# Refuerzo de temas y conceptos SENA

Lisandro Jose Rodríguez Mercado

1. ¿Qué es un computador?

Un computador es una máquina electrónica que procesa, almacena y gestiona información. Está compuesto por **hardware** (partes físicas como CPU, memoria RAM, disco duro) y **software** (programas y sistema operativo). Su función principal es ejecutar tareas, procesar datos y permitir la comunicación, siendo una herramienta esencial en el trabajo, educación, entretenimiento y más.

Algunos ejemplos de uso de un computador:

1. Enciendes el computador y abres un programa como Word o Google Docs.

2. Escribes el texto usando el teclado y le das formato.

3. Guardas el archivo en el disco duro o en la nube.

4. Finalmente, lo compartes o imprimes.

1. Qué es un programa - Concepto básico de instrucciones para la computadora.

Un **programa** es un conjunto de instrucciones escritas en un lenguaje que la computadora puede entender. Estas instrucciones le indican a la computadora cómo realizar tareas específicas, como abrir un archivo, realizar cálculos o mostrar información en pantalla. En resumen, un programa es como una "receta" que la computadora sigue paso a paso para ejecutar una acción.

Algunos ejemplos de uso de un programa es abrir un navegador web:

1. Instrucción: El programa (como Chrome o Firefox) contiene instrucciones para mostrar una interfaz gráfica.

2. Ejecución: Al hacer clic en el ícono del navegador, la computadora sigue esas instrucciones.

3. Resultado: El navegador se abre, permitiéndote buscar información, ver videos o visitar páginas web.

1. Lógica de programación - Pensamiento estructurado para resolver problemas.

La **lógica de programación** es el proceso de pensar de manera estructurada y organizada para resolver problemas, utilizando pasos claros y secuenciales que una computadora pueda entender. Es la base para crear programas eficientes.

Ejemplo de uso:

Problema: Calcular el área de un rectángulo.

1. Pensamiento lógico:
   * Identificar los datos necesarios (base y altura).
   * Saber la fórmula: área = base \* altura.
   * Mostrar el resultado.
2. Algoritmos - Pasos para resolver un problema (ej. pseudocódigo).

Un **algoritmo** es una serie de **pasos ordenados y definidos** para resolver un problema o realizar una tarea. Es como una receta detallada que indica qué hacer paso a paso. Los algoritmos se pueden representar en pseudocódigo (un lenguaje sencillo similar al humano) antes de convertirlos en código de programación.

### **Ejemplo de uso**:

**Problema**: Determinar si un número es par o impar.

**Algoritmo en pseudocódigo**:

Inicio

Leer número

Si número % 2 == 0 entonces

Mostrar "El número es par"

Sino

Mostrar "El número es impar"

Fin Si

1. Variables - Concepto y tipos (enteros, decimales, texto).

Una variable es un espacio en la memoria de la computadora que se usa para almacenar datos. Cada variable tiene un nombre (identificador) y un tipo, que define qué tipo de dato puede almacenar. Los datos pueden cambiar (de ahí el nombre "variable") durante la ejecución de un programa.

### **Ejemplo de uso**:

Supongamos que queremos calcular el área de un rectángulo y mostrar el resultado.

base = 10

altura = 5.5

mensaje = "El área es:"

area = base \* altura

print(mensaje, area)

1. **Tipos de datos - Números, cadenas, booleanos**

* **Números**: Son los valores que representan cantidades. Pueden ser enteros (sin decimales) o flotantes (con decimales).  
  Ejemplo:

python

Copiar

x = 10 # entero

y = 3.14 # flotante

* **Cadenas**: Son secuencias de caracteres que representan texto. Se colocan entre comillas simples (') o dobles (").  
  Ejemplo:

python

Copiar

saludo = "Hola, ¿cómo estás?"

nombre = 'Juan'

* **Booleanos**: Son valores que solo pueden ser True (verdadero) o False (falso). Se utilizan en expresiones lógicas y condicionales.  
  Ejemplo:

python

Copiar

es\_mayor = True

es\_adulto = False

**7. \**Operadores - Aritméticos (+, -, , /), lógicos (AND, OR, NOT)***

* **Aritméticos**: Son utilizados para realizar operaciones matemáticas.  
  Ejemplo:

python

Copiar

a = 10

b = 5

suma = a + b # 15

resta = a - b # 5

multiplicacion = a \* b # 50

division = a / b # 2.0

* **Lógicos**: Operan sobre valores booleanos.
  + **AND**: Devuelve True si ambas condiciones son True.

