

MisiónTIC 2022-Ciclo 2

Fundamentos de Programación

Jairo Armando Riaño Herrera

16 de julio de 2021



Contenidos

- 1 Introducción al Lenguaje JAVA y su Sintaxis
 - Generalidades de Java
 - Elementos Básicos
 - Variables
 - Operadores
 - Cadenas de Caracteres
- 2 Sentencias de Control de Flujo
 - Condicionales
 - Estructuras Iterativas
- 3 Vectores y Matrices
- 4 Referencias

Origen de Java

Java es un Lenguaje de Programación de propósito general, concurrente y orientado a objetos, fue desarrollado por *James Gosling* de *Sun Microsystems* en el año de 1995, la sintaxis del lenguaje es derivada del C/C++. Inicialmente se concibió para ser ejecutado como software empujado en electrodomésticos, automoviles, entre otros; por lo cual debía ser muy ligero. Con la llegada de la *World Wide Web*, Internet se convirtió en la plataforma que le dió impulsó a la evolución de Java como lenguaje para crear componentes multimedia para la web. En 2010 la empresa *Sun Microsystems* fue adquirida por *Oracle Corporation* por lo cual los productos pasaron a ser de propiedad de la nueva empresa (Java, MySQL, Solaris). La organización internacional TIOBE (<http://www.tiobe.com/>) ubica al Java como el Lenguaje de programación de mayor aceptación por las empresas desarrolladoras a nivel mundial[1]

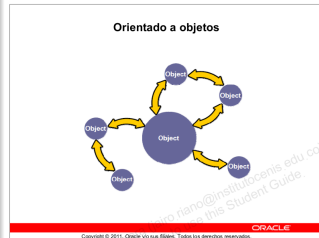
Características del Lenguaje

- Lenguaje de Propósito General
- Fuertemente tipado
- Imperativo
- Orientado a Objetos
- Permite resolver diferentes tipos de problemas
- Se pueden crear aplicaciones de consola, de escritorio, para la web y móviles

Características del Lenguaje

Orientado a Objetos

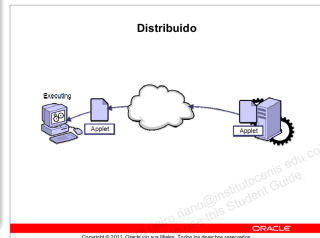
Java permite aplicar la técnica de Programación Orientada a Objetos (POO), en la cual una aplicación es un conjunto de objetos que tienen propiedades (atributos) y comportamiento (métodos), los cuales interactúan entre sí a través de paso de mensajes para cumplir con el propósito del programa. La POO permite aprovechar características como la ocultación de información (encapsulamiento), reutilización de código (herencia), usar una clase sin conocer su estructura interna ni su lógica de programación (abstracción), entre otras



Características del Lenguaje

Distribuido

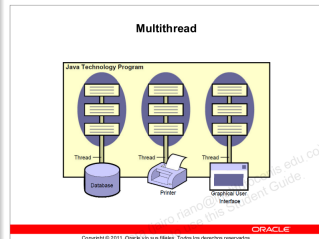
El lenguaje de programación Java proporciona soporte de tecnologías de redes distribuidas, a través de protocolos como *Remote Method Invocation* (RMI), *Common Object Request Broker Architecture* (CORBA) y *Uniform Resource Locator* (URL)



Características del Lenguaje

Multithread

En Java se puede realizar programas multiproceso (hilos), de manera que se ejecuten varias tareas en forma simultánea. Lo anterior permite que, por ejemplo, mientras se lee o escribe un archivo en disco, se pueda ejecutar un cálculo o se atienda una petición del usuario. Lo anterior hace los programas más robustos, de mejor calidad, que aprovechan eficientemente el hardware de la máquina evitando espacios de inactividad del procesador. El multiproceso o gestión de hilos es imprescindible en la programación de juegos.

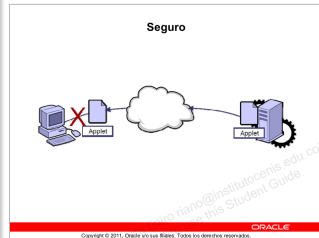


Características del Lenguaje

Seguro

Los programas en Java implementan mecanismos de seguridad, haciendo que sean seguros y protegiendo los programas de posibles ataques:

- Prohibición de que programas distribuidos como Applets escriban en la máquina local.
- Validación de que los programas de tecnología Java contienen código válido
- Soporte de firmas digitales
- Prohibición del uso de punteros para gestión de memoria.

