

Taller

Juan Pablo Restrepo Uribe

October 28, 2024

1 Semana 1

1. Investigar ejemplos de algoritmos en la vida cotidiana (mínimo 5).
2. Seleccionar un problema cotidiano y escribir una descripción clara y precisa del problema.
3. Tomar el problema seleccionado y descomponerlo en partes más pequeñas. Por ejemplo, dividir la organización de una fiesta en tareas como invitaciones, comida, música, etc.
4. Escribir un algoritmo detallado para cada subproblema identificado en el paso anterior. Utilizar pseudocódigo o diagramas de flujo.
5. Revisar el algoritmo y buscar maneras de hacerlo más eficiente. Por ejemplo, reducir el número de pasos necesarios o encontrar soluciones más rápidas.
6. Escribir una guía que explique el algoritmo, cómo funciona, y cómo usarlo.

2 Semana 2

Si X , Y y Z son variables de tipo booleano con valores $X = \text{True}$, $Y = \text{False}$, $Z = \text{True}$, determina el valor de las siguientes expresiones lógicas:

1. $X \text{ and } Y$
2. $X \text{ and } Z$
3. $X \text{ or } Y$
4. $X \text{ or } Z$
5. $\text{!} X$
6. $(X \text{ and } Y) \text{ or } (X \text{ and } Z)$
7. $X \text{ or } Y \text{ and } Z$

Si W , X , Y y Z son variables de tipo booleano con valores $W = \text{False}$, $X = \text{True}$, $Y = \text{True}$, $Z = \text{False}$, determina el valor de las siguientes expresiones lógicas:

1. $(X \text{ and } Y) \text{ or } (X \text{ and } Z)$
2. $(X \text{ and } \text{!}Y) \text{ or } (\text{!}X \text{ or } Z)$
3. $W \text{ or } Y \text{ and } X \text{ and } W \text{ or } Z$
4. $X \text{ and } \text{!}Y \text{ and } \text{!}X \text{ or } \text{!}W \text{ and } Y$
5. $(W \text{ or } \text{!}Y) \text{ and } X \text{ or } Z$
6. $X \text{ and } Y \text{ and } W \text{ or } Z \text{ or } X$

Si A , B y C son variables de tipo booleano con valores $A = \text{False}$, $B = \text{True}$, $C = \text{False}$, determina el valor de las siguientes expresiones lógicas:

1. A and B
2. A and C
3. A or B
4. A or C
5. !A
6. (A and B) or (A and C)
7. A or B and C

Si P , Q y R son variables de tipo booleano con valores $P = True$, $Q = True$, $R = False$, determina el valor de las siguientes expresiones lógicas:

1. P and Q
2. P and R
3. P or Q
4. P or R
5. !Q
6. (P and Q) or (P and R)
7. P or Q and R

Si M , N y O son variables de tipo booleano con valores $M = False$, $N = False$, $O = True$, determina el valor de las siguientes expresiones lógicas:

1. M and N
2. M and O
3. M or N
4. M or O
5. !O
6. (M and N) or (M and O)
7. M or N and O

3 Semana 3

Realizar por medio de pseudocódigo y diagramas de flujo los siguientes ejercicios. Use de preferencia PSEint

1. Crear un algoritmo que me diga los números del 1 al 10
2. Realizar un algoritmo para realizar las 4 operaciones básicas
3. Algoritmo para hallar el VOLUMEN y el AREA de un cilindro
4. Algoritmo que lea dos números, calculando y escribiendo el valor de su suma, resta, producto y división.
5. Diseñar un algoritmo que pida por teclado tres números; si el primero es negativo, debe imprimir el producto de los tres y si no lo es, imprimirá la suma.
6. Escribir un nombre y saludar
7. Un algoritmo que diga si el año es bisiesto

8. Un algoritmo que determine el día siguiente (consulte como saber si un año es bisiesto, tenga en cuenta que hay meses de 30, 31, 28 o 29 días).
9. Escribe un pseudocódigo que lea 5 números enteros ingresados por el usuario y calcule su suma.
10. Escribe un pseudocódigo que lea 3 números y determine cuál es el mayor.
11. Escribe un pseudocódigo que determine si un número ingresado por el usuario es par o impar.
12. Un algoritmo que calcule la nota final de la asignatura.
13. Escribe un pseudocódigo que lea 10 números y cuente cuántos son positivos, cuántos son negativos y cuántos son ceros.

4 Semana 4

Ejercicios de variables

1. Escribir un programa que muestre por pantalla la cadena ¡Hola Mundo!.
2. Escribir un programa que almacene la cadena ¡Hola Mundo! en una variable y luego muestre por pantalla el contenido de la variable.
3. Escribir un programa que pregunte el nombre del usuario y muestre el mensaje “Hola -nombre-”
4. Una panadería vende barras de pan a 3.49€ cada una. El pan que no es el día tiene un descuento del 60 %. Escribir un programa que comience leyendo el número de barras vendidas que no son del día. Después el programa debe mostrar el precio habitual de una barra de pan, el descuento que se le hace por no ser fresca y el coste final total.
5. Escribe un programa que defina dos variables a y b con valores numéricos y luego intercambie sus valores sin utilizar una tercera variable.
6. Desarrolla un programa que solicite al usuario ingresar su nombre y su edad, luego imprime un mensaje que diga “Hola, [nombre], tienes [edad] años”.
7. Crea un programa que calcule el área de un triángulo. El usuario debe ingresar la base y la altura.
8. Desarrolla un programa que convierta grados Celsius a Fahrenheit. El usuario debe ingresar la temperatura en Celsius y el programa debe imprimir la equivalente en Fahrenheit.
9. Escribe un programa que calcule el volumen de una esfera. El usuario debe ingresar el radio.
10. Crea un programa que calcule el precio final de un producto después de aplicarle un descuento. El usuario debe ingresar el precio original y el porcentaje de descuento.

