

**MASTER EN MODELIZACIÓN
MATEMÁTICA, ESTADÍSTICA Y
COMPUTACIÓN
2013-2014**

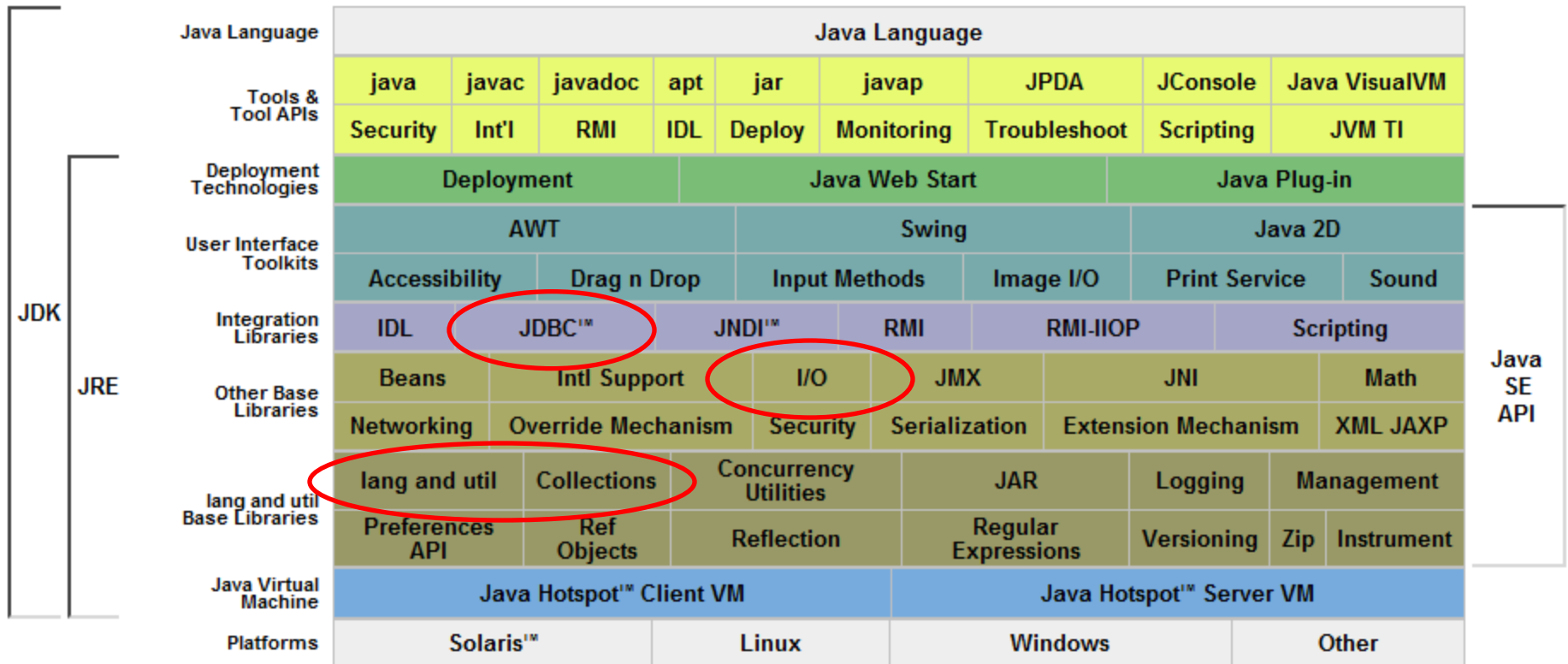
Curso: Bases de datos y programación
orientada a objetos
Parte POO

**Librerías, estructura disponibilidad y uso.
Clases básicas
Entrada / salida (I/O)**

**El navegador como capa de cliente (HTML, javascript) y la capa de presentación(JSPs,...)
Acceso a bases de datos (MySQL)**

Entorno de desarrollo de Java

Imagen global del marco de trabajo Java



Overview Package **Class** Use Tree Deprecated Index Help

Prev Class Next Class Frames No Frames

Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method

java.lang

Class Object

java.lang.Object

```
public class Object
```

Class `Object` is the root of the class hierarchy. Every class has `Object` as a superclass. All objects, including arrays, implement the methods of this class.

