



Bootcamp de Desarrollo Web

Bootcamp Desarrollo Web 1

1



Bootcamp de Desarrollo Web

Clase 1

Introducción a la programación

Bootcamp Desarrollo Web 2

2



1 - ¿Qué es la programación?

La programación es el proceso de crear un conjunto de instrucciones que le indican a una computadora cómo llevar a cabo una tarea específica.

3



¿Qué es un lenguaje de programación?

Un lenguaje de programación es un conjunto de reglas y símbolos utilizados para escribir programas informáticos. Estos lenguajes permiten a los programadores comunicarse con las computadoras y darles instrucciones para llevar a cabo tareas específicas.

4



Clasificación de los lenguajes de programación

- Lenguaje de Máquina
- Ensamblador
- Alto Nivel



Lenguaje máquina

Es conjunto de instrucciones y códigos binarios que una computadora puede entender y ejecutar directamente.

Está compuesto por una serie de instrucciones binarias (0 y 1) que **representan operaciones** fundamentales como sumar, restar, entre otros, y **direcciones de memoria** donde se realizan estas operaciones.





Lenguaje Ensamblador

El lenguaje ensamblador (también conocido como lenguaje Assembly) es un lenguaje de programación de bajo nivel que se encuentra entre el lenguaje máquina y los lenguajes de alto nivel.

Es más legible que el lenguaje máquina, pero está estrechamente vinculado a la arquitectura de la CPU.

```
section .data
msg db "Hello World!", 0x0
len equ $ - msg
section .text
global _start
_start:
    mov     eax, 4
    mov     ebx, 1
    mov     edx, msg
    mov     ecx, len
    int     0x80

    mov     eax, 1
    int     0x80
```



Lenguaje de Alto Nivel

El lenguaje de alto nivel es un tipo de lenguaje de programación que se acerca más al lenguaje humano, utilizando palabras y estructuras más legibles y comprensibles para los programadores.

Abstrae detalles de la arquitectura de la computadora y ofrece comandos más cercanos al pensamiento humano, facilitando la escritura y comprensión del código.



2. Lenguaje de desarrollo web

Un lenguaje de desarrollo web es un lenguaje de programación o tecnología que se utiliza para crear aplicaciones, sitios web y contenido web interactivo.



Ejemplos de lenguajes de desarrollo web

- 1. HTML (HyperText Markup Language):** Es el lenguaje base para crear la estructura y contenido de las páginas web.
- 2. CSS (Cascading Style Sheets):** Se utiliza para diseñar y dar estilo al contenido HTML, controlando aspectos visuales como el diseño, colores, fuentes, etc.



Ejemplos de lenguajes de desarrollo web

3. PHP (Hypertext Preprocessor): Un lenguaje de programación del lado del servidor que se usa para desarrollar aplicaciones web dinámicas, como sistemas de gestión de contenidos (CMS) como WordPress, Joomla o Drupal.

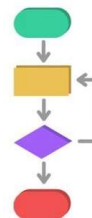
4. Python: Aunque no específicamente diseñado para el desarrollo web, Python es utilizado en el desarrollo web a través de frameworks como Django o Flask para construir aplicaciones web robustas y escalables.



3. Algoritmo

Es un conjunto de pasos o reglas precisas que se siguen para resolver un problema.

Es como una receta que describe cómo hacer algo, desde sumar números hasta ordenar una lista.





Pseudo-código

El pseudocódigo es una representación de alto nivel de un algoritmo o proceso, que utiliza un lenguaje sencillo y cercano al lenguaje humano para describir la lógica y los pasos de un programa.

No sigue una sintaxis específica de ningún lenguaje de programación y se utiliza principalmente para planificar y diseñar algoritmos antes de ser implementados en un lenguaje de programación específico.

```
Proceso Suma
  Definir A,B,C como Reales;
  Escribir "Ingrese el primer numero:";
  Leer A;
  Escribir "Ingrese el segundo numero:";
  Leer B;
  C <- A+B;
  Escribir "El resultado es: ",C;
FinProceso
```

13



Bootcamp de Desarrollo Web

Clase 1

Variables, tipos de dato y operadores

14



¿Qué es un dato?

Un dato es una representación simbólica de información que puede ser procesada por una computadora.

Puede ser un número, una palabra, un símbolo o cualquier unidad de información que tenga significado.



Variable

Una variable es un contenedor o espacio de memoria con un nombre asociado, utilizado para almacenar y representar datos dentro de un programa de computadora.

Constante

Una constante es un valor que no cambia durante la ejecución de un programa.

