RECUPERACIÓN LÓGICA DE PROGRAMACIÓN PHP

Aprendiz: Laura Isabela Ausecha Ortiz

Instructor: Carlos Sterling

Id: 2900810-ADSO

Centro de Gestión y Desarrollo Sostenible Surcolombiano

12 de julio de 2024

**INTRODUCCIÓN**

El desarrollo de programas informáticos para resolver diversos problemas matemáticos y lógicos es fundamental en la informática y la programación. En este contexto, se han planteado una serie de ejercicios que abarcan desde cálculos matemáticos simples hasta operaciones más complejas y análisis de datos.

OBJETIVOS

El objetivo de los programas desarrollados es aplicar conceptos de programación y algoritmos para resolver una variedad de problemas matemáticos, lógicos y de procesamiento de datos. Estos programas buscan:

* Aplicación de Conceptos Matemáticos
* Análisis de Datos y Toma de Decisiones
* Desarrollo de Habilidades de Programación

Al cumplir con estos objetivos, se busca mejorar las habilidades de programación, el pensamiento lógico y la capacidad para abordar diversos problemas.

1. Suponiendo que pi = 3.1416. Escribe un programa que pida al usuario que introduzca el radio, y presente por pantalla el cálculo del perímetro de la circunferencia (2\*pi\*r), el área del círculo (pi\*r2), y el volumen de la esfera (V = 4\*pi\*r3 /3).

Paso a paso:

1. Inicio:

Este es el punto de partida del programa. Aquí es donde se inicia la ejecución del mismo.

2. Entrada del Radio:

En este paso, el programa solicita al usuario que introduzca el valor del radio. El usuario proporciona este valor a través del teclado.

3. Calcular Perímetro:

Después de recibir el valor del radio, el programa utiliza la fórmula 2 \* pi \* radio para calcular el perímetro de la circunferencia. Aquí, "pi" se refiere al valor 3.1416.

4. Calcular Área:

Una vez que se ha calculado el perímetro de la circunferencia, el programa utiliza la fórmula pi \* radio al cuadrado para calcular el área del círculo.

5. Calcular Volumen:

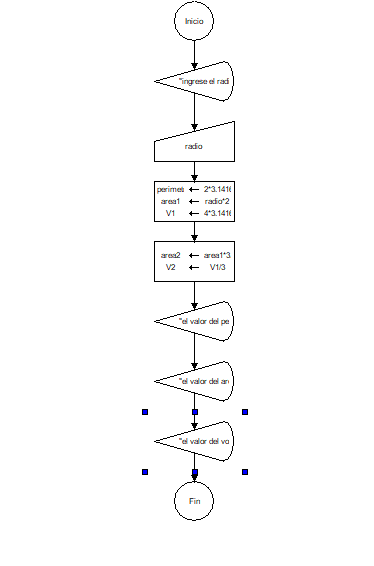
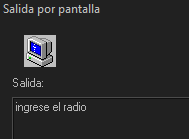
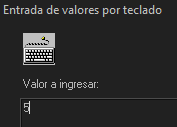
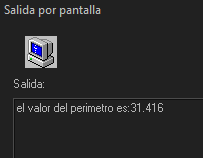
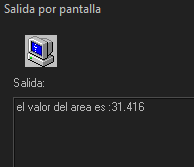
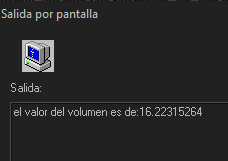
Después de calcular el área del círculo, el programa utiliza la fórmula (4/3) \* pi \* radio al cubo para calcular el volumen de la esfera.

6. Salida:

En este punto, el programa muestra los valores calculados del perímetro de la circunferencia, el área del círculo y el volumen de la esfera en la pantalla para que el usuario los vea.

7. Fin:

Este es el final del programa. Después de mostrar los valores calculados, el programa llega a su término.

.

1. Realizar un programa que lea por teclado dos números, si el primero es mayor al segundo informar su suma y diferencia, en caso contrario informar el producto y la división del primero respecto al segundo.

Paso a paso:

1. Inicio: Comienza el programa.

2. Salida 1: Muestra el mensaje "Ingrese el primer número" para solicitar al usuario que ingrese el primer número.

3. Entrada 1: El usuario ingresa el primer número.

4. Salida 2: Muestra el mensaje "Ingrese el segundo número" para solicitar al usuario que ingrese el segundo número.

5. Entrada 2: El usuario ingresa el segundo número.

6. Decisión: Se compara si el primer número es mayor o igual al segundo número.

- Verdadero: Si el primer número es mayor o igual al segundo número, se ejecutan las siguientes acciones:

Asignación Suma: Calcula la suma de los dos números.

Salida 3: Muestra el resultado de la suma con el mensaje "El resultado de la suma es, Suma".

Asignación Resta: Calcula la resta del primer número menos el segundo número.

Salida 4: Muestra el resultado de la resta con el mensaje "El resultado de la resta es, Resta".

- Falso: Si el primer número es menor que el segundo número, se ejecutan las siguientes acciones:

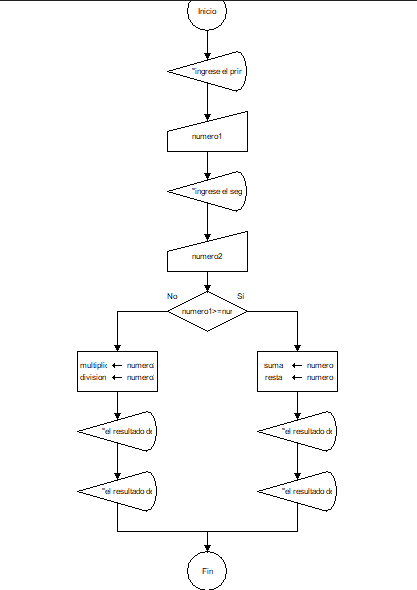
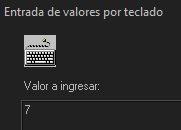
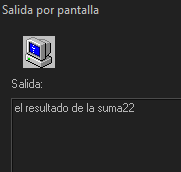
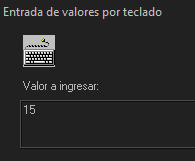
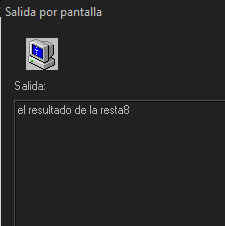
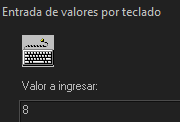
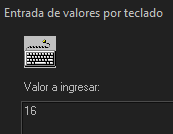
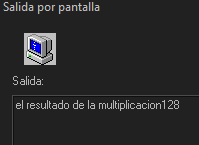
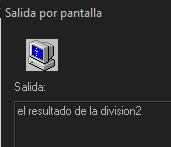
Asignación Multiplicación: Calcula la multiplicación de los dos números.

- Salida 5: Muestra el resultado de la multiplicación con el mensaje "El resultado de la multiplicación es, Multiplicación".

Asignación División: Calcula la división del segundo número entre el primer número.

- Salida 6: Muestra el resultado de la división con el mensaje "El resultado de la división es, División".

7. Fin: Finaliza el programa.



1. Confeccionar un programa que permita cargar un número entero positivo de hasta tres cifras y muestre un mensaje indicando si tiene 1, 2, o 3 cifras.

Paso a paso:

1. Inicio:

Este es el punto de partida del programa.

2. Ingresar un número:

* En este paso, el usuario ingresa un número entero positivo.

3. Número mayor a 99:

* Verifica si el número ingresado es mayor que 99.
* Verifica si el número ingresado es menor que 1000.

