

UNIVERSIDAD TÉCNICA DEL NORTE

**FACULTAD DE INGENIERÍA EN CIENCIAS
APLICADAS**

**ESCUELA DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES**

TEMA:

**Desarrollo de Aplicaciones Web con el Framework
Java Server Faces (JSF).**

APLICATIVO:

**Sistema de Administración Estudiantil para los
Colegios Fiscales de la provincia de Imbabura con
Herramientas de Software Libre.**

Autor: Dayce Gabriela Tulcán Suquitana

Director: Ing. Miguel Orquera

Ibarra, Febrero de 2010

INTRODUCCIÓN

En los últimos años el avance de la tecnología ha sido a grandes pasos, en el área de la Informática la evolución ha sido grande, encontramos nuevas arquitecturas y patrones de diseño, nuevos Frameworks para los distintos lenguajes de programación; mismos que nos dan una base sobre la estructura que debe tener una aplicación y como se relacionan los diferentes componentes; entre éstos se encuentra JavaServer Faces que se basa en el patrón Modelo Vista Controlador, este Framework nos permite desarrollar aplicaciones web como si se tratara de aplicaciones de escritorio, manejar los componentes de una página como objetos con estado en el lado del servidor, provee conversores y validadores que se los puede asociar de manera individual a cada componente, permite tener varios eventos por página y algo muy importante manejar la navegación entre páginas de una forma declarativa, mediante reglas y casos de navegación descritos en un archivo de configuración XML. Este Framework es un estándar de Java, lo que asegura su existencia y evolución, además podemos encontrar implementaciones de JSF tanto libres como comerciales y la mayoría de IDEs soportan este Framework.

Actualmente los sistemas informáticos están presentes en la mayoría de las actividades laborales del hombre, facilitando su trabajo y disminuyendo el tiempo que le lleva realizarlo. Encontramos sistemas informáticos en instituciones financieras, hospitales, aeropuertos, medianas y grandes empresas, sin ser una excepción las instituciones educativas; la mayoría de éstas debido al costo que implica adquirir un software no lo han hecho, en vista de esta necesidad por parte de la mayoría de colegios fiscales, seleccioné un colegio fiscal al azar que no tenía un sistema informático, del cual se obtuvo los requerimientos para desarrollar el sistema.

1. APLICACIONES WEB

En un principio la web era sencillamente una colección de páginas estáticas, documentos, etc., para su consulta o descarga. El paso inmediato en su evolución fue la inclusión de un método para elaborar páginas dinámicas es decir, generado a partir de los datos de la petición. Este método fue conocido como CGI y definía un mecanismo mediante el que se podía pasar información entre el servidor y ciertos programas externos. Los CGIs siguen utilizándose ampliamente; la mayoría de los servidores web permiten su uso debido a su sencillez. Además, dan total libertad para elegir el lenguaje de programación que se desea emplear.

El funcionamiento de los CGIs tenía un punto débil: cada vez que se recibía una petición, el servidor debía lanzar un proceso para ejecutar el programa CGI. Como la mayoría de CGIs estaban escritos en lenguajes interpretados, como Perl o Python, o en lenguajes que requerían un entorno de ejecución, como Java o Visual Basic, el servidor se veía sometido a una gran carga y la concurrencia de múltiples accesos al CGI podía ocasionar graves problemas.

Para solucionar este inconveniente se llega a dos posibles soluciones:

1. Diseñar sistemas de ejecución de módulos que estén mejor integrados con el servidor y que evitan la instanciación y ejecución de varios programas.
2. Se dota a los servidores de un intérprete de algún lenguaje de programación que permita incluir el código en las páginas de forma que lo ejecute el servidor, reduciendo el intervalo de respuesta.

Debido a esto se experimenta un aumento del número de arquitecturas y lenguajes que permiten desarrollar aplicaciones web. Todas siguen alguna de las dos soluciones propuestas. Las más útiles y las más utilizadas son las que permiten mezclar las 2 opciones: un lenguaje integrado que permita al servidor interpretar comandos "incrustados" en las páginas HTML y además, un sistema de ejecución de programas mejor enlazado con el servidor, que no implique los problemas de rendimiento propios de los CGIs.

Una de las más potentes es la seguida por Sun Microsystems con su Java integrado por 2 componentes; un lenguaje que permite la incrustación de código en las páginas

HTML que el servidor convierte en programas ejecutables, JSP o Páginas de Servidor de Java, y un método de programación ligado al servidor, con un rendimiento superior a los CGI, denominado Java Servlet. [www 01]

1.1. APLICACIONES WEB EN LA ACTUALIDAD

Con la introducción de Internet y de la Web en concreto, se han abierto infinidad de posibilidades en cuanto al acceso a la información desde casi cualquier sitio. Esto representa un desafío a los desarrolladores de aplicaciones, ya que los avances en tecnología demandan cada vez aplicaciones más rápidas, ligeras y robustas que permitan utilizar la Web.

La creación de aplicaciones web requiere la existencia de software ejecutándose en el servidor, que genere automáticamente los ficheros HTML que se visualizan en el navegador del usuario. La comunicación entre el cliente y el servidor se realiza a través del protocolo HTTP y éste delega a otros módulos la generación dinámica de páginas HTML que se envían al cliente, el navegador cliente es independiente de la tecnología que se utilice en el servidor para generar páginas de forma dinámica.



Figura 1.2: El contenido que se le muestra al usuario se genera dinámicamente para cada solicitud proveniente del navegador web instalado en la máquina del cliente. [www 02]

Actualmente existen potentes herramientas para generar dinámicamente el contenido que se le ofrece al usuario, puede optar por desarrollar software que se ejecute en el servidor, o en la propia máquina del usuario.

Básicamente las distintas alternativas disponibles para el desarrollo de aplicaciones web ofrecen la misma funcionalidad, sin embargo en función de las necesidades de cada proyecto algunas resultarán más apropiadas que otras, y el problema está en decidir cual tecnología es la apropiada para cada situación.

Entre estas nuevas tecnologías tenemos las Java Server Pages o Páginas de Servidor Java que permiten juntar HTML, aplicaciones JAVA y componentes como Java Beans creando una página Web especial que el Servidor Web compila dinámicamente en un Servlet la primera vez que es llamada.

Las Aplicaciones Web pueden considerarse como un sitio web al que se accede mediante un navegador o browser, y que está dotado de: interactividad, una base de datos, comunicaciones encriptadas y contraseñas de acceso.

En las Aplicaciones Web tanto el software como los datos se encuentran en una máquina remota, centralizada que permite trabajar de manera estable, continua e ininterrumpida. El usuario se conecta a este servidor a través de un navegador web.

La plataforma Web aporta múltiples ventajas frente al software de escritorio lo que hace de ésta una interesante alternativa:

- ✚ **Multiusuario.**- en las aplicaciones web puede haber varios usuarios conectados al sistema simultáneamente, cada uno a través de un computador diferente y en distintas ubicaciones geográficas; y todos utilizar la aplicación con absoluta normalidad.
- ✚ **Independencia de software.**- para acceder a la aplicación sólo es requisito un navegador web, sin necesidad de instalar en cada equipo otro software especial. Debido a estos bajos requerimientos, la aplicación incluso puede ser utilizada desde ordenadores poco potentes.
- ✚ **Seguridad.**- al encontrarse la aplicación en un servidor remoto, el funcionamiento de la misma y los valiosos datos que contiene son totalmente independientes del ordenador utilizado para acceder a la aplicación; por lo que si se presenta un daño de hardware, un virus informático o cualquier otro problema local bastará con iniciar una conexión al servidor desde cualquier otro ordenador.

- + **Multiplataforma e Interoperabilidad.**- a diferencia de las aplicaciones de escritorio que solo pueden funcionar bajo el sistema para el cual fueron diseñadas, las aplicaciones web son multiplataforma por diseño; esto significa que igual podrá ingresar a la aplicación desde un ordenador con un Sistema Operativo Windows, Linux u otro.

1.2. PROCESAMIENTO DE PÁGINAS DINÁMICAS

Cuando un servidor Web recibe una petición para mostrar una página Web estática, el servidor la envía directamente al navegador que la solicita. Cuando el servidor Web recibe una petición para mostrar una página dinámica, reacciona de distinta forma: transfiere la página a un software especial encargado de finalizar la página. Este software especial se denomina servidor de aplicaciones.

El servidor de aplicaciones lee el código de la página, y finaliza la página en función de las instrucciones del código y elimina el código de la página. El resultado es una página estática que el servidor de aplicaciones devuelve al servidor Web, que a su vez la envía al navegador solicitante. Lo único que el navegador recibe cuando llega la página es código HTML puro.

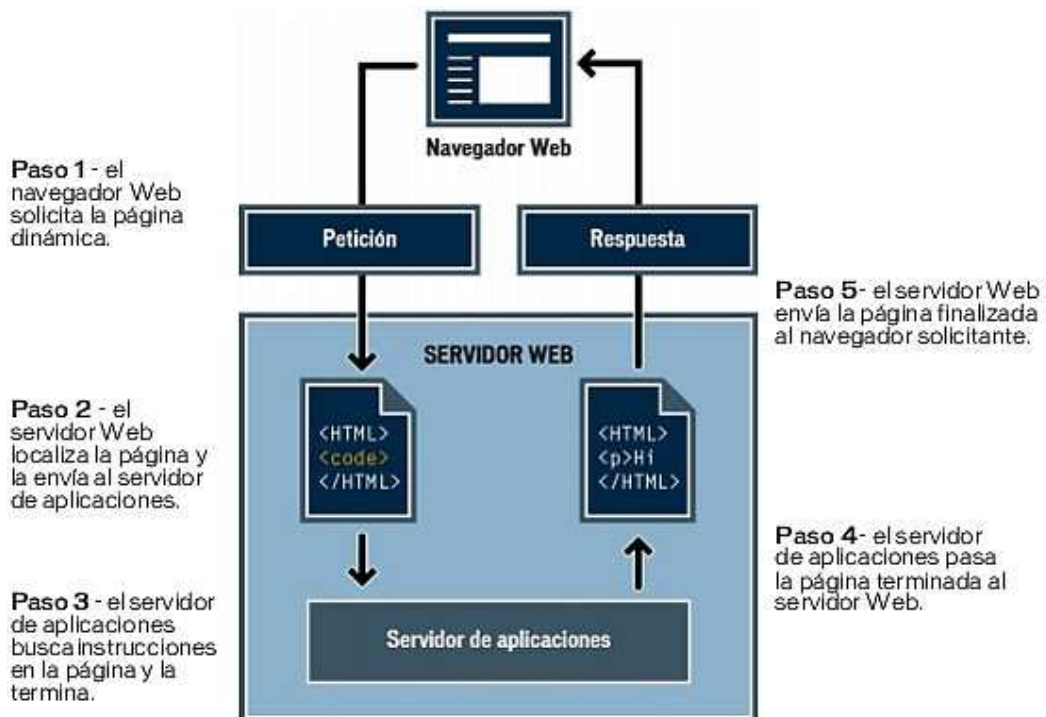


Figura 1.3: Esquema de Procesamiento de Páginas Dinámicas [www 03]

1.3. JAVA COMMUNITY PROCESS

El Proceso de la comunidad Java o Java Community Process, establecido en 1998, es un proceso formalizado el cual permite a las partes interesadas involucrarse en la definición de futuras versiones y características de la plataforma Java.

El JCP conlleva el uso de JSR, las cuales son documentos formales que describen las especificaciones y tecnologías propuestas para que sean añadidas a la plataforma Java. Las revisiones públicas formales de JSRs son controladas antes de que los JSR se conviertan en final y sean votados por el Comité Ejecutivo JCP. Un JSR final suministra una implementación de referencia la cual da una implementación libre de la tecnología en código fuente y un Kit de Compatibilidad de Tecnología para verificar la especificación de la API.

El mismo JCP está descrito por un JSR; desde el año 2006 la versión actual de JCP es la 2.6 como se describe en el JSR 215. [www 04]

Hay más de 300 JSRs algunas de las más importantes son:

JSR #	ESPECIFICACIÓN O TECNOLOGÍA
5	Java API for XML Processing (JAXP) 1.0
31	Arquitectura Java para Enlazado XML (JAXB) 1.0
37	Perfil de Dispositivo de Información Móvil (MIDP) 1.0 para Java ME
52	Librería de Etiquetas Estándar de Páginas Java Server (JSTL) 1.0 y 1.1
53	Especificaciones Java Servlet 2.3 y Java Server Pages (JSP) 1.2
54	Conectividad de Base de Datos Java (JDBC) 3.0
152	Java Server Pages (JSP) 2.0
198	Una API de Extensión Estándar para Entornos de Desarrollo Integrados (IDE)
215	Proceso de Comunidad Java (JCP) 2.6
220	Enterprise Java Beans (EJB) 3.0
224	Plataforma Java, Edición Empresas (Java EE) 5
252	Java Server Faces (JSF) 1.2

924	Especificación de la Máquina Virtual Java, Segunda Edición (JVM) (para J2SE 5.0)
-----	--

Tabla 1.1: Algunas de las JSRs más importantes

1.4. PLATAFORMA JAVA

Plataforma JAVA es el nombre de un entorno de computación de Sun Microsystems, capaz de ejecutar aplicaciones desarrolladas usando el lenguaje de programación JAVA u otros lenguajes que compilen a bytecode y un conjunto de herramientas de desarrollo.

La plataforma JAVA está constituida por una máquina virtual encargada de la ejecución de aplicaciones, y un conjunto de librerías estándar que ofrecen funcionalidad común.

La plataforma JAVA incluye:

- 🚧 Plataforma Java, Edición Estándar JAVA SE
- 🚧 Plataforma Java, Edición Empresa JAVA EE
- 🚧 Plataforma Java, Edición Micro JAVA ME

1.5. JAVA RUNTIME

Un programa destinado a la Plataforma Java necesita dos componentes en el sistema donde se va a ejecutar:

- 🚧 Una máquina virtual de Java JVM.
- 🚧 Un conjunto de librerías para proporcionar los servicios que pueda necesitar la aplicación.

La JVM que proporciona Sun Microsystems, junto con su implementación de las librerías estándar, se conocen como JRE o Entorno en tiempo de ejecución para Java. El JRE es lo mínimo que debe contener un sistema para poder ejecutar una aplicación Java sobre el mismo.

1.5.1. MÁQUINA VIRTUAL DE JAVA

El corazón de la Plataforma Java es el concepto común de un “procesador virtual” que ejecuta programas escritos en el lenguaje de programación Java. En concreto, ejecuta

el código resultante de la compilación del código fuente, conocido como bytecode. Este “procesador” es la máquina virtual de Java, que se encarga de traducir bytecode en instrucciones nativas de la plataforma destino. Esto permite que una misma aplicación Java pueda ser ejecutada en una gran variedad de sistemas con arquitecturas distintas, siempre que tenga una implementación adecuada de la JVM. Este hecho es lo que ha dado lugar a la famosa frase: “Write once, run anywhere” que significa “Escribir una vez, ejecutar en cualquier parte”.

Desde la versión 1.2 de JRE, la implementación de la máquina virtual de Sun incluye un compilador JIT. De esta forma, en vez de la tradicional interpretación del código bytecode, que da lugar a una ejecución lenta de las aplicaciones, el JIT convierte el bytecode a código nativo de la plataforma destino. Esta segunda compilación del código penaliza en cuanto a tiempo, pero el código nativo resultante se ejecuta de forma más eficaz y rápida que si fuera interpretado. Otras técnicas de compilación dinámica del código durante el tiempo de ejecución permiten optimizar más el código, dejando atrás lo que sobre Java recaía en cuanto lentitud y en sus últimas versiones la JVM se ha optimizado a tal punto que ya no se considera una plataforma lenta en cuanto a ejecución de aplicaciones.

