

Capítulo VI

Desarrollo del aplicativo web XJDN2004



- Análisis y diseño del sistema**
- Análisis de requerimientos**
- Definición del sistema**
- Diagramas UML**
- Modelo conceptual**
- Modelo físico**
- Arquitectura del aplicativo**
- Implementación del aplicativo web**
- Validación y pruebas**

6.1. Análisis y diseño del sistema

A continuación se detalla una descripción de la metodología que se utilizó para el desarrollo del aplicativo Web XJDN2004, además se indica las herramientas software utilizadas en la implementación del mismo, como: sistema operativo, servidores, manejadores de bases de datos, front-ends, etc.

Para la construcción del software, existen diferentes etapas que generan su propia documentación, a continuación se describe cada una de ellas.

6.1.1. Análisis de requerimientos

El primer paso para el desarrollo de la aplicación es definir el problema que deseamos resolver y hasta qué punto la aplicación puede ayudar a resolverlo, posteriormente hay que definir las necesidades que serán cubiertas por ella, el siguiente paso es comenzar a diseñar el esquema general de la aplicación y realizar un prototipo que se usará para corroborar que el cumplimiento de las necesidades sea completo

A través de la Federación Deportiva de Imbabura, que es un organismo encargado de planificar, fomentar, controlar, desarrollar y supervisar las actividades deportivas de los organismos que están bajo su jurisdicción, se ha conseguido realizar un aplicativo web para los X Juegos Nacionales.

Para empezar a realizar nuestro proyecto primeramente nos informamos de todo lo concerniente a la Federación Deportiva de Imbabura como: quien es la FDI, su estructura, infraestructura, presidentes y demás información.

La Federación Deportiva de Imbabura estará integrada por los siguientes organismos:

- Las Asociaciones Deportivas Provinciales por Deporte.
- Los Comités Deportivos y las Ligas Deportivas Cantonales, Barriales.
- Los Clubes Deportivos que tengan personería jurídica y que deben obligatoriamente afiliarse a la Matriz del Deporte.

Posteriormente nos enfocamos a los X Juegos Deportivos Nacionales Ibarra 2004, fuimos revisando el manejo de cada una de las disciplinas deportivas, es decir obteniendo la mayor cantidad de información posible, para continuar con el diseño del aplicativo.

El Comité Organizador de los X Juegos Deportivos Nacionales resolvió que:

- La categoría en la cual se competirá en los X Juegos Deportivos Nacionales del año 2004, será la CATEGORIA JUVENIL y de acuerdo a las edades internacionales por deporte.
- Las disciplinas en las que se va a competir son las siguientes:

Ajedrez	Atletismo	Ciclismo	Halterofilia	Tenis de Mesa
Boxeo	Ecuestres	Fútbol	Lucha	Tenis de Campo
Tiro	Escalada	Judo	Natación	Gimnasia Artística
Baloncesto	Patinaje	Karate Do	Tae Kwon Do	Pelota Nacional
	Triatlón	Voleibol		

El website que se implementó manipula la información en base a datos suministrados, por lo que, en el aplicativo se determinó que se deben diseñar dos grandes partes, la una destinada a la administración de datos, es decir, el ingreso, eliminación y modificación de resultados de las competencias que se realicen en cada disciplina; la segunda parte del aplicativo se debe enfocar a lo que un cliente común desearía consultar en el website, es decir resultados finales de las competencias, ganadores de medallas, provincias participantes, horarios de competencia, entre otras.

Como primer paso se determinó las características de los datos requeridos para cada una de las disciplinas, debido a que cada disciplina maneja sus datos de manera diferente. Se estableció las categorías existentes para cada disciplina, además de la existencia de participación en equipos en algunas disciplinas deportivas.

Con todo este análisis de requerimientos, nuestro WebSite involucra el manejo de información referente a la Federación Deportiva de Imbabura, resultados de juegos deportivos nacionales anteriores, inscripción de jugadores, resultados producidos diariamente en cada una de las disciplinas a competir, resultados generales, records, mejores deportistas, cuadros generales de medallas y consultas en las que el cliente puede interactuar.

6.1.2. Definición del sistema

Para desarrollar este aplicativo se decidió utilizar únicamente software libre, todos los elementos constitutivos de la aplicación debían cumplir dicho requisito; en los capítulos anteriores se han indicado distintas posibilidades disponibles, a continuación se mostrarán los elementos software que se seleccionaron en función de los requisitos y características del proyecto a desarrollar.