Since:

JDK1.0

See Also:

[Class](#)

Constructor Summary

Constructors

Constructor and Description
<code>Object()</code>

Method Summary

Methods

Modifier and Type	Method and Description
protected <code>Object</code>	<code>clone()</code> Creates and returns a copy of this object.
boolean	<code>equals(Object obj)</code> Indicates whether some other object is "equal to" this one.
protected void	<code>finalize()</code> Called by the garbage collector on an object when garbage collection determines that there are no more references to the object.
<code>Class<T></code>	<code>getClass()</code>

Clases básicas

java.lang.Object

- Clone
- Equals
- toString
- hashCode
- Finalize
- getClass

java.lang.System

- getenv
- getProperty, getProperties, setProperty, setProperties y clearProperty
- Exit
- Arraycopy
- setIn, setOut y setErr

java.lang.<objetos asociados a tipos>

java.lang.String (operador +, main); java.lang.StringBuffer

java.lang.Math; java.lang.StrictMath;

java.lang.Throwable

(otras) Thread, Process, SecurityManager,ClassLoader,Compiler, Runtime,etc

java.lang

•Java.util.Date; java.util.Calendar;

•Java.util.BitSet

•Java.util.Random

•Java.util.Timer; java.util.TimerTask

•Java.util.Properties

•Java.util.ResourceBundle

Java.util.Scanner; java.util.Formatter

FRAMEWORK COLECCIONES

- Interfaz List y clases Vector, Stack, ArrayList y LinkedList

- Interfaces Map y SortedMap, y clases Hashtable, HashMap, linkedHashMap y TreeMap

- Interfaces Set y SortedSet, y clases HashSet, LinkedHashSet y TreeSet

- El interfaz Queue y la clase PriorityQueue.

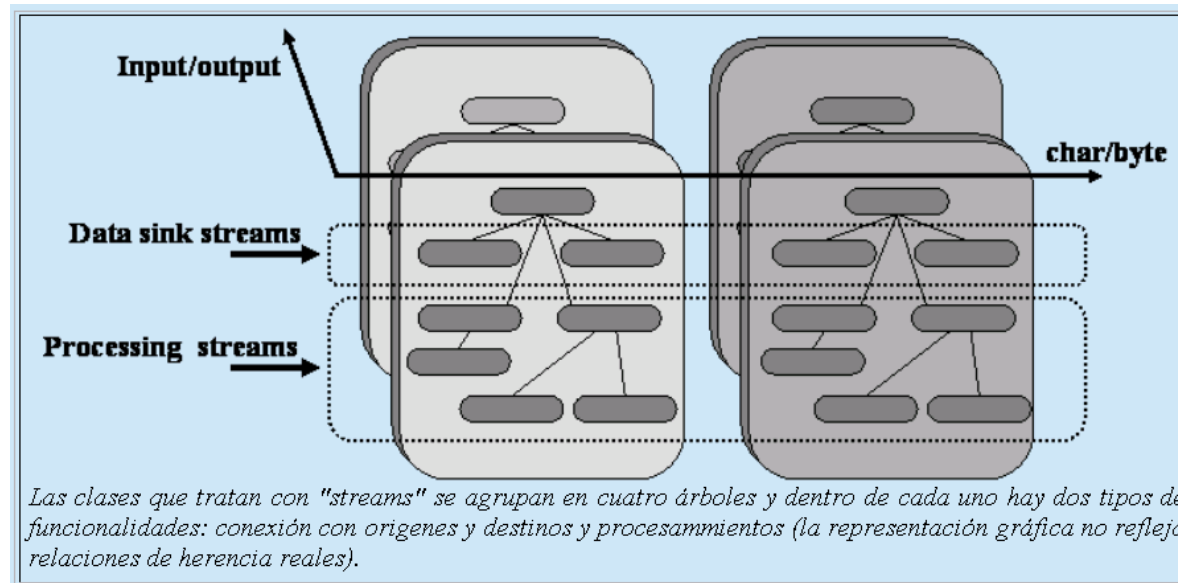
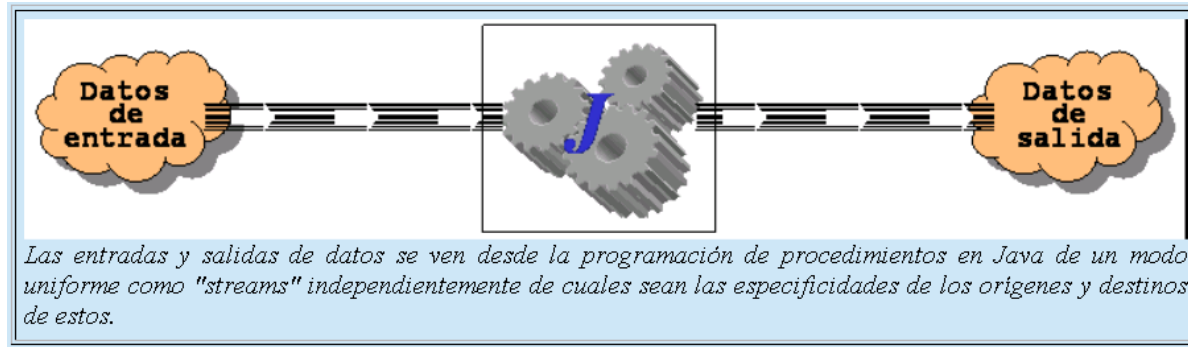
- El interfaz Comparator.