La JVM fue diseñada para ser implementada completamente en software, y al mismo tiempo hacer que fuera portable a todo tipo de hardware. [www 05]

1.5.2. LIBRERÍAS DE JAVA

En la mayoría de los sistemas operativos actuales, se ofrece una cantidad de código para simplificar la tarea de programación. Este código es un conjunto de librerías dinámicas que las aplicaciones pueden llamar cuando lo necesiten. Pero la Plataforma Java está pensada para ser independiente del sistema operativo, por lo que las aplicaciones no pueden apoyarse en funciones dependientes de cada sistema en concreto. Lo que hace la Plataforma Java, es ofrecer un conjunto de librerías estándar, que contienen muchas de las funciones reutilizables disponibles en los sistemas operativos actuales.

Las librerías de Java tienen tres propósitos dentro de la Plataforma Java. Al igual que otras librerías estándar, ofrecen al programador un conjunto bien definido de funciones para realizar tareas comunes, como manejar listas de elementos u operar de forma sofisticada sobre cadenas de caracteres. Además, las librerías proporcionan una interfaz abstracta para tareas que son altamente dependientes del hardware de la plataforma destino y de su sistema operativo. Tareas como acceso a ficheros, suelen depender fuertemente de la funcionalidad nativa de la plataforma destino. Finalmente, no todas las plataformas soportan todas las funciones que una aplicación Java espera. En estos casos, las librerías bien pueden emular esas funciones usando lo que esté disponible, o bien ofrecer un mecanismo para comprobar si una funcionalidad concreta está presente. [www 06]

1.6. PLATAFORMA JAVA EDICIÓN EMPRESARIAL

Java EE o Plataforma Java Edición Empresarial, es una plataforma para desarrollo de aplicaciones empresariales, escalable, robusta, multiplataforma y bien documentada. JAVA EE es parte de la plataforma Java; para desarrollar y ejecutar aplicaciones en lenguaje de programación Java.

La especificación original J2EE fue desarrollada por Sun Microsystems. Comenzando con J2EE 1.3, la especificación fue desarrollada bajo el JCP, con el JSR 58 y JSR 151 especifica J2EE 1.4. El SDK de J2EE 1.3 fue liberado inicialmente como beta en Abril de 2001. El SDK de J2EE 1.4 fue liberado como beta por Sun en Diciembre de 2002. La especificación de Java EE 5 fue desarrollada bajo el JSR 244 y la liberación final se realizó el 11 de Mayo de 2006.





La plataforma Java EE está definida por una especificación; similar a otras especificaciones del JCP.

Java EE incluye varias especificaciones de API como: JDBC, Servicios Web, XML, etc.; y define como coordinarlos. Java EE también configura algunas especificaciones únicas para Java EE para componentes; estas incluyen Enterprise Java Beans, Java Server Pages entre otros. Esto permite al desarrollador crear una Aplicación de Empresa

portable entre plataformas y escalable, que a la vez se integre con tecnologías anteriores.

Uno de los beneficios de Java EE como plataforma es que es posible empezar con poco o ningún coste. La implementación de Java EE de Sun Microsystems, puede ser descargada gratuitamente, y hay muchas herramientas de código abierto disponibles para simplificar el desarrollo en esta plataforma. [www 07]

Ejemplos de herramientas IDE de desarrollo Java de código abierto:

-  NetBeans
-  Eclipse
-  Jedit
-  JDeveloper, gratuito desarrollado por Oracle.

2. FRAMEWORKS MVC

2.1. DEFINICIÓN DE PATRÓN DE ARQUITECTURA

Un patrón de arquitectura, describe la estructura general de un sistema, identifica sus módulos, responsabilidades y la colaboración entre dichos módulos. [www 08]

2.2. ARQUITECTURA MODELO VISTA CONTROLADOR MVC

Básicamente las aplicaciones web tienen el problema de mezclar el código de interfaz de usuario con el código de la lógica del negocio; y cuando esto sucede las aplicaciones son más difíciles de mantener, menos escalables, y generalmente más frágiles; y el patrón MVC es una solución a este problema, ya que permite desacoplar el modelo de la presentación de los datos o vista.

MVC es un patrón de arquitectura de software que tuvo origen en Smalltalk; fue diseñado para reducir el esfuerzo de programación necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos; donde las responsabilidades estén bien distribuidas entre las distintas partes del diseño; así se decidió distinguir tres responsabilidades distintas:

- ✚ Lógica de Negocio.
- ✚ Gestión de eventos de usuario.
- ✚ Presentación o Interfaz de usuario.

La Vista es la interfaz de usuario y se encarga de interactuar con él, el modelo comprende la lógica de negocio y los datos y el controlador responde a eventos del usuario e integra el modelo y la vista. Con esta arquitectura logramos llegar a obtener una aplicación débilmente acoplada ya que reduce las dependencias entre sus diferentes capas.

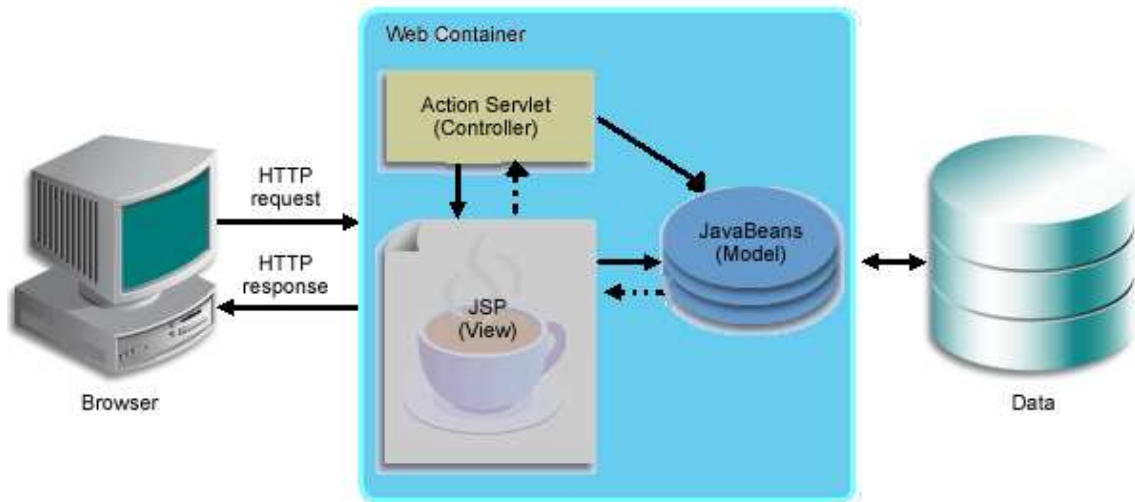


Figura 2.1: Arquitectura Modelo Vista Controlador. [REF 01]

Su principal característica es que Modelo, Vista y Controlador se tratan como entidades separadas.

Para usar MVC de manera eficiente debemos entender la división de responsabilidades en: Modelo, Vista, Controlador y como se comunican entre sí. [www 09]

2.2.1. MODELO

El Modelo representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El Modelo no tiene conocimiento específico del Controlador o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

El modelo son las operaciones que se efectúan sobre los datos que se reciben o las consultas que se hacen a la base de datos, su objetivo es preparar los datos para que la vista solo se preocupe de pintarlos, entonces si hay que realizar cambios en la estructura de la base de datos o cambiar las operaciones que se hacen con los datos lo tendríamos que hacer sólo en el modelo, evitando que un cambio se tenga que realizar en todas las vistas que utilicen unos determinados datos.

2.2.2. VISTA

La vista está representada por un conjunto de pantallas que representan la interfaz de usuario, es decir lo que se muestra a los ojos del usuario.

La Vista maneja la presentación visual de los datos representados por el Modelo. Genera una representación visual del Modelo, muestra los datos al usuario e interactúa con él.

La tecnología Java indicada para la generación de vistas en aplicaciones web son las JavaServer Pages.

2.2.3. CONTROLADOR

El Controlador es el enlace entre el usuario y la aplicación, el controlador opera dentro del ciclo de vida JSF. El controlador da significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo. Cuando se realiza algún cambio, entra en acción, ya sea por cambios en los datos del Modelo o por alteraciones de la Vista.

El controlador es el responsable de recibir los eventos, determinar el procesador del evento, invocar al procesador y finalmente provocar la generación de la vista apropiada.

Tareas del controlador:

- Prepara el contexto JSF.- objeto Java que mantiene todos los datos de la aplicación. El contexto provee a las páginas acceso a datos de la aplicación.
- Controlar la seguridad.- impidiendo accesos inadecuados o no autorizados a las páginas.
- Identificación de eventos.
- Preparar el modelo.
- Procesar el evento.
- Manejar los errores.
- Provocar la generación de la respuesta.

2.2.4. FLUJO DE CONTROL EN MVC

- a) El usuario interactúa con la interfaz de usuario de alguna forma por ejemplo, el usuario pulsa un botón.
- b) El controlador recibe por parte de los objetos de la vista la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega.
- c) El controlador accede al modelo actualizándolo o modificándolo de forma adecuada a la acción solicitada por el usuario.
- d) El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo por ejemplo, produce un listado del contenido del carro de la compra.
- e) La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente. [www 09]

2.2.5. VENTAJAS DE MVC

El utilizar un patrón como MVC en el desarrollo de Aplicaciones Web trae ventajas muy importantes como:

- ✚ Llegar a una solución de más calidad.
- ✚ Mejor mantenibilidad.
- ✚ Tener un patrón a seguir al empezar un proyecto.
- ✚ Facilidad para realizar cambios en la aplicación.
- ✚ Podemos modificar uno de los componentes, sin conocer cómo funcionan los otros.
- ✚ Hay una clara separación entre la interfaz de usuario y la programación en sí, lo cual permite implementarlos por separado.
- ✚ Hay un API muy bien definido; cualquiera que use el API, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- ✚ La conexión entre el Modelo y las Vistas es dinámica, se produce en tiempo de ejecución, no en tiempo de compilación.
- ✚ Y lo más importante normalizar y estandarizar el desarrollo de software.

2.3. DEFINICIÓN DE FRAMEWORK

El concepto Framework se emplea en muchos ámbitos del desarrollo de software, no solo en el ámbito de aplicaciones Web. Podemos encontrar Frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, y para cualquier ámbito.

En el desarrollo de software, un Framework es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado. Por lo general, un Framework puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Un Framework representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. [www 10]

2.4. DEFINICIÓN DE FRAMEWORK WEB

Un Framework Web, se define como una estructura definida, reusable en el que sus componentes facilitan la creación de aplicaciones web.

Estos Frameworks nos proveen un conjunto de componentes como por ejemplo: clases en java, descriptores y archivos de configuración en XML; que componen un diseño reutilizable que facilita y agiliza el desarrollo de Sistemas Web.

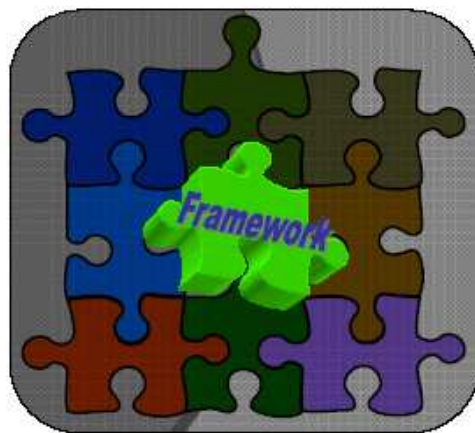


Figura 2.2: Representación gráfica de Framework

2.4.1. OBJETIVOS DE UN FRAMEWORK WEB JAVA EE

Los objetivos principales de un Framework son:

- ✚ Facilitar el desarrollo de aplicaciones empresariales.
- ✚ Acelerar el proceso de desarrollo.
- ✚ Reutilización de código.
- ✚ Promover buenas prácticas de desarrollo, como el uso de patrones.

2.4.2. TIPOS DE FRAMEWORKS WEB

Según cuál sea el tipo de aplicación web en la que se especializa un Framework tenemos los siguientes tipos:

- ✚ **Frameworks para publicación web.**- centrados en el manejo y presentación de contenidos, como: JPublish y Cocoon.
- ✚ **Frameworks para la explotación de bases de datos.**- centrados en la gestión de datos almacenados en bases de datos relacionales.
- ✚ **Frameworks de presentación.**- centrados en el desarrollo de aplicaciones con interfaces ricas y altamente interactivas.

Según la arquitectura del Framework se clasifican en:

- ✚ **Frameworks basados en transformaciones de documentos XML** la mayoría de este tipo de Frameworks están especializados en la publicación web. Su funcionamiento se basa en la transformación de documentos fuente, mezclados con datos dinámicamente que se transforman para dar lugar a un documento final que puede estar en diversos formatos, principalmente HTML y PDF. El ejemplo más característico de este tipo de Frameworks es Cocoon.
- ✚ **Frameworks basados en el encadenamiento de acciones** de recepción de información de un formulario, selección de un objeto al que pasarle esa información para que la procese y selección de la salida tras realizar el procesamiento. Todo ello basado en la configuración dada en un fichero. Esta es la arquitectura de Framework más común. Ejemplos de él son: Struts, Maverick, entre otros.

- ✚ **Frameworks basados en componentes** estos Frameworks suelen basar su funcionamiento en la definición de componentes reutilizables. Esta arquitectura es muy usada en los Frameworks de presentación, ya que el uso de componentes independientes favorece la interactividad de la aplicación. Ejemplos de estos Frameworks son Tapestry, Java Server Faces.

2.4.3. CARACTERÍSTICAS

Existen una serie de características que son comunes a todos los Frameworks como las siguientes: [www 11]

- ✚ **Abstracción de URLs y Sesiones.-** no es necesario manipular directamente las URLs ni las sesiones, ya que el Framework se encarga de hacerlo por nosotros.
- ✚ **Acceso a Datos.-** incluyen herramientas o interfaces necesarias para integrarse con herramientas de acceso a datos.
- ✚ **Controladores.-** la mayoría de Frameworks implementa controladores para gestionar eventos, como una introducción de datos mediante un formulario o el acceso a una página. Estos controladores se adaptan fácilmente a las necesidades de un proyecto en concreto.
- ✚ **Autenticación y Control de Acceso.-** incluyen mecanismos para la identificación de usuarios mediante login y password; y además permiten restringir el acceso a determinadas páginas a determinados usuarios; por ejemplo: un usuario tipo Administrador tendrá acceso a páginas a las que otro tipo de usuario no podrá acceder.
- ✚ **Internacionalización.-** permite mostrar la aplicación según el idioma del usuario.
- ✚ **Separación entre presentación y comportamiento de la aplicación.**

2.4.4. VENTAJAS DE USAR FRAMEWORKS

El uso de Frameworks en el desarrollo de aplicaciones web nos proporciona ventajas como:

- ✚ **Modularidad y reducción de la complejidad.**- la aplicación está formada por subsistemas especializados en distintos aspectos fundamentales de toda aplicación como: presentación, persistencia.
- ✚ **Fortaleza al cambio.**- los módulos pueden evolucionar o cambiar conservando la arquitectura global de la aplicación.
- ✚ **Documentación.**- la documentación del Framework promueve el uso correcto del mismo y disminuye el esfuerzo necesario para el mantenimiento.
- ✚ **Estructura.**- el desarrollo basado en Frameworks establece una estructura sobre la cual las aplicaciones pueden ser construidas, liberando al desarrollador de tomar el 100% de las decisiones de diseño.
- ✚ **Distribución de funciones.**- permite realizar en paralelo el trabajo de desarrollo, ya que la aplicación puede desarrollarse como un conjunto de piezas independientes que encajarán en el Framework usado.
- ✚ **Eficiencia.**- el desarrollador puede concentrarse en los requerimientos funcionales de la aplicación.
- ✚ **Aplicaciones ricas.**- posibilidad de dar más funcionalidad a los usuarios de la aplicación.
- ✚ **Mejoran notablemente la calidad del software resultante.**- Ya que el desarrollador puede dedicarse a resolver problemas específicos de la lógica del negocio, y no preocuparse por los detalles de programación de bajo nivel.