Sistema Operativo

El sistema operativo Linux es el sistema idóneo para aplicaciones web, en vista a que Linux tiene todas las prestaciones de un sistema operativo moderno y completamente desarrollado: multitarea real, memoria virtual, bibliotecas compartidas, carga de sistemas, compartimiento, manejo de memoria y soporte de redes TCP/IP. Linux ha alcanzado popularidad y confianza, la gran popularidad de Linux incluye los siguientes puntos: Se distribuye su código fuente, lo cual permite a cualquier persona hacer algunos cambios para resolver problemas que se puedan presentar, así como también agregar funcionalidad. Es desarrollado en forma abierta, esto permite un rápido y eficiente ciclo de desarrollo. La versión de Linux utilizada es Linux Red Hat 7.3

Servidor de base de datos

En los últimos tiempos se está viendo cómo los SGBDD libres están creciendo en prestaciones y presencia en el mercado, de tal manera que en poco tiempo podrán estar en condiciones de luchar de igual a igual con los grandes SGBDD propietarios. PostgreSQL es una base de datos relacional basada en objetos, es totalmente compatible con el ANSI SQL92, incluyendo subselects, views, foreign keys, etc. La versión de PostgreSQL es 7.2.1-5

Servidor Web

Una parte importante de cualquier aplicación web es el servidor web; que se encarga de atender las peticiones de los clientes y enviarles las páginas web solicitadas. Existen multitud de servidores web, muchos de ellos distribuidos como software libre y siendo, el más popular de todos Apache. El servidor Apache es el producto de un equipo de desarrolladores que, de forma desinteresada, construyen un servidor HTTP robusto, de alta calidad y disponible de forma libre. La versión usada de Apache es 1.3.23-11

Servidor de Servlets

Tomcat es un contenedor de servlets con un entorno JSP, un contenedor de servlets es un shell de ejecución que maneja e invoca servlets por cuenta del usuario. Tomcat es necesario para que se ejecute el servlet Cocoon.

Versión de Tomcat: 4.0.3

Estructura de publicación basada en XML/XSL

Nuestro aplicativo va a manejar el contenido y la presentación de forma separada, utilizando documentos XML que permiten describir tanto la estructura como el contenido y el uso de hojas de estilo XSL encargadas de la presentación, por lo que se requiere una estructura de publicación XML/XSL, dentro del software libre Cocoon es la opción más madura. Versión de Cocoon: 2.0.2

Cliente

Browsers, visores de pdf.

Requerimientos de Hardware Servidor

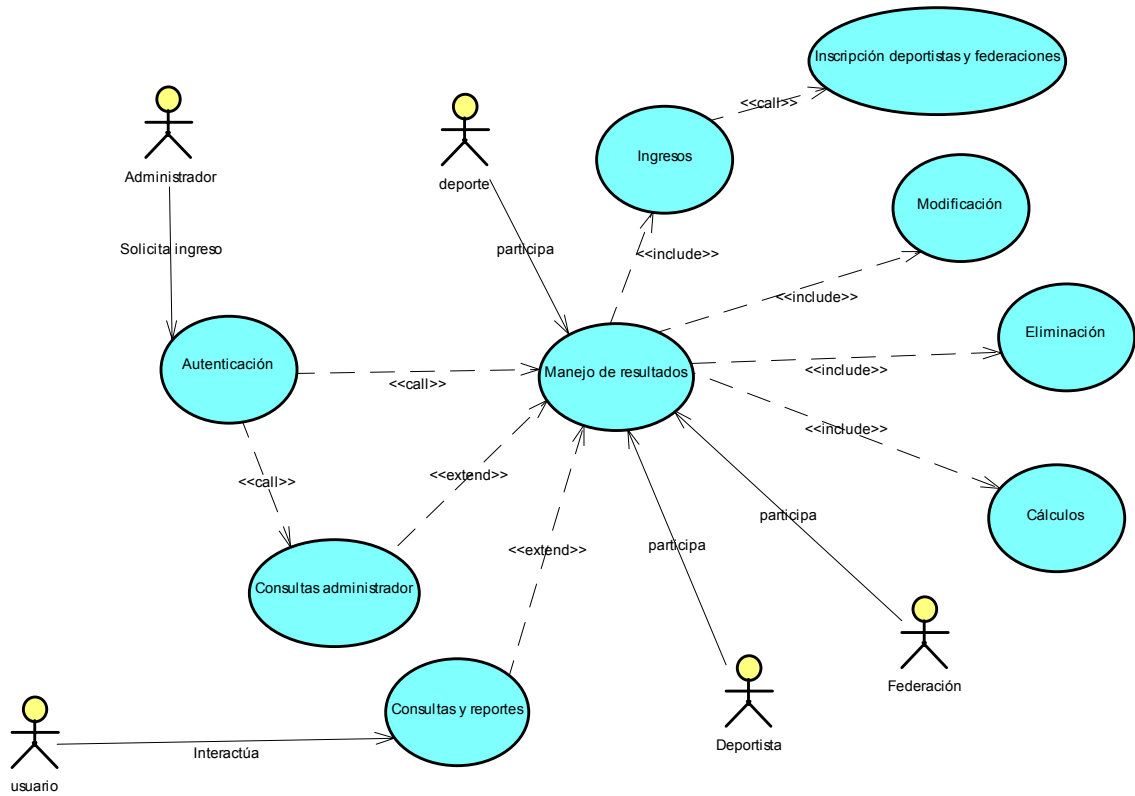
- Procesador Pentium IV 1.8Ghz
- Memoria 256 Mbytes
- 30 Gb De disco duro
- Tarjeta de red

