guia de refuerzo

Daniel mauricio galindo paz

Santiago tirado londoño

### **Refuerzo de temas y conceptos**

1. **Computador**:

**Concepto**: Un computador es un dispositivo electrónico capaz de realizar diversas tareas, como cálculos y procesamiento de datos, siguiendo instrucciones específicas dadas en forma de programas.

**Ejemplo**: Laptops, computadoras de escritorio, smartphones y tabletas son todos tipos de computadores.

**Analogía**: Piensa en un computador como un trabajador de oficina súper eficiente que puede manejar múltiples tareas simultáneamente y con gran precisión.

1. **Programa**:

**Concepto**: Un programa es un conjunto de instrucciones que una computadora sigue para realizar una tarea específica.

**Ejemplo**: Un procesador de textos como Microsoft Word es un programa utilizado para crear y editar documentos de texto.

**Analogía**: Imagina una receta que guía a un chef a través de los pasos necesarios para preparar una comida. Del mismo modo, un programa guía a la computadora a través de los pasos necesarios para completar una tarea.

1. **Lógica de programación**:

**Concepto**: La lógica de programación se refiere al proceso de pensamiento estructurado utilizado para resolver problemas y crear programas.

**Ejemplo**: Escribir código que ordene una lista de números en orden ascendente.

**Analogía**: Es como planificar una ruta en un mapa para ir del punto A al punto B, asegurándote de tomar el camino más eficiente.

1. **Algoritmos**:

**Concepto**: Un algoritmo es un procedimiento paso a paso para resolver un problema o realizar una tarea.

**Ejemplo**: Un algoritmo de ordenamiento que organiza datos en un orden particular.

**Analogía**: Piensa en ello como un conjunto detallado de instrucciones para ensamblar muebles. Cada paso debe seguirse en orden para completar con éxito la tarea.

1. **Variables**:

**Concepto**: Las variables son ubicaciones de almacenamiento en un programa de computadora que contienen datos que pueden cambiar durante la ejecución del programa.

**Tipos**: Enteros (números enteros), flotantes (números decimales), cadenas (texto).

**Ejemplo**: En un programa que calcula el área de un rectángulo, longitud y ancho serían variables.

**Analogía**: Considera las variables como contenedores etiquetados donde puedes almacenar diferentes tipos de información y cambiar el contenido según sea necesario.

1. **Tipos de datos**:

**Concepto**: Los tipos de datos son las diferentes formas en que se pueden almacenar y manipular datos en un programa de computadora.

**Tipos**: Números (enteros y flotantes), cadenas (texto), booleanos (verdadero o falso).

**Ejemplo**: En un programa que verifica si un usuario tiene acceso a una página web, podrías usar un booleano para almacenar el resultado (verdadero o falso).

1. **Operadores**:

**Concepto**: Los operadores son símbolos utilizados en la programación para realizar operaciones en datos.

**Tipos**: Aritméticos (+, -, \*, /) y lógicos (AND, OR, NOT).

**Ejemplo**: En una expresión matemática como 5 + 3, el + es un operador aritmético.

1. **Estructuras de control**:

**Concepto**: Las estructuras de control son construcciones de programación que permiten tomar decisiones y ejecutar diferentes bloques de código en función de ciertas condiciones.

**Tipos**: Condicionales (if, else).

**Ejemplo**: Un bloque de código if que verifica si una variable es mayor que un valor específico y ejecuta un bloque de código si la condición es verdadera.

1. **Bucles**:

**Concepto**: Los bucles permiten la repetición de un bloque de código hasta que se cumpla una condición específica.

**Tipos**: for, while.

**Ejemplo**: Un bucle for que recorre una lista de elementos y realiza una acción con cada uno de ellos.

1. **Entrada y salida**:

**Concepto**: La entrada y salida se refieren a la lectura de datos del usuario y la presentación de resultados.

**Ejemplo**: Leer la entrada del usuario con un comando input y mostrar resultados con un comando print.

1. **Funciones básicas**:

**Concepto**: Las funciones son bloques de código reutilizables que realizan una tarea específica.

**Ejemplo**: En Python, una función que suma dos números

**Analogía**: Piensa en una función como una receta que puedes usar una y otra vez. Cada vez que necesitas preparar el mismo platillo, solo sigues la receta en lugar de recordar cada paso.

1. **Primer lenguaje**

**Concepto**: Un lenguaje de programación es un conjunto de reglas y sintaxis que permite a los desarrolladores escribir programas que pueden ser entendidos y ejecutados por una computadora.