python

Copiar

True and False # False

* + **OR**: Devuelve True si al menos una condición es True.

python

Copiar

True or False # True

* + **NOT**: Invierte el valor de la condición (True se convierte en False y viceversa).

python

Copiar

not True # False

**8. Estructuras de control - Condicionales (if, else)**

Las estructuras condicionales permiten que el programa ejecute diferentes bloques de código según si se cumple o no una condición.

* **if**: Evalúa una condición. Si la condición es True, ejecuta el bloque de código correspondiente.
  + Ejemplo:

python

Copiar

edad = 18

if edad >= 18:

print("Eres adulto")

* **else**: Ejecuta un bloque de código si la condición en if es False.
  + Ejemplo:

python

Copiar

edad = 16

if edad >= 18:

print("Eres adulto")

else:

print("Eres menor de edad")

**9. Bucles - Repetición (for, while)**

Los bucles permiten repetir un bloque de código varias veces.

* **for**: Itera sobre una secuencia (como una lista o un rango de números).
  + Ejemplo:

python

Copiar

for i in range(5): # Rango de 0 a 4

print(i)

* **while**: Repite un bloque de código mientras se cumpla una condición.
  + Ejemplo:

python

Copiar

i = 0

while i < 5:

print(i)

i += 1

**10. Entrada y salida - Leer datos del usuario y mostrar resultados**

* **Entrada**: Usamos input() para pedir al usuario que ingrese datos desde el teclado.
  + Ejemplo:

python

Copiar

nombre = input("¿Cuál es tu nombre? ")

print("Hola " + nombre)

* **Salida**: Usamos print() para mostrar información al usuario en la pantalla.
  + Ejemplo:

python

Copiar

print("Bienvenido al programa")

**11. Funciones básicas - Reutilización de código**

Las funciones permiten agrupar bloques de código que se pueden reutilizar varias veces a lo largo de un programa.

* Ejemplo:

python

Copiar

def saludar(nombre):

print("Hola " + nombre)

saludar("Juan") # Llama a la función y pasa el argumento "Juan"

saludar("Ana") # Llama a la función con otro argumento

**12. Primer lenguaje (ej. Python) - Sintaxis básica y uso**

* **Sintaxis básica** de Python es simple y legible. No requiere declaraciones de tipo y usa indentación para definir bloques de código en lugar de llaves {}.
  + Ejemplo:

python

Copiar

x = 10

if x > 5:

print("Mayor que 5")

**13. Comentarios en el código - Documentar para claridad**

Los comentarios no son ejecutados por el programa. Se usan para explicar qué hace el código o para recordar algo importante.

* Ejemplo:

python

Copiar

# Este es un comentario

x = 10 # Este es un comentario al final de una línea

**14. Errores comunes - Cómo identificarlos (syntax error, runtime error)**

* **Syntax Error**: Ocurre cuando el código no sigue la sintaxis del lenguaje.
  + Ejemplo:

python

Copiar

print("Hola" # Error de sintaxis por falta de paréntesis

* **Runtime Error**: Ocurre durante la ejecución del programa, como intentar dividir entre 0.
  + Ejemplo:

python

Copiar

print(10 / 0) # Error en tiempo de ejecución

**15. Depuración - Uso básico de herramientas o print statements**

Se puede depurar un programa utilizando declaraciones print() para ver los valores de las variables o el flujo del programa.

* Ejemplo:

python

Copiar

x = 5

print("Valor de x:", x)

**16. Compiladores vs. intérpretes - Diferencia simple**

* **Compilador**: Traduce todo el código fuente a código máquina antes de ejecutar el programa. Los lenguajes compilados suelen ser más rápidos, pero requieren una fase de compilación previa.
* **Intérprete**: Traduce y ejecuta el código línea por línea en tiempo de ejecución (como Python).

