Parcial de lógica y Programación

Karen Sofia Suarez Hernández

Preguntas de Comprensión Temática

Variables

1. ¿Qué es una variable en programación?
Las variables son espacios en la memoria de la computadora donde se pueden guardar datos.
2. ¿Para qué sirven las variables?
Las variables se usan para guardar información que puede cambiar durante la ejecución de un programa.

3. ¿Cómo se guarda un número en una variable?
Para guardar un número en una variable, se le asigna un valor numérico a la variable usando el operador de asignación (=). Ejemplo: $edad = 18;

4. ¿Cómo se guarda un texto en una variable?
Para guardar un texto en una variable, se le asigna una cadena de caracteres (texto) entre comillas. Ejemplo: $nombre = "Juan";

5. ¿Qué tipo de datos pueden almacenar las variables?

Datos como:

- Enteros (int): guardan números enteros.
- Decimales (float): guardan números con decimales.
- Texto (string): guardan palabras o frases.
- Booleanos (bool): guardan valores de verdadero o falso.

6. ¿Cuál es la diferencia entre una variable numérica y una de texto?

La variable numérica solamente contiene números enteros o decimales y la variable de texto solamente contiene palabras.

7. ¿Se pueden cambiar los valores de las variables?

Sí, los valores de las variables se pueden cambiar durante la ejecución del programa, lo que permite modificar la información que guardan.

8. ¿Por qué es importante usar nombres claros en las variables?
Es importante usar nombres claros para las variables porque facilita la comprensión del código.

9. ¿Qué sucede si intentas usar una variable sin haberle asignado un valor?
En este caso habría un error en el programa, ya que la variable no tiene un valor definido.

10. En el siguiente código, ¿qué tipo de variable es "altura"?
$altura = 1.75;

Es una variable de tipo decimal (float).

Algoritmos y Diagramas de Flujo

11. ¿Qué es un algoritmo?

Es un conjunto de pasos para resolver un problema.

12. ¿Para qué sirven los algoritmos?

Los algoritmos sirven para resolver algún problema de manera lógica, obteniendo una solución clara.

13. Da un ejemplo de algoritmo en la vida real.

Si quieres preparar una cena, debes seguir un paso a paso (mezclar ingredientes, cocinar, servir) para al final obtener el plato.

14. ¿Cuál es la diferencia entre un algoritmo y un programa?

La diferencia es que el algoritmo es una serie de instrucciones para resolver un problema y todavía no se emplea ningún lenguaje de programación, mientras que

el programa ya implementa estos algoritmos con un lenguaje de programación para que la computadora lo ejecute.

15. ¿Cuáles son los tipos de algoritmos que existen?
Hay 3 tipos de algoritmos:

• Secuenciales: Se ejecutan en orden, uno tras otro.
• Condicionales: Toman decisiones según una condición.
• Ciclos o Iterativos: Repiten una acción varias veces.

16. Explica qué es un algoritmo condicional.

Cuando se toman decisiones según una condición. Por ejemplo:

Sí es de noche, quédate en casa.

17. Explica qué es un algoritmo con ciclos.

Cuando se repite una acción varias veces. Por ejemplo:

Mientras haya comida, sigue comiendo

18. Si tienes que lavar la ropa, ¿cómo podrías describirlo como un algoritmo?

Se podría describir paso a paso:

- Coger la canasta de la ropa.
- Llevarla al lugar donde se encuentra la lavadora.
- Dejar la canasta a un lado de la lavadora.
- Llenar la lavadora con agua y jabón.
- Meter cada prenda en la lavadora.
- Prender la lavadora.
- Dejar que se acabe el tiempo de lavado.
- Apagar la lavadora
- Sacar cada prenda de la lavadora.
- Exprimir cada prenda.
- Colgar cada prenda.
- Dejar que se seque la ropa.

- Doblar ropa.

19. ¿Se pueden hacer algoritmos sin saber programar?
Si se pueden hacer algoritmos sin saber programar ya que se pueden escribir de forma simple usando diagramas de flujo.

20. ¿Qué es pseudocódigo?
Es una forma de escribir algoritmos usando lenguaje sencillo, sin preocuparse por los detalles técnicos.

21. ¿Qué es un diagrama de flujo?

Son dibujos que representan los algoritmos de forma visual.

22. ¿Cuál es su utilidad?

Facilitan la comprensión de cada paso, también son útiles para planear el programa antes de escribir el código.

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

- Flechas: Muestran la dirección del flujo.
- Óvalos: Indica el inicio o el fin del proceso.
- Rombos: Representa una decisión.
- Rectángulos: Representa una acción o proceso.

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

Un diagrama de flujo ayuda a entender un programa al mostrar de manera visual los pasos y decisiones del algoritmo

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

[Inicio] → [Leer número 1] → [Leer número 2] → [Sumar] → [Mostrar resultado] → [Fin].

26. ¿Cómo se relacionan los algoritmos con los diagramas de flujo?

El diagrama de flujo es la representación visual del algoritmo.

27. ¿Se pueden hacer diagramas de flujo sin escribir código?
Sí, ya que son herramientas para visualizar los pasos de un algoritmo antes de implementarlo.

28. ¿Cuándo es recomendable hacer un diagrama de flujo antes de programar?
Cuando el problema es complejo o quieres tener una visión clara del proceso.

29. ¿Por qué los diagramas de flujo pueden ayudar a resolver problemas?
Porque ayudan a resolver problemas al representar visualmente el proceso y facilita la identificación de errores.

