



Sesión 1

Programación Nivel Básico



UNIVERSIDAD
LIBRE®





Sesión 1:

Fundamentos de

Programación y HTML

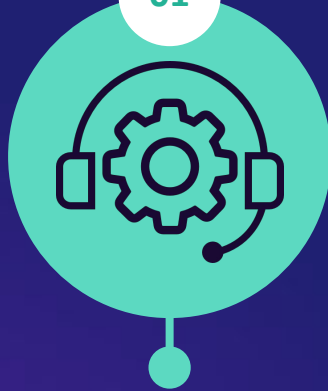
Conceptos básicos de programación



Objetivos de la sesión

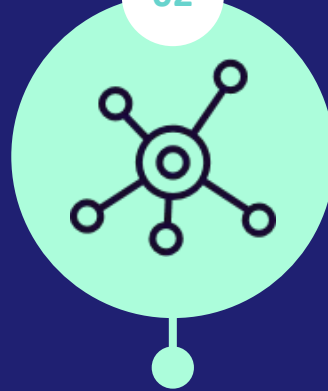
Al finalizar esta sesión estarás en capacidad de:

01



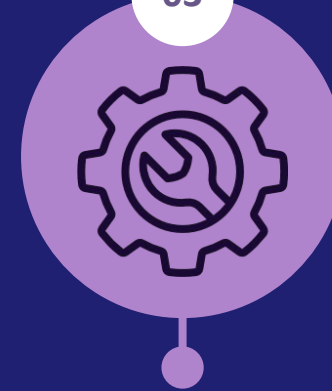
Comprender los
conceptos
fundamentales de
la programación

02



Desarrollar
habilidades para
escribir y
comprender
código

03



Aplicar
conceptos de
programación en
proyectos
prácticos.



Introducción

La programación es una habilidad fundamental en el mundo moderno, un lenguaje que nos permite interactuar con las máquinas e instruirlas para realizar tareas complejas. Es el arte de traducir ideas abstractas en instrucciones comprensibles para las computadoras. A través de la programación, creamos software, aplicaciones web, juegos y una cantidad de herramientas que hacen que nuestras vidas sean más eficientes, divertidas y conectadas.

Introducción a la programación

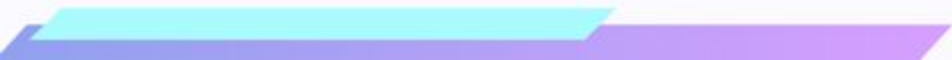
La programación es la base de la tecnología moderna. Se trata de un lenguaje que permite a las computadoras realizar tareas específicas, dando instrucciones a través de código. En esencia, la programación es como hablar con una máquina para que haga lo que queremos.





Definición de programación

La programación es el proceso de diseñar, escribir, probar y depurar código fuente para crear programas de software. Se trata de un proceso creativo que implica la resolución de problemas, la lógica y el pensamiento computacional. El código fuente se escribe en un lenguaje de programación específico que las computadoras pueden entender y ejecutar.



¿Qué es la programación?

Instrucciones para las máquinas

En esencia, la programación consiste en escribir instrucciones detalladas para las computadoras. Estas instrucciones, escritas en un lenguaje específico que la máquina entiende, le dicen qué hacer, cómo hacerlo y en qué orden.

Lógica y algoritmos

Para programar, es esencial tener una comprensión básica de la lógica y los algoritmos. Un algoritmo es una secuencia de pasos bien definidos que resuelven un problema específico. La programación consiste en traducir estos algoritmos en código.

Creación de software y aplicaciones

A través de la programación, podemos crear programas de computadora, aplicaciones web, juegos, sistemas operativos y una gran variedad de herramientas digitales que usamos a diario.



Conceptos básicos de programación

1

Variables

Las variables son contenedores que almacenan datos. Pueden contener información de diferentes tipos, como números, texto o incluso listas de elementos.

2

Operadores

Los operadores permiten realizar operaciones matemáticas, lógicas o de comparación entre variables y datos.

3

Sentencias

Las sentencias son instrucciones que la computadora ejecuta en un orden específico. Cada sentencia realiza una acción, como asignar un valor a una variable o mostrar un mensaje en la pantalla.