5. Número digitado contiene tres cifras:

* Si el número cumple las condiciones anteriores y tiene tres cifras, se sigue por este camino.

6. Número menor o igual a 99:

* Si el número no tiene tres cifras, se verifica si es menor o igual a 99.
* Si el número cumple la condición de ser menor o igual a 99, se verifica si es mayor o igual a 10.

8. Número digitado contiene dos cifras:

* Si el número tiene dos cifras, se sigue por este camino.

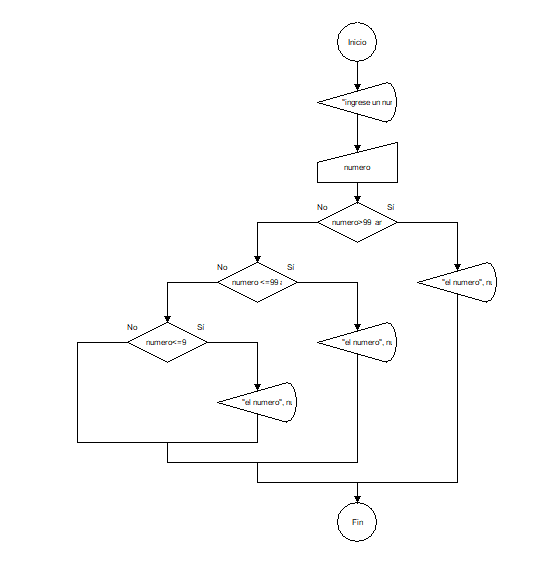
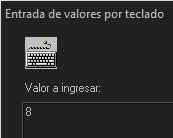
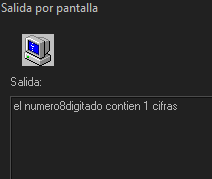
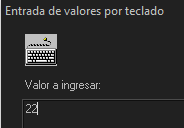
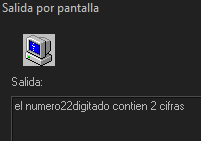
9. Número menor o igual a 9:

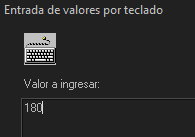
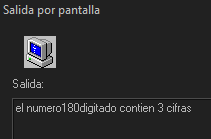
* Si el número no tiene dos cifras, se verifica si es menor o igual a 9.

10. Número digitado contiene una cifra:

* Si el número tiene una cifra, se sigue por este camino.

11. fin del programa



1. De un postulante a un empleo, que realizó un test de capacitación, se obtuvo la siguiente información: nombre del postulante, cantidad total de preguntas que se le realizaron y cantidad de preguntas que contestó correctamente. Se pide confeccionar un programa que lea los datos del postulante e informe el nivel del mismo según el porcentaje de respuestas correctas que ha obtenido, y sabiendo que:

* Nivel superior: Porcentaje>=90%.
* Nivel medio: Porcentaje>=75% y <90%.
* Nivel regular: Porcentaje>=50% y <75%.
* Fuera de nivel: Porcentaje<50%.

Paso a paso:

1. Inicio del Programa: El programa comienza.

2. Asignación de Variables: Se asigna el valor de 10 a la variable Núm\_pre, que representa el número total de preguntas en el test de capacitación.

3. Salida de Información: Se muestra la nota importante de que el número de preguntas en el test de capacitación es igual a 10.

4. Entrada de Datos: Se ingresa el número de preguntas que el postulante contestó correctamente.

5. Decisión 1: Se verifica si el número de preguntas correctas está entre 9 y 10.

Si es Verdadero: Se muestra la salida indicando que el postulante está en el nivel superior con un porcentaje mayor o igual al 90%.

Si es Falso: Se pasa a la siguiente decisión.

6. Decisión 2: Se verifica si el número de preguntas correctas está entre 7 y 8.

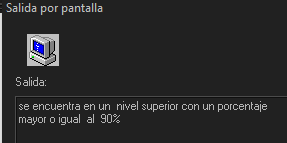
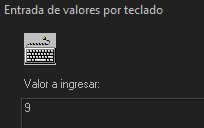
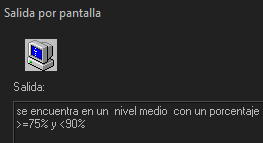
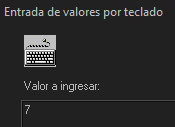
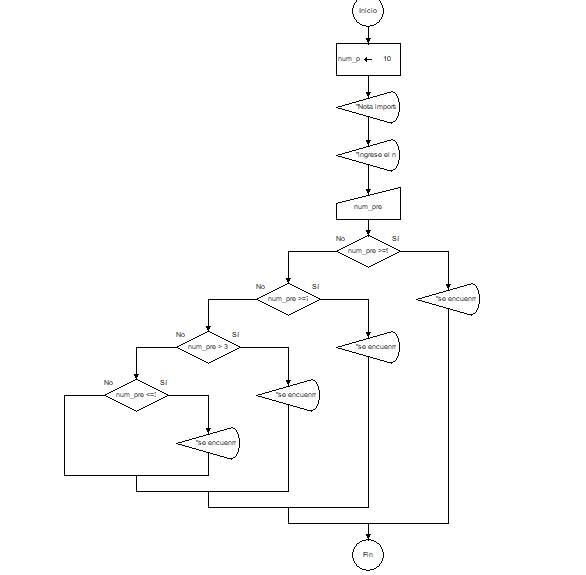
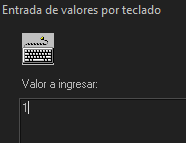
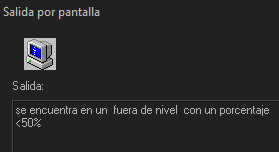
* Si es Verdadero: Se muestra la salida indicando que el postulante está en el nivel medio con un porcentaje mayor o igual al 75% y menor al 90%.
* Si es Falso: Se pasa a la siguiente decisión.

7. Decisión 3: Se verifica si el número de preguntas correctas está entre 4 y 6.

* Si es Verdadero: Se muestra la salida indicando que el postulante está en el nivel regular con un porcentaje mayor o igual al 50% y menor que el 75%.
* Si es Falso: Se pasa a la siguiente decisión.

8. Decisión 4: Se verifica si el número de preguntas correctas es igual a 3 o menos.

* Si es Verdadero: Se muestra la salida indicando que el postulante está fuera de nivel con un porcentaje menor que el 50%.
* Si es Falso: No hay más decisiones, el programa finaliza.



1. Escribir un programa que pida ingresar la coordenada de un punto en el plano, es decir dos valores enteros x e y.

Posteriormente imprimir en pantalla en qué cuadrante se ubica dicho punto. (1º Cuadrante si x > 0 Y y > 0, 2º Cuadrante: x < 0 Y y > 0, etc.).

Paso a paso:

1. Inicio del Programa: El programa comienza.

2. Entrada de Datos: Se solicita al usuario ingresar las coordenadas x e y del punto en el plano.

3. Decisión 1: Se verifica si tanto la coordenada x como la coordenada y son mayores que 0.

Si es Verdadero: El punto se encuentra en el primer cuadrante.

Se muestra la salida indicando que el punto está en el primer cuadrante.

Si es Falso: Se pasa a la siguiente decisión.

4. Decisión 2: Se verifica si la coordenada x es menor que 0 y la coordenada y es mayor que 0.

Si es Verdadero: El punto se encuentra en el segundo cuadrante.

Se muestra la salida indicando que el punto está en el segundo cuadrante.

Si es Falso: Se pasa a la siguiente decisión.

5. Decisión 3: Se verifica si tanto la coordenada x como la coordenada y son menores que 0.

Si es Verdadero: El punto se encuentra en el tercer cuadrante.