**17. Cadenas de texto - Manipulación básica (concatenar, longitud)**

* **Concatenar**: Unir cadenas de texto usando el operador +.
  + Ejemplo:

python

Copiar

saludo = "Hola"

nombre = "Juan"

mensaje = saludo + ", " + nombre

print(mensaje) # Imprime "Hola, Juan"

* **Longitud**: Usamos la función len() para obtener la longitud de una cadena.
  + Ejemplo:

python

Copiar

longitud = len("Hola")

print(longitud) # Imprime 4

**18. Hardware básico - CPU, memoria, disco duro**

* **CPU**: El procesador, el componente encargado de ejecutar las instrucciones de un programa.
* **Memoria RAM**: Memoria volátil que almacena temporalmente los datos y programas que están en ejecución.
* **Disco duro**: Almacena los datos de manera permanente, incluso cuando el computador está apagado.

**19. Software - Sistema operativo vs. aplicaciones**

* **Sistema operativo**: Es el software que gestiona los recursos del hardware y permite la interacción con otros programas (ej. Windows, Linux).
* **Aplicaciones**: Son programas diseñados para realizar tareas específicas (ej. Microsoft Word, navegadores web).

**20. Sistemas operativos - Windows, Linux, funciones básicas**

* **Windows**: Sistema operativo con una interfaz gráfica, ampliamente utilizado en PC.
* **Linux**: Sistema operativo basado en código abierto, usado especialmente en servidores.

**21. Archivos y carpetas - Organización en el sistema**

Los **archivos** contienen datos y las **carpetas** los organizan. El sistema de archivos permite guardar, buscar y acceder a los archivos.

**22. Terminal o consola - Comandos básicos (cd, dir, ls)**

* **cd**: Cambia el directorio de trabajo.
  + Ejemplo:

bash

Copiar

cd Documentos

* **dir (Windows)** o **ls (Linux/Mac)**: Muestra los archivos y carpetas en el directorio actual.
  + Ejemplo:

bash

Copiar

ls

**23. Fundamentos de Desarrollo de Software**

El desarrollo de software es el proceso de crear, diseñar, implementar, probar y mantener programas informáticos.

**24. Ciclo de vida del software - Idea básica (planificar, diseñar, codificar)**

El ciclo de vida del software incluye varias fases: planificación, diseño, codificación, pruebas y mantenimiento.

**25. Requisitos - Qué quiere el usuario**

Los **requisitos** son las necesidades o características que el software debe cumplir para satisfacer al usuario.

**26. Prototipos - Bosquejos simples de software**

Un **prototipo** es una versión preliminar de un sistema, usada para mostrar la funcionalidad básica antes de crear la versión final.

**27. Interfaz de usuario - Concepto de diseño básico**

La **interfaz de usuario (UI)** es la parte del software con la que los usuarios interactúan, como botones, menús y formularios.

**28. Pruebas - Verificar que el programa funcione**

Las **pruebas** aseguran que el software cumpla con sus requisitos y funcione correctamente.

**29. Qué es una base de datos - Almacenar información organizada**

Una **base de datos** es un sistema que almacena información de forma estructurada, como tablas. Ejemplos incluyen MySQL y MongoDB.

**30. Internet - Cómo funciona a nivel básico**

La **internet** es una red global de computadoras que se comunican entre sí utilizando protocolos de comunicación (como HTTP).

**31. Direcciones IP - Identificadores simples**

Una **dirección IP** es un número único que identifica a cada dispositivo conectado a la red.

**32. Navegadores - Qué hacen y cómo**

Los **navegadores web** permiten acceder y mostrar páginas web. Ejemplos: Google Chrome, Firefox.

**33. Cliente y servidor - Interacción básica**

En el modelo cliente-servidor, el **cliente** solicita información y el **servidor** la proporciona.

**34. Seguridad inicial - Contraseñas y riesgos**

Las **contraseñas** protegen el acceso a los sistemas. Es importante usar contraseñas fuertes y evitar riesgos como la reutilización de contraseñas.