4

Estructuras de control

Las estructuras de control determinan el flujo de ejecución de un programa. Permiten tomar decisiones, ejecutar código repetidamente y controlar el orden de las operaciones.



Conceptos ampliados de Programación



La programación incluye:



Variables

Las variables son contenedores que almacenan datos en un programa. Se les da un nombre y un tipo de datos (como entero, texto o booleano) para determinar el tipo de información que pueden contener.



Operadores

Los operadores son símbolos especiales que realizan operaciones matemáticas, lógicas o de comparación en datos. Ejemplos: +, -, *, /, =, ==, <, >.



Estructuras de control

Las estructuras de control determinan el flujo de ejecución del programa, permitiendo que las instrucciones se ejecuten en un orden específico o bajo ciertas condiciones. Entre las más comunes están las condicionales (if, else, elif) y los bucles (for, while).



Ejemplos de Operadores Lógicos

| Operador | Significado | Ejemplo |
|----------|---|----------------------|
| and | Devuelve True si ambos elementos son True | True and True = True |
| or | Devuelve True si al menos un elemento es True | True or False = True |
| not | Devuelve el contrario, True si es Falso y viceversa | not True = False |



Ejemplos de Operadores Matemáticos

| Operador | Significado | Ejemplo |
|-------------------------|----------------|------------------------|
| + (signo más) | Suma | 3+3 |
| - (signo menos) | Resta | 3-1 |
| | Negación | -1 |
| * (asterisco) | Multiplicación | 3*3 |
| / (barra oblicua) | División | 3/3 |
| % (signo de porcentaje) | Porcentaje | 20% |
| ^ (acento circunflejo) | Exponente | 3^2 (el mismo que 3*3) |



Ejemplos de Operadores de Comparación

| Operador | Significado | Ejemplo |
|------------------------|-------------------|---------|
| = (igual) | Igual a | A1=B1 |
| > (mayor que) | Mayor que | A1>B1 |
| < (menor que) | Menor que | A1<B1 |
| >= (mayor o igual que) | Mayor o igual que | A1>=B1 |
| <= (menor o igual que) | Menor o igual que | A1<=B1 |
| <> (distinto) | Distinto de | A1<>B1 |

Algoritmos y estructuras de control



1

Algoritmo

Un algoritmo es una secuencia de pasos bien definidos que resuelven un problema específico. Es la base de la programación, ya que describe cómo se debe realizar una tarea.

2

Estructuras de control

Las estructuras de control son instrucciones que controlan el flujo de ejecución de un algoritmo. Permiten tomar decisiones, repetir acciones y controlar el orden de las operaciones.

3

Ejemplos

Las estructuras de control más comunes son:

- **Sentencia IF:** Ejecuta un bloque de código si se cumple una condición.
- **Sentencia FOR:** Repite un bloque de código un número determinado de veces.
- **Sentencia WHILE:** Repite un bloque de código mientras se cumple una condición.

Tipos de datos y variables



Tipos de datos

Los tipos de datos determinan el tipo de información que puede contener una variable. Algunos tipos de datos comunes son:

- Entero: Números enteros, como 1, 2, 3.
- Decimal: Números con decimales, como 1.5, 2.75.
- Texto: Cadenas de caracteres, como "Hola" o "Programación".
- Booleano: Valores verdaderos o falsos, como True o False.

Variables

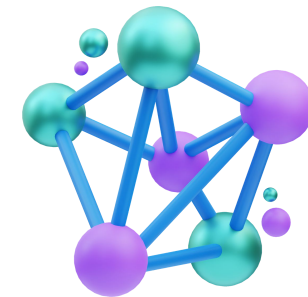
Las variables son contenedores que almacenan datos de un tipo específico. Se les asigna un nombre para poder acceder a su contenido.

