



BULLETPROOF... I WASH I WAS

Introducción a J2EE





Java



BULLETPROOF... I WASH I WAE

- Originalmente concebido para construir programas pequeños que se encuentren dentro de dispositivos electrónicos.
- Se empezó a utilizar en navegadores de Internet corriendo pequeños programas llamados Applets
- Desde entonces a tenido un gran crecimiento y esta siendo usado para construir grandes sistemas empresariales





Ejemplo de Applet

BULLECPROOF... | WASH | W&S

Microsoft Internet Explorer window titled "Listas Enlazadas". The address bar shows the local file path: C:\Documents and Settings\omar\Mis documentos>Listas Enlazadas>Listas.html.

The applet interface includes the following elements:

- Form fields: "Nombre: Enrique" and "Telefono: 9256789".
- A "Total:" label with a value of "2" in a text box.
- A list of entries: "Salvador, 9245687" and "Omar, 9252525".
- Control buttons: "Agregar", "Del * Nombre", "Ordenar", "Buscar por Tel", and "Tamaño".
- Status bar: "Estado: Salvador, 9245687 agregado".

The taskbar at the bottom shows "Subprograma ListaApplet started" and "MI PC".



Java

- Sin embargo el lenguaje Java solo, no es capaz de conocer los requerimientos de un sistema empresarial.
- Se requiere de una plataforma capaz de proporcionar algunos de los servicios que una aplicación empresarial necesita Ej.:
 - La habilidad de almacenar datos en distintas bases de datos comerciales
 - La habilidad de distribuir la aplicación en mas de una computadora
 - Soportar transacciones
 - Multihilo
 - Pool de conexiones
 - Escalable





Por lo tanto



BULLETPROOF... | WASH | W&S

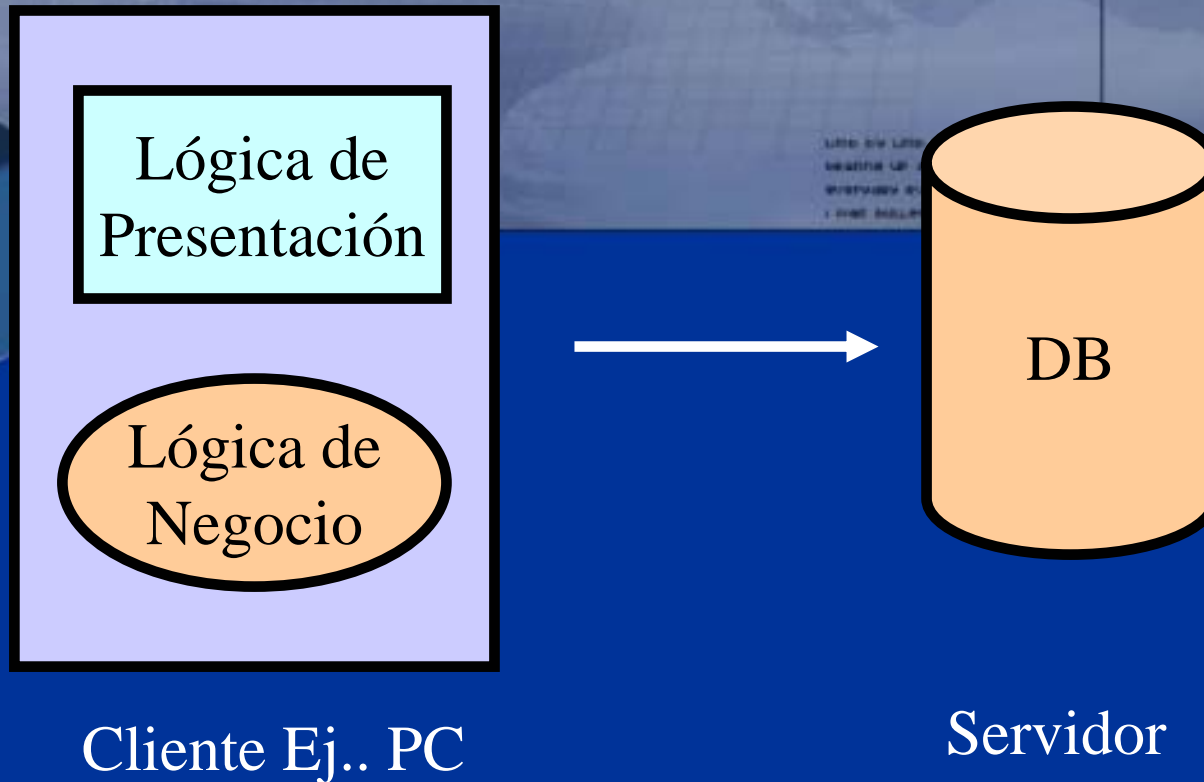
- Una aplicación empresarial no puede estar localizada en una PC.
- Es necesario utilizar algún tipo de Arquitectura o Framework donde dividamos por capas los componentes de nuestro sistema





