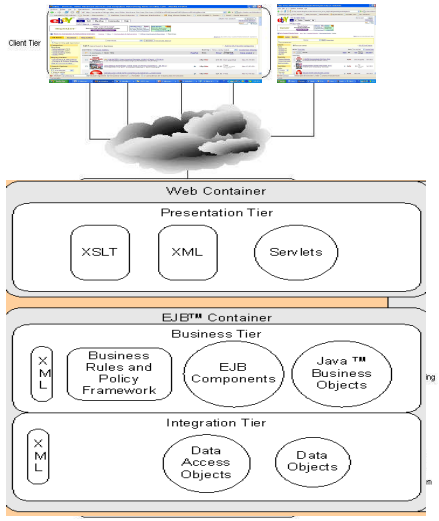
Tema 6: Arquitectura Web Multicapa

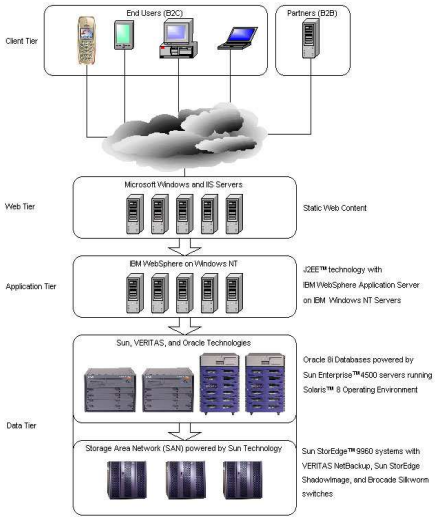
Arquitectura software en capas



(Dibujo de arquitectura de Ebay)

- **Capa Presentación:**
  - genera pantallas,
  - crea código formato.
- **Capa Lógica negocio:** objetos aplicación que generan, manipulan y organizan infomación.
- **Capa Datos:** almacenes de datos: ficheros o bbdd.

Arquitectura sistema en capas



(Dibujo de architectura de Ebay)

- **Capa Cliente:**
  - Cliente: genera pantallas: código HTML.
  - Servidor: crea código HTML.
- **Capa aplicación:**
  - Servidores que ejecutan aplicaciones: accede a datos, crea campos para HTML.
- **Capa Datos:**
  - Servidores que almacenan datos y ficheros.

Arquitecturas Web Multicapa

- Arquitectura 2 capas (Web + aplicación):
  - Apache + TOMCAT.
- Arquitecturas 3 capas(Web + aplicación +datos):
  - Datos en ficheros (Bookstore).
  - Datos en DDBB (BookStore con BBDD, LAMP).
- Arquitecturas 4 capas (Web + Servlets + Beans + datos):
  - J2EE (Ebay,….)

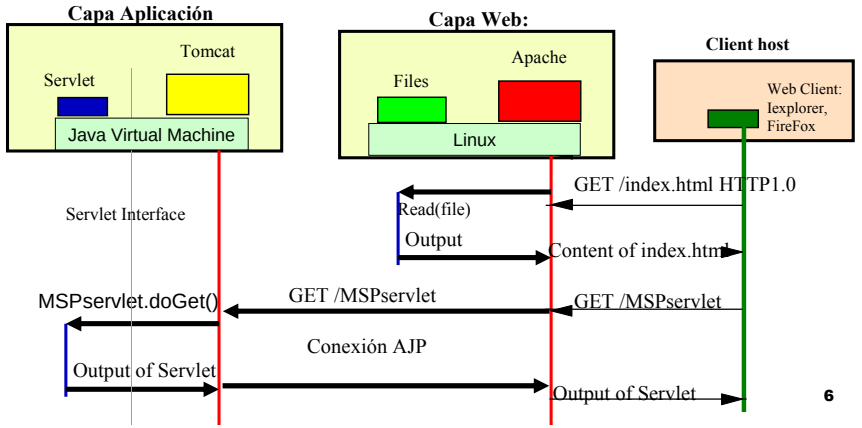
Apache vs. servidor Java (rep)

	Incovenientes	Ventajas
Apache	■Acceso a aplicaciones no eficiente (CGI) o no robusto (API)	■Acceso a ficheros eficiente. ■protocolo seguro HTTPS eficiente.
Servidor JAVA	■Lectura ficheros no eficiente. ■Protocolo HTTPS no eficiente.	■Aplicaciones JAVA. ■Aplicaciones multihilo e integradas en servidor-> bastante eficiente.