Se muestra la salida indicando que el punto está en el tercer cuadrante.

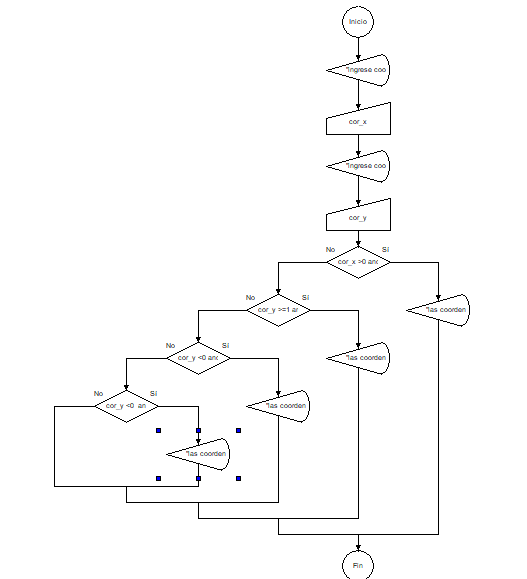
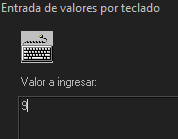
Si es Falso: Se pasa a la siguiente decisión.

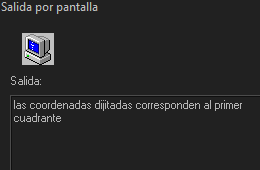
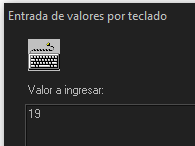
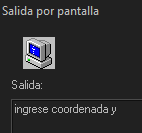
6. Decisión 4: Se verifica si la coordenada x es mayor que 0 y la coordenada y es menor que 0.

Si es Verdadero: El punto se encuentra en el cuarto cuadrante.

* Se muestra la salida indicando que el punto está en el cuarto cuadrante.

Si es Falso: No hay más decisiones, el programa finaliza.





1. De un operario se conoce su sueldo y los años de antigüedad. Se pide confeccionar un programa que lea los datos de entrada e informe

a) Si el sueldo es inferior a 500 y su antigüedad es igual o superior a 10 años, otorgarle un aumento del 20 %, mostrar el sueldo a pagar.

b) Si el sueldo es inferior a 500 pero su antigüedad es menor a 10 años, otorgarle un aumento de 5 %.

c) Si el sueldo es mayor o igual a 500 mostrar el sueldo en la página sin cambios.

Paso a paso:

1. Inicio del Programa: El programa comienza.

2. Entrada de Datos: Se solicita al usuario ingresar el sueldo y la antigüedad del operario.

3. Decisión A: Se verifica si el sueldo es menor que 500 y la antigüedad es igual o mayor a 10 años.

Si es Verdadero: Se calcula el aumento del 20% sobre el sueldo.

* Se muestra en la salida el sueldo con el aumento del 20%.

Si es Falso: Se pasa a la siguiente decisión.

4. Decisión B: Se verifica si el sueldo es menor que 500 y la antigüedad es menor a 10 años.

Si es Verdadero: Se calcula el aumento del 5% sobre el sueldo.

* Se muestra en la salida el sueldo con el aumento del 5%.

Si es Falso: Se pasa a la siguiente decisión.

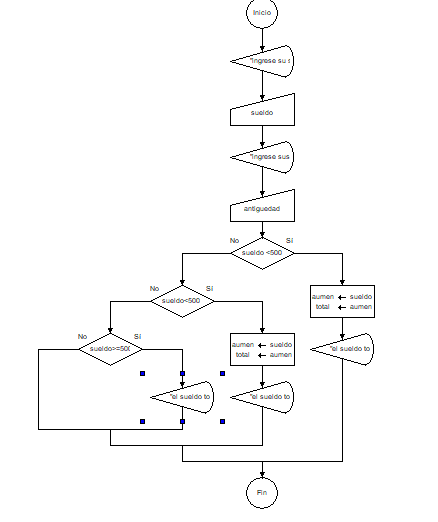
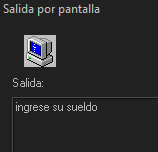
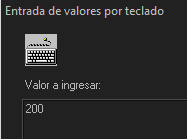
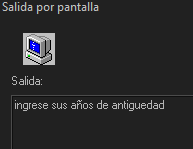
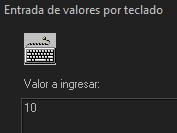
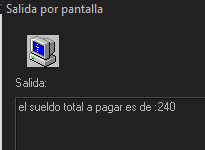
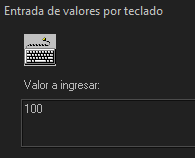
5. Decisión C: Se verifica si el sueldo es mayor o igual a 500.

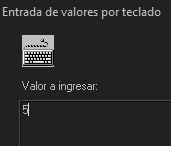
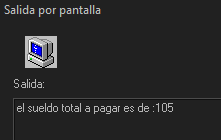
Si es Verdadero: No se aplica ningún cambio al sueldo.

* Se muestra en la salida el sueldo sin cambios.

Si es Falso: No hay más decisiones, el programa finaliza.

6. fin del diagrama





1. Ejercicio de conversión, que permita ingresar una cantidad en metros y la convierta a una de las siguientes Unidades.

1) Centímetros.

2) Kilómetros.

3) Pulgadas.

Paso a paso:

1. Inicio: El proceso comienza.

2. Una salida: Se muestra el mensaje "Ingresar una cantidad de metros para hacer la conversión" para solicitar la cantidad en metros.

3. Una entrada: El usuario ingresa la cantidad en metros, que se almacena como "Cantidadm".

4. Asignación: Se calcula "CF = Cantidadm \* Centímetros", donde Centímetros es 100/1.0.

- Verdadero, se calcula CF.

- Falso, no se realiza ninguna operación adicional.

5. Asignación: Se calcula "KF = Cantidadm \* Kilómetros", donde Kilómetros es 1.0/1000.

- Verdadero, se calcula KF.

- Falso, no se realiza ninguna operación adicional.

6. Asignación: Se calcula "PF = CantidadM \* Pulgadas", donde Pulgadas es 39.37/1.0.

- Verdadero: se calcula PF.

- Falso: no se realiza ninguna operación adicional.

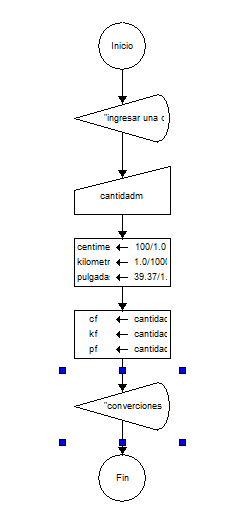
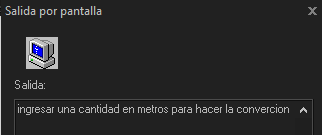
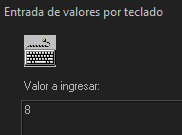
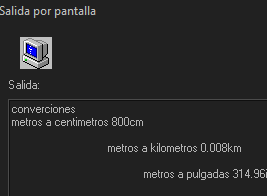
7. Una salida: Se muestra el resultado de las conversiones:

- "Metros a centímetros", CF, "cm"

- "Metros a kilómetros", KF, "km"

- "Metros a pulgadas", PF, "in"

8. Fin: El proceso termina.

1. Un hombre desea saber cuánto dinero se genera por concepto de intereses sobre la cantidad que tiene en inversión en el banco. El decidirá reinvertir los intereses siempre y cuando estos excedan a $7000, y en ese caso desea saber cuánto dinero tendrá finalmente en su cuenta

Paso a paso:

1. Iniciar: Comienza el programa.

2. Salida: Cantidad a Invertir: Muestra en pantalla "Cantidad a Invertir".

3. Entrada: Cantidad: El usuario ingresa la cantidad de dinero a invertir.

4. Salida: Tasa de Interés: Muestra en pantalla "Tasa de Interés".

5. Entrada: Tasa: El usuario ingresa la tasa de interés.

6. Asignación: Interés = Cantidad \* Tasa/100: Calcula el interés multiplicando la cantidad invertida por la tasa de interés dividida por 100.