Arquitectura de dos Capas

- Cliente – Servidor usada en los 80s y 90s





Desventajas



BULLETPROOF... I WASH I WASH

- La lógica de negocio y la lógica de presentación son mezcladas, lo que resulta difícil de entender el sistema y realizar mantenimiento



Código que mezcla la lógica de Presentación con la de Negocio



```
Horarios (shared/report2.asp*)
24 <%
25 set rs = Server.CreateObject("ADODB.Recordset")
26 rs.ActiveConnection = MM_CUCI_STRING
27 rs.Source = "SELECT * FROM dbo.Plantilla WHERE Codigo = " + Replace(rs__MMColParam, "'", "'') + ""
28 rs.CursorType = 0
29 rs.CursorLocation = 2
30 rs.LockType = 3
31 rs.Open()
32 rs_numRows = 0
33 %>
34 <html>
35 <head>
36 <title>Horarios</title>
37 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
38 </head>
39 <body bgcolor="#FFFFFF" text="#000000">
40 <br>
41 <br>
42 <%If rs.EOF or rs.BOF Then
43     Response.Write "<p align=" & "left" & ">&nbsp;";
44     Response.Write "No se encontro el codigo "&nbsp;&rs__MMColParam
45     Response.Write "<p>&nbsp;</p>"
46     Response.Write "<A HREF='plantilla.asp'>Regresar</A>"
47     Response.End
48     End If %>
49 <table width="41%" border="0">
50 <tr>
51 <td width="17%" bgcolor="#000066"><b><font face="Arial, Helvetica, sans-serif" size="-1" color="#FFFFFF">Codigo</font></b><
52 <td width="83%" bgcolor="#EEEEEE"><b><font face="Arial, Helvetica, sans-serif" size="-1" color="#000000"><%=rs.Fields.Item
53 </tr>
```

