* **PARCIAL LOGICA Y PROGRAMACIÓN**
* **Danna Ximena González Rodríguez**
* Preguntas de Comprensión Temática

1. ¿Qué es una variable en programación?

Las variables son espacios en la memoria de una computadora en la cual se pueden guardar datos.

1. ¿Para qué sirven las variables?

Son usadas para almacenar datos o información que puede cambiar durante la ejecución del programa

1. ¿Cómo se guarda un número en una variable?

Para asignarle un número a una variable

Podemos usar este tipo de código:

$edad=14; asignando la variable para guardar números enteros: int

$precio= 6.35; asignando la variable para guardar números decimales: float

1. ¿Cómo se guarda un texto en una variable?

Para guardar palabras o frases podemos utilizar

$nombre= “Maria”; asignando la variable para guardar texto: String

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

* Enteros (int)
* Decimales (float)
* Texto (String)
* Blooleanos (boof)

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

Que una variable de texto guarda datos como (stings) frases, palabras, nombre para asignar un texto especifico. Mientras que en una variable numérica se guardan datos como (int) números enteros y (float) números decimales para asignar un valor

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

Si, si se pueden cambiar ya que, al ser una variable, su valor puede estar cambiando a medida que el programa avanza

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

Es importante para que el código no tenga ningún problema ya que si no tenemos nombres claros podemos asignar mal un valor

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

Al no asignarle un valor a la variable, la variable no almacenara ni guardara ningún dato por lo tanto ala hora de pedirlo o de requerirlo el código no funcionara

1. En el siguiente código, ¿qué tipo de variable es "altura"?

$altura = 1.75;

La variable de altura es numérica, decimal (float)

1. ¿Qué es un algoritmo?

Un algoritmo es un conjunto de pasos para resolver un problema

1. ¿Para qué sirven los algoritmos?

Los algoritmos nos sirven a la hora de querer solucionar un problema, asignar el conjunto de pasos para dar una solución

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

Un ejemplo de algoritmo podría ser una receta para cocinar un pastel

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

Un algoritmo es una conjunto de pasos aun no codificado, mientras que un programa ya esta codificado

1. ¿Cuáles son los tipos de algoritmos que existen?

* Secuenciales: Se ejecutan en orden
* Condicionales: Tona decisiones mediante una condición
* Ciclos: Repiten un acción varias veces

1. Explica qué es un algoritmo condicional

Un algoritmo condicional, es aquel que tiene que tomar una decisión mediante lo que exprese una condición asignada

1. Explica qué es un algoritmo con ciclos

Un algoritmo en ciclo es aquel que se encarga que una situación suceda varias veces de forma repetitiva

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

* Inicio
* Clasificar la ropa que vaya a lavar (color, blanca, oscura)
* Luego de clasificar la ropa. Introdúzcala en la lavadora
* Prenda la lavadora y asigne el ciclo de su preferencia
* Inicio el ciclo
* Cuando la lavadora le indique que ya puede agregar el jabón, agregue la cantidad que desee según la cantidad de ropa
* Esperar que termine el ciclo
* Fin

1. ¿Se pueden hacer algoritmos sin saber programar?

Si, ya que, para hacer un algoritmo solo se necesita saber hacer la acción que nos piden describir paso a paso

1. ¿Qué es pseudocódigo?

Es un manera de escribir un algoritmo en forma de código

1. ¿Qué es un diagrama de flujo?

Son una manera de representar los algoritmos gráficamente

1. ¿Cuál es su utilidad?

Son útiles para planear un programa antes de realizar el código

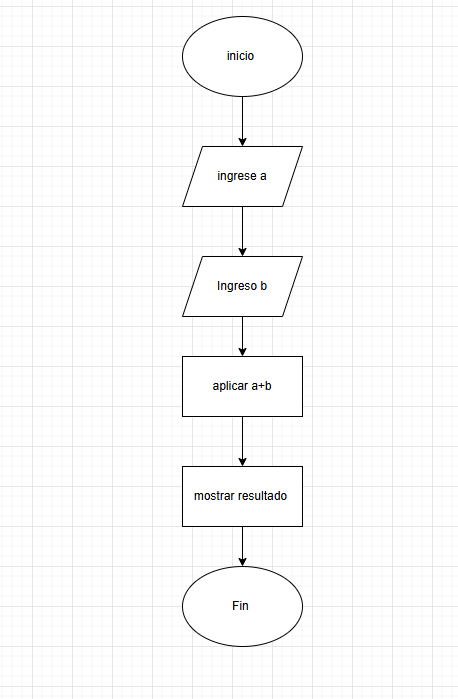
1. ¿Cuáles son los símbolos más comunes de los diagramas de flujo?