2 capas: Apache + TOMCAT (I)

Capa Aplicación:
•Sirve contenido dinámico

Capa Web:
•Sirve contenido estático
•Control de Acceso y HTTPS



2 capas: Apache + TOMCAT (II)

Configuración TOMCAT

fichero server.xml:

```
<Connector className=
"org.apache.jk.server.JkCoyoteHandler"
 port="8009" minProcessors="5“
maxProcessors="75” protocol=”AJP/1.3”>
```

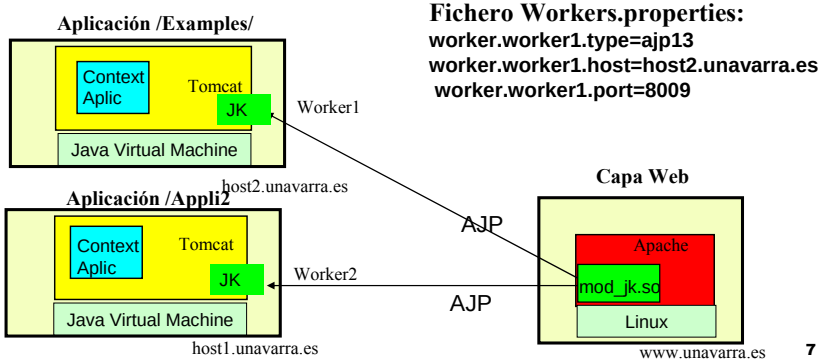
Configuración Apache

Fichero httpd.conf:

```
LoadModule jk_module libexec/mod_jk.so
AddModule mod_jk.c
JkWorkersFile .conf/workers.properties
JkMount /examples/* worker1
```

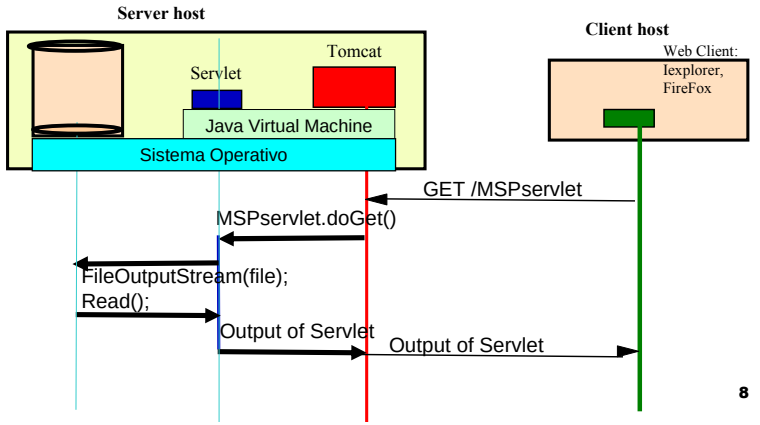
Fichero Workers.properties:

```
worker.worker1.type=ajp13
worker.worker1.host=host2.unavarra.es
 worker.worker1.port=8009
```

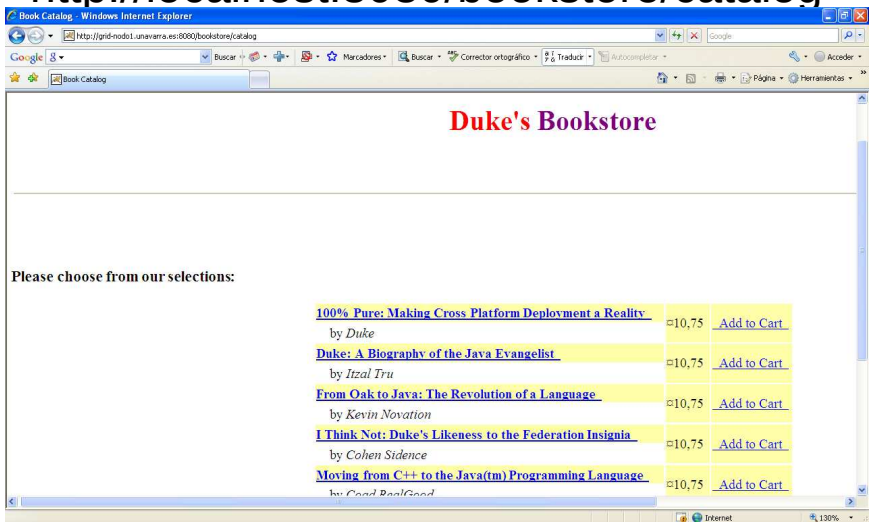


Capa de Datos, sin BBDD

- Datos en ficheros y variables.
- Aplicación lee ficheros a través de SO.



