REFUERZO DE TEMAS Y CONCEPTOS

1. ¿Qué es un computador?

Un computador es una herramienta de trabajo, que procesa gran cantidad de datos e información a altas velocidades.

---

2. ¿Qué es un programa?

Un programa es un conjunto de instrucciones que la computadora ejecuta.

Ej: Microsoft Word para procesar textos.

---

3. Lógica de programación

La lógica de programación es un pensamiento estructurado para resolver problemas con código.

Ej: Un programa que decide si un número es par o impar.

---

4. Algoritmos

Pasos definidos para resolver un problema.

Ej: Un algoritmo para ordenar una lista de números.

Analogía: Como seguir los pasos de una receta de cocina

---

5. Variables

Espacios de memoria para almacenar datos

Ej: nombre = “Ana” (variable tipo texto)

---

6. Tipos de datos

Clasificación de la información que se maneja en un programa.

Ej: Entero: 23 (int)

Cadena: “Hola” (String)

Booleano: True/False (bool)

Decimales: 2.5(float)

---

7. Operadores

Símbolos que realizan operaciones

Ej: Aritméticos (+ - * / % **)

Relacionales (comparaciones) (< > <= >= == !)

Asignación (= += -= *= /=)

Bit a bit (AND &&, OR || , NOT !(negación)

---

8. Estructuras de control

Decisiones en el código dependiendo si se cumple o no una condición.

```
let edad = 15
if (edad >= 18) {
        console.log(“Es mayor de edad”);
} else if {
console.log(“Es menor de edad”);
}
```

---

9. Bucles

Repetición de código

```
let i = 2
while( i <= 10) {
        console.log(`${i}`)
        i++
}
```

---

10. Entrada y Salida

Leer datos del usuario y mostrarlos

```
let nombre = prompt(“Ingrese su nombre”)
console.log(`Hola, ${nombre}`)
```

---

11. Funciones básicas

Bloques de código reutilizables

```
function saludar(nombre) {
return “Hola, “+ nombre;
}
console.log(saludar(“Ana”));
```

---

12. Primer lenguaje (JavaScript)

Lenguaje de programación para la web

```
Console.log(“Hola, Instru”);
```

---

13. Comentarios en el código

Explicaciones dentro del código

```
// Esto Imprime el mensaje en la consola
console.log(“Abi”);
```

---

14. Errores comunes

SyntaxError: error de escritura

ReferenceError: variable no definida

```
console.log(“Hola” // Error: falta un paréntesis de cierre
```

---

15. Depuración

Uso de console.log() o herramientas de desarrollo.

```
let x = 10;
console.log("Valor de x:", x);
```

---

16. **Compiladores vs. intérpretes**

- **Compilador:** Traduce todo el código antes de ejecutarlo.
- **Intérprete:** Ejecuta línea por línea (como JavaScript).

---

17. **Cadenas de texto**
    Manejo de texto.

```
let nombre = "Juan";

console.log(nombre.length); // Longitud

console.log(nombre.toUpperCase()); // Convertir a mayúsculas
```

---

18. Hardware básico

Definición: Componentes físicos de una computadora.

Ejemplo: CPU (procesador), RAM (memoria), disco duro (almacenamiento).

Analogía: Como el cuerpo humano: el CPU es el cerebro, la RAM es la memoria a corto plazo y el disco duro es la memoria a largo plazo.

Software

Definición: Conjunto de programas que hacen funcionar una computadora.

Ejemplo: Windows (sistema operativo), Google Chrome (aplicación).

Analogía: Como el alma de una persona: sin software, el hardware no puede hacer nada.

---

20. Sistemas operativos

Definición: Software que gestiona el hardware y permite ejecutar programas.

Ejemplo: Windows, macOS, Linux, Android.

Analogía: Como un director de orquesta que coordina todos los instrumentos (componentes del sistema).

---

21. Archivos y carpetas

Definición: Forma de organizar información en la computadora.

Ejemplo: Un archivo documento.docx dentro de la carpeta Mis Documentos.

Analogía: Como una biblioteca: los libros (archivos) están organizados en estanterías (carpetas).

---

22. Terminal o consola

Definición: Interfaz de línea de comandos para interactuar con el sistema.

Ejemplo: Comando cd para cambiar de carpeta.

Analogía: Como hablar con un asistente personal sin usar el mouse ni ventanas.

---

23. Fundamentos de Desarrollo de Software

Definición: Principios esenciales para programar y construir aplicaciones.

Ejemplo: Buenas prácticas como modularidad y eficiencia del código.

Analogía: Como los cimientos de un edificio: sin ellos, el software sería inestable.

