

Lenguaje de Programación

Un lenguaje de programación es aquel que diseña procesos utilizando lenguaje formal propio. Para ello, es necesario establecer normas semánticas y sintácticas correctas, para que el programa desarrollado tenga un buen funcionamiento y desempeño.

Estos lenguajes formales son los que, por medio de intérpretes de lenguaje, ayudan a la comunicación entre la máquina y el usuario. Estos lenguajes utilizan reglas formales que ayudan a que el desarrollador de software maneje de forma clara y estructurada cada uno de los elementos que lo componen.

Entre los lenguajes de programación se encuentran: C,C++, Pascal, Basic, Perl, Python y Java. Cada uno de estos lenguajes tienen diferente sintaxis y diferente gramática, pero cumplen un mismo fin: elaborar soluciones a problemas de la vida cotidiana y hacer más fácil la interacción entre el hombre y la máquina.

En esta unidad se va a utilizar como lenguaje de programación JAVA, ya que es uno de los más relevantes en los últimos tiempos. Cuenta con propiedades, como por ejemplo la programación estructurada.

Java es un lenguaje de programación que apareció en el año 1991, gracias a los ingenieros de Sun MicroSystem, que desarrollaron un lenguaje de programación a la medida, capaz de desarrollar cálculos muy pequeños y optimizar los mismos.

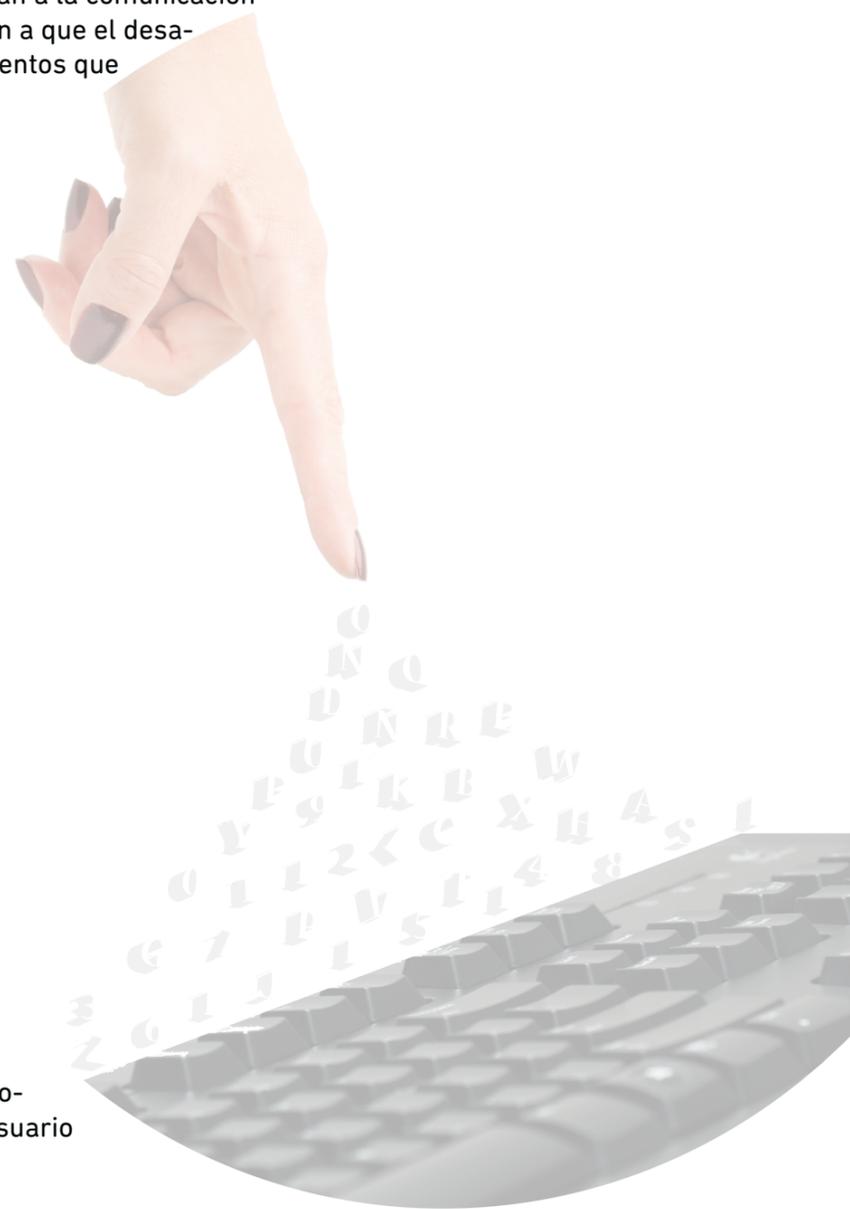
El propósito del lenguaje Java, es no tener límites. Es comunicar usuario final con todo tipo de herramientas, ya sea en la web, utilizando una base de datos o en una simple aplicación de escritorio.

