* Resume en pocas palabras el origen de Java.

El lenguaje [Java](http://www.manualweb.net/tutorial-java/) fue desarrollado en sus inicios por [James Gosling](https://www.linkedin.com/in/jamesgosling/), en el año 1991. Inicialmente [Java](http://www.manualweb.net/tutorial-java/) era conocido como **Oak** o **Green**.

La primera versión del lenguaje [Java](http://www.manualweb.net/tutorial-java/) es publicada por [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems) en 1995. Y es en la versión del lenguaje JDK 1.0.2, cuando pasa a llamarse [Java](http://www.manualweb.net/tutorial-java/), corría el año 1996.

En las primeras versiones de [*Java*](http://www.manualweb.net/tutorial-java/) 1.1, 1.2 y 1.3 es en la que el lenguaje va tomando forma, con la inclusión de tecnologías como **JavaBeans, JDBC** para el acceso a base de datos, **RMI** para las invocaciones en remoto, **Collections** para la gestión de múltiples estructuras de datos o **AWT**para el desarrollo gráfico, entre otros.

* ¿Qué es Java y por qué es tan importante en nuestro medio informático?

Es un lenguaje de programación de alto nivel que soporta variedad de dispositivos, móviles, sistemas operativos, programas y la red, y ofrece seguridad, rapidez y es estable.

* Nombra algunas características de Java
* Es MULTIPLATAFORMA.
* Es SIMPLE
* Es ORIENTADO A OBJETOS
* Es DISTRIBUIDO
* Es ROBUSTO
* Es de ARQUITECTURA NEUTRAL
* Es SEGURO
* Es PORTABLE
* Es INTERPRETADO
* Nombra las versiones de Java
* JDK 1.0 (23 de enero de 1996)
* JDK 1.1 (19 de febrero de 1997)
* J2SE 1.2 (8 de diciembre de 1998)
* J2SE 1.3 (8 de mayo de 2000)
* J2SE 1.4 (6 de febrero de 2002)
* J2SE 5.0 (30 de septiembre de 2004)
* Java SE 6 (11 de diciembre de 2006)
* Java SE 7(Julio de 2011)
* Java SE 8 (marzo de 2014)
* Java SE 9 (21 de septiembre del 2017)
* Java SE 10 (20 de marzo de 2018)
* Java SE 11 (25 de septiembre de 2018)
* Java SE 12 (19 de marzo de 2019)
* Qué significa JDK?

Java Development Kit (JDK) es un software que provee herramientas de desarrollo para la creación de programas en Java. Puede instalarse en una computadora local o en una unidad de red.

* Defina: Clase, objeto, método, atributo.

**Clase:** Las clases son la base de la Programación Orientada a Objetos. Una clase es una plantilla que define la forma de un objeto; en ella se agrupan datos y métodos que operarán sobre esos datos. En java, una clase se define con la palabra reservada class.

**Objeto:** Es un elemento declarado de un tipo de clase. Se conoce también como una instancia de clase.

**Método:** es un conjunto de instrucciones definidas dentro de una clase, que realizan una determinada tarea y a las que podemos invocar mediante un nombre.

**Atributo:** Son las características individuales que diferencian un objeto de otro y determinan su apariencia, estado u otras cualidades. Los atributos se guardan en variables denominadas de instancia, y cada objeto particular puede tener valores distintos para estas variables.

* Qué utilidad tienen las clases en Java?

El uso de clases permite proteger sus variables y métodos del acceso de otros objetos. Java proporciona cuatro niveles de acceso para las variables y métodos miembros: private, protected, public y acceso de paquete.

* Cuáles son los editores de trabajo en java?
* Eclipse
* IntelliJIDEA
* NetBeans
* Oracle JDeveloper
* Sintaxis en Java para las siguientes expresiones:

**Variables**

public static void main(String[] args) {

int n;

float a;

double x;

String x;

Char c;

}

**Constantes**

public class ConstanteEnJava {

static final int DIAS\_SEMANA = 7;

static final int DIAS\_LABORABLES = 5;

public static void main(String[] args) {

System.out.println("El número de días de la semana son " + DIAS\_SEMANA);

System.out.println("El número de días laborables de la semana son " + DIAS\_LABORABLES);

}

}

**Operador Aritmético**

suma = 3 + 7;

resta = 5 - 2;

multiplicación = 3 \* 2;

división = 4 / 2;

resto = 5 % 3;

**Operador Relacional**

== igual a

! = no igual a

> mayor que

>= mayor o igual que

< menor que

<= menor o igual que

int vble1 = 5;

int vble2 = 3;

if (vble1 == vble2)

System.out.println("Las variables son iguales");

if (vble1 != vble2)

System.out.println("Las variables son distintas");

int a= 3, b= 3;

if (a >= b && a<=b) {

// Si «a» es mayor o igual a «b» y «a» es menor o igual a «b» ejecuta la siguiente sentencia.

System.out.println (« La condición es true. «); // sentencia que SI se ejecutaría al ser las dos variables iguales.

} else {// Si no se cumple la condición, ejecuta la siguiente sentencia.

System.out.println («La condición es false. «); // sentencia que NO se ejecutaría.

}

**If**

if (edad > 25) {

System.out.println("Puede ingresar al Bar");

}

**If – else**

if (edad > 25) {

System.out.println("Puede ingresar al Bar");

} else {

System.out.println("No puede entrar");

}

**If – else –if**

if (temperatura > 25) {

System.out.println("A la playa!!!");

} else if (temperatura > 15) {

System.out.println("A la montaña!!!");

} else if (temperatura < 5 && nevando) {

System.out.println("A esquiar!!!");

} else {

System.out.println("A descansar... zZz");

}

**If anidada**

if (temperatura > 15) {

if (temperatura > 25) {

System.out.println("A la playa!!!");

} else {

System.out.println("A la montaña!!!");

}

} else if (temperatura < 5) {

if (nevando) {

System.out.println("A esquiar!!!");

}

} else {

System.out.println("A descansar... zZz");

}

**Switch case**

switch(i)

{

case 1:

System.out.println("Case1 ");

break;

case 2:

System.out.println("Case2 ");

break;

case 3:

System.out.println("Case3 ");

break;

case 4:

System.out.println("Case4 ");

break;

default:

System.out.println("Default ");

}

**Switch case Simple**

int num=2;

switch(num+2)

{

case 1:

System.out.println("Case1: Value is: "+num);

case 2:

System.out.println("Case2: Value is: "+num);

case 3:

System.out.println("Case3: Value is: "+num);

default:

System.out.println("Default: Value is: "+num);

}

**Switch case anidado**

String nombre = "Miguel";

switch(nombre) {

case "Paco":

System.out.println("Habitación 1");

break;

case "Mig" + "uel":

System.out.println("Habitación 2");

break;

case null:

System.out.println("Habitación 3");

break;

default:

System.out.println("Habitación 4");

}

**ArrayList**

List lista2 = new ArrayList();

// Añadimos nodos y creamos un Iterator

lista2.add("Madrid");

lista2.add("Sevilla");

lista2.add("Valencia");

Iterator iterador2 = lista2.iterator();

// Recorremos y mostramos la lista

while (iterador2.hasNext()) {

String elemento = (String) iterador2.next();

System.out.print(elemento + " ");

}

System.out.println("--ArrayList--");