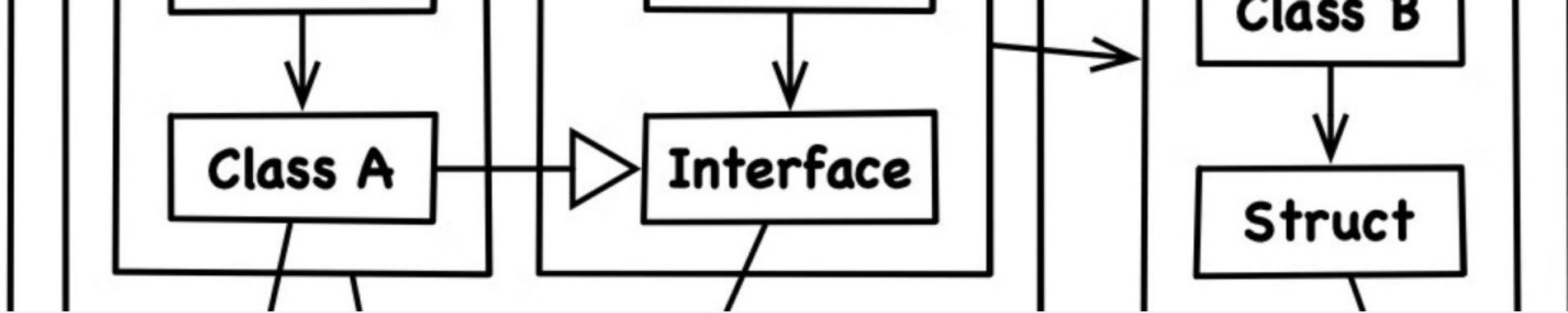
Por ejemplo, la variable "edad" podría almacenar un valor entero que representa la edad de una persona.



Operaciones con variables

Las variables se pueden usar en operaciones matemáticas, lógicas y de comparación. Por ejemplo, se puede sumar el contenido de dos variables o comparar si una variable es mayor que otra.





Funciones y subrutinas

1

Funciones

Las funciones son bloques de código que realizan una tarea específica. Se les asigna un nombre y pueden recibir parámetros como entrada. Las funciones retornan un valor como salida.

2

Subrutinas

Las subrutinas son similares a las funciones, pero no retornan un valor. Se usan para realizar tareas específicas dentro del programa principal.

3

Beneficios de usar funciones y subrutinas

- **Reusabilidad:** Se pueden reutilizar en diferentes partes del programa.
- **Organización:** Mejoran la organización y la legibilidad del código.
- **Modularidad:** Permiten dividir el programa en partes más pequeñas y manejables.



Tipos de lenguajes de programación

Existen diferentes tipos de lenguajes de programación, cada uno con sus características y aplicaciones específicas. Algunos ejemplos son:

| Lenguaje | Descripción | Aplicaciones |
|------------|---|--|
| Python | Lenguaje de alto nivel, versátil y fácil de aprender. | Ciencia de datos, aprendizaje automático, desarrollo web. |
| Java | Lenguaje orientado a objetos, robusto y ampliamente utilizado. | Desarrollo de aplicaciones móviles, aplicaciones web y software empresarial. |
| JavaScript | Lenguaje para desarrollo web, que permite la interactividad en páginas web. | Desarrollo web frontend, desarrollo de juegos, aplicaciones móviles. |
| C++ | Lenguaje de alto rendimiento, usado para desarrollo de sistemas y juegos. | Desarrollo de juegos, desarrollo de sistemas operativos, aplicaciones de alto rendimiento. |

Lenguajes de programación



Python

Un lenguaje de programación versátil y de fácil aprendizaje. Ideal para aplicaciones web, análisis de datos, aprendizaje automático e inteligencia artificial.



JavaScript

El lenguaje del web, utilizado para crear sitios web interactivos, aplicaciones web y juegos.



Java

Un lenguaje de programación orientado a objetos, conocido por su portabilidad y su uso en aplicaciones empresariales y móviles.



C++

Un lenguaje de programación potente y de alto rendimiento, utilizado para el desarrollo de sistemas operativos, juegos y aplicaciones que requieren un alto nivel de control.



Aplicaciones de la programación



Desarrollo Web

La programación es esencial para crear sitios web interactivos, aplicaciones web, plataformas de comercio electrónico y una variedad de servicios en línea.



Desarrollo de aplicaciones móviles

Con la programación, se crean aplicaciones móviles para teléfonos inteligentes y tabletas, abarcando áreas como juegos, redes sociales, compras y entretenimiento.



Desarrollo de juegos

La programación es crucial para crear juegos de computadora y videojuegos, desde juegos simples hasta títulos complejos de AAA.

