

## TECNICAS DE PROGRAMACION – TRABAJO PRACTICO DE LA MATERIA - GUIA DE EJERCICIOS

### Parte 1 – Fundamentos Básicos (Entrada, Proceso, Salida)

1. Construya un diagrama de flujo tal que dado los valores enteros A y B, muestre el resultado de la siguiente expresión:  $(A+B)^2/3$ . PS -DF
2. Dada la matricula(DNI) y 5 calificaciones de un alumno obtenidas a lo largo del semestre, construya un diagrama de flujo que imprima la matricula del alumno y el promedio de sus calificaciones. PS -DF
3. Suma de 2 números: Escribir un algoritmo que pida al usuario dos números y muestre en pantalla la suma de ambos. PS -DF
4. Calculo de Edad: Escribir un algoritmo que pida al usuario su año de nacimiento y luego muestre en su edad actual. PS -DF
5. Conversión de temperatura: Escribir un algoritmo que convierta una temperatura en grados Celsius a Fahrenheit. El usuario debe ingresar la temperatura en grados Celsius. PS -DF
6. Calculo del área de un triángulo: Escribir un programa que calcule el área de un triángulo, dados su base y altura. El usuario debe ingresar la base y la altura. PS -DF
7. Calculo del promedio de Notas: Escribir un algoritmo que calcule el promedio de tres notas ingresadas por el usuario. PS -DF
8. Conversión de moneda: Escribir un programa que convierta una cantidad de pesos argentinos a dólares estadounidenses(blue), utilizando un tipo de cambio fijo. El usuario debe ingresar la cantidad de pesos. PS -DF
9. Calculo de distancia entre 2 puntos: Escribir un programa que calcule la distancia entre dos puntos en un plano cartesiano, dados sus coordenadas. El usuario debe ingresar las coordenadas de ambos puntos. PS -DF
10. Calculo de raíz cuadrada: Escribir un algoritmo que calcule la raíz cuadrada de un número ingresado por el usuario. PS -DF
11. Conversión de tiempo: Escribir un algoritmo que convierta una cantidad de minutos a horas y minutos. El usuario debe ingresar la cantidad de minutos. PS -DF
12. Cálculo de área de un círculo:  $(\pi \times r^2)$ : Escribir un programa que calcule el área de un círculo, dado su radio. El usuario debe ingresar el valor del radio. PS -DF
13. Diseñar un algoritmo que solicite al usuario ingresar un número base y un exponente. El algoritmo debe calcular la potencia correspondiente y mostrar el resultado. **Clasificación:** Estructura Secuencial – Uso de operador de potencia. PS -DF
14. Diseñar un algoritmo que solicite al usuario ingresar un número radicando y su índice. El algoritmo debe calcular la raíz indicada y mostrar el resultado. **Clasificación:** Estructura Secuencial – Uso de operador de raíz (potencia fraccionaria). PS -DF
15. Realizar la prueba de escritorio del ejercicio 11. **Clasificación:** Verificación lógica – Tabla de trazabilidad.
16. Realizar la prueba de escritorio del ejercicio 12. **Clasificación:** Verificación lógica – Tabla de trazabilidad.

### Parte 2 – Estructuras Selectivas (Condicionales)

17. Clasificación de números: Desarrollar un algoritmo que clasifique un número entero como par o impar. PS -Py
18. Validación de edad: Crear un algoritmo que pida al usuario su edad y determine si es mayor de edad o no. PS -Py
19. Cálculo de descuento: Desarrollar un algoritmo que calcule el descuento aplicable a un producto según su precio y su categoría. PS -Py