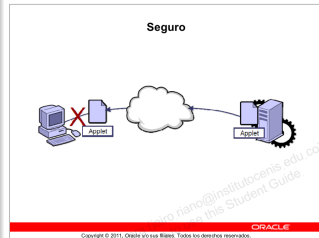


Características del Lenguaje

Seguro

Los programas en Java implementan mecanismos de seguridad, haciendo que sean seguros y protegiendo los programas de posibles ataques:

- Prohibición de que programas distribuidos como Applets escriban en la máquina local.
- Validación de que los programas de tecnología Java contienen código válido
- Soporte de firmas digitales
- Prohibición del uso de punteros para gestión de memoria.

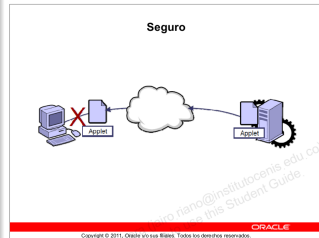


Características del Lenguaje

Seguro

Los programas en Java implementan mecanismos de seguridad, haciendo que sean seguros y protegiendo los programas de posibles ataques:

- Prohibición de que programas distribuidos como Applets escriban en la máquina local.
- Validación de que los programas de tecnología Java contienen código válido
- Soporte de firmas digitales
- Prohibición del uso de punteros para gestión de memoria.

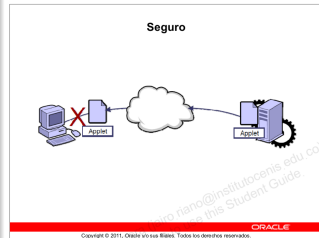


Características del Lenguaje

Seguro

Los programas en Java implementan mecanismos de seguridad, haciendo que sean seguros y protegiendo los programas de posibles ataques:

- Prohibición de que programas distribuidos como Applets escriban en la máquina local.
- Validación de que los programas de tecnología Java contienen código válido
- Soporte de firmas digitales
- Prohibición del uso de punteros para gestión de memoria.

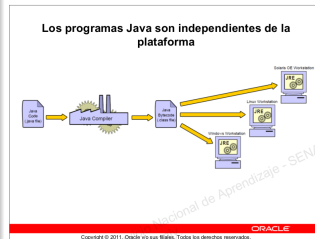


Características del Lenguaje

Independiente de la Plataforma

Un programa que se codifique en Java, se puede ejecutar sobre diferentes Sistemas Operativos sin necesidad de modificar el código fuente. Lo anterior es posible a través de la *Java Virtual Machine* (JVM) que se ejecuta sobre el Sistema Operativo, y es la encargada de interpretar el código compilado de Java (bytecode).

Para ejecutar un programa en Java, se debe tener instalada la JVM correspondiente para el Sistema Operativo sobre el cual se está trabajando.



Características del Lenguaje

Plataformas Java

- 1 *Java Micro Edition (JME)*. Versión reducida de la jerarquía de clases de Java, permite crear aplicaciones que se ejecutan en dispositivos móviles (teléfonos de baja gama) con recursos restringidos, además de otra gama de dispositivos electrónicos como Blu-Ray.
- 2 *Java Standar Edition (JSE)*. Versión estándar de Java, incluye los elementos básicos del lenguaje, así como las API y librerías estándar que permite la creación de aplicaciones de consola, de escritorio, applets y web.
- 3 *JavaFX*. Permite el desarrollo de interfaces gráficas, siguiendo estándares modernos de desarrollo de software usando diferentes componentes gráficos
- 4 *Java Enterprise Edition (Jakarta EE)*. Versión de Java, para la creación de aplicaciones empresariales de gran escala y alto rendimiento que se ejecutan en entornos distribuidos y Cliente/Servidor, con acceso a bases de datos a través de redes de computadores como Internet.