Este lenguaje es de los pocos que establece seguridad al momento de desarrollar e implementar una solución de software. una

¿Dónde generar código Java?:

para generar código fuente en este lenguaje de programación es necesario utilizar un IDE (Entorno de desarrollo integrado). Allí se puede diseñar el código Java, depurar y compilar; es decir, poner en funcionamiento el código diseñado. Algunos de estos entornos muestran herramientas gráficas que ayudan al desarrollador a generar código de forma fácil y eficiente, hacer ventanas de entorno de usuario gráfico y a hacer la interacción más fácil entre el usuario y el código fuente.

Los IDE más utilizados para el desarrollo de lenguaje de programación en Java son Eclipse IDE y Netbeans. Eclipse permite desarrollar habilidades de programación, de tal forma que para realizar el diseño de interfaces gráficas, necesita digitar cada uno de los elementos y de la sintaxis ordenada que maneja el lenguaje Java. En cambio, Netbeans tiene herramientas de diseño de forma gráfica, que ahorran tiempo en el diseño de ventanas.



Tipos de Datos Primitivos:

Los IDE más utilizados para el desarrollo de lenguaje de programación en Java son Eclipse IDE y Netbeans. Eclipse permite desarrollar habilidades de programación, de tal forma que para realizar el diseño de interfaces gráficas, necesita digitar cada uno de los elementos y de la sintaxis ordenada que maneja el lenguaje Java. En cambio, Netbeans tiene herramientas de diseño de forma gráfica, que ahorran tiempo en el diseño de ventanas.

Tipos de Datos Primitivos:

Tipo de Dato	Descripción
Boolean	Cuenta con dos valores true o false
Char	Caracteres alfa-numéricos, tipo código ASCII
Int	El intervalo de valores numéricos se encuentra desde -2147483648 hasta 2147483647
Long	El intervalo de valores se encuentra desde -9223372036854775808 hasta 9223372036854775807
Float	Números con coma flotante con una cifra de precisión, el intervalo de valores va desde 1.40239846e-45f hasta 3.40282347e+38f
Double	Número con coma, con doble cifra de precisión. El intervalo de valores va desde 4.94065645841246544e-324d hasta 1.7976931348623157e+308d.

Los caracteres especiales que se utilizan para el diseño de código Java y organización del código son los siguientes:

Carácter	Función
\n	Salto de línea
\t	Tabulador Horizontal
\r	Retorno de carro
\\	Barra invertida

Declaración de variables:

La declaración de variables en java se realiza de la siguiente forma:

<tipo variable> <nombre variable>;

Por ejemplo:

```
int primer_Número;
```

Inicialización de variables:

```
primer_Número=0;
```

Las líneas de código deben finalizar siempre con punto y coma (;).