**Ejemplo**: En Python, la sintaxis básica para imprimir "Hola, mundo" es:  
python  
print("Hola, mundo"

1. **Comentarios en el código**:

**Concepto**: Los comentarios son anotaciones en el código que explican qué hace una sección específica. No son ejecutados por la computadora.

**Ejemplo**: En Python, un comentario puede ser escrito usando el símbolo #:  
python  
# Este es un comentario

print("Hola, mundo") # Esto imprime "Hola, mundo"

**Analogía**: Los comentarios son como notas adhesivas que colocas en un informe para explicar ciertas partes del mismo.

1. **Errores comunes**:

**Concepto**: Los errores en programación son problemas que ocurren cuando el código no se ejecuta correctamente.

**Tipos**: Errores de sintaxis (syntax error) y errores en tiempo de ejecución (runtime error).

**Ejemplo**: Un error de sintaxis ocurre cuando hay un error en la escritura del código, como olvidar un paréntesis. Un error en tiempo de ejecución ocurre cuando el código se ejecuta pero produce un resultado inesperado.

1. **Depuración**:

**Concepto**: La depuración es el proceso de identificar y corregir errores en el código.

**Ejemplo**: Usar declaraciones print para verificar el valor de las variables en diferentes puntos del programa.

**Analogía**: Es como revisar cada paso de una receta para encontrar dónde te equivocaste cuando el platillo no salió como esperabas.

1. **Compiladores vs. intérpretes**:

**Concepto**: Un compilador traduce todo el código fuente de un programa a lenguaje máquina antes de ejecutarlo, mientras que un intérprete traduce y ejecuta el código línea por línea.

**Ejemplo**: Lenguajes como C++ usan compiladores, mientras que lenguajes como Python usan intérpretes.

1. **Cadenas de texto**:

**Concepto**: Las cadenas de texto (strings) son secuencias de caracteres utilizadas para almacenar y manipular texto en un programa.

**Operaciones**: Concatenar (unir) cadenas y obtener su longitud.

**Ejemplo**: En Python, puedes concatenar cadenas usando + y obtener la longitud usando len():  
python  
saludo = "Hola"

nombre = "Mundo"

mensaje = saludo + ", " + nombre

print(mensaje) Salida: Hola, Mundo

1. **Hardware básico**:

**Concepto**: El hardware se refiere a los componentes físicos de una computadora.

**Componentes**: CPU (unidad central de procesamiento), memoria (RAM), disco duro (HDD o SSD).

**Ejemplo**: La CPU es el "cerebro" de la computadora, donde se procesan todas las instrucciones.

1. **Software**:

**Concepto**: El software es el conjunto de programas y sistemas operativos que controlan el hardware y permiten a los usuarios interactuar con la computadora.

**Tipos**: Sistema operativo (OS) y aplicaciones.

**Ejemplo**: Windows es un sistema operativo, mientras que Microsoft Word es una aplicación.

1. **Sistemas operativos**:

**Concepto**: Un sistema operativo (OS) es el software que gestiona los recursos del hardware y proporciona servicios a las aplicaciones.

**Ejemplo**: Windows, Linux.

**Funciones**: Gestión de archivos, ejecución de aplicaciones, control de dispositivos periféricos.

1. **Archivos y carpetas**:

**Concepto**: Los archivos son unidades de almacenamiento de datos, mientras que las carpetas son contenedores que organizan archivos y otras carpetas.

**Ejemplo**: Un documento de texto (archivo) guardado en la carpeta "Documentos".

1. **Terminal o consola**:

**Concepto**: La terminal o consola es una interfaz de línea de comandos donde los usuarios pueden escribir

sigue generando porfa

Por supuesto, continuemos con la lista:

