1.¿Qué es una variable en programación?  
• Las variables son espacios en la memoria de la computadora donde se pueden guardar datos.   
Piensa en ellas como cajones donde guardas diferentes tipos de cosas.   
2.¿Para qué sirven las variables?  
• Las variables se usan para almacenar información que puede cambiar durante la ejecución de   
un programa.   
3.¿Cómo se guarda un número en una variable?  
• Enteros (int): guardan números enteros. Ejemplo: $edad = 18;   
• Decimales (float): guardan números con decimales. Ejemplo: $precio = 9.99;

4.¿Cómo se guarda un texto en una variable?  
• Texto (string): guardan palabras o frases. Ejemplo: $nombre = "Ana";   
5. ¿Qué tipo de datos pueden almacenar las variables?

• Números, texto, falso/verdadero

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

• Que al nosotros hacer una variable numérica no tenemos que colocar las comillas ya que el sistema lo entiende por sí mismo, al definir texto tenemos que poner las comillas para encerrar el valor de la variable.

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

• Si, desde el código podemos cambiar el valor de la variable.

8. ¿Por qué es importante usar nombres claros en las variables?

• Para poder definir bien y no tener fallos al ejecutar el código.

9. ¿Qué sucede si intentas usar una variable sin haberle asignado un valor?

• Al no tener un valor no se ejecuta la variable

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

• Numérica, decimal

11. ¿Qué es un algoritmo?

• Un algoritmo es un conjunto de pasos para resolver un problema.

12. ¿Para qué sirven los algoritmos?

• Para resolver diferentes problemas y/o operaciones

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

• Un arroz con pollo, se necesitan una serie de indicaciones para poder lograrlo

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

• El algoritmo sirve específicamente para hacer operaciones.  
15. ¿Cuáles son los tipos de algoritmos que existen?  
• Secuenciales  
• Condicionales  
• Ciclos o Iterativos

16. Explica qué es un algoritmo condicional.

• Toman decisiones según una condición. (if)

17. Explica qué es un algoritmo con ciclos

• Repiten una acción varias veces. (MIENTRAS)

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

- Inicia recogiendo toda la ropa sucia

- Separa la ropa blanca de la demás

- Revisa los bolsillos de la prenda y saca todo

- Configura la lavadora para la ropa que vas a lavar

- Pon el tiempo de lavado

- Abre la lavadora y pon la ropa en el interior

- Agrega el detergente o lo que necesite

- Cierra la lavadora e inicializa la operación

-Espera a que termine la operación

- Ya terminado, abre la lavadora

- Saca la ropa

- Cuelga la ropa

-Ya seca la ropa, descuélgala

-Plánchala

- Dóblala

19. ¿Se pueden hacer algoritmos sin saber programar?

- Si, los algoritmos son una serie de pasos

20. ¿Qué es pseudocódigo?

- Es un lenguaje de programación el cual está diseñado para que las personas que no tienen experiencia o habilidad en el tema lo puedan entender sin problema

21. ¿Qué es un diagrama de flujo?  
Son dibujos que representan los algoritmos de manera visual.   
22. ¿Cuál es su utilidad?  
• Facilitan la comprensión de los pasos.   
• Ayudan a identificar errores.   
• Son útiles para planear programas antes de escribir código.

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

- Rectangulo, rombo, paralelogramo, ovalo  
24. ¿Cómo ayuda un diagrama de flujo a entender un programa?

- Nos muestra de manera gráfica el código  
25. Dibuja un diagrama de flujo para sumar dos números.

Leer num2

Leer num1

Suma<- num1+num2

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

1. Los algoritmos son las instrucciones paso a paso.
2. Los diagramas de flujo ayudan a visualizar los algoritmos.
3. ¿Se pueden hacer diagramas de flujo sin escribir código

27 ¿Se pueden hacer diagramas de flujo sin escribir código?