Cadenas de caracteres:

```
String mensaje = "Hola Mundo";
```

Para dar inicio a un programa en Java, lo primero que se debe realizar es crear una clase. Una clase es la representación lógica en Java de un objeto o cosa del mundo real. Ahora se observará un ejemplo del inicio del código:

```
public class Saludo {
}
```

public: clase de tipo público, lo que quiere decir que se podrá utilizar en otra de las clases.

class: palabra reservada para declarar las clases en Java.

Saludo: nombre de la clase, puede cambiar dependiendo de la solución del problema.

Para imprimir un mensaje en consola o en una pantalla de símbolo del sistema, es necesario conocer esta sentencia:

```
System.out.println("Hola mundo");
```

Para ingresar texto por teclado y por consola es necesario utilizar una serie de librerías. Es necesario utilizar la palabra reservada import. Esta sentencia realiza el llamado a las diferentes librerías que contiene Java. Se utilizan antes de iniciar la clase; por lo tanto, para realizar el ingreso de estos datos, se va a utilizar la Librería Scanner. A continuación un ejemplo:



```
import java.util.Scanner;
```

Esta Liberia se encuentra en la súper biblioteca útil. Cuando realice el llamado a esta Liberia, las diferentes operaciones asociadas a esta podrán funcionar correctamente.

Para ejecutar el programa, es necesario tener una función de ejecución principal. Para ello se utiliza la siguiente línea de código:

```
Public static void main(String args[]){
//Operaciones a ejecutar
}
```

Luego de conocer los principales elementos código para la elaboración de código fuente de Java se realizará un ejemplo sencillo, donde se si identificarán los conceptos aprendidos en esta unidad.

```
import java.util.Scanner;
```



```
public class Suma {

    public static void main(String args[] ) {

        //definir variables
        int a = 0;
        int b = 0;
        int resultado = 0;
        Scanner sc = new Scanner(System.in);
        System.out.println("Ingrese Numero a: ");
        a = sc.nextInt();
        System.out.println("Ingrese Numero b: ");
        b = sc.nextInt();
        resultado = a + b;
        System.out.println("El resultado de la suma es: " + resultado);
    }
}
```

Listado de palabras reservadas:

Abstract	boolean	Break	byte	Class
Case	continue	Catch	char	Double
Else	extends	default	do	finally
Float	for	False	final	instanceof
If	implements	import	native	new
Int	interface	Long	package	private
Null	public	Switch	return	short
Protected	super	True	synchronized	this
Satatic	transient	while	try	void
Throw	volatile			

Operadores lógicos:

Símbolo	Nombre	Descripción
&&	AND	Conector lógico "y"
	Or	Conector lógico "o"
!	Negacion	Niega la sentencia



Decisiones:

If: permite ejecutar la condición que contiene y ejecuta la operación si es verdadera

```
if(numeroa<numerob){
//operación
}
else: Se ejecuta si la condición es contraria a la establecida.
else{
//operación
}
```

A continuación un ejemplo de la utilización de esta sentencia:

```
if(numeroa<numerob){
System.out.println("El número a es menor");
}
else{
System.out.println("El número a es mayor");
}
```

Sentencia Switch:

Esta sentencia ayuda para los elementos de selección. Por ejemplo, para la elaboración de un menú de opciones, la forma de codificación de esta sentencia es de la siguiente forma:

```
switch (expresión){
case value1: //condicion;
break;
case value2: //condicion;
break;
default: //condicion;
}
Bucle for:
La forma de declaración del bucle for, es de la siguiente:
For(variable de control inicializada; variable a evaluar; incremento){
//operación
}
Ejemplo:
for(int i=0;i<5;i++){
System.out.println("Ingrese el número a: ");
a=sc.nextInt();
System.out.println("Ingrese el numero b: ");
b=sc.nextInt();
suma=a+b;
}
```

La anterior operación hace cinco veces la suma de los dos números. Teniendo presente los anteriores conceptos, ahora ya puede realizar código Java y analizar el mismo.