### **Refuerzo de temas y conceptos (cont.)**

1. **Fundamentos de Desarrollo de Software**:
   * **Concepto**: Los fundamentos de desarrollo de software incluyen los principios y prácticas utilizados para diseñar, desarrollar y mantener programas de computadora.
   * **Ejemplo**: El uso de metodologías de desarrollo como el modelo ágil o en cascada.
   * **Analogía**: Piensa en el desarrollo de software como la construcción de una casa, donde necesitas planificar, diseñar, construir y mantener.
2. **Ciclo de vida del software**:
   * **Concepto**: El ciclo de vida del software describe las etapas que atraviesa un software desde su concepción hasta su retiro.
   * **Etapas**: Planificación, diseño, codificación, pruebas, implementación, mantenimiento.
   * **Ejemplo**: Desarrollar una aplicación móvil desde la idea inicial hasta su lanzamiento y actualizaciones.
   * **Analogía**: Es como el ciclo de vida de un producto, desde la idea inicial hasta su comercialización y mejoras continuas.
3. **Requisitos**:
   * **Concepto**: Los requisitos son las necesidades y expectativas del usuario que el software debe cumplir.
   * **Ejemplo**: Un requisito funcional podría ser que una aplicación permita al usuario iniciar sesión con su correo electrónico.
   * **Analogía**: Considera los requisitos como la lista de deseos que un cliente da a un arquitecto para construir su casa.
4. **Prototipos**:
   * **Concepto**: Los prototipos son representaciones simples y preliminares del software para visualizar y probar ideas antes del desarrollo completo.
   * **Ejemplo**: Un boceto de una interfaz de usuario en papel o una versión básica de una aplicación.
   * **Analogía**: Piensa en un prototipo como un borrador o maqueta que te permite experimentar antes de realizar el producto final.
5. **Interfaz de usuario**:
   * **Concepto**: La interfaz de usuario (UI) es la parte del software con la que el usuario interactúa directamente.
   * **Ejemplo**: Los botones, menús y ventanas en una aplicación.
   * **Analogía**: Es como el panel de control de un coche, donde los conductores interactúan con los diferentes controles.
6. **Pruebas**:
   * **Concepto**: Las pruebas verifican que el software funcione correctamente y cumpla con los requisitos.
   * **Ejemplo**: Pruebas unitarias que verifican funciones individuales, pruebas de integración que aseguran que los componentes funcionan juntos.
   * **Analogía**: Es como una revisión de calidad para detectar y corregir defectos antes de entregar el producto.
7. **Qué es una base de datos**:
   * **Concepto**: Una base de datos es un sistema que almacena y organiza información de manera estructurada.
   * **Ejemplo**: Una base de datos que almacena la información de clientes de una empresa.
   * **Analogía**: Piensa en una base de datos como una biblioteca organizada donde se guarda información específica en diferentes estanterías.
8. **Internet**:
   * **Concepto**: El internet es una red global de computadoras interconectadas que permite la comunicación y el intercambio de información.
   * **Ejemplo**: Navegar por la web, enviar correos electrónicos.
   * **Analogía**: Es como una autopista de la información donde puedes acceder y compartir datos en todo el mundo.
9. **Direcciones IP**:
   * **Concepto**: Una dirección IP es un identificador único asignado a cada dispositivo conectado a una red.
   * **Ejemplo**: La dirección IP 192.168.1.1 usada por un enrutador doméstico.
   * **Analogía**: Es como la dirección de una casa, que permite identificar y localizar un dispositivo en una red.
10. **Navegadores**:
    * **Concepto**: Los navegadores son programas que permiten acceder y visualizar páginas web.
    * **Ejemplo**: Google Chrome, Mozilla Firefox.
    * **Analogía**: Un navegador es como una ventana a través de la cual puedes ver y explorar el contenido del internet.
11. **Cliente y servidor**:
    * **Concepto**: La arquitectura cliente-servidor es un modelo donde los clientes (usuarios) solicitan servicios y los servidores responden a esas solicitudes.
    * **Ejemplo**: Un cliente web que solicita una página web y el servidor que la entrega.
    * **Analogía**: Piensa en un restaurante donde los clientes (usuarios) piden comida (servicios) y el personal del restaurante (servidor) la proporciona.
12. **Seguridad inicial**:
    * **Concepto**: La seguridad inicial incluye medidas básicas para proteger la información y los dispositivos.
    * **Ejemplo**: Uso de contraseñas seguras, actualización de software.
    * **Analogía**: Es como cerrar con llave las puertas y ventanas de tu casa para mantenerla segura.
13. **HTML**:
    * **Concepto**: HTML (HyperText Markup Language) es un lenguaje de marcado utilizado para estructurar y presentar contenido en la web.
    * **Ejemplo**: Crear una página web con encabezados, párrafos e imágenes.
    * **Analogía**: HTML es como el esqueleto de una página web, proporcionando la estructura básica.
14. **CSS**:
    * **Concepto**: CSS (Cascading Style Sheets) es un lenguaje utilizado para estilizar y diseñar páginas web.
    * **Ejemplo**: Cambiar el color, la fuente y el diseño de una página web.
    * **Analogía**: CSS es como la ropa y el maquillaje que usas para darle estilo y apariencia a una página web.
15. **JavaScript introductorio**:
    * **Concepto**: JavaScript es un lenguaje de programación que permite agregar interactividad y funcionalidades dinámicas a las páginas web.
    * **Ejemplo**: Crear una alerta emergente cuando se hace clic en un botón.
    * **Analogía**: JavaScript es como la animación y los efectos especiales que hacen que una página web cobre vida.
16. **Páginas estáticas**:
    * **Concepto**: Las páginas estáticas son páginas web cuyo contenido no cambia a menos que se actualice manualmente.
    * **Ejemplo**: Una página web de contacto con información fija.
    * **Analogía**: Piensa en una página estática como un cartel impreso que solo cambia cuando se vuelve a imprimir.
17. **Hosting básico**:
    * **Concepto**: El hosting es el servicio que permite publicar y almacenar páginas web en servidores accesibles desde internet.
    * **Ejemplo**: Subir una página web a un servidor para que otros la vean.
    * **Analogía**: Es como alquilar un espacio en la nube donde se aloja tu página web para que otros puedan visitarla.
18. **Editores de código**:
    * **Concepto**: Los editores de código son herramientas que permiten escribir y editar código de programación de manera eficiente.
    * **Ejemplo**: Visual Studio Code, Sublime Text.
    * **Analogía**: Un editor de código es como un cuaderno digital con funciones adicionales para facilitar la escritura de código.
19. **Control de versiones**:
    * **Concepto**: El control de versiones es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo.
    * **Ejemplo**: Git es una herramienta de control de versiones que permite rastrear modificaciones en el código.
    * **Analogía**: Es como un historial de cambios que te permite ver y revertir a versiones anteriores de tu trabajo.
20. **Repositorios**:
    * **Concepto**: Un repositorio es un lugar donde se almacena y gestiona el código fuente de un proyecto.
    * **Ejemplo**: GitHub es una plataforma para alojar repositorios de código.
    * **Analogía**: Piensa en un repositorio como una biblioteca donde se guarda todo el código de un proyecto.
21. **Línea de comandos**:
    * **Concepto**: La línea de comandos es una interfaz de texto que permite a los usuarios interactuar con el sistema operativo escribiendo comandos.
    * **Ejemplo**: Usar comandos como cd para cambiar de directorio o ls para listar archivos.
    * **Analogía**: Es como hablarle directamente a la computadora en su propio idioma para darle órdenes.
22. **Entornos de desarrollo**:
    * **Concepto**: Los entornos de desarrollo proporcionan herramientas y funcionalidades para facilitar la programación y prueba de software.
    * **Ejemplo**: Instalar y configurar Python para desarrollar aplicaciones.
    * **Analogía**: Un entorno de desarrollo es como un taller de herramientas donde tienes todo lo necesario para construir software.
23. **Metodología ágil**:
    * **Concepto**: La metodología ágil es un enfoque de desarrollo de software que se enfoca en iteraciones cortas y entrega continua de productos funcionales.
    * **Ejemplo**: Utilizar Scrum o Kanban para gestionar proyectos.
    * **Analogía**: Piensa en la metodología ágil como una serie de sprints cortos en lugar de una carrera larga, entregando valor al cliente en cada etapa.
24. **Documentación**:
    * **Concepto**: La documentación explica cómo funciona el código y cómo usarlo.
    * **Ejemplo**: Comentarios en el código, archivos