---

24. Ciclo de vida del software

Definición: Etapas en la creación de un programa.

Ejemplo: Planificación → Diseño → Desarrollo → Pruebas → Mantenimiento.

Analogía: Como fabricar un auto: primero se diseña, luego se ensambla, se prueba y finalmente se usa.

---

25. Requisitos

Definición: Lo que el usuario necesita que haga el software.

Ejemplo: Un cliente pide una app que organice sus gastos mensuales.

Analogía: Como hacer una pizza: si el cliente quiere una de pepperoni, no puedes entregarle una de champiñones.

---

26. Prototipos

Definición: Versión preliminar de un software para probar ideas.

Ejemplo: Un boceto en Figma antes de desarrollar una app.

Analogía: Como los planos de una casa antes de construirla.

---

27. Interfaz de usuario

Definición: Diseño visual e interacción entre el usuario y el software.

Ejemplo: Botones y menús en una app móvil.

Analogía: Como el tablero de un coche: si los controles están mal ubicados, el conductor se confundirá.

---

28. Pruebas

Definición: Evaluación para detectar errores en un software.

Ejemplo: Testear una app de compras antes de su lanzamiento.

Analogía: Como revisar un auto antes de salir de viaje para evitar fallas.

---

29. ¿Qué es una base de datos?

Definición: Sistema para almacenar y gestionar información estructurada.

Ejemplo: MySQL, PostgreSQL.

Analogía: Como una agenda telefónica digital donde guardas nombres y números.

---

30. Internet

Definición: Red global que conecta computadoras y dispositivos.

Ejemplo: Google, redes sociales.

Analogía: Como una telaraña gigante que une a todo el mundo digitalmente.

Direcciones IP

Definición: Identificadores únicos de dispositivos en una red.

Ejemplo: 192.168.1.1 para acceder a un router.

Analogía: Como una dirección postal en el mundo digital.

---

32. Navegadores

Definición: Programas para acceder a sitios web.

Ejemplo: Chrome, Firefox, Edge.

Analogía: Como una ventana que te permite ver y explorar internet.

---

33. Cliente y servidor

Definición: Modelo donde un dispositivo (cliente) solicita datos a otro (servidor).

Ejemplo: Un navegador solicita una página web a un servidor.

Analogía: Como pedir comida a domicilio: el restaurante (servidor) prepara y entrega la comida al cliente.

---

34. Seguridad inicial

Definición: Prácticas básicas para proteger datos y sistemas.

Ejemplo: Uso de contraseñas fuertes y autenticación en dos pasos.

Analogía: Como poner cerraduras en tu casa para evitar robos.

---

35. HTML

Definición: Lenguaje para estructurar páginas web.

Ejemplo:

<h1>Hola, mundo</h1>

Analogía: Como los ladrillos de una casa web.

---

36. CSS

Definición: Lenguaje para dar estilo a páginas web.

Ejemplo:

h1 { color: blue; }

Analogía: Como la decoración de una casa web.

---

37. JavaScript introductorio

Definición: Lenguaje de programación para hacer páginas interactivas.

Ejemplo:

alert("¡Bienvenido!");

Analogía: Como los circuitos eléctricos que dan vida a una casa web.

---

38. Páginas estáticas

Definición: Sitios web sin cambios dinámicos.

Ejemplo: Un blog sin interacción con el usuario.

Analogía: Como un cartel impreso que no se puede modificar.

---

39. Hosting básico

Definición: Servicio para subir páginas web a internet.

Ejemplo: GitHub Pages, Netlify.

Analogía: Como alquilar un local para tu negocio en línea.

---

40. Editores de código

Definición: Herramientas para escribir programas.

Ejemplo: VS Code, Sublime Text.

Analogía: Como una libreta digital para programadores.

---

41. **Control de versiones**

Definición: Sistema que registra cambios en el código para poder volver atrás si es necesario.

Ejemplo: Git permite hacer un seguimiento de las modificaciones en un proyecto.

Analogía: Como usar "Ctrl + Z" para deshacer cambios en un documento.

---

42. Repositorios

Definición: Espacios donde se almacenan los archivos y versiones del código.

Ejemplo: GitHub es una plataforma de repositorios donde los desarrolladores suben sus proyectos.

Analogía: Como una caja fuerte donde guardas copias de seguridad de tu trabajo.

---

43. Línea de comandos

Definición: Interfaz de texto donde se escriben comandos para interactuar con el sistema.

Ejemplo:

```
cd carpeta  # Cambiar de directorio
ls          # Listar archivos
```

Analogía: Como enviar mensajes de texto al sistema operativo en lugar de hacer clic con el mouse.

---

44. Entornos de desarrollo

Definición: Herramientas y configuraciones necesarias para programar.