7. Decisión: Intereses < $7,000: Verifica si los intereses calculados son menores que $7,000.

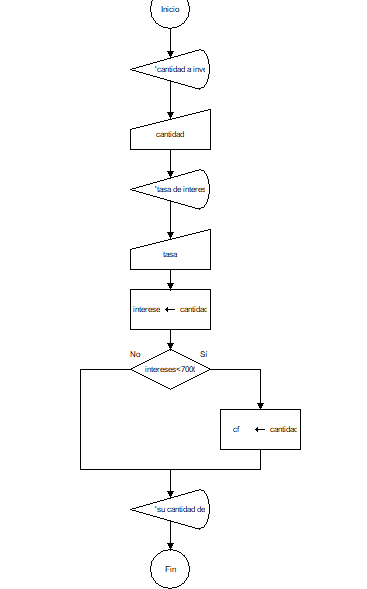
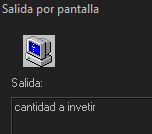
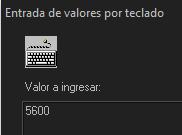
Verdadero: Si los intereses son menores que $7,000:

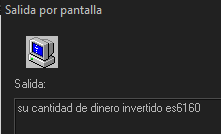
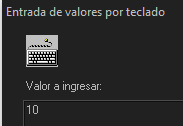
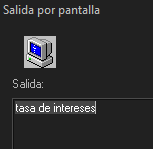
CF = Cantidad + Intereses: Calcula el total sumando la cantidad invertida y los intereses generados.

Falso: Si los intereses son iguales o mayores a $7,000, no se hace nada.

8. Salida: Su cantidad de dinero invertido es, CF: Muestra en pantalla "Su cantidad de dinero invertido es" seguido del resultado del cálculo final (CF, que es la cantidad invertida más los intereses si son menores a $7,000).

9.Fin del diagrama.





1. Una empresa quiere hacer una compra de varias piezas de la misma clase a una fábrica de refacciones. La empresa, dependiendo del monto total de la compra, decidirá qué hacer para pagar al fabricante. Si el monto total de la compra excede de $500 000 la empresa tendrá la capacidad de invertir de su propio dinero un 55% del monto de la compra, pedir prestado al banco un 30% y el resto lo pagará solicitando un crédito al fabricante. Si el monto total de la compra no excede de $500 000 la empresa tendrá capacidad de invertir de su propio dinero un 70% y el restante 30% lo pagará solicitando crédito al fabricante. El fabricante cobra por concepto de intereses un 20% sobre la cantidad que se le pague a crédito.

Paso a paso:

1. Inicio: El proceso comienza.

2. Una salida: Se muestra el mensaje "Ingresa el monto total de la compra" para solicitar el monto.

3. Una entrada: El usuario ingresa el monto, que se almacena como "Monto".

4. Una decisión: Se evalúa si el "Monto" es mayor que 500.000.

- Verdadero: Se realiza la operación "Op1: Monto - 0.55", luego se realiza la operación "Op2: Op1 + 0.30" y se muestra el resultado como "El dinero con el que la empresa cuenta para invertir es de Op2".

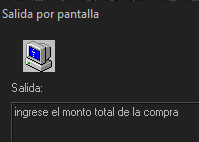
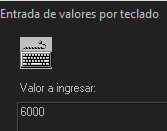
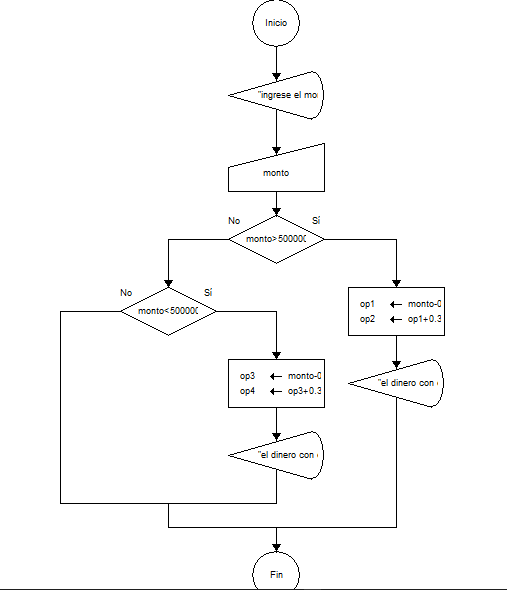
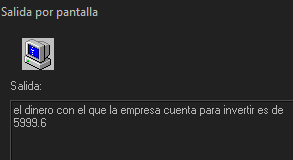
- Falso: se pasa a la siguiente decisión.

5. Una decisión: Se evalúa si el "Monto" es menor que 500.000.

- Verdadero: Se realiza la operación "Op3: Monto - 0.70", luego se realiza la operación "Op4: Op3 + 0.30" y se muestra el resultado como "El dinero con el que la empresa cuenta para invertir es de Op4".

- Falso: el flujo termina sin realizar ninguna operación adicional.

6. Fin: El proceso termina.

1. Calcular el número de pulsaciones que debe tener una persona por cada 10 segundos de ejercicio aeróbico; la fórmula que se aplica es:

cuando el sexo es femenino: núm. pulsaciones = (220 - edad) /10

y si el sexo es masculino: núm. pulsaciones = (210 - edad) /10

Paso a paso:

1. Inicio: Se solicita al usuario que ingrese su sexo (1 si es femenino, 2 si es masculino) y su edad.

2. Entradas: Se reciben los valores de sexo y edad ingresados por el usuario.

3. Decisión: Se verifica si el sexo ingresado es igual a 1 (femenino).

Verdadero: Si el sexo es 1 (femenino), se realiza la siguiente acción:

Asignación: Se calcula el número de pulsaciones como (220 - Edad) / 10.

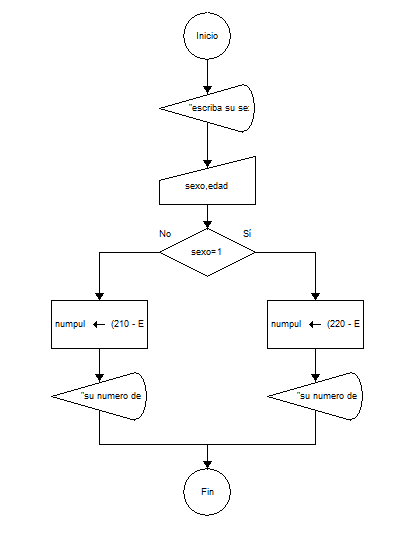
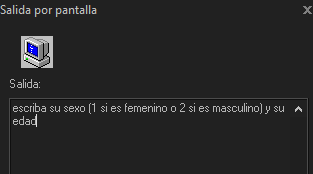
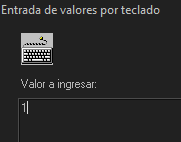
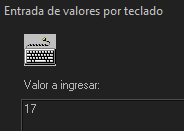
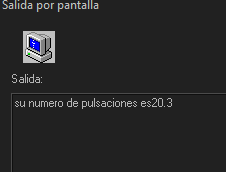
Salida: Se muestra el mensaje "Su número de pulsaciones es, Numpul donde Numpul es el resultado del cálculo anterior.