**35. HTML - Estructura de una página web**

**HTML** es el lenguaje utilizado para estructurar páginas web. Define elementos como títulos, párrafos y enlaces.

* Ejemplo:

html

Copiar

<html>

<head><title>Mi Página Web</title></head>

<body><h1>Bienvenido</h1></body>

</html>

**36. CSS - Estilo básico (colores, fuentes)**

**CSS** se usa para estilizar páginas web, cambiando colores, tamaños y disposición de los elementos.

* Ejemplo:

css

Copiar

h1 { color: blue; font-size: 24px; }

body { background-color: lightgray; }

**37. JavaScript introductorio - Interactividad simple (alertas)**

**JavaScript** agrega interactividad a las páginas web. Ejemplo de un mensaje emergente:

* Ejemplo:

javascript

Copiar

alert("¡Hola, Mundo!");

**38. Páginas estáticas - Crear algo visible**

Las **páginas estáticas** muestran contenido fijo y no cambian a menos que se modifique el archivo HTML.

**39. Hosting básico - Subir una página a la web**

El **hosting** es el proceso de poner una página web en un servidor para que sea accesible en la red.

**40. Editores de código - VS Code, uso básico**

**VS Code** es un editor de código popular que ofrece funciones como autocompletado, depuración y control de versiones.

**41. Control de versiones - Qué es Git (concepto inicial)**

**Git** es un sistema de control de versiones que permite llevar un seguimiento de los cambios en el código.

**42. Repositorios - Idea de GitHub**

**GitHub** es una plataforma donde puedes almacenar tus repositorios de código y colaborar con otros desarrolladores.

**43. Línea de comandos - Comandos útiles para programar**

En la **línea de comandos** puedes ejecutar programas y manejar archivos usando comandos textuales.

**44. Entornos de desarrollo - Instalación de Python o similar**

Un **entorno de desarrollo** es el conjunto de herramientas necesarias para programar en un lenguaje específico.

**45. Metodología ágil - Idea de iteraciones cortas**

La **metodología ágil** promueve el trabajo en ciclos cortos de desarrollo para mejorar continuamente el software.

**46. Documentación - Escribir cómo funciona el código**

La **documentación** ayuda a otros desarrolladores (o a ti mismo) a entender el código.

**47. Resolución de problemas - Dividir en partes**

**Dividir un problema en partes** hace que sea más fácil abordarlo y resolverlo paso a paso.

**48. Comunicación - Explicar ideas técnicas**

Explicar **claramente** ideas técnicas es esencial para colaborar en proyectos de desarrollo.

**49. Pensamiento crítico - Evaluar soluciones**

El **pensamiento crítico** ayuda a analizar las posibles soluciones a un problema y elegir la más adecuada.

**50. Ética en TI - Uso responsable de la tecnología**

La **ética en TI** se refiere al uso responsable de la tecnología, respetando a los usuarios y protegiendo la privacidad.

**51. Privacidad - Proteger datos básicos**

Es importante proteger la **privacidad** de los usuarios, garantizando que sus datos no sean mal utilizados.

**52. Persistencia - Lidiar con errores y fracasos**

La **persistencia** es clave para superar obstáculos y aprender de los errores.

**53. Proyecto simple - Calculadora o lista de tareas**

Un proyecto simple puede ser una **calculadora** o una **lista de tareas**, que permite aplicar conocimientos básicos.

**54. Reutilización de código - Usar funciones ya hechas**

La **reutilización de código** es clave para evitar escribir el mismo código varias veces, haciendo tu trabajo más eficiente.

**55. Inteligencia artificial - Qué es en términos simples**

La **inteligencia artificial (IA)** permite a las máquinas realizar tareas que normalmente requieren de la inteligencia humana, como el aprendizaje y la toma de decisiones.

**56. Tipos de archivos: doc, pdf**

* **.doc**: Formato de documento de Microsoft Word.
* **.pdf**: Formato de documento portátil que conserva el formato original del archivo.

**57. Aplicaciones móviles - Ejemplo de su uso**

Las **aplicaciones móviles** son programas diseñados para ejecutarse en dispositivos móviles, como WhatsApp o Instagram.

**58. Videojuegos - Introducción al desarrollo básico**

El **desarrollo de videojuegos** incluye la creación de la jugabilidad, gráficos, sonidos y mecánicas de un juego.

**59. Impacto del software - Cómo cambia el mundo**

El **software** ha transformado la sociedad, mejorando la productividad, la comunicación y el acceso a la información.

**60. Aprendizaje continuo - Importancia de seguir estudiando**

El **aprendizaje continuo** es crucial para mantenerse actualizado en un campo tan dinámico como la tecnología.