A diferencia de las variables, cuyos valores pueden modificarse, las constantes mantienen un valor fijo y no se pueden alterar una vez que se ha definido.



Tipos de datos

Los tipos de datos en programación se refieren a las categorías o estructuras que representan los diferentes tipos de información que pueden ser almacenados y manipulados por un programa.

1. **Entero (Integer):** Representa números enteros positivos o negativos sin decimales, como `-5`, `0`, `10`.
2. **Decimal o punto flotante (Float):** Representa números con decimales, como `3.14`, `-0.5`, `2.71828`.



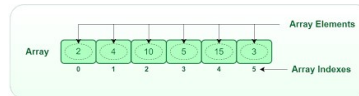
Tipos de datos

3. **Texto (String):** Almacena secuencias de caracteres, como palabras o frases, entre comillas, por ejemplo, `"Hola Mundo"`.
4. **Booleano (Boolean):** Puede ser verdadero (`true`) o falso (`false`). Se utiliza para representar valores de verdad en la lógica, como condiciones o estados lógicos.
5. **Carácter (Character):** Almacena un solo carácter, como `'a'`, `'b'`, `'5'`.

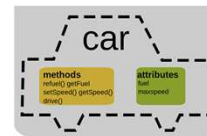


Tipos de datos

6. Arreglos (Arrays): Colecciones ordenadas de datos del mismo tipo que se almacenan juntos bajo un solo nombre.



5. Objetos: Estructuras de datos más complejas que pueden almacenar tanto datos como métodos para operar con esos datos, en lenguajes orientados a objetos como Python o Java.



Operadores de datos

Los operadores de datos son símbolos o palabras clave utilizados en programación para realizar operaciones o manipulaciones en variables u operandos.

Operadores aritméticos: Se utilizan para realizar operaciones matemáticas.

- **Suma (+):** Realiza la adición entre dos valores.
 $3 + 2$
- **Resta (-):** Realiza la sustracción entre dos valores.
 $3 - 2$
- **Multiplicación (*):** Realiza la multiplicación entre dos valores.
 $3 * 2$



Operadores de datos

Los operadores de datos son símbolos o palabras clave utilizados en programación para realizar operaciones o manipulaciones en variables u operandos.

Operadores aritméticos: Se utilizan para realizar operaciones matemáticas.

- **División (/):** Realiza la división entre dos valores.
 $2 / 2$
- **Módulo (%):** Devuelve el resto de una división entera entre dos números.
 $2 \% 2$



Operadores de datos

Operadores de asignación:
Se utilizan para asignar valores a variables.

Asignación (=): Asigna un valor a una variable.

Variable A = 3



Operadores de datos

Operadores lógicos: Utilizados para combinar expresiones lógicas.

- **AND (&&):** Devuelve verdadero si ambas condiciones son verdaderas.
`A && B`
- **OR (||):** Devuelve verdadero si al menos una de las condiciones es verdadera. `A || B`
- **NOT (!):** Niega el valor de una expresión, convirtiendo verdadero en falso y viceversa. `!A`



Operadores de datos

Operadores de comparación: Comparan dos valores y devuelven un resultado lógico (verdadero o falso).

- **Igualdad (==):** Comprueba si dos valores son iguales.
`3 == 3`
- **Desigualdad (!=):** Comprueba si dos valores son diferentes.
`3 != 3`



Operadores de datos

Operadores de comparación: Comparan dos valores y devuelven un resultado lógico (verdadero o falso).

- **Mayor que (>), Menor que (<):** Comprueban si un valor es mayor o menor que otro.
 $3 > 4$
- **Mayor o igual que (>=), Menor o igual que (<=):** Comprueban si un valor es mayor o igual, o menor o igual a otro.
 $3 >= 3$

25



Bootcamp de Desarrollo Web

Clase 1

Estructuras de control de flujo

26



Estructuras de control de flujo

Los operadores de control de flujo son estructuras utilizadas para dirigir el flujo de ejecución de un programa.

Estructuras de control condicional

- **Si-SiNo (If-else):** Permite tomar decisiones basadas en una condición. Ejecuta un bloque de código si la condición es verdadera y otro bloque si es falsa.

```
Si expresion_logica Entonces
.....
  acciones_por_verdadero
SiNo
.....
  acciones_por_falso
Fin Si
```



Estructuras de control de flujo

Los operadores de control de flujo son estructuras utilizadas para dirigir el flujo de ejecución de un programa.