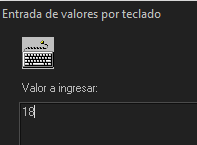
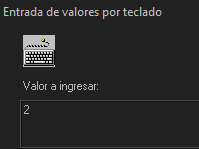
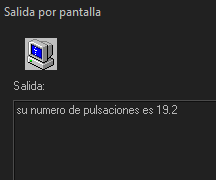
Falso: Si el sexo no es 1 (es masculino), se realiza la siguiente acción:

Asignación: Se calcula el número de pulsaciones como (210 - Edad) / 10.

Salida: Se muestra el mensaje "Su número de pulsaciones es, Numpul donde Numpul es el resultado del cálculo anterior.

4. Fin: Finaliza el diagrama.





11.Una persona desea iniciar un negocio, para lo cual piensa verificar cuánto dinero le prestara el banco por hipotecar su casa. Tiene una cuenta bancaria, pero no quiere disponer de ella a menos que el monto por hipotecar su casa sea muy pequeño. Si el monto de la hipoteca es menor que $1 000 000 entonces invertirá el 50% de la inversión total y un socio invertirá el otro 50%. Si el monto de la hipoteca es de $1 000 000 o más, entonces invertirá el monto total de la hipoteca y el resto del dinero que se necesite para cubrir la inversión total se repartirá a partes iguales entre el socio y el.

Paso a paso:

1. Inicio: Comienza el proceso.

2. Salida: Ingrese Total: Muestra en pantalla "Ingrese Total".

3. Entrada: Total: El usuario ingresa el monto total disponible.

4. Salida: Ingrese Hipoteca: Muestra en pantalla "Ingrese Hipoteca".

5. Entrada: Hipoteca: El usuario ingresa el monto de la hipoteca.

6. Decisión: Hipoteca < $1,000,000: Verifica si el monto de la hipoteca es menor que $1,000,000.

- Verdadero: Si la hipoteca es menor que $1,000,000:

Asignación: Inversión = Total \* 0.50: Calcula la inversión del 50% de cada socio.

Salida: Inversión del 50% cada socio es de, Inversión: Muestra en pantalla la inversión del 50% para cada socio.

- Falso: Si la hipoteca es igual o mayor a $1,000,000:

Decisión: Hipoteca < Total: Verifica si la hipoteca es menor que el monto total disponible.

- Verdadero: Si la hipoteca es menor que el total disponible:

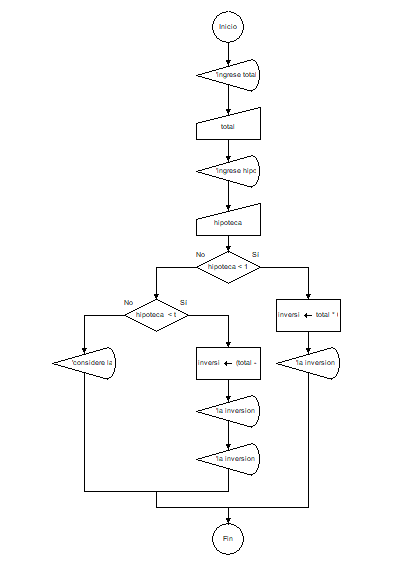
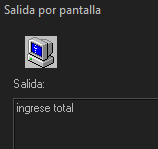
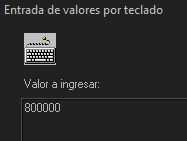
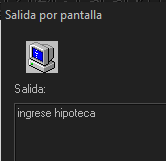
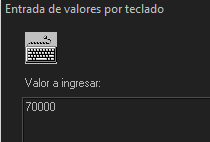
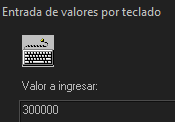
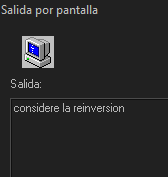
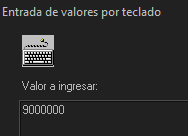
Asignación: Inversión = (Total - Hipoteca) / 2: Calcula la inversión de cada socio.

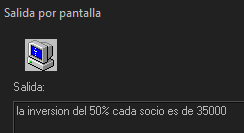
Salida: La inversión de cada socio es de, Inversión: Muestra en pantalla la inversión de cada socio.

- Falso: Si la hipoteca es igual o mayor al total disponible, se considera la reinversión.

Salida: Considera la reinversión: Muestra en pantalla un mensaje para considerar la reinversión.

7. Fin: Termina el proceso.





1. El dueño de una empresa desea planificar las decisiones financieras que tomara en el siguiente año. La manera de planificarlas depende de lo siguiente: Si actualmente su capital se encuentra con saldo negativo, pedirá un préstamo bancario para que su nuevo saldo sea de $10 000. Si su capital tiene actualmente un saldo positivo pedirá un préstamo bancario para tener un nuevo saldo de $20 000, pero si su capital tiene actualmente un saldo superior a los $20 000 no pedirá ningún préstamo. Posteriormente repartirá su presupuesto de la siguiente manera.

$5 000 para equipo de computo

$2 000 para mobiliario

y el resto la mitad será para la compra de insumos y la otra para otorgar incentivos al personal. Desplegar que cantidades se destinaran para la compra de insumos e incentivos al personal y, en caso de que fuera necesario, a cuánto ascendería la cantidad que se pediría al banco.

Paso a paso:

1. Inicio: Comienza el proceso.

2. Salida: Ingrese su capital actual, muestra en pantalla "Ingrese su capital actual".

3. Entrada: Capital, el usuario ingresa el capital actual de la empresa.

4. Decisión: Capital < 0: Verifica si el capital es negativo.

- Verdadero: Si el capital es negativo:

Asignación: Préstamo = $10,000: Se asigna un préstamo de $10,000.

Asignación: Saldo Nuevo = Capital + Préstamo: Calcula el nuevo saldo sumando el capital actual y el préstamo.

Salida: Su nuevo saldo es de, Saldo Nuevo: Muestra en pantalla el nuevo saldo.

- Falso: Si el capital es positivo o cero:

Decisión: Capital > 0 and Capital < $20,000: Verifica si el capital es mayor que cero y menor que $20,000.

- Verdadero: Si el capital está en ese rango:

Asignación: Préstamo = $20,000: Se asigna un préstamo de $20,000.

Asignación: Capital = Préstamo: El capital se iguala al préstamo.

Asignación: Saldo Nuevo = Capital: El nuevo saldo es igual al capital.

Salida: Su nuevo saldo es de, Saldo Nuevo: Muestra en pantalla el nuevo saldo.

- Falso: Si el capital es mayor que $20,000:

Asignación: Préstamo = $0: No se solicita ningún préstamo.

Asignación: Saldo Nuevo = Capital - $7,000: Calcula el nuevo saldo restando $7,000 del capital.

Decisión: Saldo Nuevo > 0: Verifica si el nuevo saldo es positivo.

- Verdadero: Si el nuevo saldo es positivo:

Asignación: Insumos = Saldo Nuevo / 2: Calcula la cantidad destinada para insumos.

Salida: La cantidad que se destinará para la compra de insumos, Insumos: Muestra en pantalla la cantidad.