Características del Lenguaje

Plataformas Java

- 1 *Java Micro Edition (JME)*. Versión reducida de la jerarquía de clases de Java, permite crear aplicaciones que se ejecutan en dispositivos móviles (teléfonos de baja gama) con recursos restringidos, además de otra gama de dispositivos electrónicos como Blu-Ray.
- 2 *Java Standar Edition (JSE)*. Versión estándar de Java, incluye los elementos básicos del lenguaje, así como las API y librerías estándar que permite la creación de aplicaciones de consola, de escritorio, applets y web.
- 3 *JavaFX*. Permite el desarrollo de interfaces gráficas, siguiendo estándares modernos de desarrollo de software usando diferentes componentes gráficos
- 4 *Java Enterprise Edition (Jakarta EE)*. Versión de Java, para la creación de aplicaciones empresariales de gran escala y alto rendimiento que se ejecutan en entornos distribuidos y Cliente/Servidor, con acceso a bases de datos a través de redes de computadores como Internet.

Características del Lenguaje

Plataformas Java

- 1 *Java Micro Edition (JME)*. Versión reducida de la jerarquía de clases de Java, permite crear aplicaciones que se ejecutan en dispositivos móviles (teléfonos de baja gama) con recursos restringidos, además de otra gama de dispositivos electrónicos como Blu-Ray.
- 2 *Java Standar Edition (JSE)*. Versión estándar de Java, incluye los elementos básicos del lenguaje, así como las API y librerías estándar que permite la creación de aplicaciones de consola, de escritorio, applets y web.
- 3 *JavaFX*. Permite el desarrollo de interfaces gráficas, siguiendo estándares modernos de desarrollo de software usando diferentes componentes gráficos
- 4 *Java Enterprise Edition (Jakarta EE)*. Versión de Java, para la creación de aplicaciones empresariales de gran escala y alto rendimiento que se ejecutan en entornos distribuidos y Cliente/Servidor, con acceso a bases de datos a través de redes de computadores como Internet.

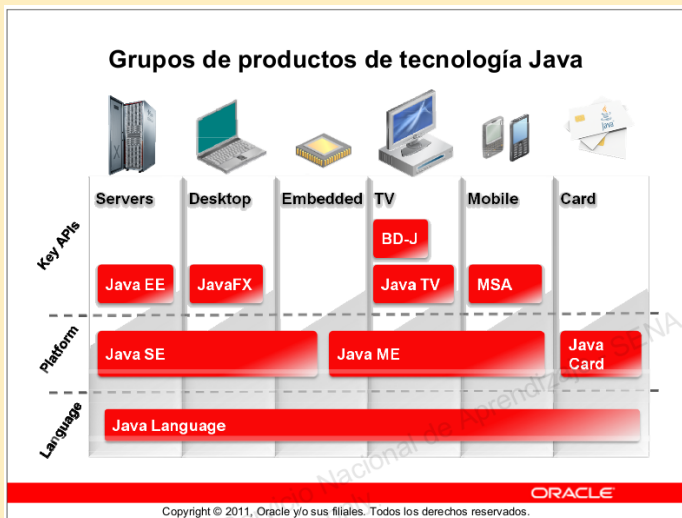
Características del Lenguaje

Plataformas Java

- 1 *Java Micro Edition (JME)*. Versión reducida de la jerarquía de clases de Java, permite crear aplicaciones que se ejecutan en dispositivos móviles (teléfonos de baja gama) con recursos restringidos, además de otra gama de dispositivos electrónicos como Blu-Ray.
- 2 *Java Standar Edition (JSE)*. Versión estándar de Java, incluye los elementos básicos del lenguaje, así como las API y librerías estándar que permite la creación de aplicaciones de consola, de escritorio, applets y web.
- 3 *JavaFX*. Permite el desarrollo de interfaces gráficas, siguiendo estándares modernos de desarrollo de software usando diferentes componentes gráficos
- 4 *Java Enterprise Edition (Jakarta EE)*. Versión de Java, para la creación de aplicaciones empresariales de gran escala y alto rendimiento que se ejecutan en entornos distribuidos y Cliente/Servidor, con acceso a bases de datos a través de redes de computadores como Internet.