Aplicación Web en Java: ejemplo http://localhost:8080/bookstore/catalog



Objetos acceso Datos, sin BBDD

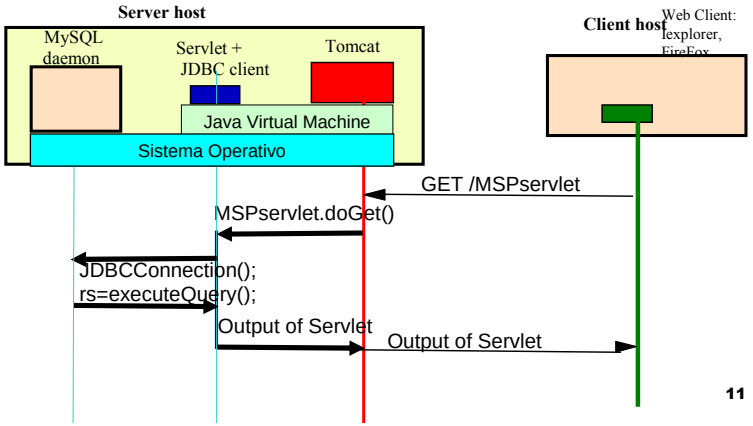
```
public class CatalogServlet extends HttpServlet {
    private BookDB bookDB;
  public void init() throws ServletException {
     bookDB =     (BookDB)getServletContext().getAttribute("examples.bookstore.database");
     if (bookDB == null) {
                                bookDB = BookDB.instance();
                              getServletContext().setAttribute("examples.bookstore.database", bookDB);
     }
  public void doGet (HttpServletRequest request,HttpServletResponse response)
  {.....
out.println("<br>  " + "<h3>Please choose from our selections:</h3>" +  "<center> <table>");
        Collection c = bookDB.getBooks();
        Iterator i = c.iterator();
        while (i.hasNext()) {
            BookDetails book = (BookDetails)i.next();
            bookId = book.getBookId();
              out.println("<tr>" + ….
  }
}
```

Ejemplo BookStore

```
public class BookDB {
  private HashMap database;

   public BookDB () {
      BookDetails book;
      database = new HashMap();
      book = new BookDetails("201", "Duke", "", "My Early Years:
                Growing up on *7*,(float)10.75, 1995, "What a book.");
      database.put(new Integer(201), book);
      }

public Collection getBooks() {
                ArrayList al = new ArrayList(database.values());
                return Collections.sort(al);

}
```

Capa de Datos, con BBDD

- Datos en BBDD.
- Aplicación hace queries a través de librería (JDBC, DAO,..)



Objetos acceso Datos, con BBDD

```
public class CatalogServlet extends HttpServlet {
    private BookDB bookDB;
  public void init() throws ServletException {
     bookDB =     (BookDB)getServletContext().getAttribute("examples.bookstore.database");
     if (bookDB == null) {
                                bookDB = BookDB.instance();
                              getServletContext().setAttribute("examples.bookstore.database", bookDB);
     }
  public void doGet (HttpServletRequest request,HttpServletResponse response)
  {.....
out.println("<br>  " + "<h3>Please choose from our selections:</h3>" +  "<center> <table>");
        Collection c = bookDB.getBooks();
        Iterator i = c.iterator();
        while (i.hasNext()) {
            BookDetails book = (BookDetails)i.next();
            bookId = book.getBookId();
              out.println("<tr>" + ….
  }
}
```

Ejemplo BookStore Avanzado

```
public class BookDB {
private EntityManager em;

public BookDB () {
       JDBCConnection();

}
public List getBooks() throws BooksNotFoundException {
       try {
          return rs=executeQuery("SELECT bd FROM Book bd
                ORDER BY bd.bookId").getResultList();
       }
```

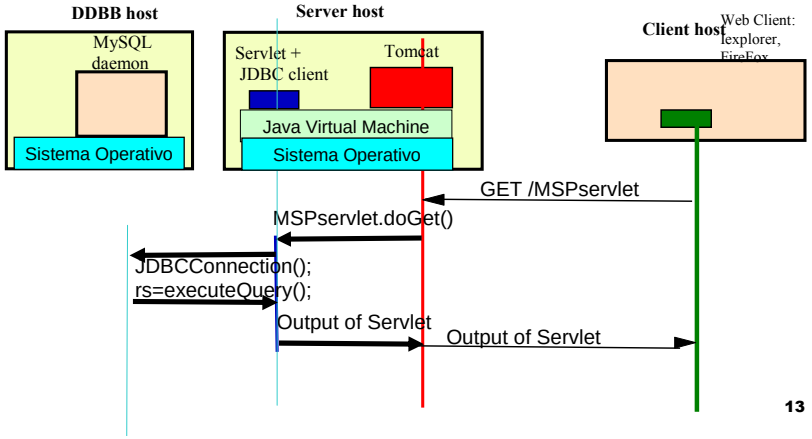
CLASE ACCESO DATOS CON MISMO INTERFACE,

Implementación usa JDBC.