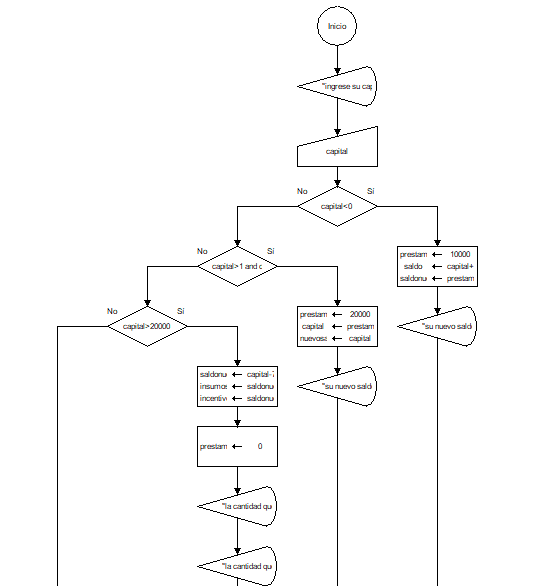
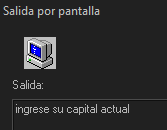
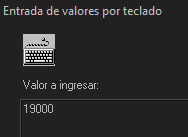
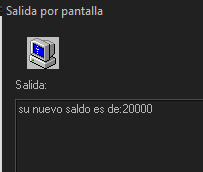
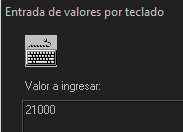
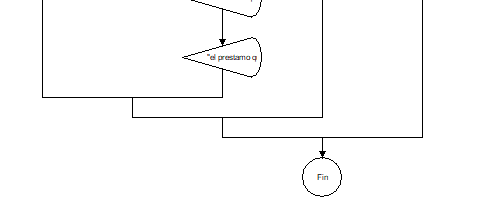
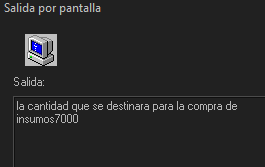
- Falso: Si el nuevo saldo es cero o negativo:

Salida: Incentivos = Saldo Nuevo / 2: Calcula la cantidad destinada para incentivos.

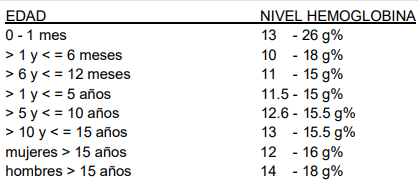
Salida: La cantidad que se destinará para la compra de incentivos, Incentivos: Muestra en pantalla la cantidad.

6. Salida: El préstamo se pidió en el banco es: Muestra en pantalla la cantidad del préstamo solicitado.

7. Fin: Termina el proceso.



1. Tomando como base los resultados obtenidos en un laboratorio de análisis clínicos, un médico determina si una persona tiene anemia o no, lo cual depende de su nivel de hemoglobina en la sangre, de su edad y de su sexo. Si el nivel de hemoglobina que tiene una persona es menor que el rango que le corresponde, se determina su resultado como positivo y en caso contrario como negativo. La tabla en la que el médico se basa para obtener el resultado es la siguiente:



Paso a paso:

1. Salida por pantalla: Solicitar al usuario ingresar el nivel de hemoglobina.

2. Entrada de valores: Permitir al usuario digitar el nivel de hemoglobina.

3.Salida por pantalla: Solicitar al usuario ingresar la edad en meses o años.

4.Entrada de valores: Permitir al usuario ingresar la edad en meses o años.

5. Salida por pantalla: Solicitar al usuario ingresar su sexo.

6. Entrada de valores: Permitir al usuario ingresar su sexo.

7. Decisión: Evaluar si la edad es mayor que 0, menor o igual a 1, y si el valor es "mes".

- Si la decisión es verdadera, continuar con la siguiente decisión.

- Si la decisión es falsa, finalizar con un mensaje de error.

8. Decisión: Evaluar si el nivel de hemoglobina es menor que 13.

- Si la decisión es verdadera, mostrar el mensaje "Tu resultado es positivo, tienes anemia."

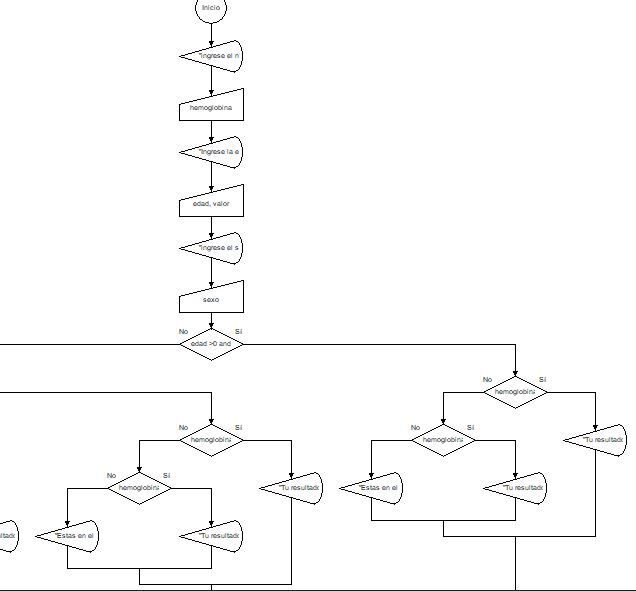
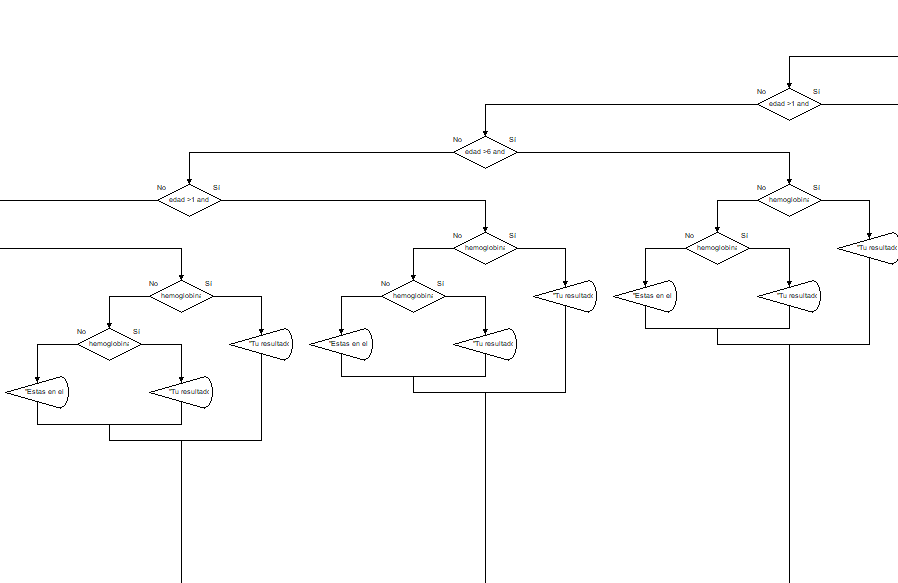
- Si la decisión es falsa, continuar con la siguiente decisión.