Lógica de Negocio

Lógica de Presentación

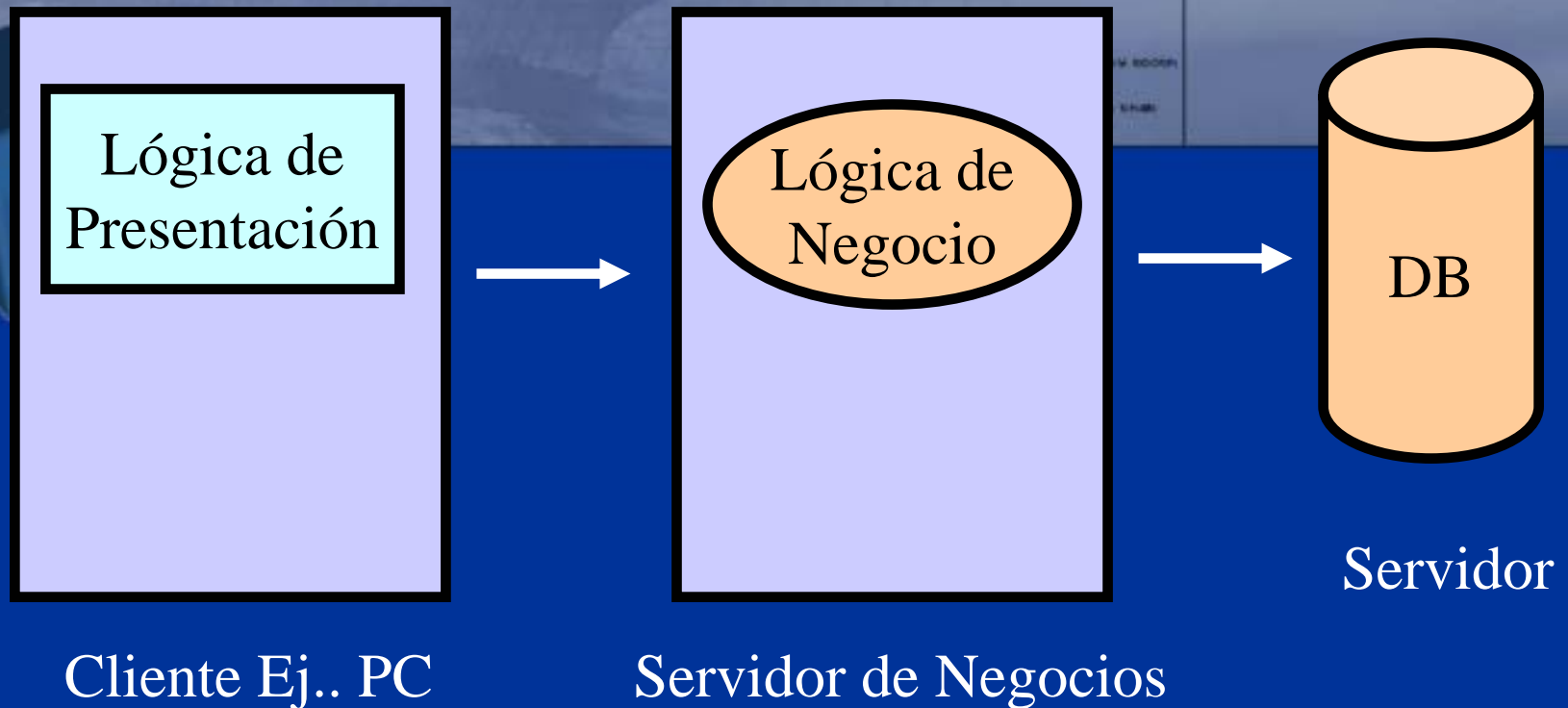
Lógica de Negocio



Modelo de tres capas

BULLETPROOF... I WASH I WASH

- Separa la lógica de presentación y la lógica de negocio en 2 capas





Desventajas



BULLETPROOF... I WASH I WAS

- Complejidad involucrada en desarrollar la aplicación
- Se requiere conocimientos de computo distribuido(Ej.. RMI, CORBA)





Servidores de Aplicación

- Pueden proporcionar un Framework que es necesario para desarrollar una aplicación multi-capa
- Proporcionan servicios tales como computo distribuido, multihilo, seguridad y persistencia
- En el pasado cada servidor de aplicación era desarrollado independientemente y por lo tanto implementaba los servicios de diferente manera
- Cambiar de un servidor de aplicaciones a otro era difícil ó imposible





Servidores de Aplicación

- En 1997 un grupo de vendedores de servidores de aplicaciones (BEA, IBM, Oracle, Sybase, SUN entre otros) comenzaron a trabajar juntos para definir un estándar en los servidores de aplicación basados en el lenguaje Java
- La visión fue crear un conjunto de servicios estandarizados y unas APIs para acceder a estos servicios





Estándar J2EE

BULLSEPPOOF... I WASH I WAE



- Define rigurosamente un conjunto de servicios que un servidor de aplicaciones debe tener, junto con una API estándar para acceder a estos servicios. Basado en Java
- J2EE no es un producto es una especificación, en base a esta especificación existen muchas implementaciones
- Las implementaciones son proporcionadas por fabricantes terceros (IBM, BEA), estas implementaciones deben de cumplir la especificación J2EE



LINE BY LINE AND BROWN BY BROWN
HEALTHY UP WITH UP THE
SUPPORT SUPPORT UP THE
I WAE BULLSEPPOOF

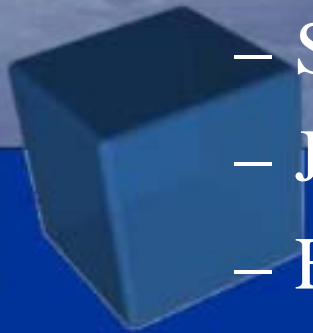


Componentes J2EE

BULLETPROOF... I WASH I WAS

- J2EE define tres tipos de componentes que un programador puede desarrollar:

- Servlets
- JSP
- Enterprise Java Beans



LINE BY LINE AND ROOM BY ROOM
MEANS UP WITH OF THE
STAFFS STAFFS UP, THE
I WAS BULLETPROOF



Java Servlets



BULLEDPPOOF... I WASH I WAS

- Proporcionan un método para escribir programas del lado del servidor
- Su uso común es la generación de paginas web dinámicas





Ejemplo Código de un Servlet

```
public class ServletEj extends HttpServlet {
    private static final String CONTENT_TYPE = "text/html";

    public void init() throws ServletException {
    }

    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType(CONTENT_TYPE);
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head><title> ServletEj</title></head>");
        out.println("<body bgcolor=' #ffffff'>");
        out.println("<p> The servlet has received a " + request.getMethod() + ". This is the reply.</p>");
        out.println("</body></html>");
    }

    public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        doGet(request, response);
    }

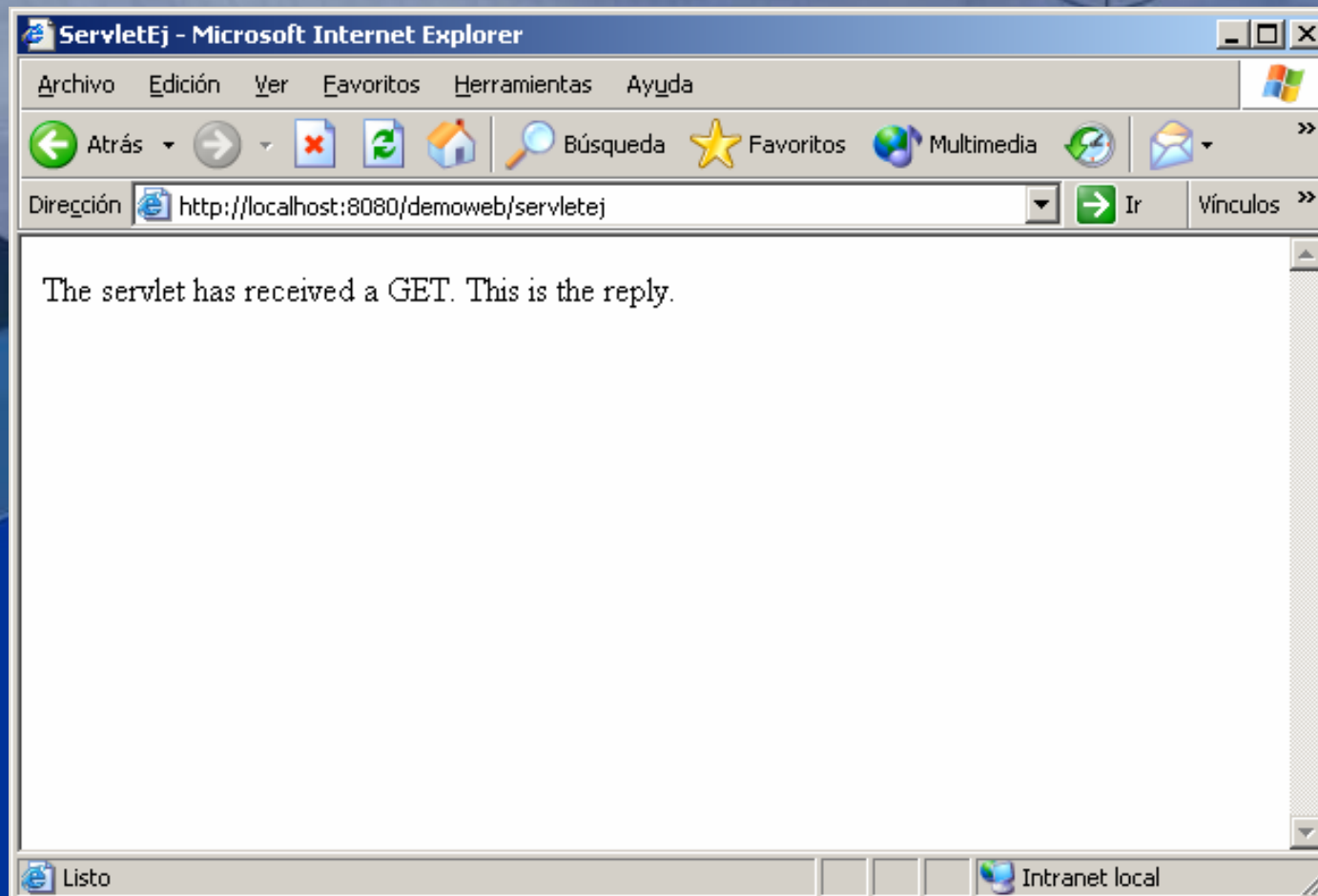
    public void destroy() {
    }
}
```