20. Clasificación de triángulos: Crear un algoritmo que determine si un triángulo es equilátero, isósceles o escaleno según la longitud de sus lados. PS - Py
21. Simulación de semáforo: Desarrollar un algoritmo que simule el funcionamiento de un semáforo, cambiando de color según la entrada del usuario. PS - Py
22. Menor o mayor: Escribir un algoritmo que determine si un número es menor o mayor a otro PS - Py
23. Diseñar un algoritmo que permita al usuario ingresar dos números e indique cuál es mayor o si son iguales.  
**Clasificación:** Estructura Selectiva Simple/Compuesta. PS - Py
24. Diseñar un algoritmo que solicite al usuario ingresar un número e indique si es par.  
**Clasificación:** Estructura Selectiva Simple – Uso del operador módulo. PS - Py
25. Diseñar un algoritmo que permita al usuario ingresar un número e informe si es múltiplo de 6.  
**Clasificación:** Estructura Selectiva Simple – Uso del operador módulo. PS - Py
26. Diseñar un algoritmo que solicite un número e informe el doble de su valor solo si es impar.  
**Clasificación:** Estructura Selectiva Simple – Validación de imparidad. PS – PY
27. Diseñar un algoritmo que solicite al usuario ingresar un número e informe si es par o impar.  
**Clasificación:** Estructura Selectiva Compuesta – Validación binaria. PS - Py
28. Diseñar un algoritmo que solicite ingresar dos números e informe si el primero es mayor, menor o igual al segundo. PS – Py
29. **Clasificación:** Estructura Selectiva Múltiple – Comparación de magnitud. PS - Py
30. Diseñar un algoritmo que permita ingresar tres números distintos e informe cuál de ellos es el mayor.  
**Clasificación:** Estructura Selectiva Múltiple – Comparación triple. PS - Py
31. Diseñar un algoritmo que calcule el monto final a pagar considerando el total de la compra y la forma de pago. Aplicar descuentos según el monto y el medio de pago. **Clasificación:** Estructura Selectiva Anidada – Múltiples condiciones. PS - Py
32. Diseñar un algoritmo que calcule el sueldo neto de un trabajador, considerando horas normales y extras (más de 160 horas mensuales se pagan al doble). **Clasificación:** Estructura Selectiva – Cálculo condicional. PS - Py
33. Diseñar un algoritmo que calcule el monto a pagar por consumo eléctrico, considerando escalas de precios según cantidad de Kw consumidos. **Clasificación:** Estructura Selectiva Anidada – Tramos tarifarios. PS - Py
34. Diseñar un algoritmo que solicite la categoría de internación (Pediatria, Maternidad u Otro) y la cantidad de días internado, y calcule el costo total. **Clasificación:** Estructura Selectiva Múltiple – Validación por categoría. PS - Py
35. Diseñar un algoritmo que solicite los importes de ventas por refacciones, servicios y vehículos, calcule el total y el promedio. Informar si se alcanzó el objetivo de \$50.000. **Clasificación:** Estructura Selectiva Simple – Validación de promedio. PS - Py
36. Diseñar un algoritmo que solicite el nombre de un artículo, su costo unitario y el número de departamento, y luego calcule un incremento en el precio final según el departamento. **Clasificación:** Estructura Selectiva – Aumento condicional. PS - Py

### Parte 3 – Estructuras Repetitivas

37. Crear un algoritmo en PseInt que pida al usuario que ingrese números enteros positivos. El algoritmo debe seguir pidiendo números hasta que el usuario ingrese un número negativo. Finalmente, debe imprimir la suma de todos los números positivos ingresados.
38. Crear un algoritmo que genera un número aleatorio entre 0 y 20 (puedes usar la función Azar de PseInt). Pídele al usuario que adivine el número. El programa debe continuar pidiéndole al usuario que ingrese un número hasta que adivine el número correcto. En cada intento, indica si el número ingresado es mayor o menor que el número secreto. Cuando adivine informarle al usuario en cuantos intentos lo logro.

39. Escribe un algoritmo que pida al usuario un número entero y muestre su tabla de multiplicar del 1 al 10 usando un bucle FOR.
40. Crea un algoritmo que muestre al usuario un menú de una calculadora con operaciones básicas (como opciones). El menú debe repetirse hasta que el usuario elija salir (por ejemplo, opción 5) bucle DO-WHILE  
1-sumar: "la suma es 23"  
2-restar: "la resta es 34"
41. Diseñar un algoritmo que, dado un rango (inferior y superior), calcule cuántos números impares existen dentro de ese rango, ya sea creciente o decreciente. **Clasificación:** Estructura Repetitiva – Conteo condicional.
42. Diseñar un algoritmo que permita validar un código alfanumérico ingresado por el usuario una cantidad de veces definida. Si el código es correcto en algún intento, mostrar "Logueo Exitoso". Si se agotan los intentos, mostrar "Ha sido bloqueado". **Clasificación:** Estructura Repetitiva con Condición de Corte.

#### Parte 4 – Estructuras de Datos (Arreglos Unidimensionales)

43. Diseñar un algoritmo que almacene en un arreglo 10 números ingresados por el usuario y los muestre en el mismo orden. **Clasificación:** Arreglo unidimensional – Lectura y recorrido secuencial.
44. Diseñar un algoritmo que almacene en un arreglo 25 números aleatorios entre 1 y 100, y luego informe la suma total y el promedio. **Clasificación:** Arreglo unidimensional – Generación, acumulación y promedio.
45. Diseñar un algoritmo que permita ingresar 6 cadenas de texto en un arreglo, copie sus elementos en otro arreglo en orden inverso y los muestre por pantalla. **Clasificación:** Arreglo unidimensional – Inversión de datos.
46. Diseñar un algoritmo que inserte la nota del alumno número 46 en un arreglo de 50 posiciones con notas ya existentes. **Clasificación:** Arreglo unidimensional – Inserción en arreglo no ordenado.
47. Diseñar un algoritmo que elimine las notas menores a 40 de un arreglo. Si existen varias, debe eliminar todas mediante una repetición del proceso. **Clasificación:** Arreglo unidimensional – Eliminación condicional.