* Flechas
* Rombos
* Ovalo
* Rectángulo

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

Por medio de diagrama de flujo podemos planificar de una manera correcta hacer el código del programa

1. Dibuja un diagrama de flujo para sumar dos números



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

Por medio de un algoritmo podemos construir el diagrama de flujo ya que, teniendo los pasos y condiciones allí dichas, podemos construir el diagrama

1. ¿Se pueden hacer diagramas de flujo sin escribir código?

Si, un diagrama se puede hacer sin necesidad de un código ya que el diagrama es lo que nos ayuda a organizar nuestras condiciones y pasos para crear el código

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

Es recomendable cuando nuestro programa tiene o lo que nos piden tiene muchas variables y condiciones por medio de un diagrama de flujo podremos organizar todos los elementos necesarios para el código

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

Nos pueden ayudar ya que por medio de estos podemos organizar toda la información para resolver el problema, lo que se puede hacer, lo que no, y de esta manera organizar toda la información y resolver el problema

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

No necesariamente todos los programas necesitaran un diagrama de flujo ya que no todos tienen el mismo grado de complejidad, sin embargo se puede hacer un diagrama de flujo para todos los programas así sea sencillo, ya que esto n os ayudara a ejecutar de mejor manera nuestro programa

**Uso Práctico y Ejercicios**

1. Escribe un algoritmo para preparar un sándwich.

1. Inicio

2. Toma dos rebanadas de pan

3. Poner mantequilla

4. Poner jamón

5. Poner queso

6. Poner tomate

7. Cubrir con la otra tajada de pan

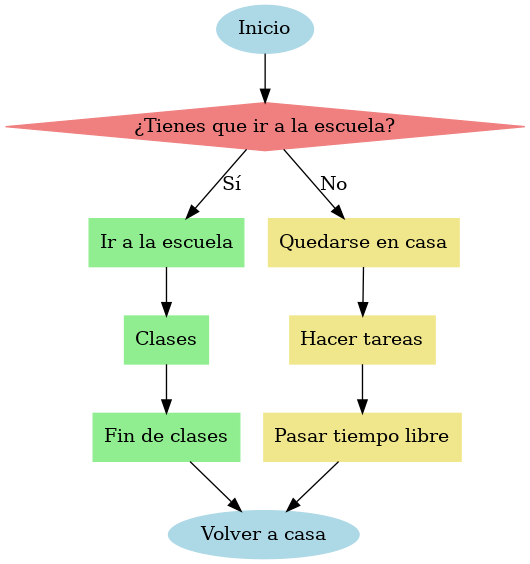
8. calentar si lo desea

9. Fin

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

* comienza con un óvalo para "Inicio"
* seguido de rectángulos para acciones como "Insertar tarjeta" y "Solicitar PIN"
* Usa un rombo para la decisión de "Verificar PIN", con flechas que indiquen los caminos para PIN correcto o incorrecto.
* Continúa con un rectángulo para el "Menú de opciones", otro para la "Selección de opción", y procesos específicos (retiro, consulta, etc.) representados también en rectángulos.
* Si hay dispensación de efectivo o impresión de recibo, úsalos. Incluye otro rombo para "¿Realizar otra transacción?" y finalmente un rectángulo para "Expulsar tarjeta"
* un óvalo para "Fin"
* Conecta todos los símbolos con flechas para mostrar el flujo lógico del proceso.

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



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

num1 = float(input("Ingrese el primer número: "))

num2 = float(input("Ingrese el segundo número: "))

suma = num1 + num2

print("La suma de los dos números es:", suma)

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

La variable del anterior código es “Carlos”

1. Escribe un algoritmo que pida dos números y los multiplique.
2. **Inicio**
3. Solicitar al usuario que ingrese el **primer número** y guardarlo en la variable num1.
4. Solicitar al usuario que ingrese el **segundo número** y guardarlo en la variable num2.
5. Multiplicar num1 por num2 y guardar el resultado en la variable resultado.
6. Mostrar en pantalla el mensaje: **"El resultado de la multiplicación es:"** seguido del valor de resultado.
7. **Fin**
8. ¿Cómo podrías representar en pseudocódigo un algoritmo que determine si una   
   persona es mayor de edad?

inicio

Leer edad

Si edad >= 18 Entonces

Escribir "Eres mayor de edad."

Sino

Escribir "Eres menor de edad."

Fin

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

