programacion en C++**, McGraw-Hill, ISBN 978-8448146436
9. Smith John, **Fundamentos de programación aplicados a C#: El manual imprescindible para la iniciación en el mundo del desarrollo de aplicaciones informáticas en C#**, ed. CreateSpace. Octubre 2015, ASIN: B0179VR0AA.
10. Zavala López Raúl Antonio, Llamas Avalos Roberto. **Fundamentos de programación para principiantes**. Kindle Edition, 2013. ASIN: B00CW4Q94I.