Datos en BBDD.

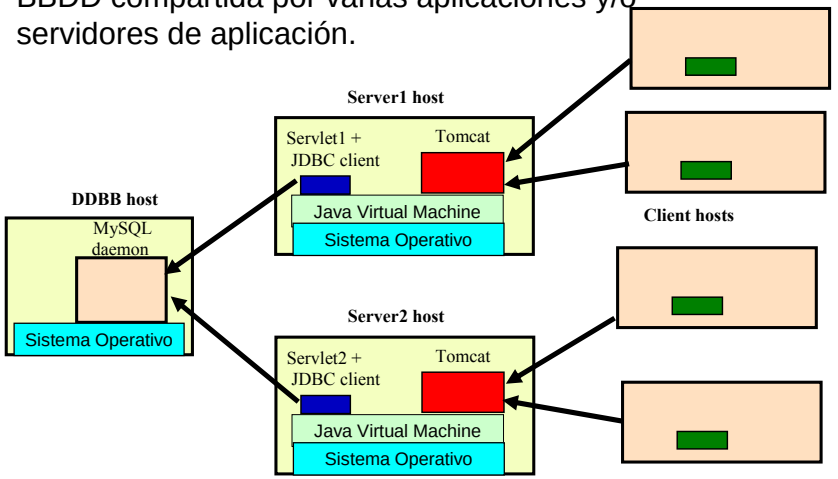
Servidores de Datos: BBDD (I)

- Arquitectura 3 capas: Web + aplicación + Datos.



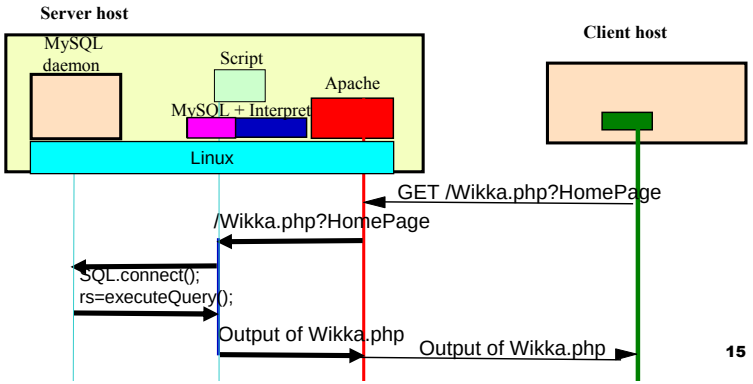
Servidores de Datos: BBDD (II)

- BBDD compartida por varias aplicaciones y/o servidores de aplicación.



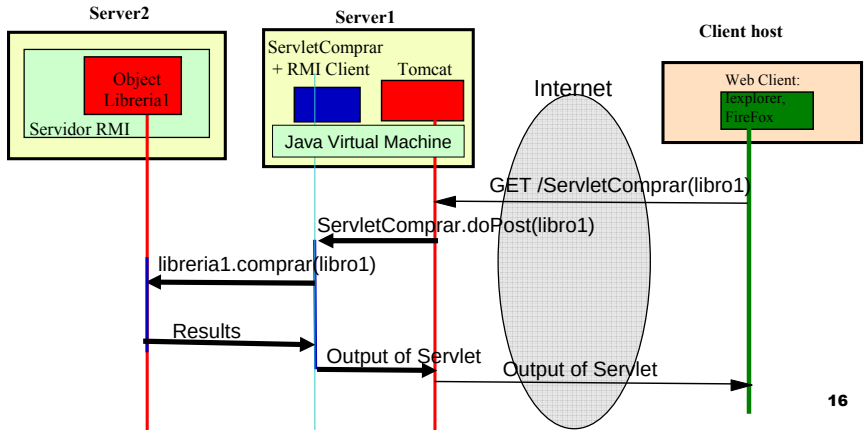
Plataforma LAMP.

- Linux + Apache + Perl + MySQL:
  - Mod_perl (API) or perl.exe (CGI).
  - MySQL library or client.