2.4.5. FRAMEWORKS MVC

LENGUAJE	LICENCIA	NOMBRE
Ruby	MIT	Ruby on Rails
Java / J2EE	Apache	Struts
Java / J2EE	Apache	Spring Framework
Java / J2EE	Apache	Tapestry
Java / J2EE	Apache	Aurora
Java / J2EE	Apache	Java Server Faces
Perl	GPL	Catalyst

PHP	BSD	Zend Framework
PHP	MIT	CakePHP
PHP	GNU / GPL	Kumbia
PHP	MIT	Symfony
PHP	MIT	QCodo
PHP	GNU / GPL	CodeIgniter
PHP	Otra	Kohana
Python	ZPL	Zope3
Python	Varias	Turbogears
Python	BSD	Django
.NET	Castle Project	MonoRail
.NET	Apache	Spring.NET
.NET	Apache	Maverick.NET

Tabla 2.1: Frameworks MVC

2.5. STRUTS

Struts es un Framework Web J2EE de software libre que implementa el patrón de arquitectura MVC y es uno de los más difundidos y utilizados por los desarrolladores Java.

Como todo Framework, Struts simplifica notablemente la implementación de una arquitectura según el patrón MVC.

Struts está construido básicamente sobre las tecnologías de Servlets y JSP. Struts combina Servlets, JSPs, custom tags propios y recursos de la aplicación en un único Framework. Struts fue creado por Craig McClanahan y forma parte del proyecto Apache Struts, de Apache Software Foundation.

Struts oculta al programador los detalles del protocolo HTTP, JSP, Servlets, etc. Un programador Struts puede desconocer estos nombres, sin embargo tener conocimiento de las tecnologías de base de Struts hace que se puedan hacer soluciones creativas.

El corazón de Struts es el Servlet Controlador u objeto `ActionServlet`, el cual intercepta todos los requerimientos HTTP entrantes, provenientes de los clientes y los delega a un manejador apropiado que es un objeto `Action`. Para determinar el flujo de la aplicación, es decir a que `Action` pasar el requerimiento, hace uso del archivo `struts-config.xml`. El Servlet Controlador, luego recibe las respuestas de los `Action` y los redirecciona a la vista apropiada. Para ello, nuevamente consulta un conjunto de mapeos definidos en el archivo de configuración XML.

Existen múltiples objetos `ActionForms`. Estos objetos son `JavaBeans` usados para mantener los datos ingresados por el usuario en las páginas JSP. Un punto clave del Framework Struts es que automáticamente llena los objetos `ActionForm` con datos de la petición del usuario.

Si bien Struts está basado en el patrón de diseño MVC2, solo provee componentes para las capas Vista y Controlador. Struts no provee componentes para el modelo, el cual generalmente se implementa con `JavaBeans`. [www 12]

2.5.1. CARACTERÍSTICAS DE STRUTS

Entre las principales características de Struts están:

- ✚ Arquitectura basada en el patrón arquitectónico MVC, en el que todas las peticiones son procesadas por el Controlador, que despacha las peticiones a los componentes apropiados de la aplicación, basándose en los identificadores lógicos que reducen el acoplamiento entre capas.
- ✚ Configuración del control centralizado.
- ✚ Las interrelaciones entre acciones y página u otras acciones se definen en un archivo XML de configuración, en lugar de codificarlas en las páginas.
- ✚ Componentes de aplicación, son el mecanismo para compartir información bidireccionalmente entre el usuario y las acciones del modelo.
- ✚ Librerías de entidades para facilitar la mayoría de las operaciones que realizan las páginas JSP.
- ✚ Struts contiene herramientas para validación de campos de plantillas y puede ser de dos formas: validación local en la página y validaciones a nivel acciones.

Struts permite al desarrollador concentrarse en el diseño de la aplicación, como una serie de componentes del Modelo y de la Vista, intercomunicados por un control centralizado; esto permite obtener una aplicación más consistente y más fácil de mantener.

2.5.2. ARQUITECTURA DE UNA APLICACIÓN STRUTS

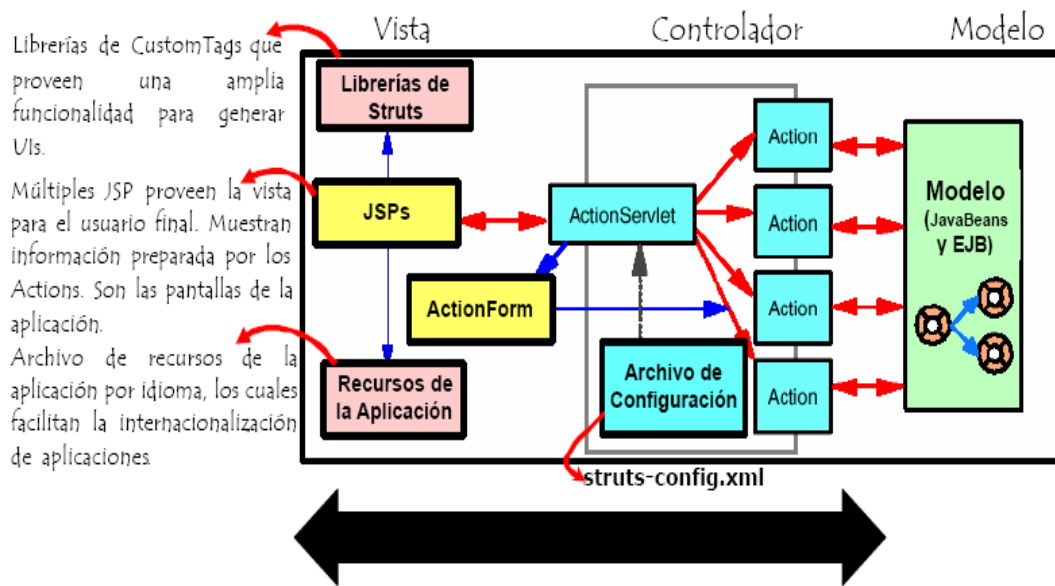


Figura 2.3: Arquitectura de una aplicación Struts. [www 13]

2.5.3. COMO FUNCIONA STRUTS

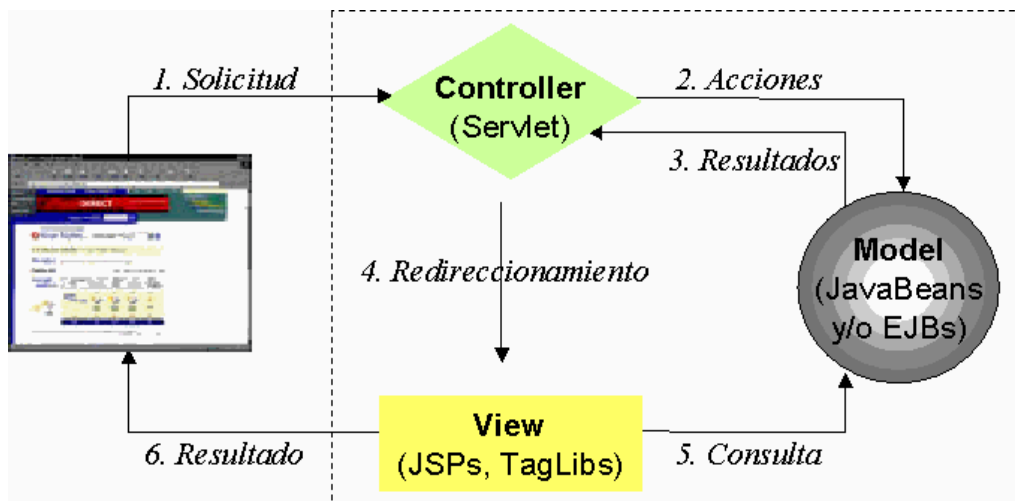


Figura 2.4: Funcionamiento de Struts. [www 14]

Cuando navegador genera una solicitud ésta es atendida por el Controlador que es un Servlet especializado y proporcionado por Struts; que se encarga de analizar la solicitud, seguir la configuración programada en el archivo Struts-config.xml y llamar al Action correspondiente pasándole los parámetros enviados. El Action instanciará y/o utilizará los objetos de negocio para realizar la tarea solicitada. Y según el resultado que retorne el Action, el Controlador derivará la generación de interfaz a una o más JSPs, las cuales podrán consultar los objetos del Modelo a fines de realizar su tarea.

2.5.4. INTERNACIONALIZACIÓN

Para dotar a una aplicación de internacionalización es necesario crear un archivo nombre_xx.properties donde xx es el código ISO de idioma por ejemplo mensajes_en.properties para Inglés, el contenido de este archivo será las claves y los valores con el formato clave.subclave = texto. Así se tendrá un archivo por cada idioma que deseemos para la aplicación. Para mostrar el mensaje se debe utilizar el Tag <bean:message key="login.usuario"> que corresponde al mensaje que se mostrará según el idioma del usuario.

Ejemplo:

Para Español	login.usuario=Usuario
Para Inglés	login.usuario=User

En el archivo struts-config.xml se debe configurar el servlet/init-param/param-name y /param-value donde irá la localización del archivo con el idioma principal. [www 15]

2.5.5. VENTAJAS DE STRUTS

- ✚ Struts es una herramienta Open Source licencia Apache.
- ✚ Es una herramienta estable, es analizado constantemente por sus creadores por lo que sus fallos son resueltos rápidamente.
- ✚ Struts cuenta con excelente documentación tanto como en libros y en Internet; además de la información disponible en su página oficial.

- ✚ En Internet se puede encontrar muchos ejemplos que son de gran ayuda al momento de empezar a desarrollar aplicaciones con este Framework.
- ✚ Promueve la reutilización de código.

2.5.6. DESVENTAJAS DE STRUTS

- ✚ Se crean nuevas versiones con mucha frecuencia por lo que si creamos una aplicación con Struts puede tener incompatibilidades con las versiones más recientes de este Framework.
- ✚ Aparición de nuevos Frameworks que son más sencillos de implementar que Struts.
- ✚ No abstrae completamente al desarrollador del funcionamiento del protocolo HTTP.
- ✚ No está diseñado para facilitar la creación de componentes propios.
- ✚ No es una especificación.
- ✚ No permite la creación de vistas independientes del dispositivo cliente.

2.6. TAPESTRY

Tapestry es un Framework de código abierto para la creación de aplicaciones web de forma dinámica, robusta y altamente escalable en Java.

Tapestry es un Framework para el desarrollo de aplicaciones web, basado en componentes, no usa JSP ni ningún motor de plantillas para las vistas, sino que es una alternativa a estas tecnologías.

En Tapestry todo son componentes y los componentes se pueden anidar unos dentro de otros para formar componentes más complejos. Estos componentes intercambian información entre ellos a partir de sus parámetros de forma que, para programar usando este Framework, tenemos que pensar en términos de propiedades, métodos y objetos.

Tapestry se encarga de manejar por nosotros las propiedades persistentes, la validación de formularios, la localización, etc., permitiendo al desarrollador centrarse en la lógica del negocio.

Tapestry divide una aplicación web en un conjunto de páginas, cada una compuesta de componentes. Esto le otorga una estructura consistente, permitiendo a Tapestry asumir responsabilidades clave como: la construcción y envío URL, almacenamiento de estado persistente en el cliente o en el servidor, validación de entrada de usuario, internacionalización, y reporte de excepciones.

Desarrollar aplicaciones Tapestry implica crear plantillas HTML usando HTML plano, y combinando las plantillas con pequeños trozos de código Java usando un descriptor de archivos XML.

En Tapestry, se crea las aplicaciones en términos de objetos, y los métodos y propiedades de estos objetos, y no especificando términos de URLs.

Tapestry ofrece un desarrollo realmente orientado a objetos a las aplicaciones web Java. Tapestry está diseñado para facilitar la creación de nuevos componentes.

2.6.1. PÁGINAS Y COMPONENTES

Tapestry divide una aplicación en un conjunto de páginas. Cada página está formada por componentes y cada componente, a su vez, puede estar formado por otros componentes, sin que exista ningún límite en la profundidad de anidamiento. Las páginas en Tapestry también son componentes, pero con algunas responsabilidades especiales.

Los componentes pueden tener parámetros, los cuales pueden ser establecidos por la página o el componente que los contiene. A diferencia de lo que ocurre con los parámetros de los métodos en Java, los parámetros de los componentes de Tapestry pueden ser bidireccionales; un componente puede leer un parámetro para obtener un valor, o escribir un parámetro para establecerlo.

Todos los componentes de Tapestry, así como las páginas que no dejan de ser componentes especiales están formados por:

- ✚ **Una especificación.**- La especificación es un documento XML en el que se indica el nombre del componente, qué parámetros y propiedades posee y de qué tipo son, por qué componentes está formado y cómo se conectan los parámetros de dichos componentes, entre otras cosas.

- ✚ **Una plantilla.**- La plantilla está formada en su totalidad por etiquetas HTML. Tapestry no añade ninguna etiqueta especial, por lo que ésta puede ser editada con cualquier editor de HTML; Tapestry añade, únicamente, el atributo `jwcid` que indica que la etiqueta en la que aparece representa a un componente. Por tanto, en la plantilla se define la apariencia visual del componente, situando los componentes contenidos en él en la posición deseada.
- ✚ **Una implementación.**- cada componente tiene una clase que implementa su comportamiento, conectando propiedades a esta clase y ejecutando los métodos adecuados cuando se produzcan los eventos.

2.6.1.1. PÁGINAS

Las páginas en Tapestry son un tipo especial de componentes. Se trata de los componentes más externos, ya que no pueden estar contenidas dentro de otros componentes y por lo tanto no pueden tener parámetros.

Ficheros que constituyen una página:

- ✚ **Especificación (*.page).**- la especificación de una página se realiza en un fichero XML, con extensión `.page`.
- ✚ **Plantilla (*.html).**- La plantilla de una página es un fichero en HTML en el cual se describe la apariencia visual de la página. La inserción del atributo `jwcid` en una etiqueta determinada provoca que Tapestry sustituya toda la etiqueta por el código HTML producido por la renderización del componente al que representa.
- ✚ **Implementación (*.java).**- cada página por ser un componente tiene una clase que la implementa. El nombre de esta clase viene dado en la especificación.

2.6.1.1.1. ESTADO DE LAS PÁGINAS

Las páginas y los componentes tienen estado. El estado se refiere un conjunto de valores para las propiedades de las páginas.

En Tapestry la duración de la vida de cada propiedad es muy importante.

Se puede distinguir tres tipos de propiedades según su duración:

- ✚ **Persistentes:** los cambios en la propiedad son almacenados y hechos persistentes entre ciclos de petición. Las propiedades persistentes se restauran cuando la página se vuelve a cargar. Este tipo de propiedades son específicas para cada usuario. Por ejemplo: nombre de usuario.
- ✚ **Transitorias:** La propiedad se establece antes de que la página sea renderizada y será reseteada a su valor por defecto al final del ciclo de petición actual. Las propiedades transitorias son comúnmente datos usados sólo una vez, por ejemplo: un mensaje de error.
- ✚ **Dinámicas:** La propiedad cambia incluso mientras la página se renderiza, pero al igual que las transitorias es reseteada al final del ciclo de petición actual.