Ejercicios de condicionales simples

1. Dado un número, decir si es positivo o negativo o 0.
2. Una juguetería tiene mucho éxito en dos de sus productos: payasos y muñecas. Suele hacer venta por correo y la empresa de logística les cobra por peso de cada paquete, así que deben calcular el peso de los payasos y muñecas que saldrán en cada paquete a demanda. Cada payaso pesa 112 g y cada muñeca 75 g. Escribir un programa que lea el número de payasos y muñecas vendidos en el último pedido y calcule el peso total del paquete que será enviado.
3. Escribir un programa que pida al usuario su peso (en kg) y estatura (en metros), calcule el índice de masa corporal y lo almacene en una variable, y muestre por pantalla la frase Tu índice de masa corporal es `jimc`, donde `jimc` es el índice de masa corporal calculado redondeado con dos decimales.
4. Escribir un programa que almacene la cadena de caracteres contraseña en una variable, pregunte al usuario por la contraseña e imprima por pantalla si la contraseña introducida por el usuario coincide con la guardada en la variable ¿Mayúsculas o minúsculas?
5. Escribir un programa que pida al usuario dos números y devuelva su división. Si el usuario no introduce números debe devolver un aviso de error y si el divisor es cero también.

6. Escribir un programa que pida al usuario dos números y la operación básica que desea hacer. El programa debe entregar el resultado de la operación indicada.
7. Escribir un programa que pida al usuario un número entero y muestre por pantalla si es par o impar.
8. Para tributar un determinado impuesto se debe ser mayor de 16 años y tener unos ingresos superiores a 1000 € mensuales. Escribir un programa que pregunte al usuario su edad y sus ingresos mensuales y muestre por pantalla si el usuario tiene que tributar o no.
9. Escriba un programa en Python que pida al usuario ingresar su edad y luego muestre un mensaje indicando si es mayor de edad o no.
10. Desarrolla un programa que pida al usuario ingresar tres números y luego indique cuál de ellos es el mayor.

5 Semana 5

1. Diseña y codifica un programa que lea un número del 0 al 6 y escriba el nombre del día de la semana correspondiente, “lunes”, “martes”, ..., “domingo”. Si el número leído no está en el intervalo previsto, escribirá “error”.
2. Escriba y codifica un algoritmo que lea tres números enteros, diga cuál es el mayor, el mediano y el menor.
3. Codifica un programa que lea un número del 1 al 12 y escriba el nombre del mes correspondiente, “enero”, “febrero”, ..., “diciembre”. Si el número leído no está en el intervalo previsto, escribirá “error”.
4. Codifica un programa que, dado el ordinal de un mes, calcule el número de días para ese mes en un año no bisiesto.
5. Diseña y codifica un programa que, dado el ordinal de un mes y el año, nos muestre el número de días para ese mes, teniendo en cuenta que puede ser bisiesto.
6. Codifica un programa que pida una fecha y calcule la fecha siguiente, mostrándola en pantalla.
7. Codifica un programa que pida una fecha y calcule la fecha anterior, mostrándola en pantalla.
8. Escriba un programa que pida dos números enteros y que calcule su división, escribiendo si la división es exacta o no. Mejore el programa anterior haciendo que tenga en cuenta que no se puede dividir por cero
9. Escriba un programa que pida dos números enteros y que escriba si el mayor es múltiplo del menor.
10. Escriba un programa que pida tres números y que escriba si son los tres iguales, si hay dos iguales o si son los tres distintos.
11. Escriba un programa que pida un año y que escriba si es bisiesto o no. Se recuerda que los años bisiestos son múltiplos de 4, pero los múltiplos de 100 no lo son, aunque los múltiplos de 400 sí.
12. Diseña y codifica un programa que lea un real x y calcule el valor de la función y definida como sigue:

$$y(x) = \begin{cases} -1 & \text{si } -5 < x < -1 \\ x(x+20) & \text{si } -1 \leq x \leq 4 \\ 2x & \text{si } 4 < x \leq 10 \\ 0 & \text{en otros casos} \end{cases} \quad (1)$$

13. Un hotel tiene tres tipos de habitaciones: estándar, de lujo y suite.

- El precio de una habitación estándar es de \$ 100 por noche
- El precio de una habitación de lujo es de \$ 200 por noche
- El precio de una suite es de \$300 por noche.

Además, el hotel ofrece un descuento del 10 % si el huésped reserva por más de 7 noches.

Escribe un programa que solicite al usuario el tipo de habitación y la cantidad de noches que desea reservar, y luego calcule el precio total de la reserva aplicando los descuentos correspondientes.

14. Un restaurante ofrece diferentes menús según el día de la semana y el horario. Los menús son los siguientes:
 - Menú del día (lunes a viernes, de 12 p.m. a 3 p.m.): \$ 10

- Menú especial (viernes y sábado, de 7 p.m. a 10 p.m.): \$ 20
- Menú estándar (resto de los días y horarios): \$ 15
- Además, el restaurante ofrece un descuento del 20 % si el cliente es miembro del club de fidelidad.

