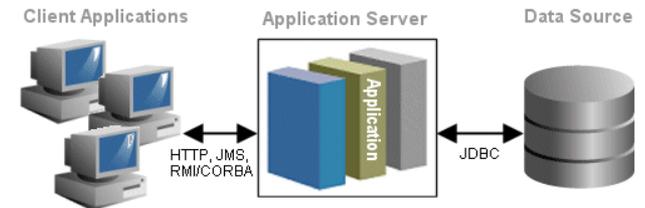
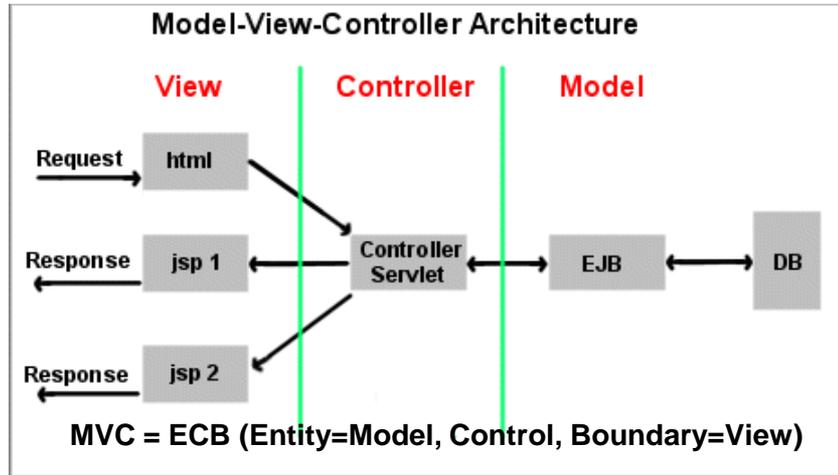


**MASTER EN MODELIZACIÓN
MATEMÁTICA, ESTADÍSTICA Y
COMPUTACIÓN
2015-2016**

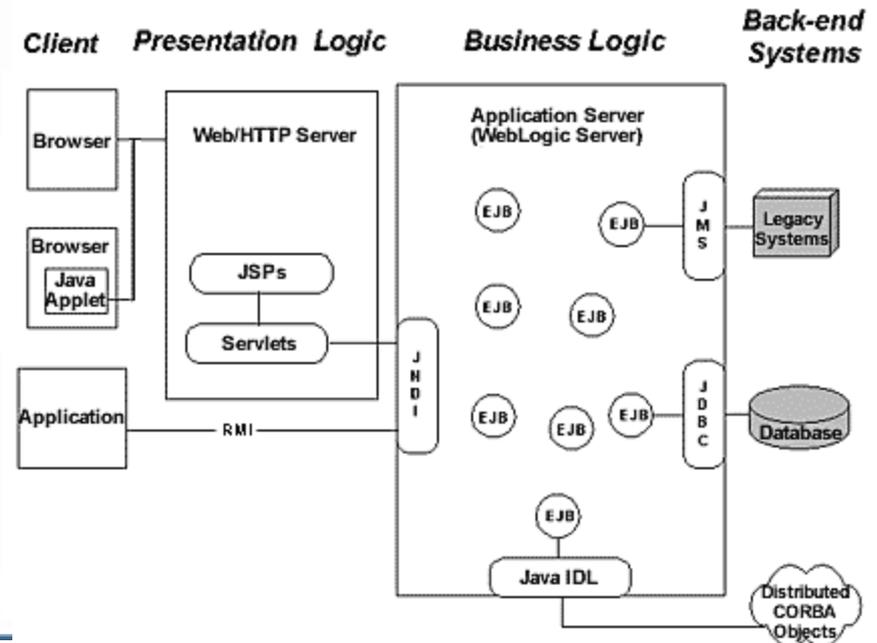
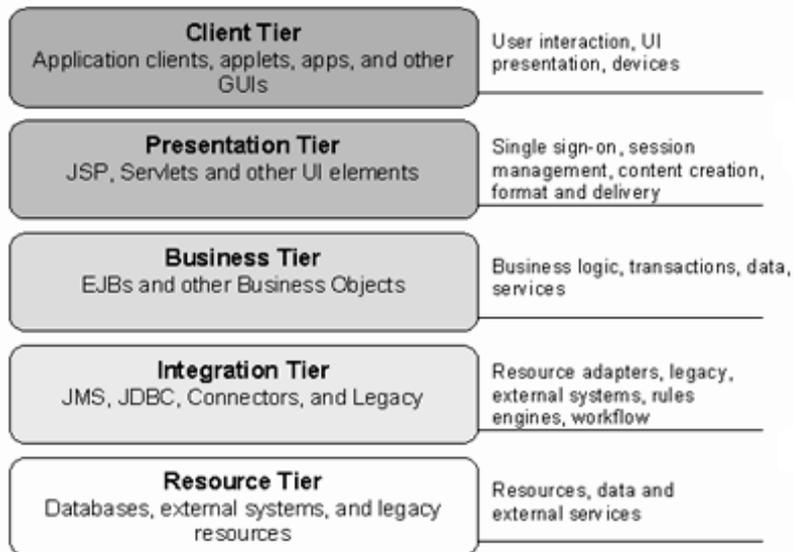
Curso: Bases de datos y programación
orientada a objetos
Parte POO

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

Estructura de una aplicación: capas (tiers)



Five Tier Model for logical separation of concerns



Tecnologías necesarias...

HTML + JavaScript : (HiperText Markup Language)

Servlets/JSPs (Java Server Pages)

Ver curso en la web

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

Tecnologías avanzadas...

AJAX (Asynchronous JavaScript And XML)

Frameworks...

WebServices...

(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.

Acceso a bases de datos (MySQL)

```
import java.sql.*
```

```
-----  
try {  
    Class.forName("com.mysql.jdbc.Driver").newInstance();  
    Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/DataBase","usuario", "clave");
```

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

```
        Int n=con.createStatement().executeUpdate(".../*insert, update, delete,...*/");
```

```
        ---
```

```
    }  
    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());  
    }  
}
```