2.6.1.1.2. CARGA DE LAS PÁGINAS

El proceso de carga de una página es decir la instanciación de la página y de todos sus componentes puede ser algo costoso ya que implica leer la especificación de la página, así como la especificación de todos los componentes que conforman la página, además de localizar, leer y analizar las plantillas HTML de todos los componentes.

Este proceso lleva tiempo, no mucho en un servidor descargado, pero potencialmente más del aceptable en un sitio con alta carga. Por tanto sería un desperdicio crear páginas para después descartarlas al final del ciclo de petición. En lugar de esto, en Tapestry las páginas se usan durante el ciclo de petición y después se almacenan en un pool para ser reutilizadas posteriormente.

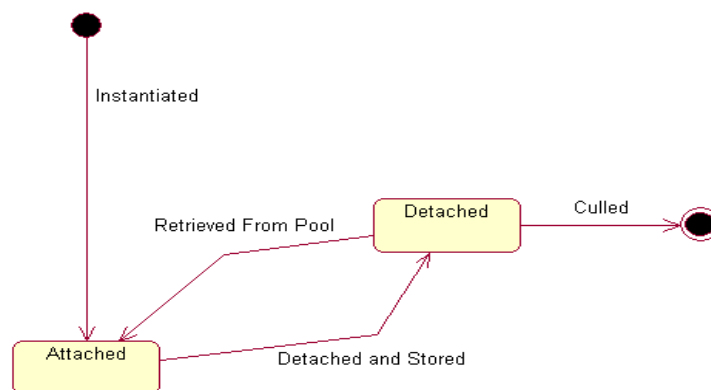


Figura 2.5: Ciclo de vida de página [www 16]

Las páginas permanecen en el pool hasta que se seleccionan para ser eliminadas, momento en el que el recolector de basura liberará la memoria usada por ella y todos los componentes que la forman. El comportamiento por defecto es eliminar las páginas no usadas después de aproximadamente diez minutos.

2.6.1.1.3. PÁGINAS REQUERIDAS

Cada aplicación requiere tener al menos cinco páginas con nombres específicos. Tapestry proporciona una implementación por defecto para cuatro de ellas.

NOMBRE DE LA PÁGINA	DESCRIPCIÓN
Exception	Es proporcionada por el Framework. Esta página es usada para presentar al usuario excepciones no capturadas.
Home	Proporcionada por el desarrollador. Es la página inicial mostrada cuando se inicia la aplicación.
Inspector	Es proporcionada por Tapestry y permite a las aplicaciones Tapestry ser interrogadas sobre su estructura.
StaleLink	Es proporcionada por Tapestry. Página mostrada cuando se lanza una excepción <code>StaleLinkException</code> durante el procesamiento de la petición. Esta excepción es lanzada cuando se detecta una URL incorrecta que no se corresponde con el estado de la aplicación en el lado del servidor.
StaleSession	Es proporcionada por Tapestry. Página mostrada cuando se lanza una excepción <code>StaleSessionException</code> durante el procesamiento de la petición. Esta excepción es lanzada cuando se detecta que la sesión ha caducado.

Tabla 2.2: Páginas de Tapestry

2.6.1.2. COMPONENTES

El desarrollo de componentes es similar al desarrollo de las páginas, pero los componentes pueden tener parámetros y las páginas no.

Ficheros que constituyen un componente:

- ✚ **Especificación (*.jwc).**- en la especificación del componente se indica que parámetros posee y de qué tipo son, además de que componentes lo forman y como se conectan los parámetros del componente padre con los hijos.
- ✚ **Plantilla (*.html).**- en la plantilla se indica en qué posición hay que renderizar cada uno de los componentes que forman el componente y las etiquetas HTML adicionales que darán formato al texto.
- ✚ **Implementación (*.java).**- en la implementación del componente creamos los métodos necesarios para resolver las expresiones indicadas usando la sintaxis de OGNL en la especificación.

2.6.1.3. PARÁMETROS Y LIGADURAS

Cada componente tiene un conjunto específico de parámetros, los cuales tienen un nombre, un tipo y pueden ser obligatorios u opcionales.

Los parámetros definen el tipo de valor necesitado, pero no el valor actual. El valor actual lo proporciona un objeto especial llamado ligadura o binding; que es un puente entre el componente y el valor del parámetro, exponiendo ese valor al componente cuando lo necesita.

Hay dos tipos de ligaduras: estática y dinámica. Las ligaduras estáticas son de sólo lectura y su valor es definido en la especificación del componente. Las ligaduras dinámicas son más usuales y útiles. Una ligadura dinámica usa el nombre de una propiedad JavaBean para obtener el valor cuando el componente lo necesita.

2.6.1.4. INTERNACIONALIZACIÓN

Cada componente puede tener un conjunto de cadenas localizadas, que se encuentran almacenadas en un fichero .properties. Este fichero debe estar en el mismo directorio que se encuentra la especificación del componente.

2.6.2. GESTIONANDO EL ESTADO EN EL LADO DEL SERVIDOR

El manejo del estado en el lado del servidor es uno de los aspectos más complicados y propensos a errores en el diseño de aplicaciones web. Hablar de estado en el lado del servidor se refiere a asegurar que determinada información sobre el usuario, adquirida durante la sesión, está disponible más tarde durante esa sesión. Un ejemplo típico se produce cuando un usuario introduce información en un formulario de login para acceder a un determinado contenido. Dicha información, que se introduce en la página de login, suele estar disponible en otras páginas.

2.6.2.1. ENGINE

Es el objeto principal encargado de mantener el estado en el servidor. El Engine se almacena como un atributo de la HttpSession.

Una aplicación Tapestry tiene un único servlet: ApplicationServlet. Cuando le llega una petición de un cliente, éste localizará el Engine asociado a la sesión del usuario que realiza la petición. Una vez localizado el Engine, el ApplicationServlet delegará en él la responsabilidad de responder a la petición. Para responder a la petición el Engine localizará el servicio que generó la URL y le solicitará que responda a la petición.

2.6.2.2. EL OBJETO VISIT

El objeto Visit es un objeto definido por la aplicación, que puede ser obtenido a partir del Engine. El nombre Visit enfatiza que cualquier dato almacenado en el Visit se refiere a una visita a la aplicación web.

Habitualmente, la clase Visit se define en la especificación de la aplicación, o como un `<init-parameter>` en el descriptor de despliegue de la aplicación web, web.xml.

2.6.2.3. EL OBJETO GLOBAL

El objeto Global es muy similar al objeto Visit con algunas diferencias importantes. Todas las instancias del Engine de la aplicación comparten el objeto Global, que se almacena como un atributo de ServletContext. El objeto Global no es persistente y

puede ser accedido usando la propiedad global tanto de la página como del Engine, ambos accesos son equivalentes.

2.6.3. CICLO DE PETICIÓN

Las aplicaciones web tienen una estructura significativamente diferente a otros tipos de aplicaciones interactivas, debido a la naturaleza del protocolo HTTP, que no mantiene el estado.

Tapestry permite a páginas y componentes creer que están en contacto continuo con el cliente. En esto juega un papel central el ciclo de petición, que es muy importante en Tapestry y que se representa mediante un objeto que se emplea durante el proceso de responder al cliente creando una página HTML.

El ciclo de petición está representado por el objeto RequestCycle. El RequestCycle es un objeto controlador que maneja un ciclo de petición. En una aplicación Tapestry, esto implica:

- ✚ Responder a la URL encontrando un objeto IEngineService.
- ✚ Determinar la página resultante.
- ✚ Renderizar la página resultante.
- ✚ Liberar cualquier recurso.

Mezcladas con estas responsabilidades, están:

- ✚ Manejar las excepciones.
- ✚ Cargar páginas y plantillas desde recursos.
- ✚ Manejar los cambios en las propiedades de las páginas y restaurar las páginas a su estado inicial.
- ✚ Pooling de páginas y objetos.

2.6.3.1. REBOBINADO Y RENDERIZACIÓN

Un ciclo de petición está dividido en dos fases:

Fase de rebobinado: Esta fase es opcional y está ligada a los componentes ActionLink y Form. En ella la página anterior se rehace hasta que se llega a un componente

específico. Este rebobinado asegura que la página se restaura al estado exacto que tenía cuando se generó la URL para el ciclo de petición. Una vez que se llega al componente que generó la URL se notifica a su `IActionListener`. El método oyente puede modificar el estado de cualquier página y seleccionar una nueva página resultante.

Fase de renderizado: La siguiente fase es la de renderizado. En ella se renderiza la página resultante y la salida se envía al navegador web del cliente.

2.6.3.2. ELECCIÓN DE LA PÁGINA A MOSTRAR

Durante el procesamiento del ciclo de petición, en cualquiera de los métodos oyentes, puede elegirse cual será la siguiente página a mostrar; para ello se puede llamar al método `activate(...)`, pasándole la página a mostrar o un `String` con el nombre de la página. La página activada será la que Tapestry renderizará para generar la respuesta. Si al final del ciclo llega la hora de renderizar la respuesta y no se ha activado otra página, se renderizará de nuevo la página actual. [www 17]

2.6.4. VENTAJAS

- Framework Open Source Licencia Apache.
- Permite la creación de componentes propios.
- El diseñador no necesita utilizar nuevos tags, ni aprender nuevos lenguajes, ya que los templates se codifican en lenguaje HTML.
- La creación de componentes es sencilla.
- Separación completa entre lógica y presentación.

2.6.5. DESVENTAJAS

- Comunidad de desarrolladores pequeña en comparación con la de otros Frameworks.
- La documentación es escasa tanto en libros como en la Web.
- Es necesario crear 3 archivos por cada página que tenga una aplicación.

2.6.6. CONCLUSIONES TAPESTRY

Tapestry es un Framework de presentación realmente basado en componentes, que facilita enormemente el desarrollo de interfaces ricas e interactivas.

En Tapestry todo son componentes. Este enfoque facilita la reusabilidad y el mantenimiento en las aplicaciones web: una vez diseñados los componentes necesarios, construir una aplicación no es más que unirlos.

Construir componentes con Tapestry es tan fácil como construir una página. La idea es muy simple: una plantilla HTML para definir la apariencia; una especificación para indicar qué componentes se usan en la página y cómo se relacionan sus parámetros y una implementación para programar métodos oyentes y proporcionar los datos necesarios.

2.7. FRAMEWORK JAVA SERVER FACES

A lo largo de todo este capítulo se menciona la palabra COMPONENTE UI y éstos son los controles que permiten construir la interfaz de usuario como: listas, botones, cajas de texto entre otros.

En JSF los componentes UI, son objetos con estado del lado del servidor, son Java Beans con propiedades, métodos y eventos. Están organizados en una vista o árbol de componentes y son desplegados en una página JSP. Los componentes UI están conformados de dos partes: el componente es sí que es el que implementa la funcionalidad y el renderer que es el encargado de dibujar el componente en el cliente.

2.7.1. INTRODUCCIÓN A JAVA SERVER FACES

Java Server Faces es un Framework que pretende facilitar el desarrollo de aplicaciones web, gestionando las acciones realizadas por el usuario en una página HTML y las traduce a eventos que son enviados al servidor, y así reflejar los cambios provocados por dichas acciones.

La estandarización de los Frameworks en Java se está realizando a través de Java Server Faces. El grupo de trabajo se formó en el año 2001 y la primera especificación

JSF 1.0 fue terminada el 11 de Marzo del 2004 y fue revisada en Mayo del mismo año JSF 1.1 estas dos versiones forman parte del JSR 127. Los objetivos de esta estandarización son:

- ✚ Crear un Framework estándar basado en componentes, de fácil integración en entornos de desarrollo.
- ✚ Definir la estructura de clases para gestionar el estado y los eventos de los componentes.
- ✚ Definir un conjunto básico de componentes, que incluyen los elementos habituales de las páginas HTML.
- ✚ Proporcionar un modelo de eventos, que permita enlazar eventos de la interfaz de usuario con la lógica del servidor.
- ✚ Definir los APIs de validación, tanto en el servidor como en el cliente.
- ✚ Especificar el modelo de internacionalización y localización.
- ✚ Generación automática de presentación en función del navegador cliente.
- ✚ Generación automática de presentación cumpliendo estándares de accesibilidad.

JSF 1.2 la actual versión es un estándar de JEE 5.0. La fuerte orientación a componentes de Java Server Faces, y la evolución de la programación WEB hacia un modelo basado en componentes, está llevando a que JSF esté entre los Frameworks MVC más populares.

Java Server Faces es un Framework Web J2EE de código fuente abierto que implementa el patrón de diseño MVC.

JSF es un Framework basado en componentes de interfaz de usuario con estado, que facilita y agiliza el desarrollo de aplicaciones web en plataforma Java lo que implica:

- ✚ Se trata de una tecnología que se ejecuta del lado del Servidor y no del lado del Cliente.
- ✚ La Interfaz de Usuario es tratada como un conjunto de componentes UI. Concepto fundamental para la comprensión de JSF.

Los principales componentes de la tecnología Java Server Faces son:

- ✚ Un API y una implementación de referencia para representar componentes UI y manejar su estado; manejo de eventos, validación del lado del servidor y conversión de datos; definir la navegación entre páginas, soportar internacionalización y además proporcionar extensibilidad para estas características.
- ✚ Una librería de etiquetas JSP personalizadas para dibujar componentes UI en una página JSP.

Está diseñado para facilitar el desarrollo y mantenimiento de las aplicaciones que se ejecutan en un servidor y devuelven el resultado en forma de interfaz al cliente.

Java Server Faces es el único Framework Web J2EE que tiene una especificación creada por el Java Community Process, esto lo transforma en un estándar y como principal consecuencia, todas las implementaciones tanto las de código fuente abierto como las comerciales deben respetar la especificación.

JSF 1.2 la versión actual del Framework, forma parte de la especificación J2EE 5.0 con lo cual todas las implementaciones de servidores J2EE 5.0 lo deben soportar.

Existen compañías y organizaciones involucradas en el desarrollo de JSF como son, Sun, IBM, Oracle, Apache Software Foundation, Macromedia entre otras.

Una de las ventajas de que JSF sea una especificación es que podemos encontrar implementaciones de distintos fabricantes y así podemos seleccionar la implementación que más convenga.

Los programadores piensan en términos de componentes de UI, eventos y sus interacciones. JSF oculta al programador los detalles de request HTTP y response HTTP, logrando un estilo de programación similar a la programación de aplicaciones de escritorio del estilo Java Swing. [www 18]

Entre las principales implementaciones de Software Libre tenemos:

- ✚ MyFaces de Apache Software Foundation.
- ✚ JSF RI de Sun

- ☒ Smile
- ☒ ICEFaces (con AJAX)

JSF tiene un alto nivel de abstracción que permite una mayor separación entre el diseño y el código permitiendo dividir realmente el desarrollo de una aplicación en diseño y desarrollo, donde los diseñadores aunque todavía necesitan tener conceptos básicos de la tecnología no tienen que preocuparse por la lógica de la aplicación.

JSF incluye un conjunto de componentes y herramientas que facilitan la creación de Interfaces de Usuario. Componentes como: cajas de texto, botones, radio botones, en fin todos los componentes que se puedan utilizar en una interfaz. La codificación de estos componentes que son usados normalmente dentro de páginas JSP como tags, es definida por la cada implementación de JSF. La comunicación entre los componentes, la entrada de datos y la validación forman parte de la tecnología, es decir, no hay que implementar nada de esto. La visualización de los mensajes de validación y errores de conversión también están estandarizados incluso soportan internacionalización.