Aplicaciones de la programación



La programación tiene aplicaciones en casi todos los aspectos de nuestra vida, desde los dispositivos que usamos hasta los servicios que consumimos. Algunos ejemplos de aplicaciones de la programación incluyen:



Desarrollo web

Crear sitios web, aplicaciones web y plataformas online.



Ciencia de datos y aprendizaje automático

Analizar datos, crear modelos predictivos y desarrollar sistemas inteligentes.



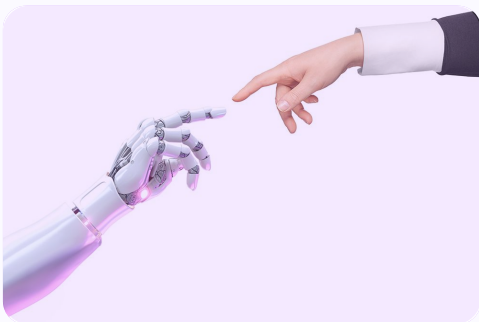
Desarrollo de aplicaciones móviles

Crear aplicaciones para teléfonos inteligentes y tablets.



Ciberseguridad

Proteger sistemas informáticos, redes y datos de amenazas cibernéticas.



Robótica e inteligencia artificial

Controlar robots, desarrollar sistemas de IA y automatizar tareas complejas.

Importancia de la programación en la actualidad

La programación es cada vez más importante en la era digital. El mundo se está volviendo cada vez más automatizado y conectado, y la programación es la clave para impulsar esta transformación. La programación es esencial para:



Innovación

Crear nuevas soluciones y tecnologías para mejorar la vida humana.



Automatización

Optimizar procesos, reducir errores humanos y aumentar la eficiencia.



Conectividad

Crear plataformas y servicios que conecten a las personas y la información.



Oportunidades laborales

Ofrece una gran demanda de profesionales calificados en el mercado laboral.



Importancia de la programación en la actualidad

1

Innovación tecnológica

La programación es el motor de la innovación tecnológica. Permite crear soluciones a los problemas del mundo real, desde la automatización de tareas hasta la creación de nuevos sistemas de comunicación.

2

Crecimiento económico

La industria de la tecnología y la información es una de las más dinámicas y de rápido crecimiento. Los programadores son altamente demandados y bien remunerados.