30. ¿Todos los programas necesitan un diagrama de flujo?
No necesariamente, pero son muy útiles para programas complejos.

Uso Práctico y Ejercicios

31. Escribe un algoritmo para preparar un sándwich.

1.Inicio

2.Toma dos rebanadas de pan

3. Poner mantequilla o mayonesa en una o en ambas rebanadas

4. Poner jamón

5. Poner queso

6. Poner tomate

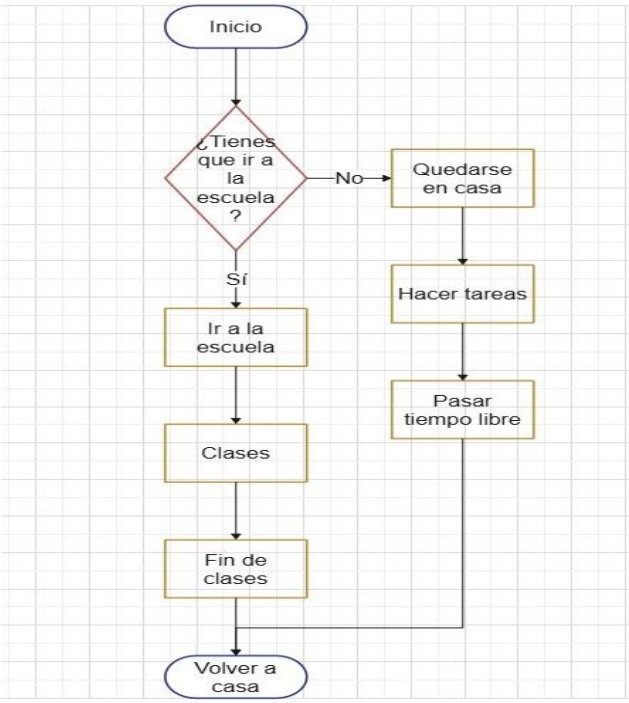
7. Cubrir con la otra tajada de pan

8. Fin

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

1. Inicio
2. Ingresar tarjeta
3. Introducir PIN
4. Verificar PIN
5. Si es correcto, continua.
6. Si es incorrecto, muestra un error y vuelve a pedir el PIN.
7. Seleccionar operación: El usuario selecciona la operación (retirar dinero, consultar saldo, etc.).
8. Realizar operación
9. ¿Operación completada?: Si la operación es exitosa, continúa.
10. Si no, muestra un mensaje de error.
11. ¿Otro retiro?
12. Si sí, repite el proceso.
13. Si no, Finaliza.
14. Fin

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
// Definir dos números
$numero1 = 5;
$numero2 = 10;

// Sumar los dos números
$suma = $numero1 + $numero2;

// Mostrar el resultado
echo "La suma de $numero1 y $numero2 es: $suma";
?>
```

35. ¿Cuál es la variable en el siguiente código?

```
 $nombre = "Carlos";
print("Hola, ". $nombre);
```

La variable es $nombre

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

Inicio

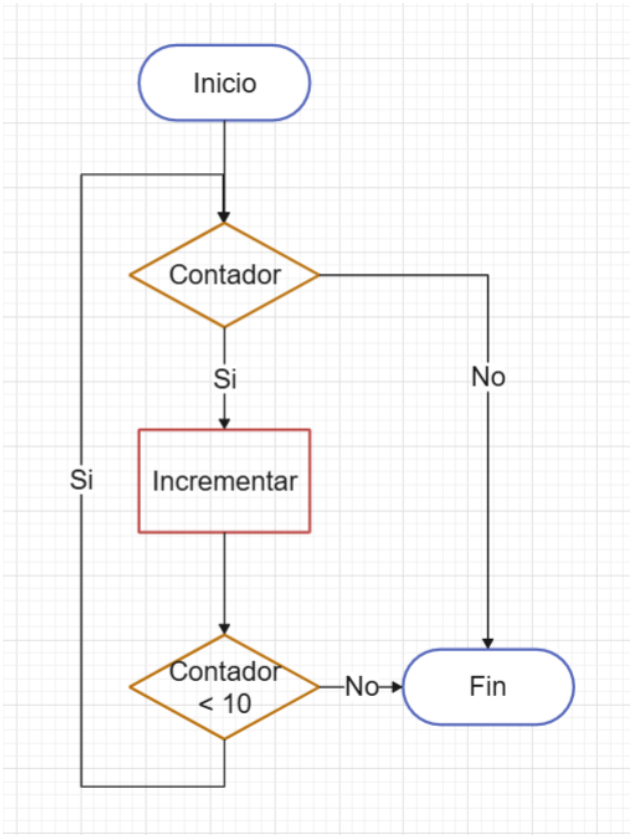
1. Solicitar al usuario que ingrese el primer número.
2. Leer el primer número ingresado por el usuario.
3. Solicitar al usuario que ingrese el segundo número.
4. Leer el segundo número ingresado por el usuario.
5. Multiplicar los dos números.
6. Mostrar el resultado de la multiplicación.

Fin

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

1. Inicio
2. Leer edad
3. Si edad >= 18 Entonces
4. Escribir "Eres mayor de edad."
5. Sino
6. Escribir "Eres menor de edad."
7. Fin

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


Inicio
Contador
Si
No
Incrementar
Si
Contador
< 10
No
Fin