JSF comparada con otros Frameworks MVC está más orientada a componentes y eventos; además de tener un mayor nivel de abstracción. JSF es la tecnología más reciente, avanzada y con un futuro prometedor.

2.7.2. IMPLEMENTACIONES DE JSF

- ☒ **JSF RI.**- de Sun es la implementación de referencia, la implementación oficial la podemos descargar de: <http://java.sun.com/j2ee/javaserverfaces>, al principio no era open source, pero luego su código se hizo disponible bajo la Java Research License.
- ☒ **Apache My Faces.**- es la primera implementación open source y gratuita de JSF, la podemos descargar de: <http://myfaces.apache.org/> , posee extensiones a los componentes estándar, también validadores y componentes personalizados y muy fáciles de usar. [www 19]

- + **Smile.-** es una implementación open source de la API JSF, la podemos descargar de: <http://smile.sourceforge.net/>, modelo de programación no es JSP. El objetivo de esta implementación es ser compatible con la especificación.

2.7.3. HERRAMIENTAS DE DESARROLLO

Al igual que las implementaciones también contamos con IDEs comerciales y gratuitos. Algunos de los IDEs que tienen soporte para JSF:

- + **Eclipse Web Tools.-** es gratuito, es un subproyecto de Web Tools Platform para manejo de JSF. Ayuda en la edición de código, cuenta con un editor para el archivo de configuración.

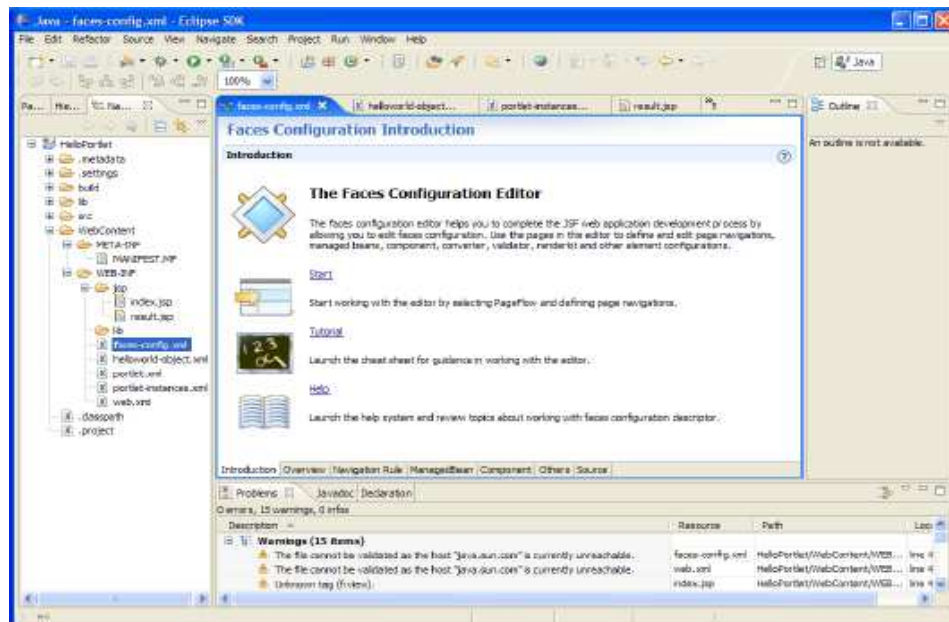


Figura 3.1: IDE Eclipse

- + **NetBeans.-** cuenta con soporte para JSF, pero no es tan extenso como otros IDEs. Tiene herramientas para el manejo del archivo de configuración de JSF, y el manejo de Managed Beans, ayuda en el código, no tiene paleta de componentes.

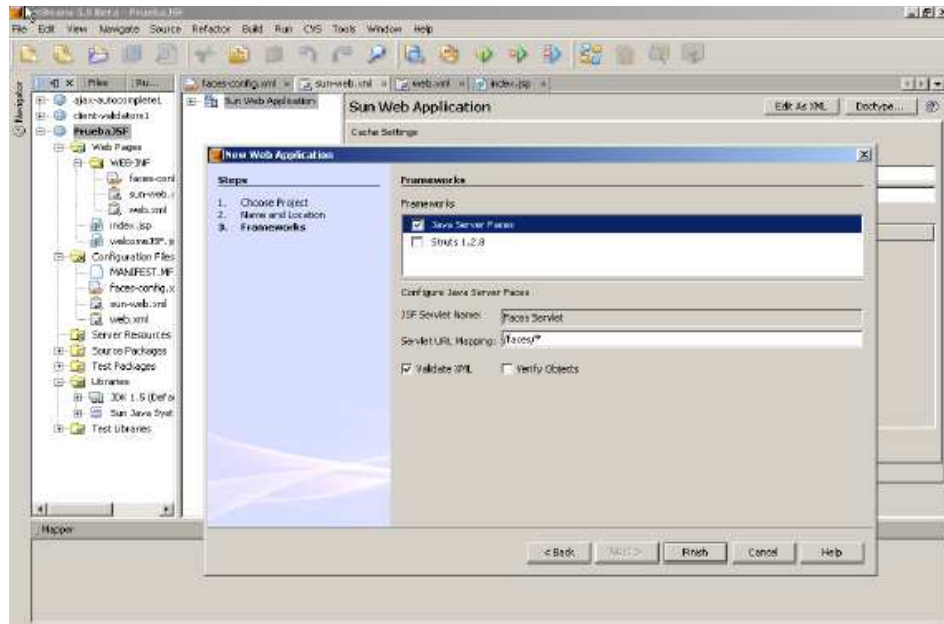


Figura 3.2: IDE NetBeans

- **JDeveloper.**- es desarrollado por Oracle, y fue liberado en el año 2005, es un IDE gratuito pero no open source.

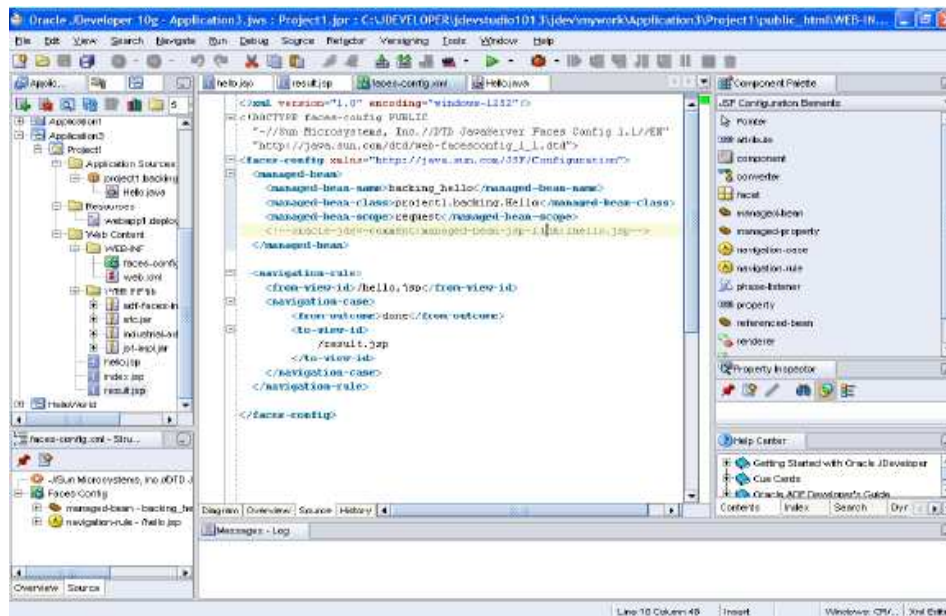


Figura 3.3: IDE JDeveloper

- **Java Studio Creator.**- se lo descarga gratuitamente después de registrarse, es desarrollado por Sun Microsystems, está basado en NetBeans, está diseñado para la implementación visual de aplicaciones web en Java.

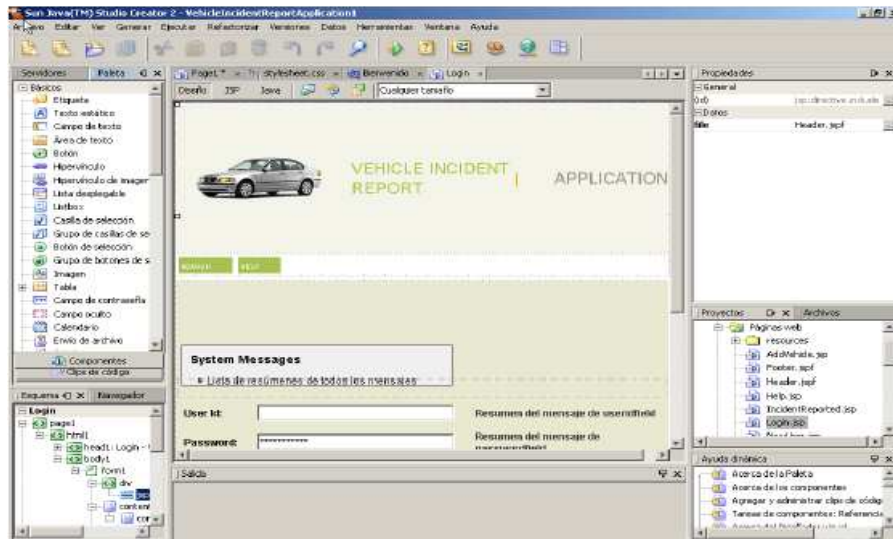


Figura 3.4: IDE Java Studio Creator [www 20]

2.7.4. ARQUITECTURA DE JSF

JSF es una tecnología que permite construir aplicaciones web, que soportan diferentes dispositivos como clientes por ejemplo: teléfonos celulares, PALMs. Los componentes UI son independientes del dispositivo cliente.

En la figura se muestra los componentes principales del Framework JSF que intervienen en la construcción de una aplicación web J2EE, y también su flexibilidad para aceptar peticiones provenientes de clientes diferentes.

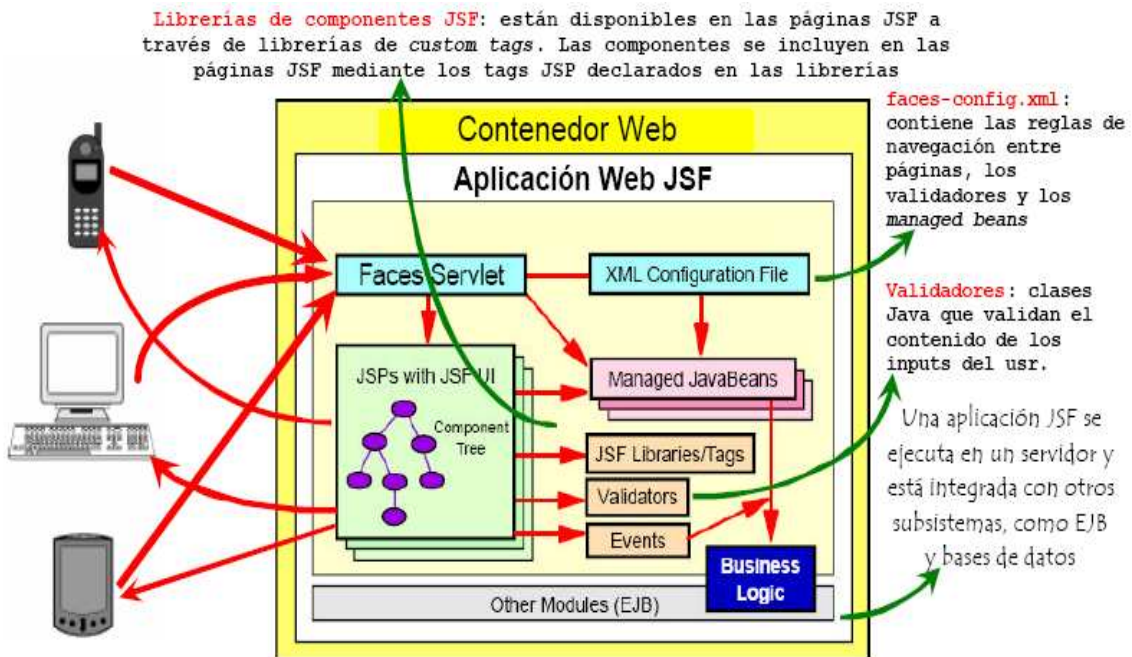


Figura 3.5: Arquitectura de una Aplicación JSF [www 21]

2.7.5. VENTAJAS DE UTILIZAR JSF

El utilizar un Framework como Java Server Faces en el desarrollo de Aplicaciones Web nos trae muchas ventajas como son:

- Una de las grandes ventajas que nos proporciona JSF es que nos ofrece una clara separación entre el comportamiento y la presentación de una aplicación. Las aplicaciones desarrolladas usando tecnología JSP conseguían esta separación parcialmente. Sin embargo una aplicación JSP no puede mapear peticiones HTTP al manejo de eventos específicos del componente, ni manejar elementos de Interfaz de Usuario como objetos con estado en el lado del servidor.

La tecnología Java Server Faces nos permite construir aplicaciones web que implementan separación entre el comportamiento y la presentación tradicionalmente ofrecida por arquitecturas de Interfaz de Usuario del lado del cliente.

- La separación entre la lógica y la presentación también le permite a cada miembro del equipo de desarrollo de una aplicación Web enfocarse en su parte

del proceso de desarrollo, y proporciona un sencillo modelo de programación para enlazar todas las piezas.

- ✚ Otro objetivo importante de la tecnología Java Server Faces es mejorar los conceptos de componente de Interfaz de Usuario y capa Web sin limitarnos a una tecnología de script particular o un lenguaje de marcas. Aunque la tecnología Java Server Faces incluye una librería de etiquetas JSP personalizadas para representar componentes en una página JSP, los APIs de la tecnología Java Server Faces se han creado directamente sobre el API Java Servlet.

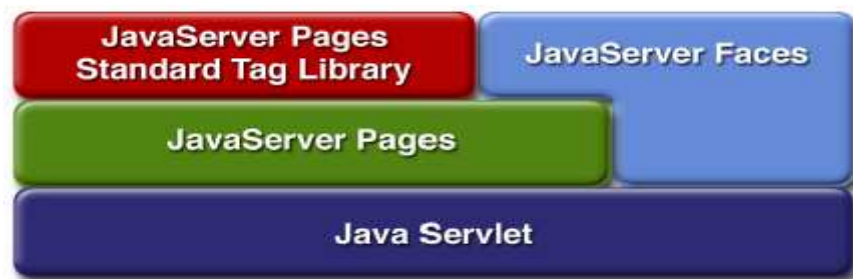


Figura 3.6: Tecnologías Java para Aplicaciones Web [www 22]

Esto nos permite hacer algunas cosas como: usar otra tecnología de presentación junto a JSP, crear nuestros propios componentes personalizados directamente desde las clases de componentes, y generar salida para diferentes dispositivos cliente.

- ✚ La tecnología Java Server Faces nos permite convertir y validar datos sobre componentes individuales y reportar cualquier error antes de que se actualicen los datos en el lado del servidor.
- ✚ Pero lo más importante, la tecnología Java Server Faces proporciona una rica arquitectura para manejar el estado de los componentes, procesar los datos, validar la entrada del usuario, y manejar eventos.
- ✚ Facilidad de integración con otras tecnologías web, como Spring, Hibernate, Struts.