Para obtener Ayuda, presione F1



Ejemplo de un Servlet

BULLECPPOF... I WASH I WBS





Java Server Pages (JSP)

- Permiten a los diseñadores web construir paginas web interactivas sin entrar en detalles del lenguaje java
- Se ven muy similares al estándar HTML
- La diferencia es que JSP permite fragmentos de código Java incrustado en la pagina web





Ejemplo de código JSP

BULLBOYPROOF... I WASH I WASH



Ejemplo de JSP

BULLECPROOF... | WASH | W&S





Enterprise Java Beans (EJB)

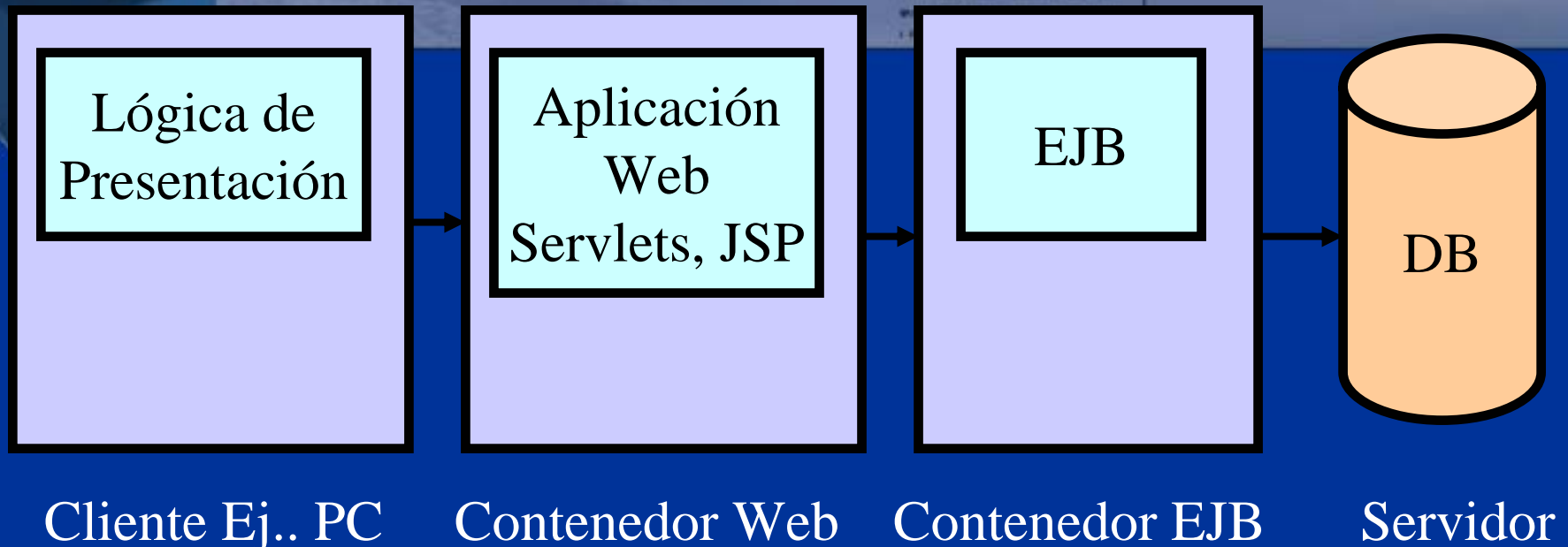
- De los tres tipos de componentes los EJBs son los mas importantes
- Un EJB es una clase java que tiene varias características por Ej.:
 - Son distribuidos
 - Usan transacciones
 - Multi Hilos
 - Persistentes
- Muchas de las características de los EJBs son proporcionadas por los distintos servidores de aplicación





Arquitectura J2EE

- La Arquitectura J2EE es muy similar al modelo de tres capas, la diferencia es que es mas extendido





Arquitectura J2EE

BULLETPROOF... I WASH I WAS

- Un servidor de aplicaciones J2EE debe proporcionar un Contenedor Web, Contenedor EJBs, Contenedor de Applets
- Una Aplicación J2EE no necesariamente debe ser una aplicación web.