Escribe un programa que solicite al usuario el día de la semana y la hora de la visita, y determine el precio del menú aplicando los descuentos correspondientes.

15. Una tienda de electrónica ofrece descuentos en la compra de productos según el tipo y la cantidad de productos comprados. Los descuentos son los siguientes:

- Si se compran 1 o 2 productos de cualquier tipo, no hay descuento.
- Si se compran 3 o más productos del mismo tipo, se aplica un descuento del 10 %.
- Si se compran al menos 5 productos en total, se aplica un descuento adicional del 5 % sobre el total de la compra.

Escribe un programa que solicite al usuario el tipo y la cantidad de productos que desea comprar, y calcule el precio total de la compra aplicando los descuentos correspondientes.

16. Implemente el siguiente pseudocódigo en Python

```

1 Algoritmo CondicionalesAnidados
2
3     Definir hace_sol, llueve, cine_abierto Como Logico
4
5     Escribir " Hacer sol? (true/false)"
6     Leer hace_sol
7
8     Si hace_sol Entonces
9         Escribir "Iremos al parque."
10    Sino
11        Escribir " Esta lloviendo? (true/false)"
12        Leer llueve
13
14        Si llueve Entonces
15            Escribir "Nos quedaremos en casa."
16
17            Escribir " Esta el cine abierto? (true/false)"
18            Leer cine_abierto
19
20            Si cine_abierto Entonces
21                Escribir "Iremos al cine."
22            Sino
23                Escribir "Nos quedaremos en casa."
24            FinSi
25
26        Sino
27            Escribir "Nos quedaremos en casa."
28        FinSi
29
30    FinSi
31
32 FinAlgoritmo

```

17. Implemente el siguiente pseudocódigo en Python

```

1 Algoritmo CondicionalesAnidados
2
3     Algoritmo RestauranteCafeCocina
4
5     Definir restaurante_lleno, restaurante_cerrado, cansancio como Logicos

```

```

6
7   Escribir " El restaurante est  lleno? (true/false)"
8   Leer restaurante_lleno
9
10  Si restaurante_lleno Entonces
11      Escribir "Podemos probar en el nuevo cafe de la esquina."
12  Sino
13      Escribir " El restaurante esta cerrado? (true/false)"
14      Leer restaurante_cerrado
15
16      Si restaurante_cerrado Entonces
17          Escribir "Cocinaremos en casa."
18      Sino
19          Escribir " Estamos demasiado cansados para cocinar? (true/false)"
20          Leer cansancio
21
22          Si cansancio Entonces
23              Escribir "Descansaremos en lugar de cocinar."
24          Sino
25              Escribir "Cocinaremos en casa."
26          FinSi
27      FinSi
28  FinSi
29
30  FinSi
31
32  FinAlgoritmo

```

18. Implemente el siguiente pseudocódigo en Python

```

1  Algoritmo CondicionalesAnidados
2
3      Algoritmo LlegadaCine
4
5          Definir llegada_tiempo, decidir_otro_cine, pelicula_en_cartelera como
            Logicos
6
7          Escribir " Llegamos a tiempo al cine? (true/false)"
8          Leer llegada_tiempo
9
10         Si no llegada_tiempo Entonces
11             Escribir "Perderemos el comienzo de la pelicula."
12         Sino
13             Escribir " Decidimos ir a otro cine? (true/false)"
14             Leer decidir_otro_cine
15
16             Si decidir_otro_cine Entonces
17                 Escribir "\ La pelicula que queremos ver esta en cartelera en
                    este cine\? (true/false)"
18                 Leer pelicula_en_cartelera
19
20                 Si pelicula_en_cartelera Entonces
21                     Escribir "Aun podremos disfrutar la pelicula."
22                 Sino
23                     Escribir "No podremos ver la pelicula que queriamos en este
                        cine."
24                 FinSi
25             FinSi
26         FinSi

```

```

27
28     FinSi
29
30 FinAlgoritmo

```

19. Supongamos que tienes una tienda de comida rápida que ofrece descuentos basados en la edad y el tipo de cliente. Los descuentos son los siguientes:

- Si el cliente es un estudiante menor de 18 años, obtiene un descuento del 20 %.
- Si el cliente es un adulto mayor de 65 años, obtiene un descuento del 25 %.
- Si el cliente es un adulto entre 18 y 65 años y es miembro del club de fidelidad, obtiene un descuento del 15 %.
- Si el cliente es un adulto entre 18 y 65 años y no es miembro del club de fidelidad, no obtiene descuento.

Ahora, escribe un programa que solicite al usuario, su edad y si es miembro del club de fidelidad, y calcule el descuento correspondiente a aplicar en su compra.

20. Supongamos que estás desarrollando un sistema de admisión para una universidad que tiene diferentes criterios de admisión según la carrera a la que los estudiantes deseen postularse. Los criterios de admisión son los siguientes:

- Para la carrera de Ingeniería, se requiere una calificación mínima en Matemáticas y Física.
- Para la carrera de Medicina, se requiere una calificación mínima en Biología y Química.
- Para la carrera de Literatura, se requiere una calificación mínima en Lengua y Literatura Universal.
- Para el resto de las carreras, se considerará la calificación promedio de todas las materias.

Escribe un programa que solicite al usuario las calificaciones en las materias correspondientes y la carrera a la que desea postularse, y determine si el estudiante cumple con los criterios de admisión.