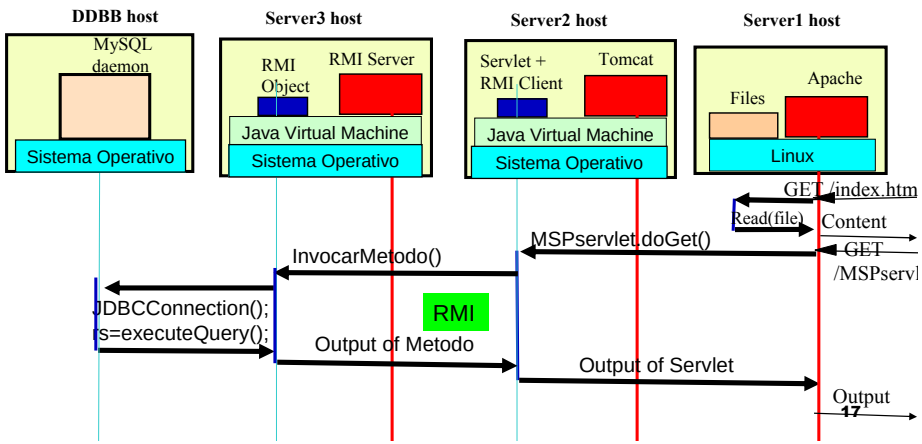
RMI en Aplicaciones Servidor

- Servlets: genera pagina web + cliente RMI.
- Servidor RMI: guarda objetos.



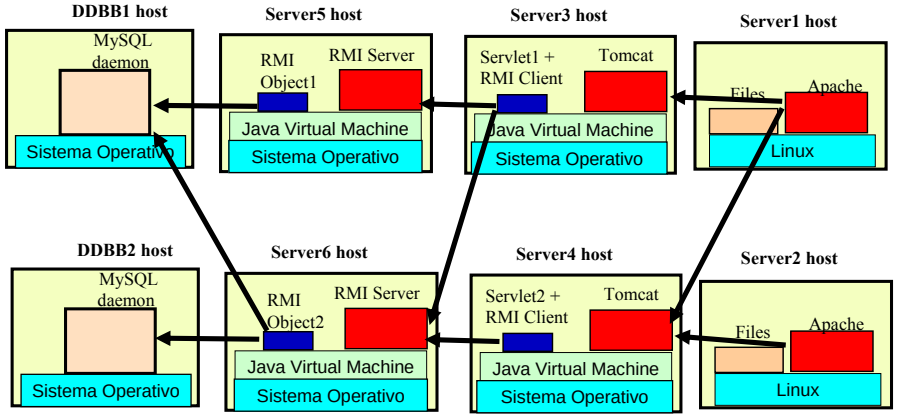
Arquitectura de 4 capas (I)

Datos + Lógica Negocio + Web Dinámico+ Web Estático



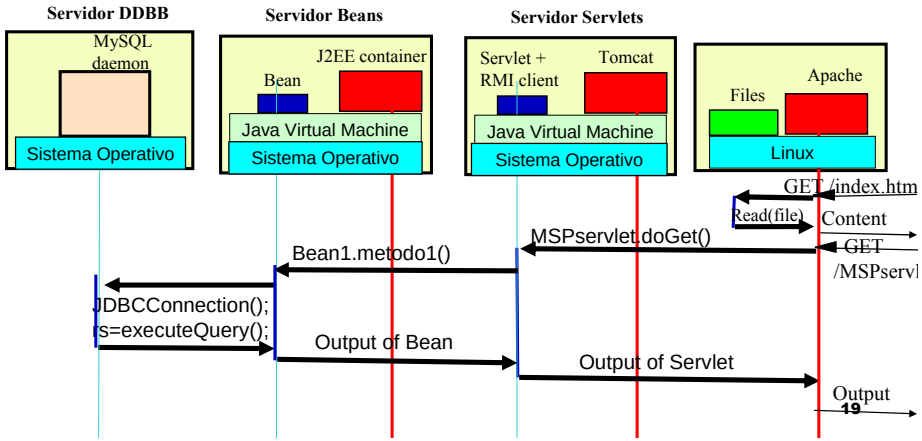
Arquitectura de 4 capas (II)

- Lógica Negocio, Web dinámica distribuida en servidores independiente.



Arquitectura J2EE en 4 capas

- J2EE: Web + Servlets + Beans + Datos
- Los beans pueden ser persistentes en BBDD.



EJB: Entreprise Java Beans

- EJB son un componente del J2EE de Sun (una arquitectura de desarrollo muy usada)
- EJB son objetos remotos:
  - Tienen un interface.
  - Internamente se invocan por RMI.
- EJB son componentes que se instalan en contenedores de beans:
  - Contenedor J2EE: SunAppserver, JBoss,…
  - Se instalan, despliegan, etc como servlets.