- ✚ Existen algunas implementaciones del estándar JavaServer Faces, tanto comerciales como implementaciones open source.
- ✚ IDEs compatibles con el estándar, tanto comerciales como libres, que permiten al desarrollador seleccionar el IDE apropiado.
- ✚ Comunidad de desarrolladores en aumento.
- ✚ Excelente documentación de JSF se puede encontrar en Internet.
- ✚ Herramientas de desarrollo y soporte en aumento. [www 23]

2.7.6. DESVENTAJAS

- ✚ La curva de aprendizaje es grande.- este es un inconveniente significativo al menos en proyectos pequeños, en los que el tiempo de desarrollo es corto y el desarrollador puede consumir este tiempo tanto en el aprendizaje de JSF como en desarrollo en sí del proyecto.
- ✚ La creación de componentes propios es compleja.

2.7.7. LENGUAJE DE EXPRESION UNIFICADO PARA JSP Y JSF

El lenguaje de expresión unificado es usado para acceder a objetos implícitos o a clases Java.

Java Server Faces y Java Server Pages cada uno tiene su propio Lenguaje de Expresiones. El lenguaje de expresiones incluido en JSP proporciona mayor flexibilidad al desarrollador de aplicaciones web, pero existe un problema ya que JSF se usa para renderizar componentes GUI y necesita evaluación run-time de expresiones. El lenguaje de expresiones unificado adquiere completamente la responsabilidad de evaluar la expresión del motor JSP y evalúa la expresión en el momento apropiado durante el ciclo de vida de la página. Pero el lenguaje de expresiones de JSP está diseñado para la evaluación inmediata de expresiones.

Otro problema es que los componentes JSF necesitan una manera de invocar a métodos de objetos del lado del servidor durante varias fases del ciclo de vida, en orden para validar datos y manejar eventos de componentes. Las funciones JSP no son suficientes porque estos solamente se pueden usar para llamar a métodos estáticos

definidos en un Tag Library Descriptor, no pueden ser usados para invocar dinámicamente a métodos públicos de algún objeto.

Por estas razones es necesario un lenguaje de expresiones poderoso para trabajar con JSP y JSF. El nuevo lenguaje de expresiones unificado esencialmente representa la unión de los lenguajes de expresión de JSP y JSF, y grandes beneficios para la tecnología JSF. El lenguaje de expresiones unificado cuenta con las siguientes características:

- Evaluación de expresiones postergada.
- Soporte para expresiones que pueden poner o signar valores y expresiones que pueden invocar métodos.

2.7.7.1. EVALUACIÓN DEL LENGUAJE DE EXPRESIONES

La evaluación del Lenguaje de Expresión es de dos tipos: evaluación inmediata y evaluación postergada. La evaluación inmediata significa que en una página JSP las expresiones se evalúan cuando la página es renderizada, con este tipo de evaluación todos los valores son de sólo lectura.

En la evaluación postergada el lenguaje de expresiones unificado adquiere completamente la responsabilidad de evaluar la expresión del motor JSP y evalúa la expresión en el momento apropiado durante el ciclo de vida de la página. En el lenguaje de expresiones JSF las expresiones tienen la forma `#{expresion}`.

El siguiente código representa una etiqueta JSF de un `textField` en el que el usuario ingresa un valor, en esta etiqueta mediante el atributo `value` la expresión hace referencia a la propiedad `titulo` del Bean `libro`.

```
<h:inputText id="txtname" value="#{libro.titulo}"/>
```

Para una petición inicial de la página que contiene la etiqueta, la implementación de JSF evalúa la expresión `#{libro.titulo}` durante la fase de renderizar la respuesta del ciclo de vida. Durante esta fase la expresión solamente accede el valor de `titulo` en el Bean `libro` como se hace en la evaluación inmediata.

Si es una petición `postback`, la implementación de JSF evalúa la expresión durante las fases de: aplicar valores de petición, procesar validaciones, actualizar el modelo

2.7.7.2. EXPRESIONES DE MÉTODOS

Una característica del lenguaje de expresiones unificado es que tiene soporte para expresiones de métodos. Una etiqueta de un componente JSF usa expresiones de métodos para invocar métodos que realizan algún proceso para el componente que la etiqueta representa en la página. Para los componentes estándar, estos métodos son necesarios para tratar los eventos que los componentes generan y validar los datos del componente.

2.7.8. UNA APLICACIÓN JAVA SERVER FACES

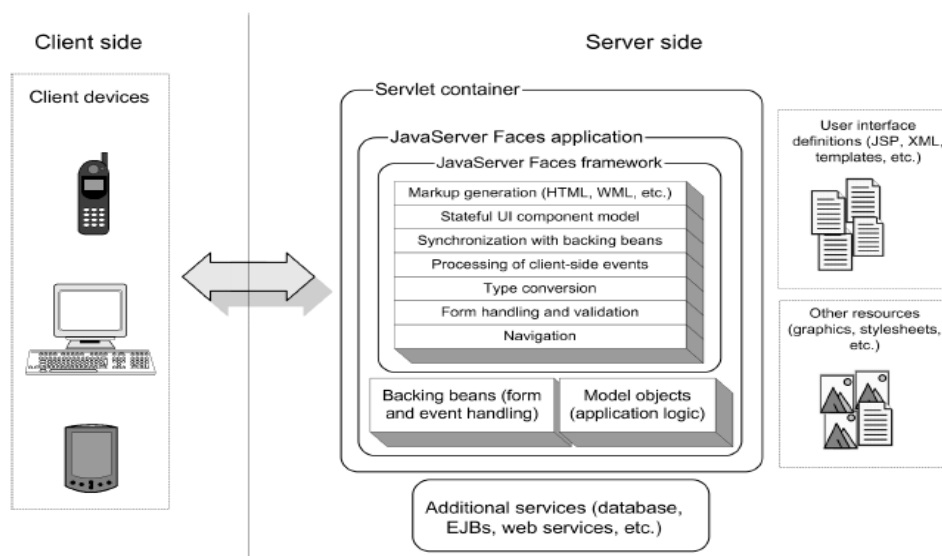


Figura 3.7: Vista de alto nivel de una Aplicación Java Server Faces [www 21]

Las aplicaciones Java Server Faces son como cualquier otra aplicación Web Java. Se ejecutan en un contenedor Servlet Java y contienen:

- ✚ JavaBeans (llamados objetos del modelo en tecnología JavaServer Faces) conteniendo datos y funcionalidades específicas de la aplicación.
- ✚ Oyentes de Eventos.
- ✚ Páginas JSP.
- ✚ Clases de utilidad del lado del servidor, como Beans para acceder a bases de datos.
- ✚ Una librería de etiquetas personalizadas para dibujar componentes UI en una página.

- ✚ Una librería de etiquetas personalizadas para representar manejadores de eventos, validadores, y otras acciones.
- ✚ Componentes UI representados como objetos con estado en el servidor.
- ✚ Validadores, manejadores de eventos y manejadores de navegación.

Toda aplicación Java Server Faces debe incluir una librería de etiquetas personalizadas que define las etiquetas que representan componentes UI y una librería de etiquetas para representar otras acciones importantes, como validadores y manejadores de eventos. La implementación de JavaServer Faces proporciona estas dos librerías.

La librería de etiquetas de componentes elimina la necesidad de codificar componentes UI en HTML u otro lenguaje de marcas, resultando en componentes completamente reutilizables. Y, la librería core hace fácil registrar eventos, validadores y otras acciones de los componentes.

En una aplicación JavaServer Faces los componentes UI de la página están representados en el servidor como objetos con estado. Esto permite a la aplicación manipular el estado del componente y conectar los eventos generados por el cliente a código en el lado del servidor.

2.7.9. MODELO DE COMPONENTES DE INTERFAZ DE USUARIO

Los componentes UI JavaServer Faces son elementos configurables y reutilizables que conforman el interface de usuario de las aplicaciones JSF.

Un componente de interfaz de usuario puede ser simple, como un botón, o compuesto como una tabla que puede estar constituida por múltiples componentes.

La tecnología JavaServer Faces proporciona una arquitectura de componentes rica y flexible que incluye:

- ✚ Un conjunto de clases `UIComponent`, para especificar el estado y comportamiento de componentes UI.

- ✚ Un modelo de renderizado que define cómo renderizar los componentes de diferentes formas.
- ✚ Un modelo eventos y oyentes que define cómo manejar los eventos de los componentes.
- ✚ Un modelo de conversión que define como conectar conversores de datos a un componente.
- ✚ Un modelo de validación que define como registrar validadores a un componente.

2.7.9.1. CLASES DE COMPONENTES DE INTERFAZ DE USUARIO

Java Server Faces proporciona un conjunto de clases de componentes UI, que especifican toda la funcionalidad del componente como: mantener el estado del componente, mantener una referencia a objetos del modelo, dirigir el manejo de eventos, y el renderizado para un conjunto de componentes estándar.

Estas clases son completamente extensibles, lo que significa que podemos extenderlas para crear nuestros propios componentes personalizados.

Todas las clases de componentes UI descienden de la clase `UIComponentBase`, que define el estado y comportamiento por defecto de un componente.

El conjunto de clases de componentes UI que incluye la última versión de la tecnología `JavaServer Faces` es:

COMPONENTE	DESCRIPCIÓN
<code>UIColumn</code>	Representa una columna de datos en un componente <code>UIData</code> .
<code>UICommand</code>	Representa un control que dispara acciones cuando se activa.
<code>UIData</code>	Representa datos que están ligados a una colección de datos representados por
<code>UIForm</code>	Encapsula un grupo de controles que envían datos de la aplicación. Este componente es análogo a la etiqueta <code>form</code> de HTML.
<code>UIGraphic</code>	Muestra una imagen.
<code>UIInput</code>	Toma datos de entrada del usuario. Esta clase es una subclase de

	UIOutput.
UIMessage	Muestra un mensaje localizado.
UIMessages	Muestra un conjunto de mensajes localizados.
UIOutput	Muestra la salida de datos en una página.
UIPanel	Muestra una tabla
UIParameter	Representa la sustitución de parámetros.
UISelectBoolean	Permite a un usuario seleccionar un valor booleano en un control, seleccionándolo o deseleccionándolo. Esta clase es una subclase de UIInput.
UISelectItem	Representa un sólo ítem de un conjunto de ítems.
UISelectItems	Representa un conjunto completo de ítems.
UISelectMany	Permite al usuario seleccionar varios ítems de un grupo de ítems. Esta clase es una subclase de UIInput.
UISelectOne	Permite al usuario seleccionar un ítem de un grupo de ítems. Esta clase es una subclase de UIInput.
UIViewRoot	Representa la raíz del árbol de componentes.

Tabla 3.1: Componentes UI

El desarrollador no tiene que utilizar estas clases directamente, sino incluir los componentes en una página usando la etiqueta correspondiente al componente.

Además de extender de `UIComponentBase`, las clases de los componentes también implementan uno o más comportamientos de interfaz, cada uno de los cuales define cierto comportamiento para un conjunto de componentes cuyas clases implementan la interfaz.

Estos comportamientos de interfaz son:

- ✚ `ActionSource`: indica que el componente puede disparar un evento o acción.
- ✚ `EditableValueHolder`: extiende de `ValueHolder` y adicionalmente especifica las características editables de los componentes, como son validación y emisión de eventos de cambio de valor.
- ✚ `NamingContainer`: cada componente debe tener un ID único.

- ✚ StateHolder: indica que el estado de un elemento debe guardarse entre peticiones.
- ✚ ValueHolder: indica que el componente tiene un valor local y cómo acceder a los datos en el modelo.
- ✚ Un UICommand se puede renderizar como un botón o como un hipervínculo.

2.7.9.2. MODELO DE RENDERIZADO DE COMPONENTES

La arquitectura de componentes en JSF es diseñada de tal manera que la funcionalidad de los componentes se definida mediante las clases de componentes, mientras que el renderizado de los componentes se puede definir mediante renderizadores separados.

Este diseño aporta beneficios como:

- ✚ Se puede definir el comportamiento de un componente sólo una vez, pero crear varios renderizadores, cada uno de los cuales define una forma diferente de dibujar el componente para el mismo cliente o para diferentes clientes.
- ✚ Se puede cambiar la apariencia de un componente de la página seleccionando la etiqueta que represente la combinación apropiada de componente y renderizador.

Un render kit define como se mapean las clases de los componentes a las etiquetas de componentes apropiadas para un cliente en particular. La implementación de JavaServer Faces incluye un render kit estándar para renderizar a un cliente HTML.

Para cada componente UI que soporte un render kit, éste define un conjunto de objetos Render. Cada objeto Render define una forma diferente de dibujar el componente en la salida definida por el renderkit. Por ejemplo un componente UISelectOne tiene tres renderizadores diferentes; uno de ellos dibuja el componente como un conjunto de radio buttons, otro dibuja el componente como un ComboBox y el tercero dibuja el componente como un ListBox.

Cada etiqueta JSP personalizada en el render kit de HTML, está compuesta por la funcionalidad del componente, definida en la clase UIComponent; y los atributos de

renderizado definidos por el Renderer. Por ejemplo en la tabla se representa un componente UICommand renderizado de dos formas diferentes.


ETIQUETA	SE RENDERIZA COMO
commandButton	
commandLink	hyperlink

Tabla 3.2: Etiqueta UICommand

La parte command de las etiquetas mostradas en la tabla anterior corresponde a la clase UICommand; y especifica la funcionalidad que es disparar un Action; las partes Button y Link de las etiquetas corresponden a un rederizador independiente que define como dibujar componente.

La implementación de JavaServer Faces proporciona una librería de etiquetas personalizadas para renderizar componentes en HTML; soporta todos los componentes listados en la siguiente tabla:

ETIQUETA	FUNCIÓN	SE RENDERIZA COMO	APARIENCIA
column	Representa una columna de datos en un componente UIData.	Una columna de datos en tabla HTML.	Una columna en una tabla.
commandButton	Enviar un formulario a la aplicación.	Un elemento HTML <code><input type="type"></code> donde el valor de type puede ser: submit, reset o imagen.	Un botón.

commandLink	Enlazar con otra página o localización en otra página.	Un elemento HTML <code><a href></code> .	Un Hipervínculo o enlace.
dataTable	Representa un contenedor de datos.	Un elemento <code><table></code> .	Una tabla que se puede actualizar dinámicamente.
form	Representa un formulario de entrada. Las etiquetas internas del formulario reciben los datos que serán enviados con el formulario.	Un elemento HTML <code><form></code> .	No tiene apariencia
graphicImage	Despliega una imagen.	Un elemento HTML <code></code> .	Una imagen.
inputHidden	Permite incluir una variable oculta en la página.	Un elemento HTML <code><input type=hidden></code> .	Sin apariencia
inputSecret	Le permite al usuario ingresar una cadena o String sin que aparezca el String real en el campo.	Un elemento HTML <code><input type=password></code> .	Un campo de texto que muestra una fila de caracteres en lugar del texto real introducido.
inputText	Permite al usuario ingresar una cadena o String.	Un elemento HTML <code><input type=text></code> .	Un campo de texto.
inputTextarea	Permite al usuario ingresar un texto multilíneas.	Un elemento HTML <code><textarea></code> .	Un campo de texto multilíneas.
message	Despliega un mensaje localizado.	Texto normal.	Texto normal.
messages	Despliega mensajes	Texto normal.	Texto normal.

	localizados.		
outputLabel	Muestra un componente anidado, como una etiqueta para un campo de texto específico.	Un elemento HTML <code><label></code> .	Texto normal.
outputLink	Enlazar con otra página o ubicación de una página sin generar una acción.	Un elemento HTML <code><a></code>	Un Hipervínculo o enlace.
outputFormat	Despliega un mensaje localizado.	Texto normal.	Texto normal.
outputText	Muestra una línea de texto.	Texto normal.	Texto normal.
panelGrid	Muestra una tabla.	Un elemento HTML <code><table></code> con elementos <code><tr></code> y <code><td></code> .	Una tabla.
panelGroup	Agrupar un conjunto de paneles bajo un padre.		Una fila en una tabla.
selectBoolean Checkbox	Permite al usuario cambiar el valor de una elección booleana.	Un elemento HTML <code><input type=checkbox></code> .	Un checkbox
selectItem	Representa un ítem de una lista de ítems en un componente <code>UISelectOne</code> .	Un elemento HTML <code><option></code> .	Sin apariencia.
selectItems	Representa una lista de ítems en un componente <code>UISelectOne</code> .	Una lista de elementos HTML <code><option></code> .	Sin apariencia.
selectMany Checkbox	Muestra un conjunto de Checkboxes en los que	Un conjunto de elementos HTML	Un conjunto de checkboxes.


	el usuario puede hacer una selección múltiple.	<input type=checkbox>.	
selectMany Listbox	Permite al usuario seleccionar múltiples ítems de un conjunto de ítems, todos mostrados a la vez.	Un elemento HTML <select>.	Un list box.
selectMany Menu	Permite al usuario seleccionar varios ítems, de un grupo de ítems.	Un elemento HTML <select>.	Un combo box.
selectOne Listbox	Permite al usuario seleccionar un ítem de un conjunto de ítems.	Un elemento HTML <select>	Un list box.
selectOneMenu	Permite al usuario seleccionar un ítem de un conjunto de ítems.	Un elemento HTML <select>.	Un combo box.
selectOneRadio	Permite al usuario seleccionar un ítem de un conjunto de ítems.	Un elemento HTML <input type=radio>.	Un conjunto de botones de radio.