Características del Lenguaje

Productos



Características del Lenguaje

Componentes

- *Java Runtime Environment - JRE* Entorno de ejecución de Java, incluye las herramientas y librerías necesarias para ejecutar un programa escrito en Java
- *Java Development Kit - JDK* Kit de desarrollo de Java, comprende las aplicaciones, herramientas y librerías necesarias para el desarrollo, compilación y empaquetado de aplicaciones. Incluye en su instalación el JRE. Es necesario que el JDK se encuentre instalado antes de proceder a instalar un IDE.
- *Java Virtual Machine - JVM* Máquina Virtual de Java, es una capa intermedia entre el Sistema Operativo y la aplicación en Java, la cual no se ejecuta directamente sobre el sistema operativo, sino que será necesario primero contar con la Máquina Virtual para su ejecución. Lo anterior permite que un programa escrito en Java se pueda ejecutar en cualquier Sistema Operativo que cuente con la JVM

Características del Lenguaje

Componentes

- *Java Runtime Environment - JRE* Entorno de ejecución de Java, incluye las herramientas y librerías necesarias para ejecutar un programa escrito en Java
- *Java Development Kit - JDK* Kit de desarrollo de Java, comprende las aplicaciones, herramientas y librerías necesarias para el desarrollo, compilación y empaquetado de aplicaciones. Incluye en su instalación el JRE. Es necesario que el JDK se encuentre instalado antes de proceder a instalar un IDE.
- *Java Virtual Machine - JVM* Máquina Virtual de Java, es una capa intermedia entre el Sistema Operativo y la aplicación en Java, la cual no se ejecuta directamente sobre el sistema operativo, sino que será necesario primero contar con la Máquina Virtual para su ejecución. Lo anterior permite que un programa escrito en Java se pueda ejecutar en cualquier Sistema Operativo que cuente con la JVM

Características del Lenguaje

Componentes

- *Java Runtime Environment - JRE* Entorno de ejecución de Java, incluye las herramientas y librerías necesarias para ejecutar un programa escrito en Java
- *Java Development Kit - JDK* Kit de desarrollo de Java, comprende las aplicaciones, herramientas y librerías necesarias para el desarrollo, compilación y empaquetado de aplicaciones. Incluye en su instalación el JRE. Es necesario que el JDK se encuentre instalado antes de proceder a instalar un IDE.
- *Java Virtual Machine - JVM* Máquina Virtual de Java, es una capa intermedia entre el Sistema Operativo y la aplicación en Java, la cual no se ejecuta directamente sobre el sistema operativo, sino que será necesario primero contar con la Máquina Virtual para su ejecución. Lo anterior permite que un programa escrito en Java se pueda ejecutar en cualquier Sistema Operativo que cuente con la JVM

Características del Lenguaje

Tipo de Aplicaciones Java

- *Applets* Son aplicaciones que se ejecutan en un navegador web, permiten dotar de dinamismo a las páginas web. Hoy en día no se usa suficientemente ya que ha sido reemplazado por otras tecnologías (JavaScript)
- *Aplicaciones de Consola.* Programas en los cuales el usuario interactúa con el programa usando la entrada y salida estándar (teclado y consola). No se recomienda para software funcional, se usa en la etapa de pruebas de las clases de la lógica del negocio.
- *Servlets.* Son aplicaciones que se ejecutan en un entorno web. Se requiere un servidor de aplicaciones que convierta las clases de Java a Servlets como Apache Tomcat o GlashFish

Características del Lenguaje

Tipo de Aplicaciones Java

- *Applets* Son aplicaciones que se ejecutan en un navegador web, permiten dotar de dinamismo a las páginas web. Hoy en día no se usa suficientemente ya que ha sido reemplazado por otras tecnologías (JavaScript)
- *Aplicaciones de Consola*. Programas en los cuales el usuario interactúa con el programa usando la entrada y salida estándar (teclado y consola). No se recomienda para software funcional, se usa en la etapa de pruebas de las clases de la lógica del negocio.
- *Servlets*. Son aplicaciones que se ejecutan en un entorno web. Se requiere un servidor de aplicaciones que convierta las clases de Java a Servlets como Apache Tomcat o GlashFish

Características del Lenguaje

Tipo de Aplicaciones Java

- *Applets* Son aplicaciones que se ejecutan en un navegador web, permiten dotar de dinamismo a las páginas web. Hoy en día no se usa suficientemente ya que ha sido reemplazado por otras tecnologías (JavaScript)
- *Aplicaciones de Consola*. Programas en los cuales el usuario interactúa con el programa usando la entrada y salida estándar (teclado y consola). No se recomienda para software funcional, se usa en la etapa de pruebas de las clases de la lógica del negocio.
- *Servlets*. Son aplicaciones que se ejecutan en un entorno web. Se requiere un servidor de aplicaciones que convierta las clases de Java a Servlets como Apache Tomcat o GlashFish

