

# PARCIAL LOGICA Y PROGRAMACION

- Sara Ximena López Fonseca

## Preguntas de Comprensión Temática

1. ¿Qué es una variable en programación?  
Son espacios o lugares donde podemos almacenar datos.
2. ¿Para qué sirven las variables?  
Para guardar información ala hora de ejecutar un programa.
3. ¿Cómo se guarda un número en una variable?  
Enteros (int): \$edad=?; o puede ser decimales (float): \$precio= ?
4. ¿Cómo se guarda un texto en una variable?  
Texto (string): \$nombre=?
5. ¿Qué tipo de datos pueden almacenar las variables?  
Enteros (int)  
Decimales (float)  
Texto (string)  
Booleanos (bool)
6. ¿Cuál es la diferencia entre una variable numérica y una de texto?  
Depende la que se utilice o especifique, el programador va a interpretar e imprimir, si utilizamos una numérica solo va a leer números, y una de texto solo texto.
7. ¿Se pueden cambiar los valores de las variables?  
Si se pueden cambiar.
8. ¿Por qué es importante usar nombres claros en las variables?  
Para que a la hora de identificación de o que se le asigno a la variable, el programa lo pueda leer e interpretar correctamente.
9. ¿Qué sucede si intentas usar una variable sin haberle asignado un valor?  
El código no será leído correctamente, por ende, aparecerá error.
10. En el siguiente código, ¿qué tipo de variable es "altura"?

Una variable decimal (string).

## **Algoritmos y Diagramas de Flujo**

11. ¿Qué es un algoritmo?  
Son el paso a paso para que resuelva una instrucción.
12. ¿Para qué sirven los algoritmos?  
Para seguir una indicación en orden y llegar al resultado requerido.
13. Da un ejemplo de algoritmo en la vida real.  
Un niño cuando va a aprender a caminar, debe primero seguir unos pasos para poder lograrlo.
14. ¿Cuál es la diferencia entre un algoritmo y un programa?  
El algoritmo son los pasos de lo que se va a ejecutar en el programa, este lo que hace es mostrar las indicaciones requeridas.
15. ¿Cuáles son los tipos de algoritmos que existen?  
Secuenciales  
Condicionales  
Ciclos o Iterativos
16. Explica qué es un algoritmo condicional.  
Sirven para tomar decisiones dependiendo de la situación descrita.
17. Explica qué es un algoritmo con ciclos.  
Repiten una operación varias veces.
18. Si tienes que lavar la ropa, ¿cómo podrías describirlo como un algoritmo?
  - Reunir la ropa sucia
  - Separar por colores y tipos de tela (Ejemplo: ropa blanca, de color, delicada)
  - Verificar etiquetas de cuidado en la ropa
  - Colocar la ropa en la lavadora
  - Agregar detergente y suavizante según la cantidad de ropa
  - Seleccionar el ciclo de lavado adecuado
  - Encender la lavadora y esperar a que termine
  - Sacar la ropa y escurrir si es necesario
  - Colgar o meter en la secadora según el tipo de tela
  - Doblar o planchar la ropa cuando esté seca
  - Guardar la ropa en su lugar

19. ¿Se pueden hacer algoritmos sin saber programar?

Si

20. ¿Qué es pseudocódigo?

Es una indicación que le estamos dando para ser ejecutada completamente.

21. ¿Qué es un diagrama de flujo?

Son diseños que muestran los algoritmos de manera visual.

22. ¿Cuál es su utilidad?

Mejorar el entendimiento de los pasos, identificar errores y diseñar el programa antes de escribir el código.

23. ¿Cuáles son los símbolos más comunes de los diagramas de flujo?