Estructuras de control condicional

- **Según-opción(Switch-case):** Evalúa diferentes valores de una expresión y ejecuta bloques de código dependiendo del valor coincidente.

```
Segun variable_numerica Hacer
opcion_1:
.....
  secuencia_de_acciones_1
opcion_2:
.....
  secuencia_de_acciones_2
opcion_3:
.....
  secuencia_de_acciones_3
De Otro Modo:
.....
  secuencia_de_acciones_dom
```



Estructuras de control de flujo

Los operadores de control de flujo son estructuras utilizadas para dirigir el flujo de ejecución de un programa.

Estructuras de control iterativo o bucles:

- **For:** Ejecuta un bloque de código un número específico de veces.

```
Para variable_numerica<-valor_inicial Hasta valor_final Con Paso paso Hacer  
...  
secuencia_de_acciones  
Fin Para
```



Estructuras de control de flujo

Los operadores de control de flujo son estructuras utilizadas para dirigir el flujo de ejecución de un programa.

Estructuras de control iterativo o bucles:

- **While:** Repite un bloque de código mientras una condición sea verdadera.

```
Mientras expresion_logica Hacer  
...  
secuencia_de_acciones
```



Estructuras de control de flujo

Los operadores de control de flujo son estructuras utilizadas para dirigir el flujo de ejecución de un programa.

Estructuras de control iterativo o bucles:

- **Do-while:** Ejecuta un bloque de código al menos una vez y luego repite mientras la condición sea verdadera.

```
Repetir
    secuencia_de_acciones
Hasta Que expresion_logica
```



Estructuras de control de flujo

Los operadores de control de flujo son estructuras utilizadas para dirigir el flujo de ejecución de un programa.

Estructuras de control de salto:

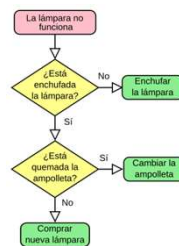
- **Break:** Sale de un bucle o estructura de control.
- **Continue:** Salta a la siguiente iteración en un bucle, omitiendo el resto del código dentro de esa iteración.



Diagrama de flujo

Un diagrama de flujo es una representación gráfica y visual que utiliza símbolos y formas para ilustrar el flujo secuencial de algoritmo

Se utiliza en programación para visualizar paso a paso la secuencia de acciones o decisiones que deben tomarse para completar una tarea.

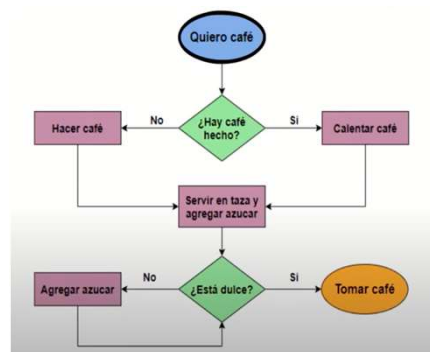


Simbología de un Diagrama de flujo

Símbolo		Función
Líneas de flujo		Conectan los pasos, etapas, decisiones y otros elementos que intervienen en los diagramas
Decisión		Se usan para indicar las elecciones y decisiones realizadas.
Datos		Ofrecen información nueva, de interés o de gran valor para el desarrollo del proceso representado.
Actividad		Indican las acciones que se transforman en datos que dan continuidad al proceso.
Inicio / final		Se utiliza cada vez que se indica el problema/solución en el diagrama de flujo marcando el inicio y cierre de mismo.



Ejemplo de un Diagrama de flujo



35



Bootcamp de Desarrollo Web

Clase 1

¿Que es un IDE?

36



¿Que es un IDE?

Un IDE (Entorno de Desarrollo Integrado, por sus siglas en inglés) es una herramienta de software que proporciona un conjunto completo de facilidades para el desarrollo de software en un solo entorno integrado.

Un IDE típicamente incluye un editor de código, un compilador/intérprete, herramientas de depuración, un gestor de proyectos, herramientas de automatización y otras utilidades necesarias para desarrollar software de manera eficiente.



¿Que es PSeInt?

PSeInt es un entorno de desarrollo y aprendizaje de la lógica de programación.

Su nombre significa "**Pseudo Intérprete**", y está diseñado para ayudar a estudiantes y principiantes en programación a comprender los conceptos básicos de algoritmos y lógica sin la necesidad de aprender un lenguaje de programación en particular.

```
1 Algoritmo suma
2   Escribir "Teclea el primer numero a sumar";
3   Leer num1;
4
5   Escribir "Teclea el segundo numero a sumar";
6   Leer num2;
7
8   sumar <- num1 + num2;
9
10  FinAlgoritmo
```



Bootcamp de Desarrollo Web

Clase 1

Fin