Requerimientos de Hardware Cliente

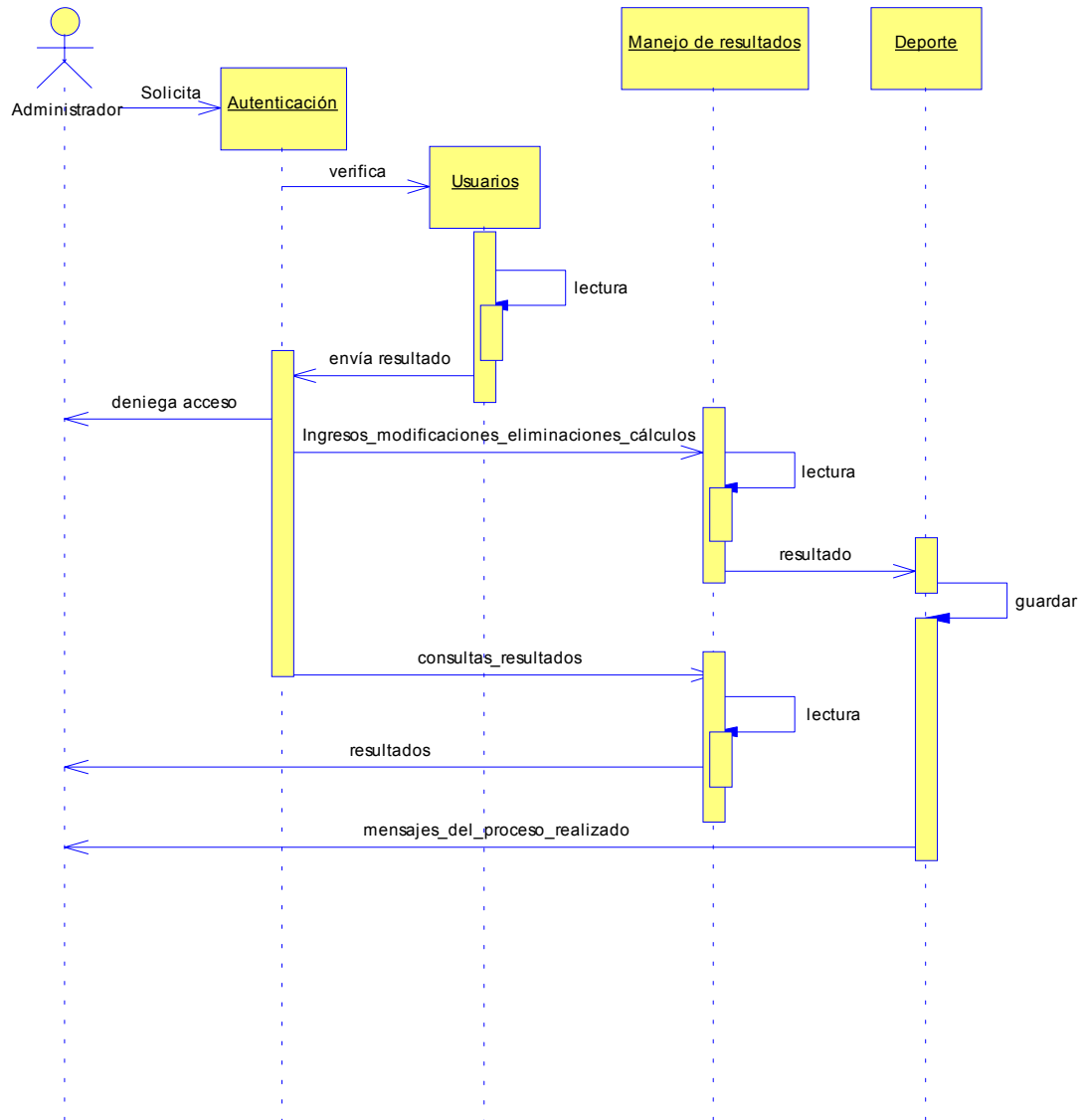
- Cualquier computador que tenga instalado un navegador: Internet Explorer 5.0 en adelante, Netscape Navigator 4.5. o superior.