Ovalo

Flechas

Rectángulos

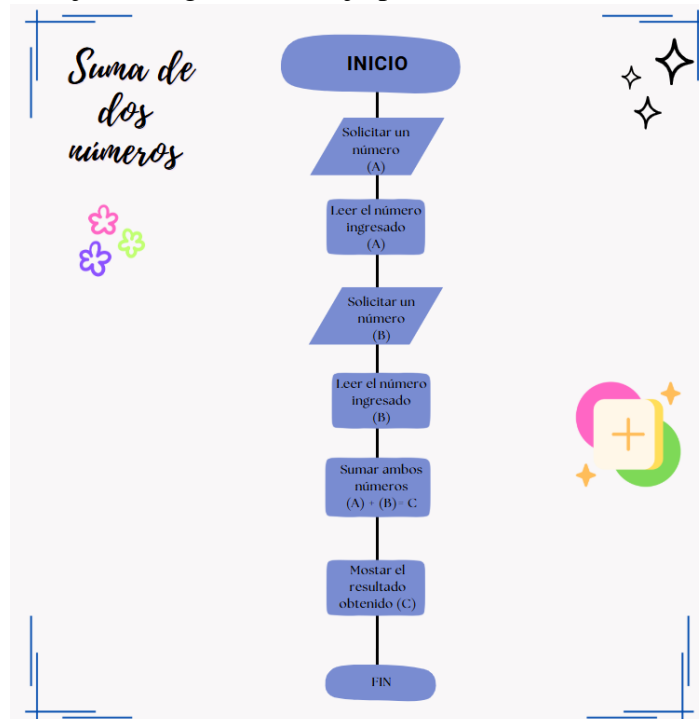
Rombos

Paralelogramo

24. ¿Cómo ayuda un diagrama de flujo a entender un programa?

Al dibujar un diagrama planificamos y evidenciamos de mejor manera como escribir nuestro código.

25. Dibuja un diagrama de flujo para sumar dos números.



26. ¿Cómo se relacionan los algoritmos con los diagramas de flujo?  
En los diagramas de flujo evidenciaremos las instrucciones paso a paso llamadas algoritmos, que debemos emplear, de esta manera identificamos mejor el paso a paso.
27. ¿Se pueden hacer diagramas de flujo sin escribir código?  
Si, lo más importante es tener claro el algoritmo que es el paso a seguir, para crear de mejor manera nuestro diagrama.
28. ¿Cuándo es recomendable hacer un diagrama de flujo antes de programar?  
Cuando no entendemos muy bien el problema o indicación dada, debemos dibujarlo para crear de mejor manera los pasos.
29. ¿Por qué los diagramas de flujo pueden ayudar a resolver problemas?  
Porque representan de manera visual el problema dado, así de esta forma será más fácil identificarlo.
30. ¿Todos los programas necesitan un diagrama de flujo?  
No, si desde el primer momento es entendido el problema su ejecución será fácil, de lo contrario es recomendable crear un diagrama para evidenciar correctamente el paso a seguir.

### **Uso Práctico y Ejercicios**

31. Escribe un algoritmo para preparar un sándwich.
1. Inicio
  2. Reunir los ingredientes:
    - 2 rebanadas de pan
    - 1 ingrediente de proteína (puede ser jamón)
    - 1 tipo de queso
    - Lechuga, tomate
    - Salsas
  3. Colocar las rebanadas de pan sobre una superficie plana, como una tabla de cocina.
  4. Aplicar salsa:
  5. Colocar la proteína sobre una de las rebanadas de pan.
  6. Añadir queso sobre la proteína.
  7. Coloca las hojas de lechuga, rodajas de tomate.
  8. Cubrir con la otra rebanada de pan.
  9. Presionar ligeramente el sándwich para que se compacte.
  10. Servir y disfrutar.

32. Explica cómo se puede representar en un diagrama de flujo el algoritmo de un cajero automático.

1. Inicio

2. Solicitar la tarjeta e ingresar PIN

Si el PIN es incorrecto, mostrar error y permitir reintentar (hasta 3 intentos).

Si se exceden los intentos, retener la tarjeta y finalizar.