Tabla 3.3: Etiquetas de componentes UI

2.7.9.3. MODELO DE CONVERSIÓN

Una aplicación JavaServer Faces puede opcionalmente asociar un componente con datos del objeto del modelo del lado del servidor. Este objeto del modelo es un componente JavaBeans, que encapsula los datos de un conjunto de componentes. Una aplicación obtiene y configura los datos del objeto modelo para un componente llamando a las propiedades apropiadas del objeto modelo para ese componente.

Cuando un componente se une a un objeto modelo, la aplicación tiene dos vistas de los datos del componente:

-  **Vista de modelo**, los datos son representados como tipos de datos, como int o long.

- ✚ **Vista de presentación**, representa los datos de una forma que el usuario puede verlos o modificarlos.

Una aplicación JavaServer Faces debe asegurarse de que los datos del componente puedan ser convertidos entre la vista de modelo y la vista de presentación. Esta conversión normalmente la realiza automáticamente el renderizador del componente. JavaServer Faces convierte automáticamente datos de componentes, cuando la propiedad del Bean asociada con el componente es de uno de los tipos soportados por los datos del componente. Por ejemplo, si un componente `UISelectBoolean` está ligado a la propiedad de un Bean, y esta propiedad es de tipo Boolean, entonces Java Server Faces automáticamente convierte los datos del componente de String a Boolean.

Se puede usar los conversores proporcionados por JavaServer Faces o crear nuestros propios conversores personalizados, en caso de necesitar convertir un dato de un componente a un tipo no soportado por el renderizador del componente. Para facilitar esto la tecnología JavaServer Faces incluye un conjunto de implementaciones estándar de Converter que nos permite crear nuestros conversores personalizados. La implementación de Converter convierte los datos del componente entre las dos vistas.

2.7.9.4. MODELO DE EVENTOS Y OYENTES

El modelo de eventos y oyentes de JavaServer Faces mejora el diseño del modelo de eventos de JavaBeans.

Al igual que la arquitectura de componentes JavaBeans, la tecnología JavaServer Faces define las clases Listener y Event que una aplicación puede utilizar para manejar eventos generados por componentes UI.

Un objeto Event identifica al componente que lo generó y almacena información sobre el propio evento. Para ser notificado de un evento, una aplicación debe proporcionar una implementación de la clase Listener y registrarla con el componente que genera el evento. Cuando el usuario activa un componente, como cuando pulsa un botón, entonces se dispara un evento. Y la implementación de JavaServer Faces invoca al método oyente que procesa el evento.

JavaServer Faces soporta dos tipos de eventos:

- ✚ **Eventos Action.-** ocurre cuando el usuario pulsa un botón o un hipervínculo. El componente `UICommand` genera este evento.
- ✚ **Eventos value-changed.-** ocurre cuando el usuario cambia el valor de un componente. Por ejemplo, seleccionar un checkbox, que resulta en que el valor del componente cambia a `true`. Los tipos de componentes que generan este tipo de eventos son: `UIInput`, `UISelectOne`, `UISelectMany` y componentes `UISelectBoolean`. Los `value-changed` sólo se disparan si no se ha detectado errores de validación.

Dependiendo del valor inmediato de la propiedad del componente que emite el evento, los eventos Action pueden ser procesados durante la fase de Invocar a la aplicación o la fase de aplicar valores de petición. Y los eventos value-changed pueden ser procesados durante la fase de proceso de validación o la fase de aplicar valores de petición.

Una aplicación puede reaccionar de dos maneras a eventos Action o eventos value-changed emitidos por un componente estándar.

- ✚ Implementar una clase oyente de eventos, para que se ocupe del evento y registrar el oyente en el componente anidando una etiqueta `valueChangeListener` o una etiqueta `actionListener` dentro de la etiqueta del componente.
- ✚ Implementar un método de Bean de respaldo, para que se ocupe del evento y referirse al método con una expresión `method-binding` del atributo apropiado de la etiqueta del componente.

2.7.9.5. MODELO DE VALIDACIÓN

La tecnología Java Server Faces soporta un mecanismo para validar el dato local de un componente durante la fase del Proceso de Validación, antes de actualizar los datos del objeto modelo.

El modelo de validación define un conjunto de clases estándar para realizar la validación de datos comunes. La librería de etiquetas JSF-core define un conjunto de etiquetas que corresponden con las implementaciones estándar de Validator.

La mayoría de las etiquetas tienen un conjunto de atributos para configurar las propiedades del validador, como los valores mínimo y máximo permitidos para el dato del componente. El desarrollador registra el validador con un componente anidando la etiqueta del validador dentro de la etiqueta del componente.

Al igual que el modelo de conversión, el modelo de validación nos permite crear nuestros propios validadores con sus correspondientes etiquetas para realizar validaciones personalizadas.

2.7.10. MODELO DE NAVEGACIÓN

Las aplicaciones web constan de un conjunto de páginas es por eso que uno de los aspectos que más preocupan a un desarrollador es la gestión de la navegación entre las páginas.

El modelo de navegación de JavaServer Faces facilita la definición de la navegación entre las páginas y el manejo de cualquier procesamiento adicional necesario para elegir la secuencia en que se cargan las páginas. En JSF la navegación se define en el fichero de configuración de la aplicación mediante un conjunto de reglas para lo cual se usan un pequeño conjunto de elementos XML; y proporcionar el código necesario si necesitamos un procesamiento adicional para determinar que página mostrar luego.

Para cargar la siguiente página en una aplicación web, el usuario pulsa un botón lo que genera un evento Action. La implementación de JavaServer Faces proporciona un oyente de eventos Action por defecto para manejar este evento. Este oyente determina la salida del evento Action, como success o failure; esta salida se puede definir como una propiedad String del componente que generó el evento o como el resultado de un procesamiento adicional realizado en un objeto Action asociado con el componente. Después de determinar la salida el oyente la pasa al NavigationHandler asociado a la aplicación; basándose en la salida de vuelta, el NavigationHandler

selecciona la página apropiada consultando el fichero de configuración de la aplicación.

Cada regla de navegación define como navegar de una página en particular a otras páginas en la aplicación. Cada caso de navegación dentro de una regla de navegación define la página a desplegar y un resultado lógico.

2.7.10.1. NAVEGACIÓN ESTÁTICA

En una aplicación web simple la navegación es estática, ya que al pulsar un botón se suele redirigir al navegador a una misma página para dar la respuesta. En este tipo de navegación se le da un valor al atributo Action de un botón o un hipervínculo:

```
<h:commandButton label="Ingresar" action="login"/>
```

Esta acción debe concordar con la etiqueta outcome del fichero de configuración faces-config.xml, dentro de sus reglas de navegación. Por ejemplo:

```
<navigation-rule>
  <from-view-id>/index.jsp</from-view-id>
  <navigation-case>
    <from-outcome>login</from-outcome>
    <to-view-id>/hola.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
```

En esta regla de navegación se indica que si en la página index.jsp hay una acción "login" la página que se debe mostrar es hola.jsp.

2.7.10.2. NAVEGACIÓN DINÁMICA

En la mayoría de las aplicaciones web la navegación no es estática. El flujo de la página no solo depende de que botón se pulse, sino también de los datos ingresados por el usuario; por ejemplo una página de autenticación puede tener dos resultados: éxito o fallo.

Para utilizar la navegación dinámica el botón ingresar debe tener un método de referencia por ejemplo:

```
<h:commandButton label="Ingresar" action="Autenticacion.VerificarUser"/>
```


En este caso Autenticación es la referencia a un Bean, el cual debe tener un método llamado VerificarUser.

Un método de referencia en un atributo action no tiene parámetros de entrada, y devuelve una cadena de caracteres que será utilizada para activar una regla de navegación. Suponiendo que el método VerificarUser devuelva un String “exito” o “fallo” después de analizar los datos ingresados por el usuario, éste String será utilizado por el manejador de navegación, para buscar la regla de navegación apropiada para el resultado devuelto; la regla de navegación es la siguiente:

```
<navigation-rule>
  <from-view-id>/login.jsp</from-view-id>
  <navigation-case>
    <from-outcome>exito</from-outcome>
    <to-view-id>/exito.jsp</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>fallo</from-outcome>
    <to-view-id>/error.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
```

Mediante esta regla de navegación tiene dos casos de navegación en los cuales se indica que si el resultado devuelto es “exito” la siguiente página que se debe mostrar es éxito.jsp, y si el resultado es “fallo” se debe mostrar la página error.jsp.

2.7.10.3. NAVIGACIÓN AVANZADA

- ✚ **Redirección.**- si se añade la etiqueta <redirect> después de <to-view-id>, JSF envía un redirección HTTP al usuario de una vista nueva. Esto quiere decir que el usuario ve el URL de la página que actualmente mira, en lugar de la dirección de la página anterior.
- ✚ **Comodines.**- se puede usar comodines en las etiquetas <from-view-id>; solo se permite un * que debe estar al final de la cadena del <from-view-id> por ejemplo:

```

<navigation-rule>
  <from-view-id>/app/*</from-view-id>
  <navigation-case>
    ...
  </navigation-case>
</navigation-rule>

```

Esta regla se aplicará a todas las páginas que empiecen con el prefijo **app**.

- ✚ **Etiqueta <from-action>**.- se la coloca antes de la etiqueta <from-out-come>, la etiqueta <from-action> es útil si en la misma regla de navegación tenemos dos casos de navegación con la misma cadena de acción, o dos métodos de referencia de acción que devuelven la misma cadena.

2.7.11. BACKING BEAN O BEAN DE RESPALDO

Otra función crítica de las aplicaciones Web es el manejo apropiado de los recursos. Esto incluye la separación de la definición de objetos componentes UI de los objetos de datos y almacenar y manejar estados de objetos en el ámbito apropiado.

Una aplicación JavaServer Faces incluye uno o más Backing Beans, que son componentes Java Beans asociados a componentes UI usados en una página. Un Backing Bean define propiedades de componentes UI, y cada una de estas propiedades está ligada al valor de un componente o a la instancia de un componente. Un Bean de respaldo también puede definir métodos que realicen funciones asociadas con un componente, incluyendo validación, manejo de eventos y proceso de navegación.

Para ligar los valores del componente UI y las propiedades del Backing Bean o métodos del Backing Bean con etiquetas de componentes UI el desarrollador usa la sintaxis de Lenguaje de Expresión (EL).

2.7.12. CICLO DE VIDA DE UNA PÁGINA JAVA SERVER FACES

El ciclo de vida de una página JSF es similar al de una página JSP. El cliente hace una petición HTTP y el servidor responde con una página traducida a HTML. Sin embargo, debido a las características extras que ofrece la tecnología JavaServer Faces, el ciclo de

vida proporciona algunos servicios adicionales mediante la ejecución de algunos pasos extras.

Una página JSF está representada por un árbol de componentes UI, llamada vista. Cuando un cliente hace una petición a una página empieza el ciclo de vida. Cuando un cliente envía una página JSF realiza varias tareas como validar los datos ingresados en los componentes de la vista y convertir estos datos a los tipos especificados en el lado del servidor. JSF realiza estas tareas como una serie de pasos en el ciclo de vida.

Los pasos del ciclo de vida se ejecutan dependiendo de si la petición se originó o no desde una aplicación JSF y si la respuesta es o no generada con la fase de renderizado del ciclo de vida de JavaServer Faces

2.7.12.1. ESCENARIOS DE PROCESAMIENTO DEL CICLO DE VIDA DE UNA PETICIÓN

Una aplicación JSF soporta dos tipos diferentes de responses (respuestas) y dos tipos diferentes de requests (peticiones), la idea es que en una aplicación JSF se pueda mezclar páginas JSF y páginas JSP sin componentes UI JSF.

- ✚ **Respuesta Faces:** Una respuesta servlet que se generó mediante la ejecución de la fase Renderizar la Respuesta del ciclo de vida de procesamiento de la respuesta.
- ✚ **Respuesta No-Faces:** Una respuesta Servlet que no se generó mediante la ejecución de la fase Renderizar la Respuesta. Un ejemplo es una página JSP que no incorpora componentes Java Server Faces.
- ✚ **Petición Faces:** Una petición Servlet que fue enviada desde una Respuesta Faces previamente generada. Un ejemplo es un formulario enviado desde un componente de interface de usuario JavaServer Faces, donde la URI de la petición identifica el árbol de componentes JavaServer Faces para usar el procesamiento de petición.
- ✚ **Petición No-Faces:** Una petición Servlet que fue enviada a un componente de aplicación como un Servlet o una página JSP, en vez de directamente a un componente JavaServer Faces.

La combinación de estas peticiones y respuestas resulta en tres posibles escenarios del ciclo de vida que pueden existir en una aplicación Java Server Faces:

✚ **Una petición No-Faces genera una respuesta Faces:** un ejemplo de este escenario es cuando se pulsa un enlace de una página HTML que abre una página que contiene componentes Java Server Faces. Para dibujar una respuesta Faces desde una petición No-Faces, la aplicación debe proporcionar un mapeo FacesServlet en la URL de la página que contiene componentes JavaServer Faces. FacesServlet acepta peticiones entrantes y pasa a la implementación del ciclo de vida para su procesamiento.

✚ **Una petición Faces genera una respuesta No-Faces:** algunas veces una aplicación JavaServer Faces puede necesitar generar una respuesta que no contiene componentes Java Server Faces.

En esta situación el desarrollador debe saltarse la fase de renderizado de respuesta llamando a `FacesContext.responseComplete`; `FacesContext` contiene toda la información asociada con una petición Faces particular. A este método se lo puede invocar durante las fases: Aplicar valores de petición, Procesar validaciones, o Actualizar los valores del modelo.

✚ **Una petición Faces genera una respuesta Faces:** éste es el escenario más común en una aplicación Java Server Faces. Este escenario implica componentes Java Server Faces enviando una petición a una aplicación JavaServer Faces utilizando el `FacesServlet`. Como la petición ha sido manejada por la implementación JavaServer Faces, la aplicación no necesita pasos adicionales para generar la respuesta. Todos los oyentes, validadores y conversores serán invocados automáticamente durante la fase apropiada del ciclo de vida estándar.