6 Semana 6

1. Crear un ciclo for que cuente de 0 a 100
2. Crea un bucle for que imprima la tabla del 5 (del 5 al 50).
3. Cree un programa que pida un número y calcule todos sus múltiplos hasta 100 Haz una tabla de multiplicar utilizando el ciclo for
4. Imprima los números del 1 a 10 al revés utilizando el ciclo for
5. Imprima los números pares del 2 al 20 utilizando un bucle for.
6. Crear un bucle que cuente todos los números pares hasta el 100
7. Cree un bucle que sume los números del 100 al 200
8. Cree un bucle pida dos números, y me muestre la suma acumulada de todos los numero contenidos en ese rango
9. Dado un número, cuente el número total de dígitos de un número: Por ejemplo, el número es 75869, por lo que la salida debería ser 5.
10. Escriba un programa que pida dos números enteros y me muestre todos los números que son pares entre los dos enteros solicitados
11. Escriba un programa que pida dos números enteros y me muestre todos los números que son impares entre los dos enteros solicitados
12. Escriba un programa que pida un número entero mayor que cero y me muestre todos sus divisores.
13. Escriba un programa que:
 - Pregunte cuántos números se van a introducir
 - Pida esos números
 - Muestre un mensaje cada vez que un número no sea mayor que el primero.

14. Escriba un programa que:

- Pregunte cuántos números se van a introducir
- Pida esos números
- Muestre un mensaje cada vez que un número no sea mayor que el anterior.

15. Escriba un programa que pregunte cuántos números se van a introducir, pida esos números y escriba cuántos negativos ha introducido y muestre la suma acumulada de todos los negativos.

16. Cree un programa que muestre los primeros 10 elementos de la serie de Fibonacci.

$$F(n) = \begin{cases} 0 & \text{si } n = 0 \\ 1 & \text{si } n = 1 \\ F(n-1) + F(n-2) & \text{en otros casos} \end{cases}$$

17. Cree un programa que pida un número y muestre todos los números de la serie de Fibonacci que sean menores que él.

18. Escribe un programa que calcule el factorial de un número ingresado por el usuario utilizando un bucle for.

$$n! = n \times (n-1) \times (n-2) \times \dots \times 3 \times 2 \times 1$$

19. Escribe un bucle for que imprima los caracteres de una cadena de texto uno por uno.

20. Escribe un programa que encuentre todos los números primos menores que 100 utilizando un bucle for.

21. Crea un bucle for anidado para imprimir un patrón de asteriscos en forma de pirámide.

22. Implementa un juego de adivinanza donde el programa elige un número aleatorio y el usuario debe adivinarlo

```
1 import random
2 numeroAleatorio = random.randint(1, 100)
3 print(numeroAleatorio)
```

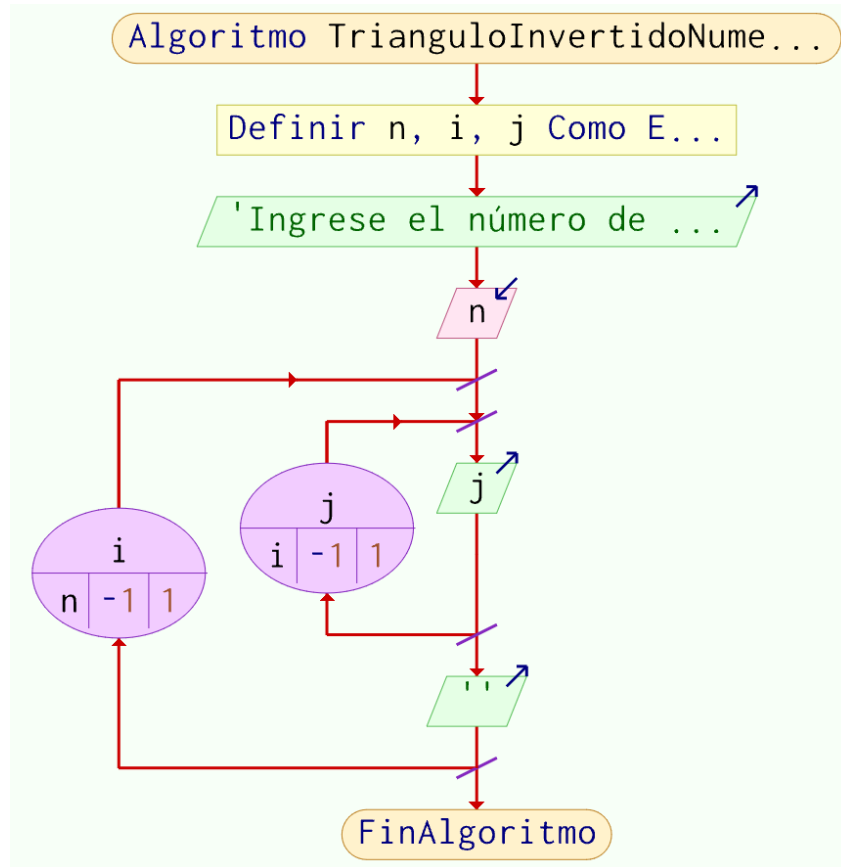
23. Cree un programa que pida una contraseña y la compare con una previamente almacenada, debe tener límite de intentos.