Características del Lenguaje

Tipo de Aplicaciones Java

- *Sitios web Dinámicos*. Es la evolución de los Servlets permitiendo usar código Java embebido con HTML para la presentación o interfaz de usuario, permite aplicar el Modelo Vista Controlador (MVC) con tecnologías Java. Dentro de las tecnología de Java para sitios dinámicos se puede usar *Java Server Pages* (JSP), *Java Server Faces* (JSF) entre otras. Se comunica con motores de bases de datos a través de *Java Database Connectivity* (JDBC)[2] o usando Frameworks de persistencia como *Hibernate*[3] o *Java Persistence API* (JPA)[4]

Características del Lenguaje

Entornos Servidor

Java en entornos de servidor



Java se suele usar en entornos de empresa:

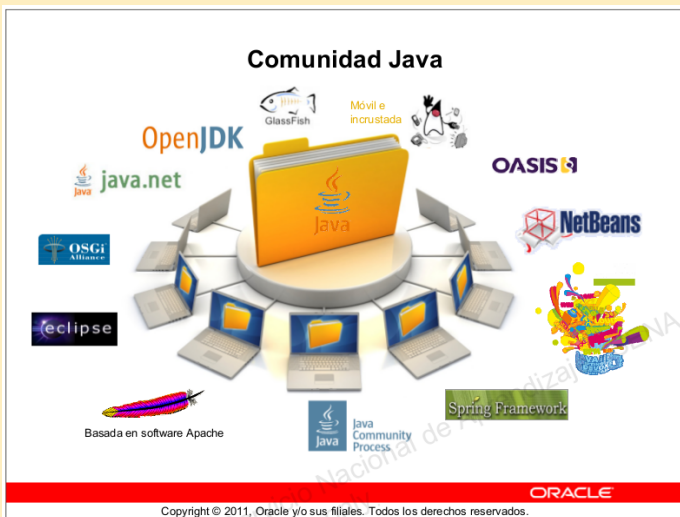
- Oracle Fusion Middleware
 - Servidores de aplicaciones Java
 - GlassFish
 - WebLogic
- Servidores de base de datos
 - MySQL
 - Oracle Database

ORACLE

Copyright © 2011, Oracle y/o sus filiales. Todos los derechos reservados.

Características del Lenguaje

Comunidad Java



Programa de Ejemplo

```
public class Example{  
    public static void main( String[] args ){  
        System.out.println("Hello World!!");  
    }  
}
```

Estructura de una clase

```
package <package_name>;

import <other_packages>;

public class ClassName {
    <variables(also known as fields)>;

    <constructor method(s)>;

    <other methods>;
}
```

Bloques de Código

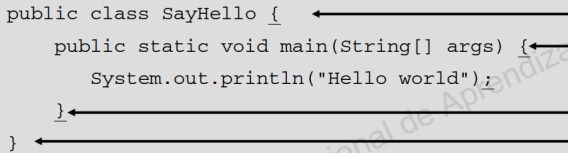
Bloques de código

- Todas las declaraciones de clase se incluyen en un bloque de código.
- Las declaraciones de métodos se incluyen en bloques de código.
- El ámbito de los campos y los métodos Java es el bloque (o la clase).
- Los bloques de códigos se definen entre corchetes:

```
{ }
```

- Ejemplo:

```
public class SayHello {  
    public static void main(String[] args) {  
        System.out.println("Hello world");  
    }  
}
```

**ORACLE**

Copyright © 2011, Oracle y/o sus filiales. Todos los derechos reservados.

Variables

Una variable es un espacio de memoria, identificado con un nombre y que puede contener datos o valores. En java es necesario definir el tipo de variable, lo cual determina el tipo de información que puede contener, el rango de valores y el espacio que ocupa en la memoria RAM. Antes de poder usar una variable es necesario definirla, asignándole un nombre y un tipo.