3. Mostrar opciones de operación:

Retiro

Consulta de saldo

Depósito

Transferencia

4. Si el usuario elige "Retiro":

Ingresar monto

Verificar si hay saldo suficiente

Si no hay saldo, mostrar error y volver al menú

Si hay saldo, entregar dinero y actualizar saldo

5. Si el usuario elige "Consulta de saldo"

Mostrar saldo disponible

6. Si el usuario elige "Depósito"

Pedir el monto

Confirmar depósito

7. Si el usuario elige "Transferencia"

Pedir cuenta de destino

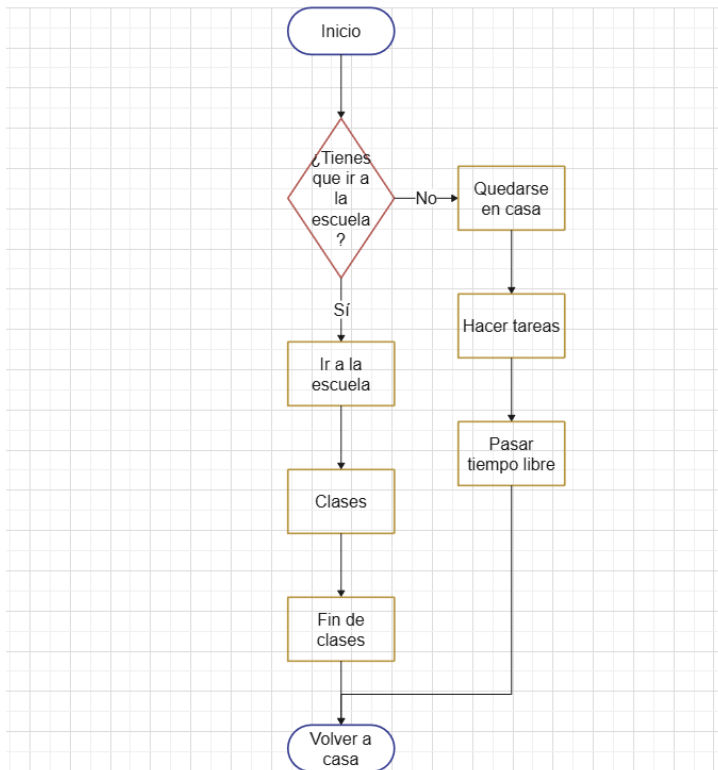
Confirmar monto y realizar la transacción

8. Preguntar si desea otra operación:

Sí: Volver al menú

No: Expulsar tarjeta y finalizar

33. Dibuja un diagrama de flujo para tomar una decisión como "Ir a la escuela o quedarse en casa".



34. Escribe un código sencillo que sume dos números y muestre el resultado.

```
<?php
// Solicitar al usuario que ingrese dos números
$num1 = readline("Ingresa el primer número: ");
$num2 = readline("Ingresa el segundo número: ");

// Sumar los dos números
$suma = $num1 + $num2;

// Mostrar el resultado
echo "La suma de los dos números es: " . $suma . "\n";
?>
```

35. ¿Cuál es la variable en el siguiente código? 36. \$nombre = "Carlos";  
print("Hola, " . \$nombre);

Nombre

37. Escribe un algoritmo que pida dos números y los multiplique.

Inicio: El algoritmo comienza.

Entrada de datos: Se solicita al usuario que ingrese dos números.

Operación de multiplicación: Los dos números ingresados se multiplican y el resultado se guarda en la variable multiplicación.

Salida de datos: El resultado de la multiplicación se muestra al usuario.

Fin: El algoritmo termina.

38. ¿Cómo podrías representar en pseudocódigo un algoritmo que determine si una persona es mayor de edad?

Inicio

Leer edad

Si edad  $\geq$  18 Entonces

Mostrar "Eres mayor de edad."

Sino

Mostrar "Eres menor de edad."

Fin

39. Dibuja un diagrama de flujo que represente un ciclo que cuente del 1 al 10.