24. Escribe un programa que calcule e imprima la secuencia de Fibonacci hasta el término n utilizando un bucle for.

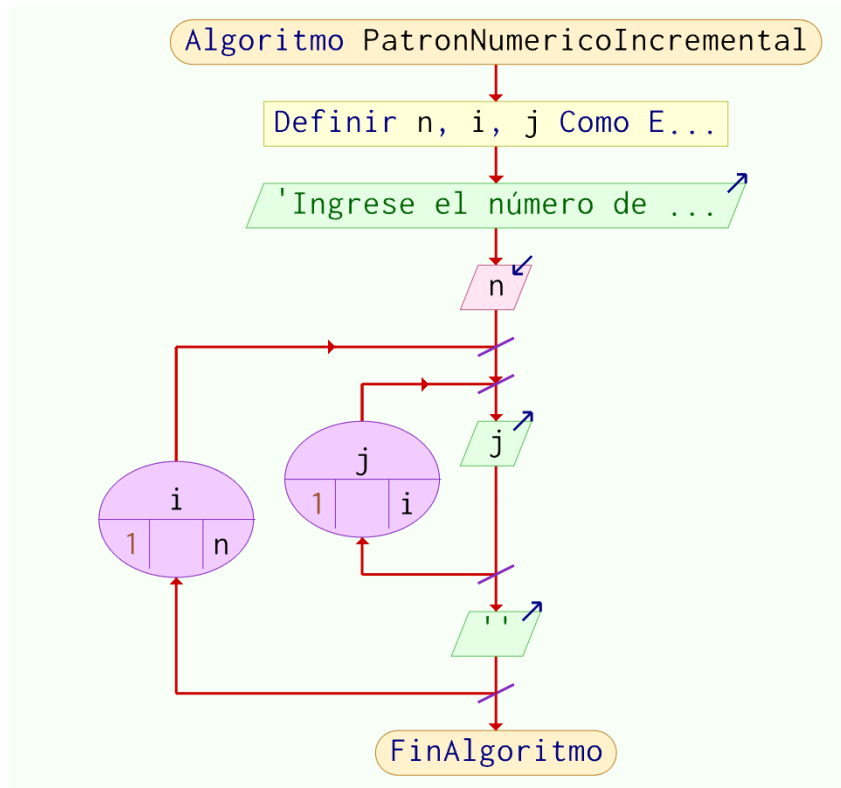
25. Crea un programa que simule una carrera entre varios corredores. Utiliza un bucle for para avanzar a los corredores en cada iteración.

26. Escribe un programa que genere un tablero de ajedrez de tamaño n x n, donde n es un número par dado por el usuario, utilizando caracteres como X y O

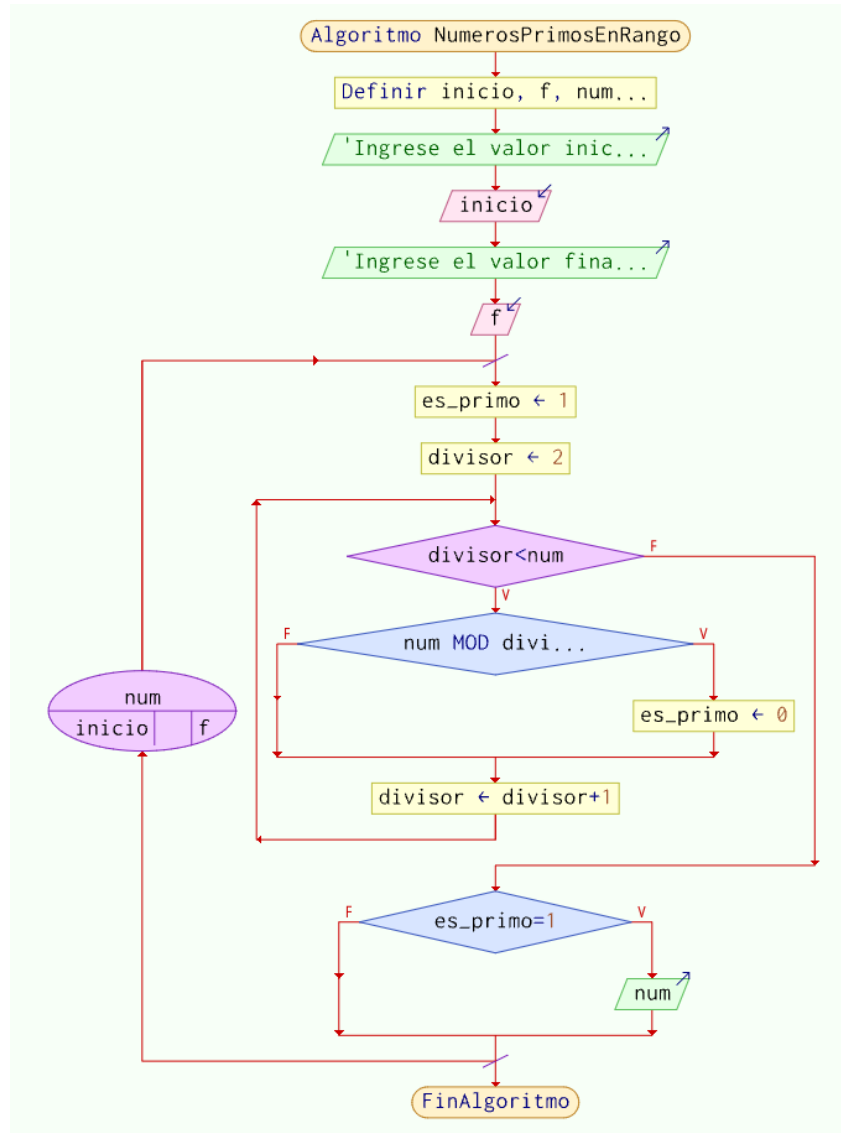
27. Implemente el código en Python del siguiente diagrama de flujo