6.1.3. Diagramas UML

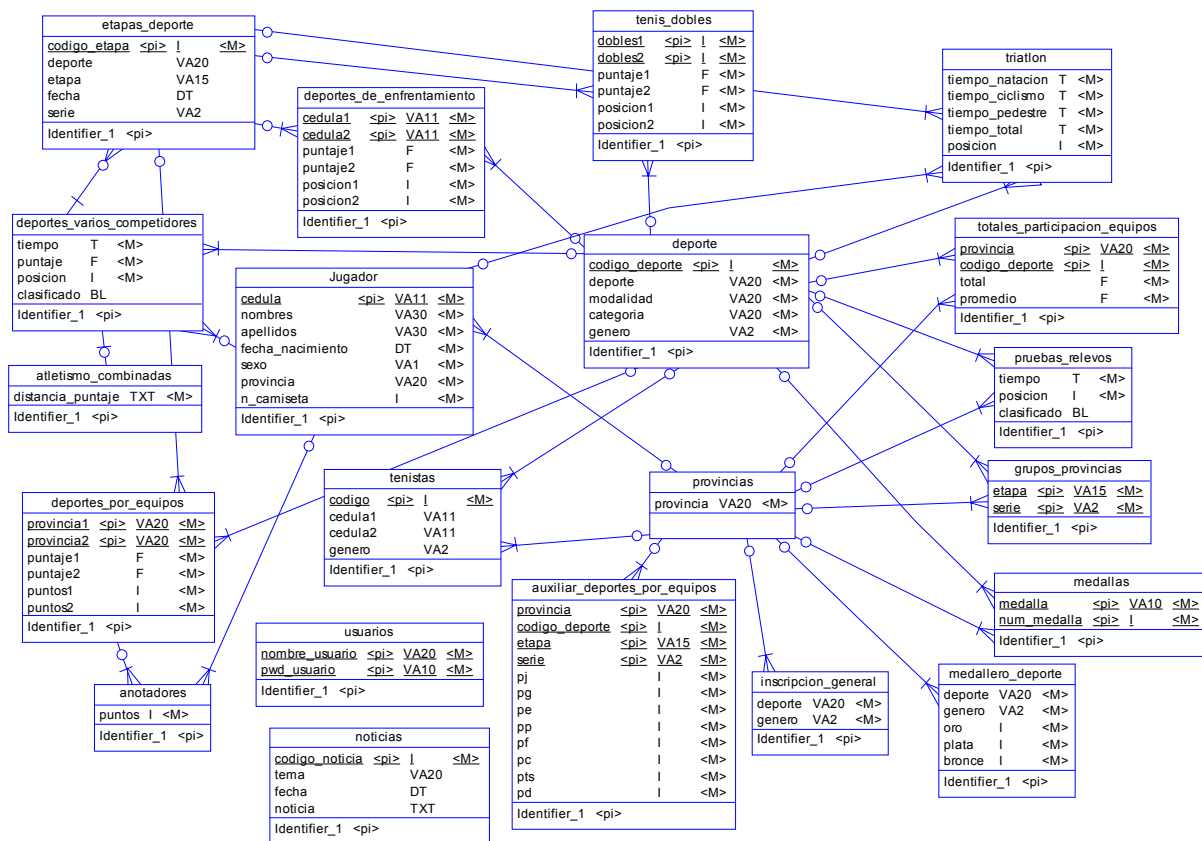
6.1.3.1. Diagrama de Casos de usos



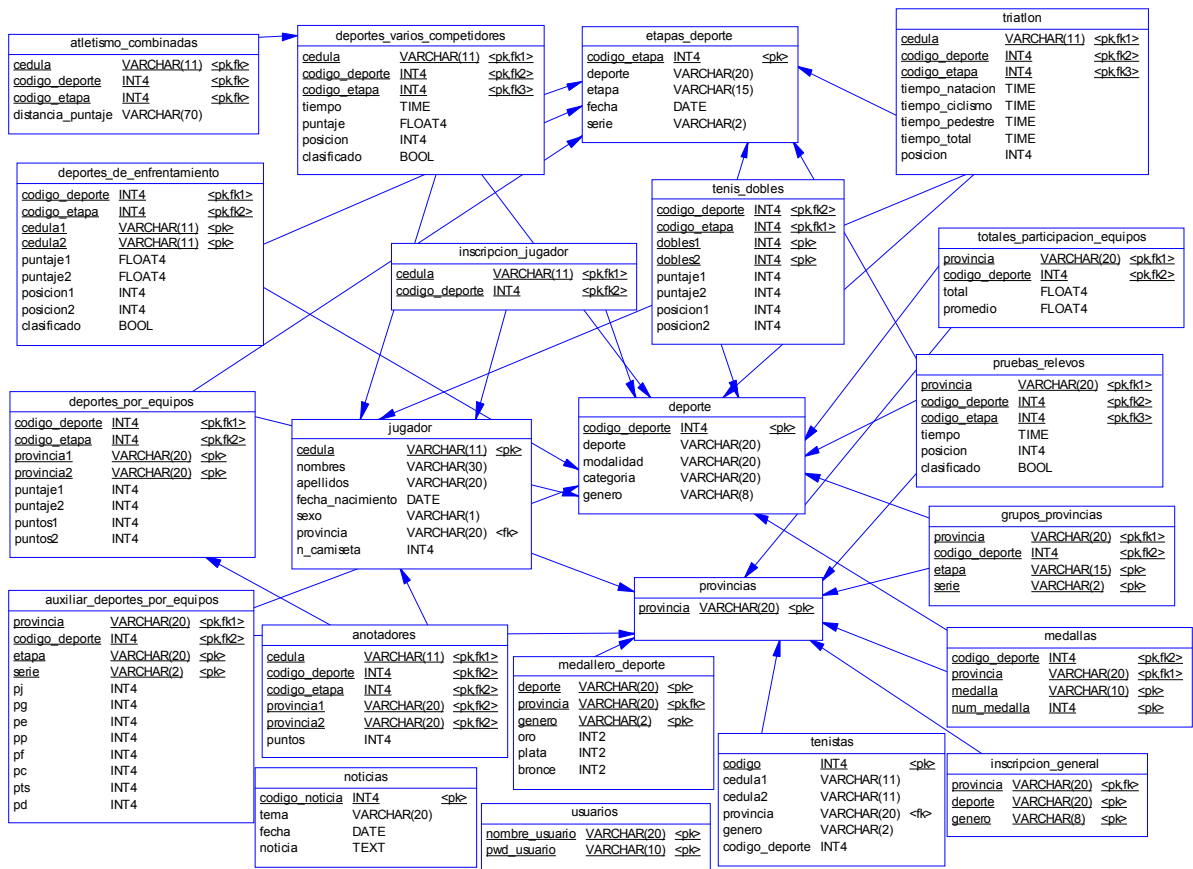
6.1.3.2. Diagrama de secuencia



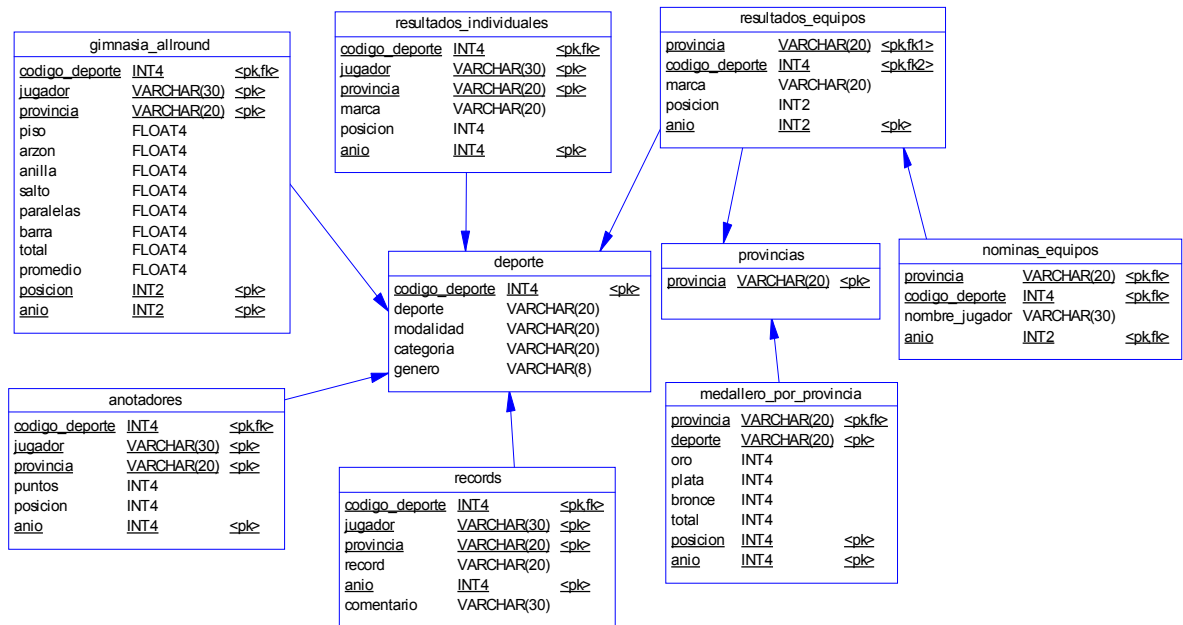
6.1.4. Modelo conceptual



6.1.5. Modelo físico (Base de Datos para los X juegos Deportivos Nacionales 2004)



Modelo físico de la base histórica de datos



6.1.6. Arquitectura del aplicativo.

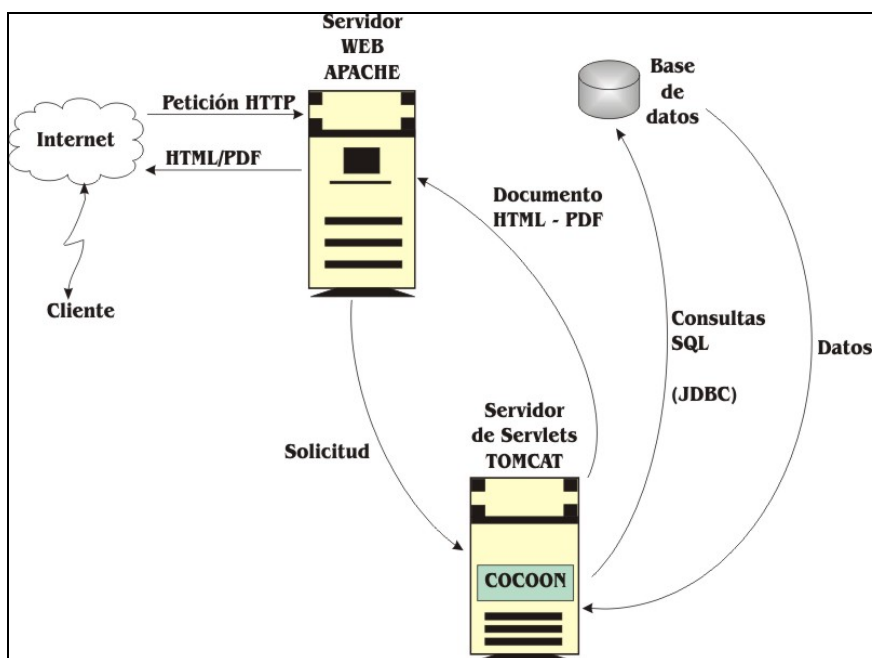


Figura 6.1.: Arquitectura del aplicativo.

Un cliente a través de Internet realiza una petición HTTP al servidor web (APACHE), éste hace la solicitud al servidor de servlets Tomcat, a través del cual se ejecuta Cocoon encargándose de realizar el proceso de acceder a la base de datos PostgreSQL para realizar las consultas y extraer la información a un documento XML y aplicarle una hoja de estilo XSL y devolver el resultado como un documento HTML o PDF al servidor web para que este sirva al cliente los datos solicitados.

En el aplicativo se divide el proceso de servir una petición en dos fases principales:

- La primera fase consiste en generar un documento XML desde la lógica de la aplicación, a través de XSP.
- La segunda fase consiste en procesar el documento XML a través de una hoja de estilo XSL de forma que el resultado de la transformación sea la interfaz deseada.

Estas fases las realiza Cocoon y este proceso es transparente al usuario que recibe como respuesta HTML o PDF.

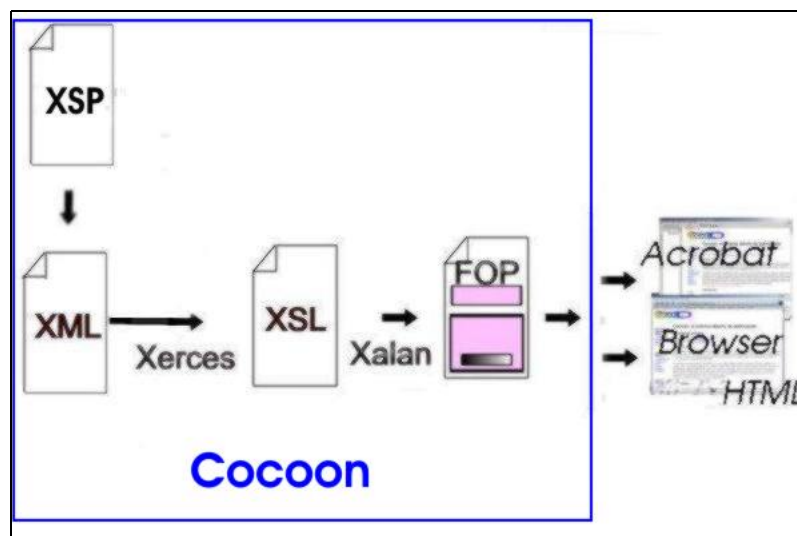


Figura 6.2. Procesamiento de XML/XSL a través de Cocoon

Analicemos un poco más en detalle cada una de las fases:

Generación del documento XML: En esta parte los documentos XML se generan a través de XSP, mediante XSP realizamos el acceso a base de datos, los datos obtenidos se van etiquetando generando así un documento XML, al que posteriormente se le aplica una hoja de estilo XSL para su presentación.

Procesamiento del documento XML con XSL: Este proceso lo realiza Cocoon. Una vez generado el documento XML o tomando el documento XML requerido, Cocoon analiza (parsea) el documento XML, mediante el analizador Xerces, posteriormente aplica la hoja de estilo, proceso realizado por Xalan (procesador XSL) obteniendo como resultado HTML o PDF según sea la petición del usuario.

6.2. Implementación del web site

La implementación del aplicativo web XJDN2004 empezó utilizando para su análisis y diseño los diagramas de objetos basados en UML, generando el diagrama de casos de uso, diagrama de secuencia, el modelo conceptual y el modelo físico.

Una vez escogido el motor de base de datos, en nuestro caso PostgreSQL, implementamos tablas, vistas y otras estructuras relacionales, con esto empezamos a crear el prototipo funcional, para lo cual desarrollamos páginas estáticas con XML/XSL, produciendo como resultado HTML y PDF, que se visualizarán en el cliente.

La conexión a la base de datos se la realizó vía JDBC, y para poder realizar las consultas necesarias se utilizó XSP, que involucra páginas dinámicas con código java embebido. Las validaciones de datos se las implementó en JavaScript, la validación en el lado del cliente hace que la aplicación sea más amigable para el usuario, y no tenga que esperar la respuesta del servidor con mensaje de error, para volver a reescribir los datos.