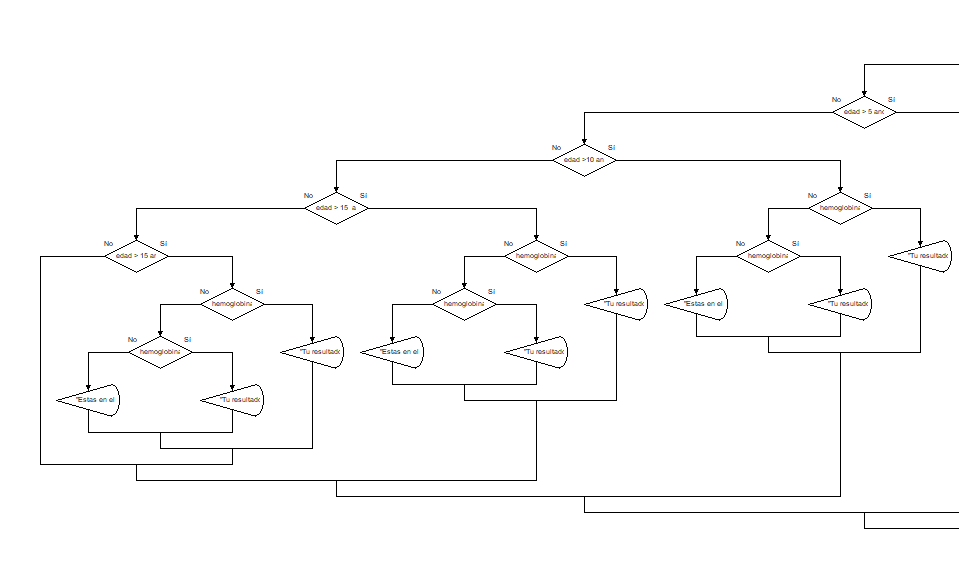
9. Decisión: Evaluar si el nivel de hemoglobina es mayor que 26.

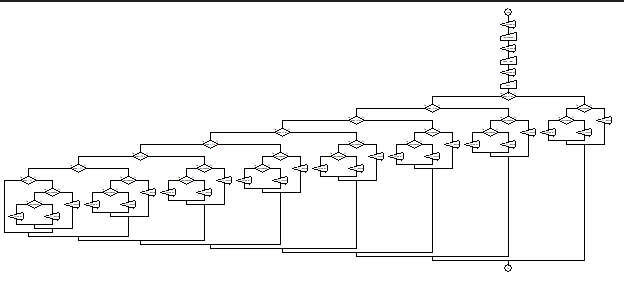
- Si la decisión es verdadera, mostrar el mensaje "Tu resultado es negativo, por lo tanto, no tienes anemia."

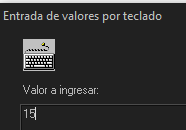
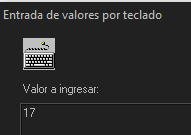
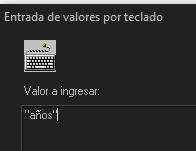
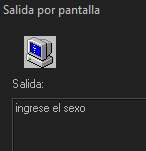
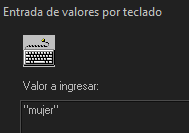
- Si la decisión es falsa, mostrar el mensaje "Estás en el rango establecido, te encuentras estable."

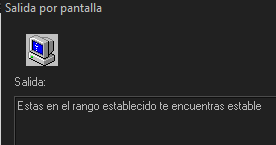
10. El resultado final depende de las condiciones evaluadas y de los valores ingresados por el usuario.









1. Escribe un programa que imprima los 50 primeros números de la sucesión de Fibonacci empezando en 0.

• La serie Fibonacci se compone por una sucesión de números en la que el siguiente siempre es la suma de los dos anteriores.

0, 1, 1, 2, 3, 5, 8, 13...

Paso a paso:

1. Inicio: Comienza el proceso.

2. Asignación: número = 0, N = 0, X = 1, Y = 1: Inicializa las variables necesarias para el cálculo de la secuencia de Fibonacci.

3. Asignación: igual a 1: Asigna el valor 1 a una variable temporal para mostrar el mensaje "Digite un número de elementos".

4. Salida: "Digite un número de elementos": Muestra en pantalla el mensaje solicitando al usuario que ingrese el número de elementos de la secuencia de Fibonacci.

5. Entrada: Número: El usuario ingresa el número de elementos deseado para la secuencia.

6. Ciclo Para: i = 1, número, 1: Inicia un ciclo que se repetirá tantas veces como el número ingresado por el usuario.

Asignación: N = X + Y: Calcula el siguiente número de Fibonacci sumando los dos números anteriores.

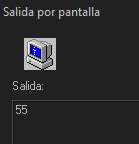
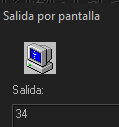
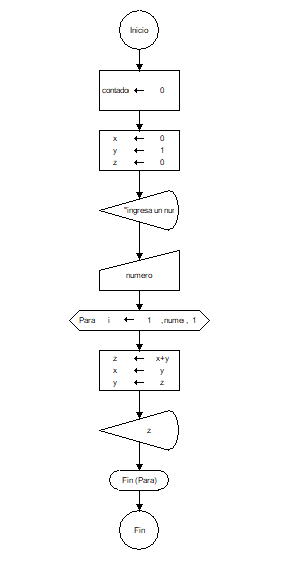
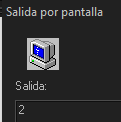
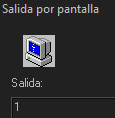
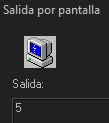
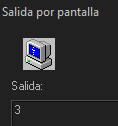
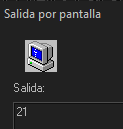
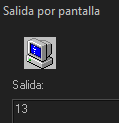
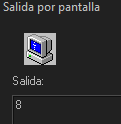
Asignación: X = Y: Actualiza el valor de X con el valor de Y para la siguiente iteración.

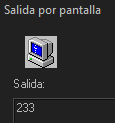
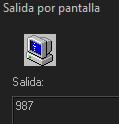
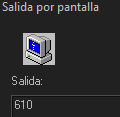
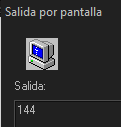
Asignación: Y = N: Actualiza el valor de Y con el nuevo número de Fibonacci para la siguiente iteración.

Salida: N: Muestra en pantalla el número Fibonacci calculado en esta iteración.

7. Fin (Para): Finaliza el ciclo para después de haberse ejecutado el número de veces especificado por el usuario.

8. Fin del proceso.





Así sucesivamente…

1. Crea un programa que invierta el orden de una cadena de texto sin usar funciones propias del lenguaje que lo hagan de forma automática.

Si le pasamos "Hola mundo" nos retornaría "odnum aloH".

Para mis estos ejercicios no se pueden realizar ya que involucran funciones específicas de Python, como el slicing o el manejo de arrays, no es compatible con la representación y el flujo de datos en un diagrama de flujo (DFD). Ya que los DFD no están diseñados para representar la lógica detallada de un lenguaje de programación.

1. Crea un programa que cuente cuantas veces se repite cada palabra y que muestre el recuento final de todas ellas.

• Los signos de puntuación no forman parte de la palabra.

• Una palabra es la misma, aunque aparezca en mayúsculas y minúsculas.

• No se pueden utilizar funciones propias del lenguaje que lo resuelvan automáticamente.

Este ejercicio no se puede realizar en un diagrama de flujo de datos (DFD) porque requiere realizar varias acciones que no son compatibles con un DFD. Por ejemplo, la eliminación de signos de puntuación y el uso de la función predeterminada en Python (lower) para convertir una cadena a minúsculas no se pueden representar fácilmente en un DFD. Además, el uso de diccionarios, que es fundamental en este ejercicio para almacenar el recuento de cada palabra, tampoco es viable en un DFD.

1. Crea un programa que sea capaz de transformar texto natural a código morse y viceversa.

• Debe detectar automáticamente de qué tipo se trata y realizar la conversión.

• En morse se soporta raya "—", punto ".", un espacio " " entre letras o símbolos y dos espacios entre palabras " ".

• El alfabeto morse soportado será el mostrado en

https://es.wikipedia.org/wiki/Código\_morse.

No pude completar este ejercicio en un diagrama de flujo de datos (DFD) porque requería un diccionario con el abecedario en código Morse, y al intentar igualar el abecedario al código Morse en el DFD, recibí un mensaje de error que decía "se espera un constante lógico". Los valores del alfabeto Morse impedían que el flujo del código funcionara correctamente en el DFD.