28. Implemente el código en Python del siguiente diagrama de flujo



29. Implemente el código en Python del siguiente diagrama de flujo



7 Semana 7

1. Haga un ciclo while que imprima los números del 6 al 14.
2. Haga un ciclo while que imprima los números pares del 2 al 20.
3. Haga un ciclo while que imprima el mensaje “Bienvenido a Python” 8 veces.
4. Crea un programa que cuente desde 1 hasta un número ingresado por el usuario utilizando un ciclo while.
5. Solicita al usuario un número y muestra su tabla de multiplicar del 1 al 10 utilizando un ciclo while.
6. Pide al usuario un número y luego cuenta desde ese número hasta 1 utilizando un ciclo while.
7. Crea un programa que permita a dos usuarios jugar. Uno de los usuarios introduce un número que será desconocido para el segundo jugador, el segundo jugador deberá adivinar el número. Agregue pistas (frío, tibio, caliente)
8. Escribe un programa que le pida al usuario ingresar una serie de números enteros positivos. Utiliza una variable bandera para determinar si el número ingresado es par o impar y lleva un contador de números pares y otro de números impares. El programa debe permitir al usuario ingresar números hasta que decida detenerse.
9. Crea un programa que le pida al usuario ingresar una serie de números enteros positivos. Utiliza una variable centinela para permitir al usuario continuar ingresando números hasta que ingrese un número negativo. Luego, muestra la suma de todos los números positivos ingresados.

10. Crea una calculadora simple que permita al usuario realizar operaciones matemáticas básicas (suma, resta, multiplicación, división) con dos números ingresados por el usuario. Utiliza una variable centinela para permitir al usuario realizar múltiples cálculos hasta que decida salir.
11. Crea un juego en el que el programa elija una palabra al azar y muestre guiones bajos (-) que representen las letras de la palabra. El usuario debe adivinar las letras una por una y tiene un número limitado de intentos.
12. Escribe un programa que simule un sistema de autenticación de usuarios. El programa debe permitir al usuario ingresar un nombre de usuario y una contraseña. Si los datos son incorrectos, deberá pedir nuevamente la información hasta que el usuario acierte o se hayan realizado 3 intentos fallidos.
13. Escribe un programa que simule un cajero automático. El usuario tiene un saldo inicial y puede retirar dinero. El cajero debe validar si el monto a retirar es válido y si hay suficiente saldo disponible. Si no lo hay, debe mostrar un mensaje adecuado. El programa debe seguir pidiendo montos hasta que el usuario decida no retirar más.
14. Desarrolla una calculadora que acepte dos números y una operación (suma, resta, multiplicación, división). El programa debe repetirse hasta que el usuario ingrese "salir" y debe manejar los siguientes casos:
 - Si se ingresa una división por cero, debe mostrar un error y permitir ingresar otra operación.
 - Si se ingresa una operación no válida, debe mostrar un error y volver a solicitar una operación.
15. Crea un programa que permita al usuario ingresar calificaciones y los respectivos créditos de una cantidad indefinida de materias. El programa debe calcular el promedio ponderado y terminar cuando el usuario ingrese "0" en los créditos.
16. Simula un sistema de control de inventario donde el usuario puede añadir, retirar o consultar el inventario de un producto. El programa debe pedir la acción a realizar y la cantidad, actualizando el inventario en cada paso. Si el usuario intenta retirar más de lo que hay en inventario, debe mostrar un error (Use listas o diccionarios -consulte-).
17. Crea un programa que permita a un maestro ingresar las respuestas correctas de un examen (sí/no o verdadero/falso). Luego, debe ingresar las respuestas de los estudiantes uno por uno, calculando la cantidad de respuestas correctas y mostrando si el estudiante pasó (mínimo 60 %) o falló el examen. El proceso debe repetirse hasta que se evalúen todos los estudiantes.
18. Desarrolla un juego de "Piedra, Papel o Tijera" donde el usuario juegue contra la computadora. El programa debe repetirse hasta que el usuario decida salir. Lleva un registro de las victorias del usuario, las de la computadora y los empates.
19. Crea un programa que convierta entre diferentes unidades de medida (por ejemplo, metros a kilómetros, libras a kilogramos, etc.). El usuario debe seleccionar la conversión que desea realizar, ingresar el valor y el programa debe mostrar el resultado. El programa se repetirá hasta que el usuario decida salir.
20. Crea un programa que permita al usuario ingresar la temperatura diaria de una semana. El programa debe calcular la temperatura promedio de la semana y determinar cuántos días estuvieron por encima del promedio.

8 Semana 8

1. Escribe un programa que solicite al usuario ingresar una cadena y luego cuente cuántas vocales (a, e, i, o, u) hay en la cadena. Luego, muestra el resultado.
2. Escribe un programa que verifique si una palabra ingresada por el usuario es un palíndromo (se lee igual de izquierda a derecha y de derecha a izquierda - palabra[::-1]).
3. Escribe un programa que reemplace todas las letras "a" en una cadena ingresada por el usuario por el carácter "x" y muestre la cadena resultante.
4. Escribe un programa que cuente cuántas letras mayúsculas y cuántas letras minúsculas hay en una cadena ingresada por el usuario (ayuda 1, ayuda 2)
5. Escribe un programa que tome una frase ingresada por el usuario y genere un acrónimo utilizando la primera letra de cada palabra en mayúsculas.
6. Escribe un programa que cuente cuántas palabras hay en una oración ingresada por el usuario. Puedes asumir que las palabras están separadas por espacios.