Tipos de Datos

Tipo	Tamaño	Rango	Literal (ejemplo)
byte	8 bit	[-128 , 127]	100
short	16 bit	[-32.768 , 32.767]	10000
int	32 bit	[-231 , -231-1]	100000
long	64 bit	[-263 , 263-1]	100000l ó 100000L
float	32 bit		4.25f ó 4.25F
double	64 bit		42.5 ó 42.5d ó 42.5D ó 4.25e1
boolean	1 bit	true / false	false
char	16 bit	['\u0000' , '\uffff']	'a'

Literales Numericas

Literales numéricas de Java SE 7

En Java SE 7 (y versiones posteriores), puede aparecer cualquier número de caracteres subrayados (_) entre dígitos en un campo numérico. Esto puede mejorar la lectura del código.

```
long creditCardNumber = 1234_5678_9012_3456L;
long socialSecurityNumber = 999_99_9999L;
float pi = 3.14_15F;
long hexBytes = 0xFF_EC_DE_5E;
long hexWords = 0xCAFE_BABE;
long maxLong = 0x7fff_ffff_ffff_ffffL;
byte nybbles = 0b0010_0101;
long bytes = 0b11010010_01101001_10010100_10010010;
```


Literales Binarios

Literales binarios de Java SE 7

En Java SE 7 (y versiones posteriores), los literales binarios también se pueden expresar con el sistema binario agregando los prefijos `0b` o `0B` al número:

```
// An 8-bit 'byte' value:  
byte aByte = (byte)0b00100001;  
  
// A 16-bit 'short' value:  
short aShort = (short)0b1010_0001_0100_0101;  
  
// Some 32-bit 'int' values:  
int anInt1 = 0b1010_0001_0100_0101_1010_0001_0100_0101;  
int anInt2 = 0b101;  
int anInt3 = 0B101; // The B can be upper or lower case.
```

Operadores

En un programa las instrucciones manipulan los valores de las variables para procesar los datos y obtener los resultados, a través de los operadores se realizan los procesos de actualización de los valores de las variables. El lenguaje de programación Java cuenta con un conjunto de operadores básicos, para determinar las acciones a realizar con las variables del programa. Los operadores usados son los aritméticos, relacionales (de comparación) y los lógicos.

Operadores	Precedencia
Postfix	expr++ expr--
Unario	++expr --expr +expr -expr ~!
Multiplicativo	* / %
Aditivo	+ -
Shift	<< >> >>>
Relacional (lógico)	< > <= >= instanceof
Igualdad (lógico)	== !=
Bitwise AND	&
Bitwise exclusive OR	^
Bitwise inclusive OR	
AND lógico	&&
OR lógico	
Ternario	? :
Asignación	= += -= *= /= %= &= ^= = <<= >>= >>>=

Cadenas de Caracteres

Una cadena de caracteres es un conjunto o secuencia de caracteres de cualquier tipo y que puede contener una palabra o una frase combinando letras y números. Java no cuenta con un tipo de dato primitivo para la gestión de cadenas de caracteres, pero implementa la clase *String* que contiene una variedad de métodos para la gestión de cadenas, las cuales son tratadas como objetos de la aplicación [5].

Sentencias if-then

Permite que con base en el resultado de la evaluación de una expresión lógica, que puede ser verdadero o falso, se ejecuten o no el conjunto de sentencias. Si el resultado es verdadero, se ejecutan las sentencias del bloque, y el flujo del programa continúa después del fin de la sentencia; en caso contrario, el flujo de programa continúa después del fin de la sentencia.

Sentencias if-then-else

Permite que con base en el resultado de la evaluación de una expresión lógica, que puede ser verdadero o falso, se ejecuten uno, o el otro conjunto de sentencias. Si el resultado es verdadero, se ejecutan las sentencias del primer bloque (if) y el flujo del programa continúa después del fin de la sentencia; en caso contrario, se ejecuta el segundo bloque (else) y el flujo de programa continúa después del fin de la sentencia.