Ejemplo de una página dinámica XSP, en la que se realiza una consulta sobre el cuadro general de medallas de los X Juegos.

```
<?xml version="1.0"?>
<xsp:page language="java"
  xmlns:xsp="http://apache.org/xsp" xmlns:xsp-request="http://apache.org/xsp/request/2.0"
  xmlns:esql="http://apache.org/cocoon/SQL/v2">
<pagina>
  <xsp:logic>
    int ubicacion=0, oro=0, plata=0, bronce=0, total=0;
    String provincia="", deporte1="";
  </xsp:logic>
  <title>DISTRIBUCIÓN GENERAL DE MEDALLAS POR FEDERACIONES</title>
  <imagenesup src="images/superior.jpg"/>
  <opcionsup>
    <opcion label="FEDERACION" img="images/fdi_icon.gif" enlace="../index.xml"/>
    <opcion label="X JUEGOS" img="images/xjn_icon.gif" enlace="index_xjn.xml"/>
    <opcion label="CALENDARIO XJN" img="images/horario.gif" enlace="horario.xml"/>
    <opcion label="SEDES XJN" img="images/sedes.gif" enlace="xjn2004_sedes.xml"/>
    <opcion label="ATRAS" img="images/atras.gif" enlace="xjn_resultados_generales.xsp"/>
  </opcionsup>
  <esql:connection>
  <esql:pool>xjn2004_user</esql:pool>
  <resultados>
    <titulo>CUADRO DE MEDALLAS POR FEDERACIONES XJDN2004</titulo>
    <label width="5">No.</label>
    <label width="47">FEDERACION</label>
    <label width="12">ORO</label>
    <label width="12">PLATA</label>
    <label width="12">BRONCE</label>
    <label width="12">TOTAL MEDALLAS</label>
    <esql:execute-query>
    <esql:query> select * from medallero_general order by oro, plata, bronce </esql:query>
    <esql:results>
      <esql:row-results>
        <resultado>
          <xsp:logic>
            ubicacion++;
            provincia = <esql:get-string column="provincia"/>;
            oro = <esql:get-int column="oro"/>;
            plata = <esql:get-int column="plata"/>;
            bronce = <esql:get-int column="bronce"/>;
            total = oro + plata + bronce;
          </xsp:logic>
          <ubicacion><xsp:expr>ubicacion</xsp:expr></ubicacion>
```

```

    <provincia><xsp:expr>provincia</xsp:expr></provincia>
    <medallas_oro><xsp:expr>oro</xsp:expr></medallas_oro>
    <medallas_plata><xsp:expr>plata</xsp:expr></medallas_plata>
    <medallas_bronce><xsp:expr>bronce</xsp:expr></medallas_bronce>
    <medallas_total><xsp:expr>total</xsp:expr></medallas_total>
  </resultado>
</esql:row-results>
</esql:results>
</esql:execute-query>
</resultados>
</esql:connection>
<autor>DAYB - TESIS XML</autor>
</pagina>
</xsp:page>

```

Ejemplo de la hoja de estilo XSL, utilizada para dar presentación a los datos generados por la página XSP anterior.

```

<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="pagina">
<html>
  <head><title><xsl:value-of select="title"/></title></head>
<body>
  <table width="763" border="1" align="center" bordercolor="#003366">
    <tr>
      <td width="100%" bordercolor="#D8D8CB">
        <table border="0" cellpadding="0" cellspacing="0" width="100%">
          <xsl:apply-templates select="imagenesup"/>
          <tr> <td width="100%">
            <table border="0" cellpadding="0" cellspacing="0" width="100%">
              <tr> <td width="23%" bgcolor="#FFFFFFD9" valign="top"></td>
                <td width="77%" bgcolor="#003366" >  </td>
            </tr>
          </table>
        </td> </tr>
      <tr> <td width="100%">
        <table border="0" width="100%" background="images/sup3.jpg">
          <tr> <td width="20%" rowspan="2"> </td>
            <td width="80%" valign="top"> </td>
          </tr>
        </table>
      </td> </tr>
      <tr> <td width="100%">
        <table border="1" width="100%" bordercolor="#003366">
          <tr> <td width="100%" bordercolor="#8CC2FF">
            <table border="0" cellpadding="0" cellspacing="0" width="100%" bgcolor="#003366">
              <tr> <td width="20%">
                <table border="0" cellpadding="0" cellspacing="0" width="102%">
                  <tr> <xsl:apply-templates select="opcionsup/opcion"/> </tr>
                </table>
              </td> </tr>
            </table>
          </td> </tr>
        </table>
      </td> </tr>
      <tr> <td>
        <table width="100%" border="1" bordercolor="#003366">