9 Semana 9

1. Escribe un programa que calcule la suma de todos los elementos en una lista.
2. Crea un programa que multiplique todos los elementos de una lista y devuelva el resultado.
3. Encuentra el número más grande y el más pequeño en una lista.
4. Escribe un programa que elimine los elementos duplicados de una lista.
5. Crea un programa que cuente cuántas veces aparece un elemento específico en una lista (sin métodos).
6. Escribe un programa que tome una lista de números y devuelva una nueva lista solo con los números pares.
7. Dada una lista de palabras, encuentra la palabra más larga.
8. Escribe un programa que elimine todos los elementos iguales a un valor específico de una lista.
9. Combina dos listas en una tercera lista sin duplicados. Si tienes una lista de listas, elimina las listas duplicadas basándote en su contenido.
10. Toma una lista de palabras y ordénalas de acuerdo a su longitud, de la más corta a la más larga.

10 Semana 10

1. Crea un programa que pida dos números enteros al usuario y diga si alguno de ellos es múltiplo del otro. Crea una función EsMultiplo que reciba los dos números, y devuelve si el primero es múltiplo del segundo.
2. Crear una función que calcule la temperatura media de un día a partir de la temperatura máxima y mínima. Crear un programa principal, que utilizando la función anterior, vaya pidiendo la temperatura máxima y mínima de cada día y vaya mostrando la media. El programa pedirá el número de días que se van a introducir.
3. Crea una función “ConvertirEspaciado”, que reciba como parámetro un texto y devuelve una cadena con un espacio adicional tras cada letra. Por ejemplo, “Hola, tú” devolverá “H o l a , t ú “. Crea un programa principal donde se use dicha función.
4. Escribe una función llamada encontrar_palindromos que tome una lista de palabras como argumento y devuelva una lista con las palabras que son palíndromos. Un palíndromo es una palabra que se lee igual hacia adelante y hacia atrás (por ejemplo, “reconocer” o “anilina”).
5. El día juliano correspondiente a una fecha es un número entero que indica los días que han transcurrido desde el 1 de enero del año indicado. Queremos crear un programa principal que al introducir una fecha nos diga el día juliano que corresponde. Para ello podemos hacer las siguientes subrutinas:
 - LeerFecha: Nos permite leer por teclado una fecha (día, mes y año).
 - DiasDelMes: Recibe un mes y un año y nos dice los días de ese mes en ese año.
 - EsBisiesto: Recibe un año y nos dice si es bisiesto.
 - Calcular_Dia_Juliano: recibe una fecha y nos devuelve el día juliano.

Se puede apoyar en el siguiente pseudocódigo

11 Semana 11

1. Escribe un programa que reciba una cadena de texto como entrada y devuelva un diccionario que contenga cada palabra única de la cadena como clave y el número de veces que aparece como valor.
2. Escribe un programa que solicite al usuario ingresar el nombre y la calificación de varios estudiantes y luego genere un diccionario donde las claves sean los nombres de los estudiantes y los valores sean sus calificaciones.
3. Crea un programa que permita al usuario ingresar palabras en inglés y sus traducciones en otro idioma. Al final, el programa debería imprimir un diccionario con las palabras en inglés como claves y sus traducciones como valores.
4. Escribe un programa que reciba una lista de palabras como entrada y genere un diccionario que cuente la cantidad de veces que aparece cada letra en todas las palabras combinadas.

5. Escribe un programa que reciba una cadena de texto como entrada y devuelva un diccionario que cuente la frecuencia de cada carácter en el texto, ignorando espacios en blanco y distinguiendo entre mayúsculas y minúsculas.
6. Escribe una función que tome dos diccionarios como entrada y devuelva un nuevo diccionario que contenga todas las claves y valores de ambos diccionarios. Si una clave está presente en ambos diccionarios, el valor en el nuevo diccionario debería ser una lista que contenga ambos valores.
7. Escribe una función que tome dos palabras como entrada y determine si son anagramas (es decir, si las letras de una palabra pueden reorganizarse para formar la otra). Utiliza un diccionario para contar la frecuencia de cada letra en ambas palabras y compara los diccionarios resultantes.
8. Crea un programa que permita gestionar un inventario de productos. Debe permitir al usuario agregar nuevos productos, eliminar productos existentes, actualizar la cantidad de productos y mostrar el inventario completo.

12 Semana 12

1. Escribe una función que tome un array NumPy de números enteros y calcule la suma de todos los elementos pares en el array.
2. Escribe una función que cuente cuántos elementos en un array NumPy son mayores que un valor dado.
3. Escribe una función que tome un array NumPy y cambie todos los valores negativos por cero.
4. Escribe una función que calcule el promedio de los valores en un array NumPy. Si el valor es negativo, no debe ser incluido en el cálculo del promedio.