- Arrays (search sort)

EL SUBPAQUETE java.util.zip

java.util

Clases básicas: entrada/salida



**Reader
Writer**

**InputStream
OutputStream**

Origen/Destino	Streams de caracteres	Streams de Bytes
Memoria	CharArrayReader CharArrayWriter	ByteArrayInputStream ByteArrayOutputStream
	StringReader StringWriter	StringBufferInputStream
“Pipes”	PipedReader PipedWriter	PipedInputStream PipedOutputStream
Ficheros	FileReader FileWriter	FileInputStream FileOutputStream

Clases de entrada y salida de los orígenes y destinos básicos.

Procesamientos	Streams de caracteres	Streams de Bytes
Conversión de bytes a caracteres	InputStreamReader OutputStreamWriter	
Buffering	BufferedReader BufferedWriter	BufferedInputStream BufferedOutputStream
Filtrado	FilterReader FilterWriter	FilterInputStream FilterOutputStream
Concatenación		SequenceInputStream
Conversión de datos		DataInputStream DataOutputStream
Conteo	LineNumberReader	LineNumberInputStream
Peeking Ahead	PushbackReader	PushbackInputStream
Impresión	PrintWriter	PrintStream
Serialización de objetos		ObjectInputStream ObjectOutputStream

Clases para entradas y salidas con procesamiento de datos.

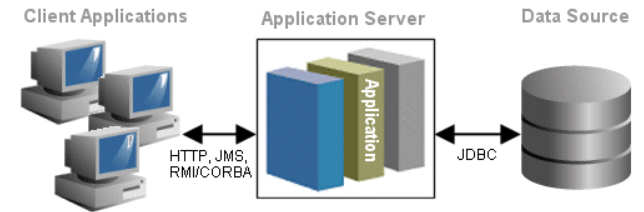
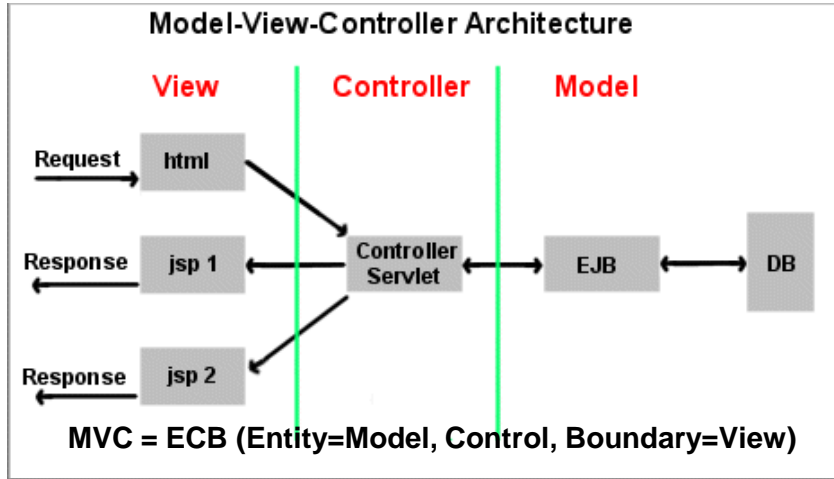
```
1- InputStreamReader entrada=new InputStreamReader(System.in);
2- BufferedReader entradaB=new BufferedReader(entrada);
3- //....
4- String s=entradaB.readLine();
```

```
1- BufferedReader entradaB=new BufferedReader(new InputStreamReader(System.in));
2- //....
3- String s=entradaB.readLine();
```