-Sí, no solo se utilizan los diagramas de flujo para códigos

28. ¿Cuándo es recomendable hacer un diagrama de flujo antes de programar?

-Siempre, para entender lo que estamos haciendo

29. ¿Por qué los diagramas de flujo pueden ayudar a resolver problemas?

-Nos permite realizar de manera más clara lo que tenemos que hacer y nos permite ver la estructura del código con mayor claridad

30. ¿Todos los programas necesitan un diagrama de flujo?

-No, para problemas sencillos no es necesario ya que con hacerlo de forma clara es suficiente

31. Escribe un algoritmo para preparar un sándwich.

**Inicio**

Obtener los ingredientes necesarios:

* Pan (2 rebanadas)
* Relleno (jamón, queso, vegetales, etc.)
* Condimentos (mantequilla, mayonesa, mostaza, etc.)

Colocar las rebanadas de pan sobre una superficie limpia.

Untar los condimentos deseados en una o ambas rebanadas de pan.

Agregar el relleno en una de las rebanadas:

* Colocar el jamón y/o el queso.
* Añadir vegetales como lechuga, tomate, etc.

Cubrir el relleno con la otra rebanada de pan (lado untado hacia abajo).

(Opcional) Cortar el sándwich por la mitad o en cuartos según la preferencia.

Servir el sándwich en un plato.

**Fin**

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

* **Inicio**: Representado por un óvalo, marca el comienzo del proceso.
* **Insertar tarjeta**: Un paralelogramo indica la entrada de la tarjeta por parte del usuario.
* **Verificar tarjeta**: Un rombo representa una decisión, por ejemplo, si la tarjeta es válida o no:
* Si **no es válida**, se muestra un mensaje de error (rectángulo) y se devuelve al usuario.
* Si **es válida**, se pasa al siguiente paso.
* **Ingresar contraseña**: Un paralelogramo muestra que el usuario debe proporcionar su clave personal (PIN).
* **Validar contraseña**: Otro rombo verifica si la contraseña es correcta:
* Si **es incorrecta**, el diagrama puede mostrar un límite de intentos (por ejemplo, 3 veces).
* Si **es correcta**, se continúa.
* **Seleccionar operación**: Un rectángulo indica que el usuario puede elegir entre opciones como consultar saldo, retirar dinero, o realizar depósitos.
* **Procesar la operación seleccionada**:
* Para retiros: Se valida si hay suficiente saldo.
* Para consultas: Se muestra el saldo disponible.
* Para depósitos: Se pide al usuario que inserte dinero o un cheque.
* **Finalizar operación**:
* Si se completa con éxito, el cajero entrega el dinero, el comprobante o un mensaje (rectángulo).
* Si ocurre un error, se devuelve un mensaje al usuario.
* **Devolver tarjeta**: Se asegura que el cajero automático entregue la tarjeta.
* **Fin**: Un óvalo marca el término del proceso.

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

No ir

NO

Ir

SI

Tienes tareas importantes

No ir

SI

NO

Estas enfermo

SI

No ir

NO

Es estre semana

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

<html>

<head></head>

<body>

<?php

$numero1 = 5;

$numero2 = 7;

$resultado = $numero1 + $numero2;

echo "La suma de $numero1 y $numero2 es: $resultado";

?>

</body>

</html>

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

- La variable es $nombre y es una variable de texto

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

<html>

<head></head>

<body>

<?php

$numero1 = rand(1, 100);

$numero2 = rand(1, 100);

$resultado = $numero1 \* $numero2;

echo "El primer número es: $numero1\n";

echo "El segundo número es: $numero2\n";

echo "El producto de $numero1 y $numero2 es: $resultado\n";

?>

</body>

</html>

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

<html>

<head></head>

<body>

<?php

$edad = rand(1, 100);

if ($edad >= 18) {

echo "La edad generada es $edad. Eres mayor de edad.\n";

} else {

echo "La edad generada es $edad. No eres mayor de edad.\n";

}

?>

</body>

</html>

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