Ejemplo: Visual Studio Code (VS Code) es un entorno usado por desarrolladores.

Analogía: Como tener un escritorio bien organizado con todas las herramientas necesarias para trabajar.

---

45. Metodología ágil

Definición: Forma de desarrollar software en pequeñas iteraciones para mejorar continuamente.

Ejemplo: Scrum, donde el equipo trabaja en ciclos cortos llamados *sprints*.

Analogía: Como mejorar una receta poco a poco en cada intento hasta que quede perfecta.

---

46. Documentación

Definición: Instrucciones que explican cómo funciona un código o un software.

Ejemplo: MDN Web Docs es una guía para aprender sobre tecnologías web.

Analogía: Como el manual de un electrodoméstico, que explica cómo usarlo correctamente.

---

47. Resolución de problemas

Definición: Habilidad para analizar y solucionar errores en el código.

Ejemplo: Depurar un programa que no funciona correctamente usando console.log().

Analogía: Como ser detective y analizar pistas hasta encontrar la causa del problema.

---

48. Comunicación

Definición: Capacidad de explicar ideas técnicas de forma clara y efectiva.

Ejemplo: Un desarrollador que explica cómo usar una API a su equipo.

Analogía: Como traducir un idioma técnico para que todos puedan entenderlo.

---

49. Pensamiento crítico

Definición: Evaluar diferentes soluciones antes de elegir la mejor.

Ejemplo: Comparar dos algoritmos para ver cuál es más eficiente.

Analogía: Como analizar varias rutas antes de elegir la más rápida para llegar a un destino.

---

50. Ética en TI

Definición: Uso responsable de la tecnología para evitar daños o abusos.

Ejemplo: No robar datos personales ni crear software malicioso.

Analogía: Como respetar las reglas de tránsito para evitar accidentes.

---

51. Privacidad

Definición: Protección de la información personal de los usuarios.

Ejemplo: Usar contraseñas seguras y cifrado en aplicaciones.

Analogía: Como cerrar bien las cortinas de tu casa para que nadie pueda espiar dentro.

---

52. Persistencia

Definición: No rendirse ante los errores y seguir aprendiendo.

Ejemplo: Un programador que prueba varias soluciones hasta que su código funcione.

Analogía: Como aprender a andar en bicicleta: caerse muchas veces hasta lograrlo.

---

53. Proyecto simple

Definición: Desarrollo de una pequeña aplicación para practicar programación.

Ejemplo:

```
function sumar(a, b) {
   return a + b;
}
console.log(sumar(2, 3)); // 5
```

Analogía: Como construir un prototipo antes de hacer un producto final.

---

54. Reutilización de código

Definición: Usar funciones y bibliotecas en lugar de escribir código desde cero.

Ejemplo:

```
let texto = "Hola";

console.log(texto.toUpperCase()); // HOLA
```

Analogía: Como usar piezas de Lego para armar algo nuevo sin empezar desde cero.

---

55. Inteligencia artificial

Definición: Sistemas que aprenden y toman decisiones sin intervención humana.

Ejemplo: Chatbots como **ChatGPT** o asistentes como **Siri**.

Analogía: Como un robot que aprende observando cómo actúan los humanos.

---

56. Tipos de archivos

Definición: Diferentes formatos de archivos según su uso.

Ejemplo:

- .docx (documento de texto)
- .png (imagen)
- .mp4 (video)

Analogía: Como tener distintos tipos de papel para escribir, dibujar o imprimir fotos.

---

57. Aplicaciones móviles

Definición: Programas diseñados para ejecutarse en teléfonos o tablets.

Ejemplo: WhatsApp, Instagram, Google Maps.

Analogía: Como herramientas portátiles que llevas en el bolsillo.

---

58. Videojuegos

Definición: Software interactivo para entretenimiento.

Ejemplo: Juegos como **Minecraft**, **Fortnite** o **Super Mario**.

Analogía: Como crear un mundo virtual donde las personas pueden jugar y explorar.

---

59. **Impacto del software**
El software ha revolucionado la forma en que trabajamos, nos comunicamos y vivimos. Desde la inteligencia artificial hasta las aplicaciones móviles, su impacto es global.

Ejemplo: Plataformas como **Uber** cambiaron el transporte, **Netflix** transformó el entretenimiento y **ChatGPT** revolucionó la asistencia digital.

Analogía: El software es como la electricidad moderna: aunque no lo veas, está en todas partes facilitando la vida diaria.

---

60. Aprendizaje continuo
El aprendizaje continuo es clave en tecnología porque las herramientas, lenguajes de programación y metodologías evolucionan constantemente.