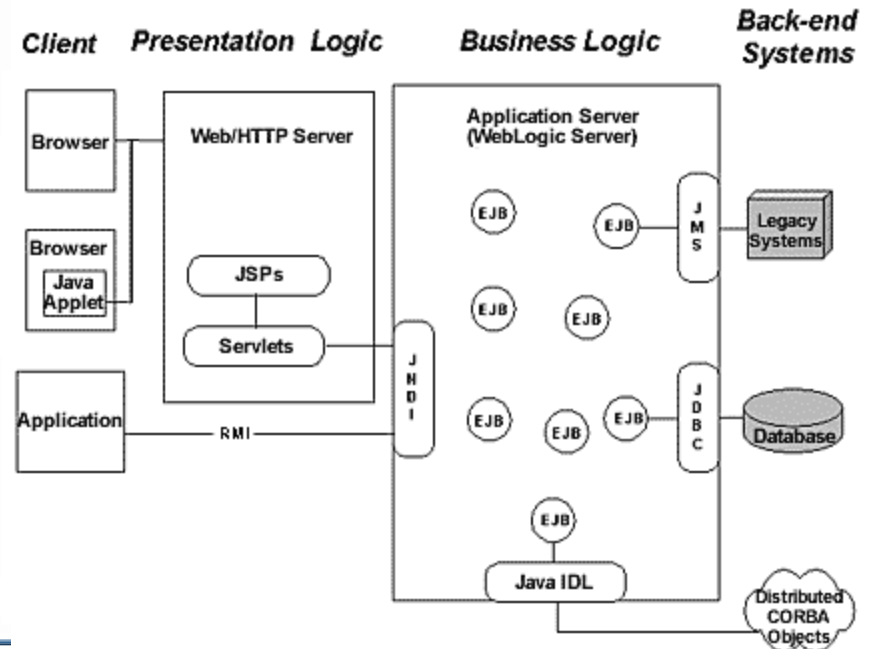
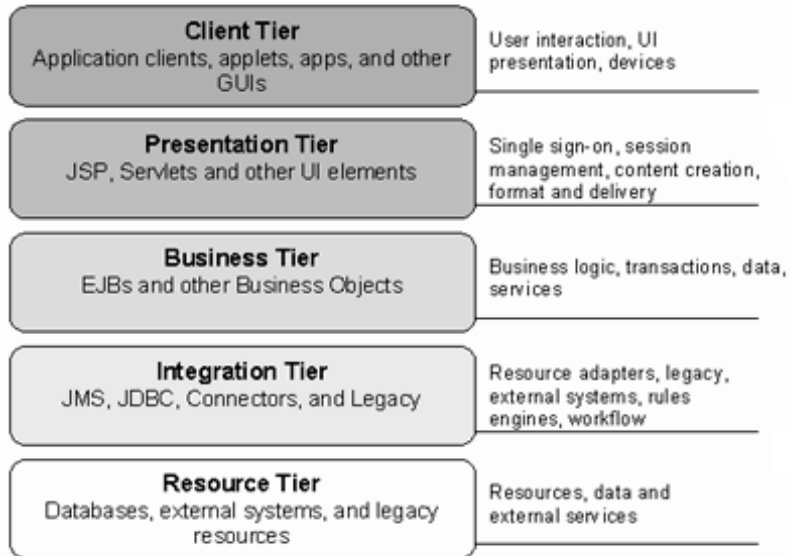
Comentarios sobre

- java.io.File
- java.io.StreamTokenizer
- java.io.RandomAccessFile

Estructura de una aplicación: capas (tiers)



Five Tier Model for logical separation of concerns



Tecnologías necesarias...

HTML + JavaScript : (HiperText Markup Language)

Ver curso páginas web

http://gtts.ehu.es/German/Docencia/Curso_paginas_web/

Servlets/JSPs (Java Server Pages)

Ver curso en la web

<http://www.jsptut.com/>

Tecnologías avanzadas...

AJAX (Asynchronous JavaScript And XML)

Frameworks...

WebServices...

Acceso a bases de datos (MySQL)

(1) Disponer de la base de datos. El modo más general de acceder consiste en configurarla como un servicio en un determinado puerto (si reside en la misma máquina que la aplicación puede accederse también igualmente a través de "pipes").

(2) Establecer la conexión desde nuestra aplicación con la base de datos, cosa que se hará a través de un driver específico para la base concreta que usemos. (si pensamos en MySQL, el driver lo encontramos en la librería Connector/J)

El driver se cargará mediante su nombre (no se instancia ningún objeto, por lo que el ClassLoader ha de "buscarlo" en las librerías disponibles. Nos servimos para ello del método "forName" de la clase "Class"). Una vez que se carga, queda a disposición del "DriverManager", a quién le podemos pedir la conexión mediante una especificación de la URL de la base de datos, del usuario y la clave de acceso.

```
try {
    import java.sql.*
    Class.forName("com.mysql.jdbc.Driver").newInstance();
    Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/DataBase","usuario", "clave");
    //Continúa la aplicación
}
catch(SQLException ex) {
    System.err.println("Problema de conexión con la base de datos: "+ex.getMessage());
}
catch(ClassNotFoundException ex) {
    System.err.println("No se encuentra el driver JDBC: "+ex.getMessage());
}
catch(InstantiationException ex) {
    System.err.println("No puede instanciarse el driver JDBC: "+ex.getMessage());
}
catch(IllegalAccessException ex) {
    System.err.println("Intento de acceso ilegal al driver JDBC: "+ex.getMessage());
}
}
```

Y luego...

```
ResultSet rs = con.createStatement().executeQuery("select * from autores");
```