1. Crea un programa que dibuje un cuadrado o un triángulo con asteriscos "\*".

• Indicaremos el tamaño del lado y si la figura a dibujar es una u otra.

• EXTRA: ¿Eres capaz de dibujar más figuras?

Paso a paso:

el ejercicio no se puede realizar porque al establecer el ciclo, se generan varios asteriscos en cada salida a la pantalla, lo cual depende del valor inicial ingresado. Esto significa que los asteriscos no se agruparán en una sola salida para formar la figura cuadrada. Por lo tanto, llegamos a la conclusión de que es imposible representar este proceso de forma efectiva en un diagrama de flujo de datos (DFD).

1. Crea un programa se encargue de transformar un número binario a decimal sin utilizar funciones propias del lenguaje que lo hagan directamente.

Paso a paso:

1. Inicio: Comienza el proceso.

2. Entrada de valores: N - Se introduce el número binario N.

3. Asignación: Cadena = X - Se inicializa una cadena vacía llamada "Cadena".

4. Ciclo Mientras: N > 0 - Se inicia un ciclo que continuará mientras N sea mayor que 0.

Decisión: N MOD 2 = 0 - Se verifica si el número binario es divisible por 2.

- Verdadero: Cadena = "0" + Cadena - Si es divisible, se agrega "0" al principio de la cadena.

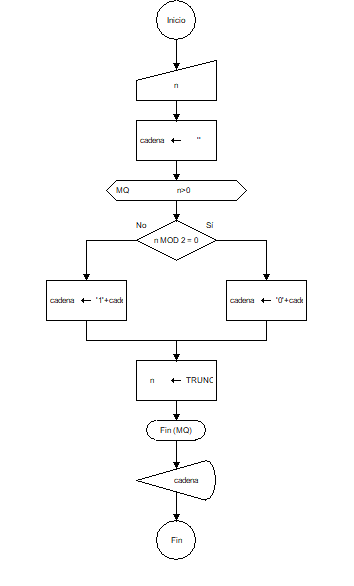
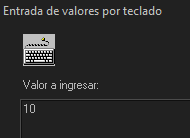
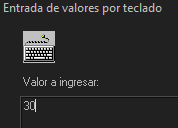
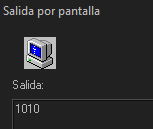
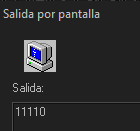
- Falso: Cadena = "1" + Cadena - Si no es divisible, se agrega "1" al principio de la cadena.

Asignación: N = TRUNC(N / 2) - Se actualiza el valor de N dividiéndolo por 2 y tomando la parte entera.

5. Fin (Mientras) - Termina el ciclo cuando N sea igual a 0.

6. Salida: Cadena - Se muestra la cadena resultante que representa el número binario convertido a decimal.

7. Fin del proceso.



1. Escribe un programa que se encargue de comprobar si un número es o no primo. Hecho esto, imprime los números primos entre 1 y 100.

Paso a paso:

1. Inicio: Comienza el proceso.

2. Salida: "Ingrese un número" - Se solicita al usuario que ingrese un número.

3. Entrada: n - Se lee el número ingresado por el usuario y se asigna a la variable n.

4. Asignación: conta = 0 - Se inicializa el contador conta en 0.

5. Ciclo Para: i = 1, n, 1 - Se inicia un ciclo que va desde 1 hasta el número n.

Decisión: N MOD i = 0 - Se verifica si n es divisible por el valor actual de i.

- Verdadero: Asignación: conta = conta + 1 - Si n es divisible por i, se incrementa el contador conta en 1.

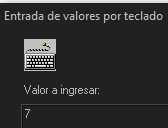
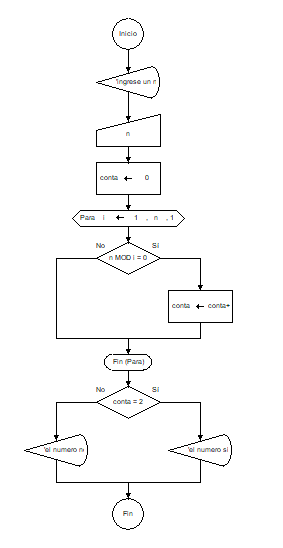
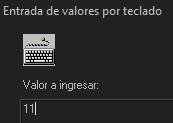
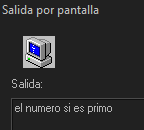
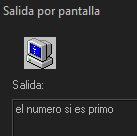
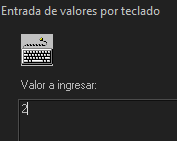
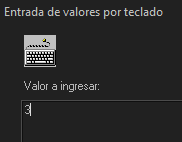
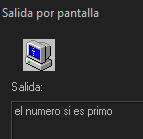
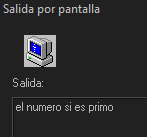
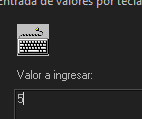
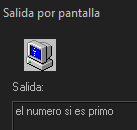
- Fin (Para) - Se termina el ciclo.

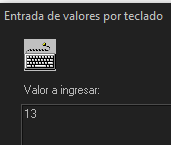
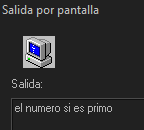
6. Decisión: conta = 2 - Se verifica si conta es igual a 2.

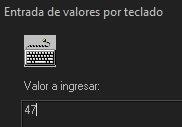
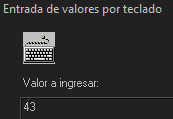
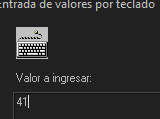
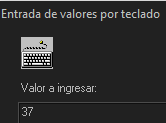
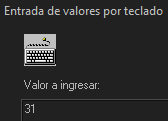
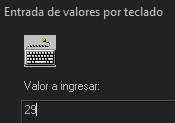
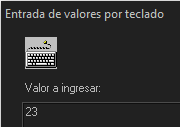
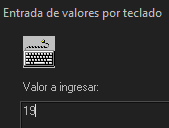
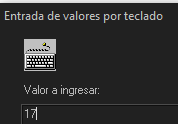
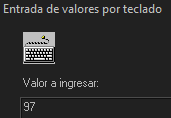
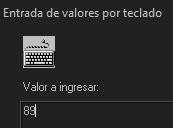
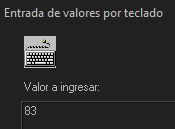
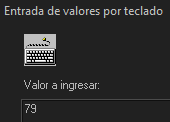
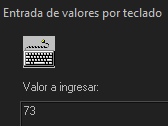
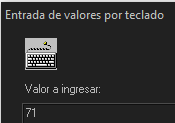
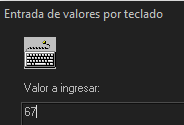
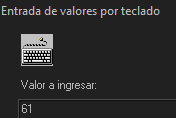
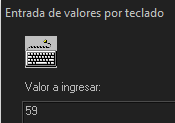
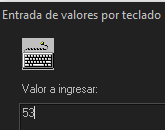
- Verdadero: Salida: "El número es primo" - Si conta es igual a 2, se muestra en pantalla que el número es primo.

- Falso: Salida: "El número no es primo" - Si conta no es igual a 2, se muestra en pantalla que el número no es primo.

7. Fin: Fin del proceso.





c

CONCLUSIONES

El desarrollo de estos programas ha permitido abordar una variedad de problemas, desde cálculos matemáticos y análisis de datos. Estos ejercicios refuerzan la comprensión de algoritmos, estructuras de control y manipulación de datos, aspectos fundamentales en el aprendizaje de la programación y la resolución de problemas.