Ejemplo de una Arquitectura



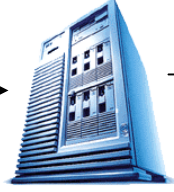
Cliente Web



Servidor J2EE
Linux



Servidor J2EE
Windows



Contenedor EJB

Facade EJB

usa

usa

Categoría EJB

1

1..*

Empresa EJB

Contenedor Web

Servlet
inicio



index.jsp

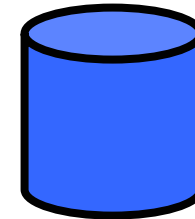


Cliente J2ME

Servidor DB



dbdirectorio.jds





Ejemplo de una Aplicación J2EE



El Buscador





Descripción



BULLECPROOF... | WASH | WAE

- Se trata de un Directorio de Empresas y Clasificados el cual nos sirva para consultar direcciones, teléfonos de empresas ó anuncios clasificados de las ciudades de toda la Republica Mexicana por medio de una sencilla búsqueda sobre nuestro celular o a través de un navegador de Internet.



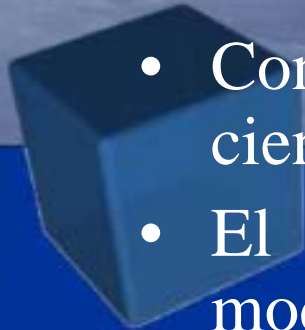


Desarrollo



BULLECPROOF... I WASH I WAE

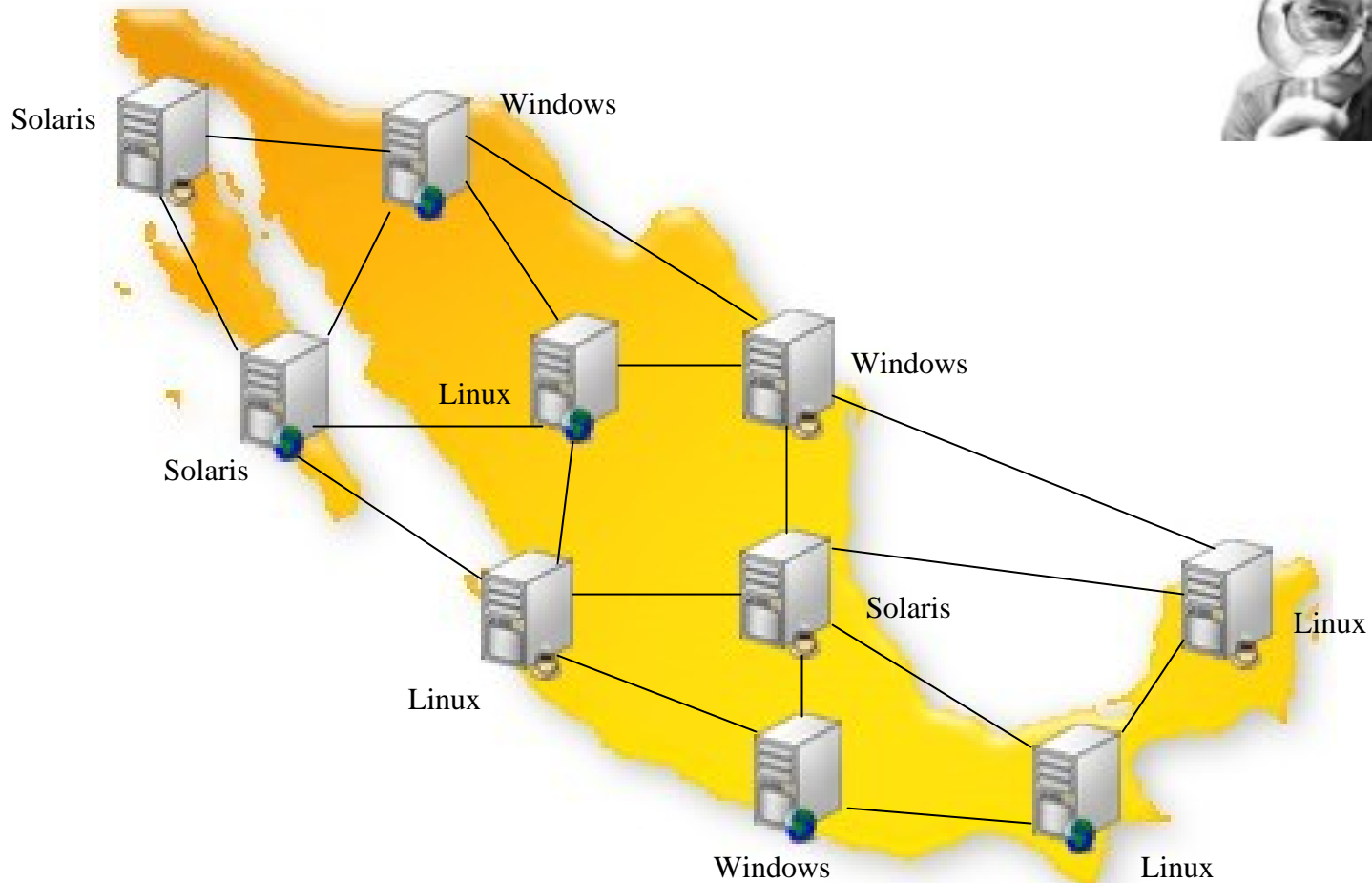
- Hecho completamente en Java utilizando J2EE para demostrar el gran potencial de usar esta tecnología por ejemplo: Alta disponibilidad, Redundancia a Fallas, Alto desempeño, Balanceo de Cargas.
- Con estas características El Buscador puede manejar cientos de miles de registros sin ningún problema.
- El Buscador puede ir creciendo sin hacer alguna modificación ya que esta hecho bajo J2EE.
- Fácilmente puede ser migrado a cualquier servidor de aplicaciones que cumpla con la especificación J2EE.
- De igual manera la Base de Datos puede ser migrada por una distinta.