**Documentación**:

* **Concepto**: La documentación explica cómo funciona el código y cómo usarlo.
* **Ejemplo**: Comentarios en el código, archivos README que explican el propósito y el uso del proyecto.
* **Analogía**: La documentación es como un manual de instrucciones que te guía sobre cómo ensamblar y utilizar un producto.

**48. Resolución de problemas**:

* **Concepto**: La resolución de problemas implica identificar, analizar y encontrar soluciones a los problemas que surgen durante el desarrollo del software.
* **Ejemplo**: Diagnosticar y solucionar un error de código que causa un fallo en una aplicación.
* **Analogía**: Es como descomponer un rompecabezas en piezas más pequeñas para encontrar la solución.

**49. Comunicación**:

* **Concepto**: La comunicación en TI implica la capacidad de explicar ideas técnicas y colaborar con otros miembros del equipo.
* **Ejemplo**: Presentar un informe de progreso del proyecto al equipo o explicar una solución técnica a un cliente.
* **Analogía**: Es como ser un traductor que convierte lenguaje técnico en términos comprensibles para todos.

**50. Pensamiento crítico**:

* **Concepto**: El pensamiento crítico es la capacidad de analizar y evaluar información para tomar decisiones informadas y resolver problemas.
* **Ejemplo**: Evaluar diferentes enfoques para optimizar el rendimiento de una aplicación.
* **Analogía**: Es como un detective que examina todas las pistas antes de llegar a una conclusión.

**51. Ética en TI**:

* **Concepto**: La ética en TI implica el uso responsable y ético de la tecnología, considerando el impacto en la sociedad y los individuos.
* **Ejemplo**: Proteger la privacidad de los usuarios y evitar prácticas engañosas o dañinas.
* **Analogía**: Es como seguir un código de conducta que garantiza que tus acciones beneficien a todos y no causen daño.

**52. Privacidad**:

* **Concepto**: La privacidad se refiere a la protección de datos personales y sensibles contra el acceso no autorizado.
* **Ejemplo**: Implementar medidas de seguridad para proteger la información de los usuarios, como cifrado y autenticación de dos factores.
* **Analogía**: Es como poner un candado en tu diario para mantener tus pensamientos personales seguros y privados.

**53. Persistencia**:

* **Concepto**: La persistencia es la capacidad de continuar trabajando en un problema y no rendirse ante los fracasos y errores.
* **Ejemplo**: Continuar depurando un programa hasta que funcione correctamente, a pesar de los obstáculos.
* **Analogía**: Es como un corredor de maratón que sigue avanzando a pesar del cansancio y las dificultades.

**54. Proyecto simple**:

* **Concepto**: Un proyecto simple es una tarea de desarrollo que permite aplicar los conceptos aprendidos y practicar habilidades de programación.
* **Ejemplo**: Crear una calculadora básica o una lista de tareas.
* **Analogía**: Es como un proyecto escolar que te ayuda a aplicar lo que has aprendido en clase.

**55. Reutilización de código**:

* **Concepto**: La reutilización de código implica usar funciones y módulos existentes en lugar de escribir código desde cero para tareas similares.
* **Ejemplo**: Usar una función de biblioteca para leer archivos en lugar de escribir tu propio código.
* **Analogía**: Es como reciclar materiales para crear algo nuevo y evitar desperdicios.

**56. Inteligencia artificial**:

* **Concepto**: La inteligencia artificial (IA) es una rama de la informática que busca crear sistemas capaces de realizar tareas que normalmente requieren inteligencia humana.
* **Ejemplo**: Sistemas de reconocimiento de voz, asistentes virtuales como Siri o Alexa.
* **Analogía**: Piensa en la IA como un robot que puede aprender y tomar decisiones basadas en datos.

**57. Tipos de archivos**:

* **Concepto**: Los tipos de archivos son diferentes formatos en los que se pueden almacenar datos.
* **Ejemplo**: Documentos (doc), imágenes (png, jpg), videos (mp4).
* **Analogía**: Es como tener diferentes tipos de recipientes para almacenar diferentes productos: botes para líquidos, cajas para sólidos, etc.

**58. Aplicaciones móviles**:

* **Concepto**: Las aplicaciones móviles son programas diseñados para ejecutarse en dispositivos móviles como smartphones y tabletas.
* **Ejemplo**: Aplicaciones de redes sociales, juegos móviles, aplicaciones de banca en línea.
* **Analogía**: Piensa en una aplicación móvil como una versión digital y portátil de un servicio o herramienta que puedes llevar contigo a todas partes.

**59. Videojuegos**:

* **Concepto**: El desarrollo de videojuegos implica crear juegos interactivos que se ejecutan en computadoras, consolas y dispositivos móviles.
* **Ejemplo**: Crear un juego sencillo como el clásico "Pong" o un juego más complejo como "Fortnite".
* **Analogía**: Es como contar una historia interactiva donde los jugadores pueden tomar decisiones y afectar el resultado.

**60. Impacto del software**:

* **Concepto**: El software tiene un impacto significativo en la sociedad y en cómo se realizan las tareas cotidianas.
* **Ejemplo**: Aplicaciones de telemedicina que permiten a los pacientes consultar a sus médicos en línea.
* **Analogía**: Piensa en el software como una herramienta que transforma y mejora la forma en que vivimos y trabajamos.

**61. Aprendizaje continuo**:

* **Concepto**: El aprendizaje continuo es la idea de seguir adquiriendo conocimientos y habilidades a lo largo de la vida.
* **Ejemplo**: Tomar cursos en línea, asistir a conferencias, leer libros y artículos relacionados con tu campo.
* **Analogía**: Es como una planta que necesita agua y nutrientes constantes para seguir creciendo y floreciendo.