13 Semana 13

1. Escribe una función en Python que tome dos matrices (listas bidimensionales) como entrada y devuelva la suma de esas matrices como resultado. Asegúrate de verificar que las matrices tengan las mismas dimensiones antes de sumarlas.
2. Escribe una función en Python que tome dos matrices como entrada y devuelva su producto como resultado. Asegúrate de que las matrices sean compatibles para la multiplicación (el número de columnas de la primera matriz debe ser igual al número de filas de la segunda matriz).
3. Escribe una función en Python que tome una matriz como entrada y devuelva su matriz transpuesta como resultado. La matriz transpuesta se obtiene intercambiando filas por columnas.

14 Final

14.1 Gestión de Inventario

Tome como ejemplo el diccionario que se encuentra a continuación, y considere la realización de funciones que permitan la gestión del inventario.

Implementar funciones para:

- Agregar productos al inventario
- Actualizar la cantidad de productos (una por nuevos productos en el inventario (nuevo código de producto), otra por venta de productos)
- Eliminar productos del inventario
- Hacer una consulta específica de un producto.

```

base_de_datos_productos = {
    ``001``: { ``nombre``: ``Producto 1``, ``precio``: 10.99, ``cantidad``: 50 },
    ``002``: { ``nombre``: ``Producto 2``, ``precio``: 5.99, ``cantidad``: 100 },
    ``003``: { ``nombre``: ``Producto 3``, ``precio``: 20.49, ``cantidad``: 25 },
    ``004``: { ``nombre``: ``Producto 4``, ``precio``: 15.99, ``cantidad``: 75 },
    ``005``: { ``nombre``: ``Producto 5``, ``precio``: 7.99, ``cantidad``: 60 },
    ``006``: { ``nombre``: ``Producto 6``, ``precio``: 12.99, ``cantidad``: 40 },
    ``007``: { ``nombre``: ``Producto 7``, ``precio``: 6.49, ``cantidad``: 90 },
    ``008``: { ``nombre``: ``Producto 8``, ``precio``: 25.99, ``cantidad``: 30 },
    ``009``: { ``nombre``: ``Producto 9``, ``precio``: 8.99, ``cantidad``: 70 },
    ``010``: { ``nombre``: ``Producto 10``, ``precio``: 14.99, ``cantidad``: 55 },
    ``011``: { ``nombre``: ``Producto 11``, ``precio``: 9.99, ``cantidad``: 45 },
    ``012``: { ``nombre``: ``Producto 12``, ``precio``: 18.99, ``cantidad``: 35 },
    ``013``: { ``nombre``: ``Producto 13``, ``precio``: 11.99, ``cantidad``: 80 },
    ``014``: { ``nombre``: ``Producto 14``, ``precio``: 13.99, ``cantidad``: 70 },
    ``015``: { ``nombre``: ``Producto 15``, ``precio``: 16.99, ``cantidad``: 60 }
}

```

14.2 Generación de Facturas

Deberá generar facturas (puede ser por la terminal, pero puede consultar como generar archivos pdf o Excel, o el tipo de archivo que considere más adecuado). Cada factura debe contener

- Información del cliente (si considera relevante, se puede apoyar en el diccionario que se presenta en la última sección)
- Lista de productos comprados
- Número de productos comprados por producto, el precio individual y el precio total
- Además, incluya un programa de fidelización (puntos, descuentos, etc)

Nota: Recuerde que al hacer una compra se debe consultar el precio del producto y si a compra se realiza se debe modificar la cantidad de productos de la base de datos de productos.

Implementar funciones para:

- Crear descuentos
- Calcular el IVA
- Eliminar productos de la factura

14.3 Base de datos cliente

Deberá tener una base de datos de clientes, como la que se ve a continuación, con la información que usted considere importante, recuerde que deberá tener una campaña de fidelización, y esta debe estar relacionada en la base de datos de clientes.

Implementar funciones para:

- Crear cliente nuevo
- Actualizar información de cliente ya existente
- Eliminar cliente
- Consultar información de cliente

Notas generarles

- Usted deberá agregar las funciones que considere necesarias (obligatorio crear mínimo 9 funciones más, 3 por cada apartado de proyecto).
- Si bien, los ejemplos dados con bases de datos en estructuras tipo diccionario, usted tiene libertad con el tipo de dato que podrá usar.

```

base_de_datos_clientes = {
    ``cliente1``: {
        ``nombre``: ``Juan Perez``,
        ``email``: ``juan.perez@email.com``,
        ``telefono``: ``123-456-7890``,
        ``direccion``: ``Calle 123, Ciudad, Pais``
    },
    ``cliente2``: {
        ``nombre``: ``Maria Rodriguez``,
        ``email``: ``maria.rodriguez@email.com``,
        ``telefono``: ``987-654-3210``,
        ``direccion``: ``Avenida ABC, Ciudad, Pais``
    },
    ``cliente3``: {
        ``nombre``: ``Luis Gonzalez``,
        ``email``: ``luis.gonzalez@email.com``,
        ``telefono``: ``555-555-5555``,
        ``direccion``: ``Plaza XYZ, Ciudad, Pais``
    },
    ``cliente4``: {
        ``nombre``: ``Ana Sanchez``,
        ``email``: ``ana.sanchez@email.com``,
        ``telefono``: ``777-777-7777``,
        ``direccion``: ``Calle ZZZ, Ciudad, Pais``
    },
    ``cliente5``: {
        ``nombre``: ``Pedro Morales``,
        ``email``: ``pedro.morales@email.com``,
        ``telefono``: ``111-222-3333``,
        ``direccion``: ``Paseo ABC, Ciudad, Pais``
    }
}

```