Ciclo While

Se valida una expresión lógica, si el resultado del proceso de validación es verdadero, se ejecutan las instrucciones del bucle, si es falso, el flujo del programa continúa después del fin del ciclo. Luego de ejecutar la última instrucción del bucle, se vuelve a validar la expresión lógica. Es posible que el bucle no se ejecute ni una sola vez.

```
1 int cont = 1;
2 while ( cont <= 10 ){
3     System.out.println("El contador vale... " + cont );
4     cont++;
5 }
```

Ciclo Do..While

Se inicia ejecutando las instrucciones del ciclo iterativo, al finalizar la ejecución de la última instrucción se valida la expresión lógica, si el resultado es verdadero, vuelve a ejecutar las instrucciones del ciclo.; si el resultado es falso, el flujo del programa continúa después del fin del ciclo.

```
1 int cont = 1;
2 do{
3     System.out.println("El valor del contador es..." + cont );
4     cont++;
5 }while( cont <= 10 );
```

Ciclo For

La estructura de control incluye tres elementos:

- Inicialización Variable. Se asigna el valor a la variable que va a controlar la ejecución del ciclo. Si no existe la variable, se puede definir
- Expresión Lógica. Indica la operación relacional de la expresión
- Modificación variable. Indica la expresión aritmética que permite modificar la variable que controla la ejecución del ciclo

```
1 for( int cont = 1 ; cont <= 10 ; cont++ ){  
2     System.out.println("El Valor del contador es... " + cont)  
3 }
```

Las instrucciones del bucle se ejecutan mientras el resultado del proceso de validación de la Expresión Lógica sea verdadero, cuando es falso, finaliza la ejecución del ciclo y el control de flujo del programa continúa después del fin del ciclo

Interrumpir Ejecución Ciclo

Es posible interrumpir la secuencia de ejecución de un ciclo con las instrucciones **continue**, **break** y **exit**

- *continue*. Interumpe el flujo de ejecución del ciclo volviendo a validar la expresión lógica del ciclo iterativo
- *break*. Interrumpe el flujo de ejecución del ciclo y continúa después del fin del mismo.
- *exit*. Finaliza la ejecución del programa y retorna el control al Sistema Operativo

Los vectores y matrices son arreglos (arrays) o estructuras de datos del mismo tipo que ocupan posiciones seguidas en la memoria RAM. En Java es necesario definirlos y dimensionarlos antes de poderlos usar. Una vez dimensionadas, ocupan un espacio fijo en memoria. Para acceder a una posición del arreglo se hace a través de un índice que indica el número de elemento iniciando desde cero (0). Para los vectores se usa solo un índice, y para las matrices se debe indicar la fila y columna de la celda a la que se desea acceder. El acceso a un arreglo permite escribir o leer valores

Vectores

Un vector es un arreglo de una dimensión que contiene datos del mismo tipo. En el siguiente ejemplo se visualiza un vector de datos de valores enteros.

65	54	18	67	32
----	----	----	----	----

Ejemplo de Vector

En el siguiente código en Java se define, dimensiona, inicializa y visualiza los elementos de un vector de cinco elementos.

```
1 class Example{
2     public static void main(String[] args){
3         int vector[] = new int[5];
4         vector[0] = 65;
5         vector[1] = 54;
6         vector[2] = 18;
7         vector[3] = 67;
8         vector[4] = 32;
9
10        for (int i = 0 ; i < vector.length ; i++){
11            System.out.printf("vector [%d]=%d\n",i,vector[i]);
12        }
13    }
14 }
```

matrices

Una matriz es un arreglo de dos dimensiones, se accede a través de un par de índices que indican la fila y la columna. En el siguiente código de ejemplo se define e inicializa una matriz de tres filas y tres columnas

```

1 class Example{
2     public static void main(String[] args){
3         int matriz[][] = new int[][]{{8,1,6},{3,5,7},{4,9,2}};
4
5         for (int i = 0 ; i < matriz.length ; i++ ){
6             for( int j = 0 ; j < matriz[i].length; j++ ){
7                 System.out.printf("%d\t",matriz[i][j]);
8             }
9             System.out.println();
10        }
11    }
12 }

```

Si se intenta acceder fuera de los límites de un arreglo, el lenguaje Java lanzará una excepción, deteniendo la ejecución del programa

Referencias

- [1] http://www.tiobe.com/tiobe_index.
Tiobe index for may 2016.
[Web; accedido el 06-03-2016].
- [2] <http://www.oracle.com/technetwork/java/javase/jdbc/index.html>.
Java se technologies - database.
[Web; accedido el 06-07-2016].
- [3] <http://hibernate.org>.
Hibernate.
[Web; accedido el 06-07-2016].
- [4] https://es.wikipedia.org/wiki/Java_Persistence_API.
Java persistence api.
[Web; accedido el 06-07-2016].
- [5] <https://docs.oracle.com/javase/7/docs/api/java/lang/String.html>.
Class string.