```

```

        <tr> <td> <xsl:apply-templates select="resultados"/></td></tr>
    </table>
</td> </tr>
<tr> <xsl:apply-templates select="autor"/> </tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>
</xsl:template>

<xsl:template match="imagenesup">
<tr><td width="100%"></td></tr>
</xsl:template>

<xsl:template match="opcionsup/opcion">
    <td width="20%"> <table border="0" cellpadding="0" cellspacing="0" width="100%">
        <tr> <td width="65%"> <font face="Arial" color="#FFFFFF">
            <xsl:value-of select="@label"/></font></td>
            <td width="21%"><a href="{@enlace}"> </a></td>
            <td width="14%"></td>
        </tr>
    </table></td>
</xsl:template>

<xsl:template match="resultados">
<table width="75%" align="center" border="0" cellpadding="1" cellspacing="1" bgcolor="#000000">
    <xsl:apply-templates select="resultados/titulo"/>
    <tr bgcolor="#6699CC"><xsl:apply-templates select="resultados/label"/></tr>
    <xsl:apply-templates select="resultados/resultado"/>
    <xsl:apply-templates select="resultados/totales"/>
    <xsl:apply-templates/>
</table>
</xsl:template>

<xsl:template match="resultados/label">
    <td width="{@width}%" bgcolor="#6699CC"> <font color="#000000" face="Arial">
        <strong><xsl:apply-templates/> </strong> </font> </div> </td>
</xsl:template>

<xsl:template match="resultados/titulo">
    <tr bgcolor="#6699CC" ><td colspan="6" bgcolor="#F9EAB7"><div align="center">
        <font size="3" face="Verdana, Arial"><strong> <xsl:apply-templates/></strong></font></div>
    </td></tr>
</xsl:template>

<xsl:template match="resultados/resultado">
    <tr bgcolor="#FFFFFF">
        <xsl:apply-templates select="resultados/resultado/ubicacion"/>
        <xsl:apply-templates select="resultados/resultado/provincia"/>
        <xsl:apply-templates select="resultados/resultado/medallas_oro"/>
        <xsl:apply-templates select="resultados/resultado/medallas_plata"/>
        <xsl:apply-templates select="resultados/resultado/medallas_bronce"/>
        <xsl:apply-templates select="resultados/resultado/medallas_total"/>
    </tr>

```

```
</tr>
<xsl:apply-templates/>
</xsl:template>

<xsl:template match="resultados/resultado/ubicacion">
  <td bgcolor="#EAF1FF"> <font size="2" face="Verdana, Arial"><xsl:apply-templates/></font></td>
</xsl:template>

<xsl:template match="resultados/resultado/provincia">
  <td bgcolor="#C2D7FF"> <font size="2" face="Verdana, Arial"><xsl:apply-templates/></font></td>
</xsl:template>

<xsl:template match="resultados/resultado/medallas_oro">
  <td bgcolor="#EAF1FF"> <font size="2" face="Verdana, Arial"> <xsl:apply-templates/></font></td>
</xsl:template>

<xsl:template match="resultados/resultado/medallas_plata">
  <td bgcolor="#EAF1FF"> <font size="2" face="Verdana, Arial"> <xsl:apply-templates/></font></td>
</xsl:template>

<xsl:template match="resultados/resultado/medallas_bronce">
  <td bgcolor="#EAF1FF"> <font size="2" face="Verdana, Arial"> <xsl:apply-templates/></font></td>
</xsl:template>

<xsl:template match="resultados/resultado/medallas_total">
  <td bgcolor="#EAF1FF"> <font size="2" face="Verdana, Arial"> <xsl:apply-templates/></font></td>
</xsl:template>

<xsl:template match="autor">
  <td width="100%" bgcolor="#003366"> <p align="center">
    <font size="1" face="Arial" color="#FFFFFF"> <xsl:apply-templates/></font></p>
  </td>
</xsl:template>
</xsl:stylesheet>
```


Ubic.	FEDERACION	ORO	PLATA	BRONCE	TOTAL MEDALLAS
1	PICHINCHA	15	10	20	45
2	GUAYAS	12	15	10	37
3	ESMERALDAS	8	10	16	34
4	EL ORO	5	8	12	25
5	IMBABURA	2	10	15	27
TOTAL		42	53	73	168

Figura 6.3. Visualización del resultado la página XSP con XSL en el browser.

6.3. Validación y pruebas.

La principal debilidad en las aplicaciones Web es la insuficiente validación de los datos de entrada. Esto quiere decir que para cada dato que el usuario a suministrado a la aplicación Web, antes de procesarla se debe tener control sobre ella; es decir, sanearla de datos no autorizados.

La validación de los datos se puede efectuar del lado del servidor como del lado del cliente, en el lado del cliente se implementaron funciones escritas en JavaScript, permitiendo que la aplicación sea más amigable para el usuario, y no tenga que esperar la respuesta del servidor con mensaje de error, para volver a reescribir los datos.

Las pruebas con el aplicativo se realizaron utilizando datos de anteriores Juegos Nacionales, comprobando con esto la funcionalidad del aplicativo en lo referente al manejo de la información generada en las competencias.