Puede estar corriendo sobre distintas Arquitecturas



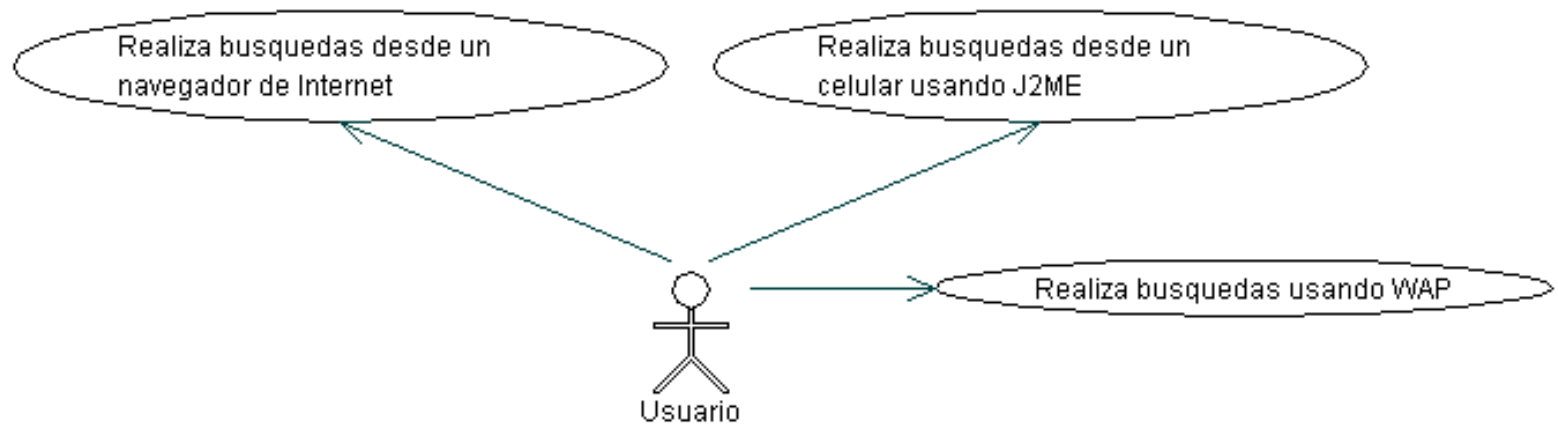
BULETIN DE LA UNIVERSIDAD DE VALENCIA





Tipos de Clientes

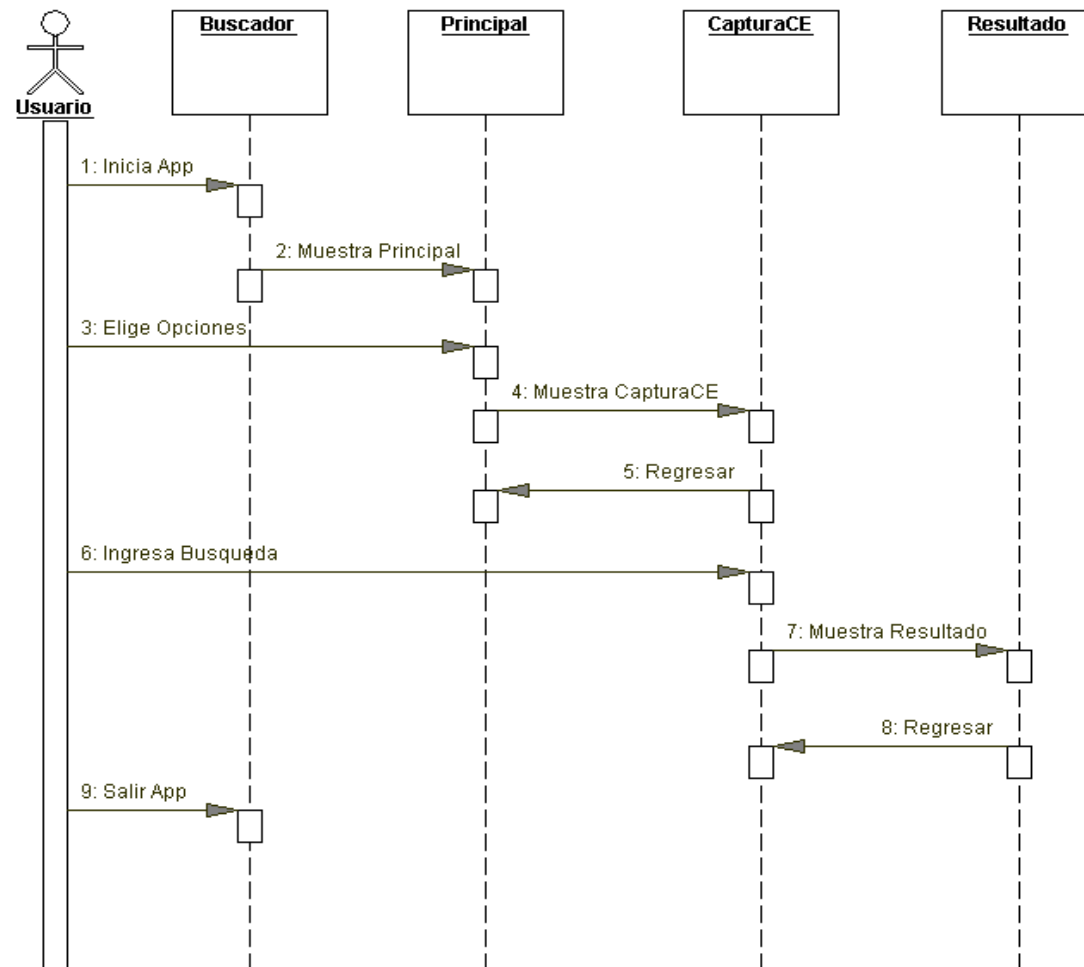
BULLETPROOF... | WASH | W&S



Secuencia para realizar alguna Búsqueda usando J2ME



BULLSEPPOOF... I WASH I WASH





¿Cómo funciona?

BULLECPROOF... I WASH I WASH

- Los clientes Web, J2ME, WAP, se conectan a un Servlet llamado inicio el cual direcciona diferentes contenidos de presentación de acuerdo al cliente.
- El Servlet inicio utiliza un EJB llamado FacadeBean que contiene las rutinas necesarias para que funcione El Buscador.
- El EJB Facade sirve para realizar la comunicación entre el Servlet inicio y los demás EJBs.
- Los EJBs Empresa y Categoría realizan un mapeo a una Base de Datos que es donde se guarda la información.





BULLETPROOF... I WASH I WAS

Gracias !!!

USE BY USE AND ROOM BY ROOM
WASH UP WITH UP THE
WASH UP WITH UP THE
I WAS BULLETPROOF