3

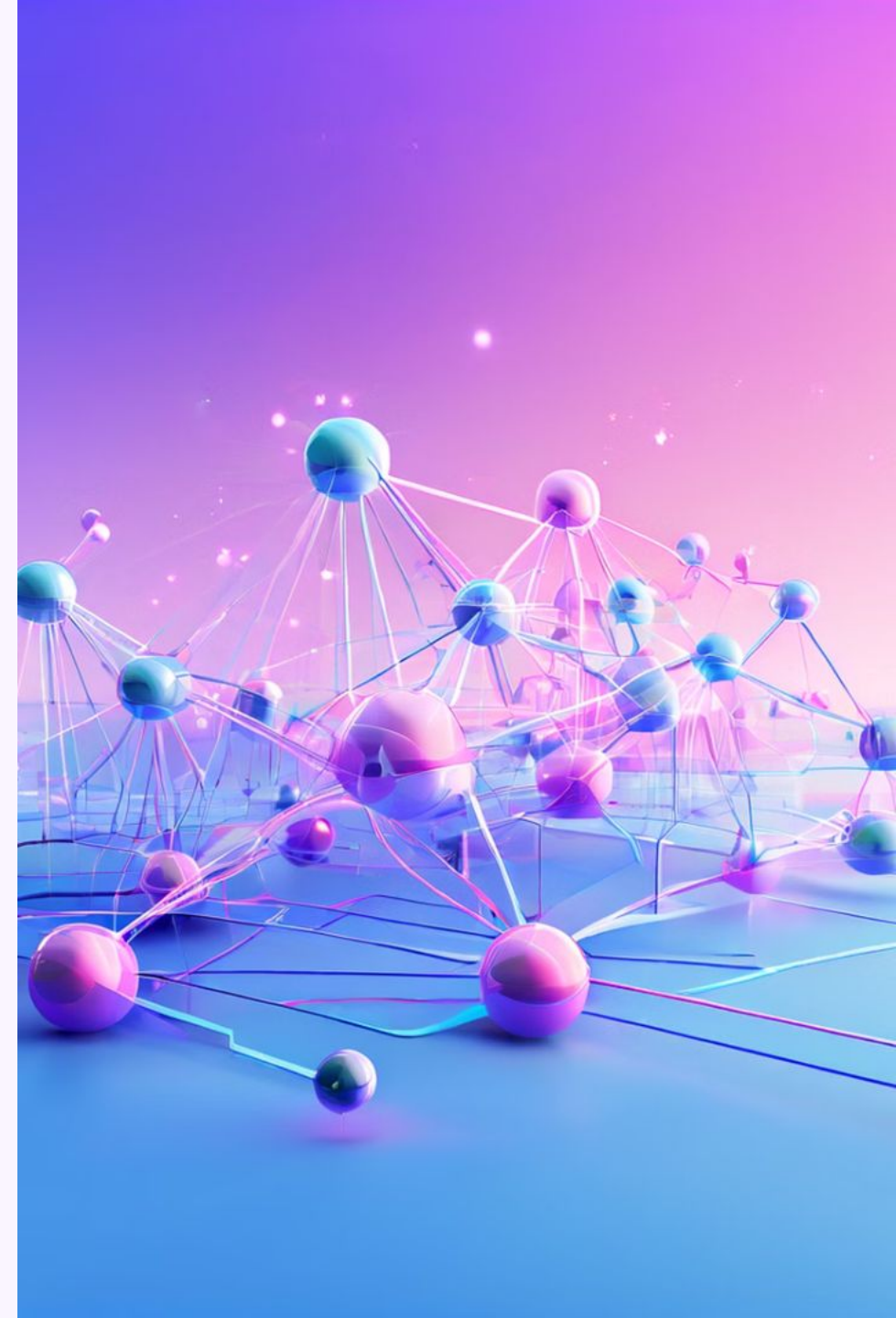
Resolución de problemas

La programación desarrolla habilidades de pensamiento crítico, lógica y resolución de problemas. Es una habilidad valiosa en cualquier campo profesional.

4

Creatividad y expresión

La programación es una forma de expresión creativa. Permite a las personas dar vida a sus ideas y construir nuevas realidades digitales.





Habilidades necesarias para la programación

Para convertirse en un programador exitoso, se requieren habilidades técnicas y de pensamiento crítico. Algunas de las habilidades más importantes incluyen:

1

Pensamiento lógico

La capacidad de descomponer problemas complejos en pasos más pequeños y solucionarlos de forma sistemática.

2

Habilidades de resolución de problemas

La capacidad de identificar problemas, analizarlos y encontrar soluciones creativas.

3

Dominio de lenguajes de programación

La capacidad de escribir código en diferentes lenguajes de programación y entender cómo funciona el código.

4

Conocimiento de estructuras de datos y algoritmos

La capacidad de organizar datos de manera eficiente y desarrollar algoritmos para resolver problemas.

Pasos para comenzar a programar

Si estás interesado en aprender a programar, puedes empezar con estos pasos:

01

Elige un lenguaje de programación

Hay muchos lenguajes de programación disponibles, elige uno que sea adecuado para tus intereses y objetivos. Algunos lenguajes populares son Python, Java, JavaScript y C++.

02

Aprende los conceptos básicos

Estudia los conceptos básicos de la programación, como variables, operadores, estructuras de control y funciones. Hay muchos recursos disponibles en línea y en libros.

03

Practica regularmente

La práctica es fundamental para aprender a programar. Elige proyectos pequeños para empezar y ve aumentando la complejidad gradualmente. Hay muchos desafíos y proyectos en línea que puedes probar.

04

Únete a una comunidad de programadores

Busca comunidades en línea o grupos de programación locales donde puedes compartir conocimientos, obtener ayuda y colaborar con otros programadores.

05

Mantente actualizado

La tecnología evoluciona rápidamente, por lo que es importante mantenerte actualizado con las últimas tendencias y herramientas en programación.





Ejercicios de práctica



iGracias
por ser parte de
esta experiencia
de aprendizaje!