2.7.12.2. CICLO DE VIDA ESTÁNDAR DE PROCESAMIENTO DE PETICIONES

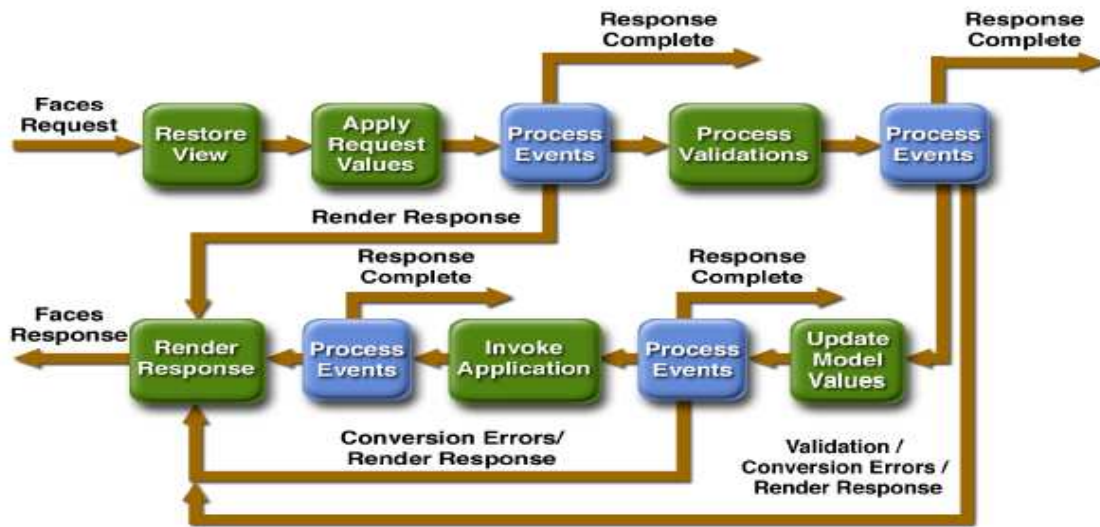


Figura 3.8: Pasos del ciclo de vida petición – respuesta Java Server Faces [www 23]

2.7.12.2.1. RECONSTRUIR EL ÁRBOL DE COMPONENTES

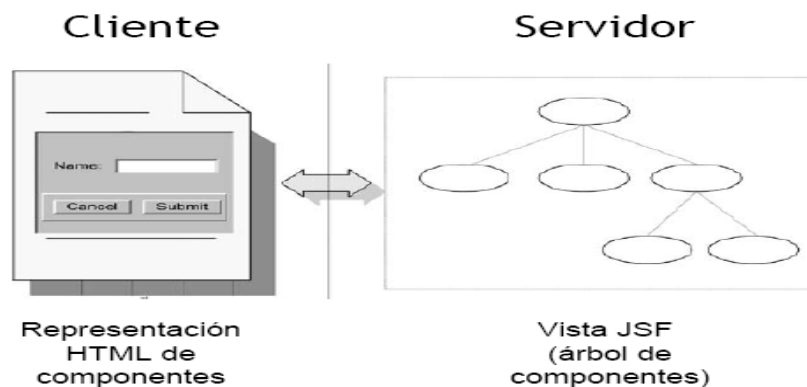


Figura 3.9: Árbol de componentes de una página solicitada por el cliente. [www 23]

Cuando se hace una petición para una página JavaServer Faces, como cuando se pulsa sobre un enlace o un botón, la implementación JavaServer Faces comienza la fase de Reconstruir el árbol de componentes.

Durante esta fase, JavaServer Faces construye el árbol de componentes de la página, conecta los manejadores de eventos y los validadores y guarda el estado en el FacesContext. El FacesContext contiene toda la información necesaria para procesar una petición.

Si la petición para la página es una petición inicial, JSF crea una vista vacía durante esta fase y el ciclo de vida adelanta a la fase de renderizar la respuesta.

Si la petición para la página es una petición postback, la vista que corresponde a esta página ya existe. Durante esta fase, Java Server Faces restaura la vista usando información del estado guardado en el cliente o el servidor.

2.7.12.2.2. APLICAR VALORES DE LA PETICIÓN

Una vez construido el árbol de componentes, cada componente del árbol extrae su nuevo valor desde los parámetros de la petición con el método `decode` y el valor es almacenado localmente en el componente. Si falla la conversión del valor, se genera un mensaje de error asociado con el componente y se pone en la cola de `FacesContext`; este mensaje se mostrará durante la fase de renderizar la respuesta, junto con cualquier error de validación resultante de la fase Procesar validaciones.

Si en esta fase se produce un evento, JSF emite al evento al oyente correspondiente.

Si algunos componentes de la página tienen atributos inmediatos, entonces la validación, conversión y los eventos asociados a estos componentes son procesados en esta fase.

En este punto si la aplicación necesita a un recurso de aplicación web diferente, o generar una respuesta que no tenga componentes JSF puede llamar a `FacesContext.responseComplete`.

Al final de esta fase, se han puesto los nuevos valores en los componentes y los mensajes y eventos se han puesto en sus respectivas colas.

2.7.12.2.3. PROCESAR VALIDACIONES

Durante esta fase, JavaServer Faces procesa todas las validaciones registradas en los componentes del árbol. Examina los atributos del componente que especifican las reglas de validación y compara esas reglas con el valor local almacenado en el componente. Si el valor local no es válido, JavaServer Faces añade un mensaje de error al `FacesContext` y el ciclo de vida avanza directamente hasta la fase Renderizar la Respuesta, para que la página sea dibujada de nuevo incluyendo los mensajes de error.

Si había errores de conversión de la fase Aplicar Valores de la Petición, también se mostrarán.

En este momento si la aplicación necesita redirigirse a un recurso de aplicación web diferente, o generar una respuesta que no contenga componentes JSF, puede llamar a `FacesContext.responseComplete`.

Si en esta fase se ha disparado eventos, JSF los emite a los oyentes correspondientes.

2.7.12.2.4. ACTUALIZAR LOS VALORES DEL MODELO

Cuando la implementación JavaServer Faces determina que el dato es válido, puede pasar por el árbol de componentes y configurar los valores del objeto de modelo correspondiente con los valores locales de los componentes. Sólo se actualizarán las propiedades del Bean asociadas a los componentes UI mediante el atributo `value`. Si el dato local no se puede convertir a los tipos especificados por las propiedades del objeto del modelo, el ciclo de vida avanza directamente a la fase Renderizar la respuesta y en ésta fase se dibujará de nuevo la página mostrando los errores, similar a lo que ocurre con los errores de validación.

En este momento si la aplicación necesita a un recurso de aplicación web diferente, o generar una respuesta que no contenga componentes Java Server Faces, se puede llamar a `FacesContext.responseComplete`.

Si se han disparado eventos durante esta fase, la implementación Java Server Faces los emite a los oyentes interesados.

2.7.12.2.5. INVOCAR APLICACIÓN

Durante esta fase, la implementación JavaServer Faces maneja cualquier evento a nivel de aplicación, como enviar un formulario o enlazar a otra página.

En este momento si la aplicación necesita a un recurso de aplicación web diferente, o generar una respuesta que no contenga componentes Java Server Faces, se puede llamar a `FacesContext.responseComplete`.

Si la vista que se está procesando ya existe se reconstruirá con información del estado de una petición anterior y si un componente ha disparado un evento, estos eventos se emiten a los oyentes interesados.

A continuación la implementación Java Server Faces, configura el árbol de componentes de la respuesta a esa nueva página. Finalmente la implementación transfiere el control a la fase Renderizar la Respuesta.

2.7.12.2.6. RENDERIZAR LA RESPUESTA

Durante esta fase, la implementación JavaServer Faces invoca las propiedades de codificación de los componentes y dibuja los componentes del árbol grabado en el FacesContext.

Si se encontraron errores en las fases de: aplicar los valores de petición, procesar validaciones o actualizar los valores del modelo, se dibujará la página original. Si las páginas contienen etiquetas message, cualquier mensaje de error que haya en la cola se mostrará en la página.

Se pueden añadir nuevos componentes en el árbol si la aplicación incluye renderizadores personalizados, que definen cómo renderizar un componente. Después de que se haya renderizado el contenido del árbol, éste se graba para que las siguientes peticiones puedan acceder a él y esté disponible para la fase Restaurar la vista de las siguientes llamadas. [www 24]

2.7.13. INTERNACIONALIZACIÓN

La internacionalización se puede definir como un proceso para diseñar sitios web adaptables a diferentes idiomas y regiones sin necesidad de realizar cambios en el código. Se utiliza archivos con extensión *.properties, se tiene uno de estos archivos por cada idioma que se desee añadir a la aplicación.

2.7.14. ARCHIVO DE CONFIGURACIÓN FACES-CONFIG.XML

Al desarrollar con Frameworks se minimiza la escritura de código Java y se maximiza la configuración mediante archivos XML.

El archivo de configuración de JavaServer Faces es `faces-config.xml` y es leído por el Framework cuando arranca la aplicación.

Entre los principales aspectos que podemos configurar mediante el `faces-config.xml` tenemos los siguientes:

- ✚ Declaración de los Managed Beans y las propiedades de los mismos.
- ✚ Declaración de reglas de navegación y sus respectivos casos de uso.
- ✚ Declaración de los archivos de idiomas que soporta la aplicación para la internacionalización.
- ✚ Convertidores y validadores personalizados.

2.8. COMPARACIÓN DE FRAMEWORKS MVC

2.8.1. JAVA SERVER FACES VS STRUTS Y TAPESTRY

2.8.1.1. MADUREZ

JSF al igual que Struts es un Framework Web que surge en el año 2004, tiene una gran ventaja con respecto a otros Frameworks Web; ya que es el primer Framework J2EE que tiene una especificación incluida en la versión J2EE 5.0, lo que lo convierte en el primer estándar y como consecuencia obliga a todas las implementaciones de servidores J2EE a soportarlo. Actualmente JSF es considerado el rival más fuerte de Struts.

Struts es un Framework Web que nace a principios del año 2000 y en el tiempo transcurrido ha alcanzado un alto grado de madurez. Cuenta con una comunidad activa de desarrolladores, gran cantidad de documentación disponible como son libros, y muchos tutoriales y artículos en la Web. Los IDEs Java más populares lo soportan.

2.8.1.2. DESAROLLO DE LA VISTA

JSF cuenta con un conjunto de componentes UI, que pueden personalizarse y extenderse para crear nuevos componentes UI.

La arquitectura flexible y extensible de JSF permite asociar renders diferentes para distintas tecnologías clientes como son teléfonos celulares, PDAs, además del típico cliente web.

Struts tiene un conjunto de custom tags que facilitan al desarrollador la creación de formularios HTML para ingreso de datos y que interactúa con Struts. Struts soporta un único cliente que es un navegador web.

En Tapestry las vistas son archivos HTML estándar, no tiene custom tags.

La única diferencia es la existencia de algunos atributos extras como jwcid, que significa Java Web Component ID.

2.8.1.3. FLEXIBILIDAD DEL CONTROLADOR Y MANEJO DE EVENTOS

JSF, Struts y Tapestry centralizan el manejo de peticiones de los clientes.

Con JSF cada uno de los componentes que conforman una página JSF puede tener asociados comportamientos personalizados como son conversiones, validaciones y procesamiento de eventos. JSF da muchos beneficios al controlador, como la capacidad de manejar múltiples eventos sobre una página.

El controlador de Struts está implementado siguiendo el concepto de caja gris, que permite definir puntos de extensión y proveer un comportamiento particular para procesar peticiones y manejar errores. Struts solo puede manejar un evento por página.

En Tapestry el manejo de eventos se realiza a través de oyentes de eventos que se asocian a los componentes que generan dicho evento.

2.8.1.4. VALIDACIÓN Y CONVERSION DE DATOS

JSF permite validar individualmente cada componente de un formulario ya sea utilizando validadores estándares proporcionados por JSF o creando nuestros propios métodos validadores. En cuanto a la conversión de datos es posible asociar conversores específicos a los componentes y si es necesario crear conversores personalizados.

Struts valida un objeto ActionForm completo que representa todos los campos del formulario de entrada. Y para la conversión de datos utiliza la estándar de JavaBeans.

Tapestry también permite validar individualmente un componente, asociando el validador requerido al componente.

2.8.1.5. NAVEGACIÓN

La navegación en JSF es manejada por objetos oyentes de eventos, que se encargan de procesar los eventos generados por los componentes de UI. Los oyentes realizan algún proceso que devuelve un resultado lógico, que luego es usado por el sistema de navegación para seleccionar la siguiente página a mostrar. En JSF la navegación se la define en el archivo de configuración faces-config.xml mediante reglas de navegación con sus respectivos casos de uso.

La navegación en Struts se basa en objetos ActionForward; éstos son los encargados de definir a qué página ir o pasar el control una vez que se ha completado el Action.

En Tapestry la navegación se define en el método oyente y no en un archivo de configuración.

2.8.1.6. MODELO DE PROGRAMACIÓN JAVA

JSF ofrece mayor flexibilidad sobre dónde poner el código de las aplicaciones ya que no se limita a utilizar la clase java que corresponde a cada página JSP. JSF tiene un

mecanismo más intuitivo para el manejo de sesiones, cualquier página se puede conectar a un Managed Bean, así el ámbito sea de petición, sesión o aplicación.

Tapestry limita a la aplicación a usar las propiedades persistentes de una página.

2.8.1.7. INTERNACIONALIZACIÓN

En JSF y Struts permiten internacionalizar una aplicación mediante archivos *.properties que contienen pares de cadenas clave= valor y haciendo referencia a éstas claves en las vistas.

Para la internacionalización Tapestry utiliza catálogos de mensajes similares a los archivos .properties de JSF y Struts. Cada componente puede tener un conjunto de catálogos de mensajes; éstos llevan el mismo nombre del componente pero con extensión properties.

2.8.1.8. CICLO DE VIDA

JSF tiene un único ciclo de vida definido que consta de 6 fases. En cada punto después de la segunda fase es posible pasar directamente a la última fase que es mostrar la respuesta.

En Tapestry el ciclo de vida depende del Engine Service que lo invoca, esto se debe a que cada Engine Service tiene su propio ciclo de vida el mismo que está diseñado para adaptarse a la tarea que está tratando de lograr.

2.8.2. CONCLUSIONES

- ✚ Struts es más maduro que JavaServer Faces y desde que apareció ha tenido gran aceptación entre los desarrolladores y en la actualidad sigue evolucionando.
- ✚ La arquitectura flexible y extensible de JSF es una ventaja con respecto a otros Frameworks como Struts ya que permite tener distintos tipos de clientes y

construir interfaces de usuario más ricas; estas cosas son imposibles de lograr con Struts.

- ✚ En cuanto al manejo de eventos JSF tiene una gran ventaja sobre Struts ya que puede manejar múltiples eventos por página; al contrario de Struts que solamente maneja un evento por página.
- ✚ JSF permite validar y convertir individualmente datos de cada componente y además crear nuestros propios validadores y conversores. Mientras que Struts valida un ActionForm completo y para la conversión utiliza la estándar de JavaBeans.
- ✚ JSF y Struts manejan un modo de navegación declarativo y definen la navegación usando reglas en un archivo de configuración XML. Los dos Frameworks soportan dos tipos de navegación: estática y dinámica.
- ✚ JSF permite definir un control más fino sobre las reglas de navegación en una página. En Struts una petición se corresponde con una acción y cuando ésta finaliza se aplica un regla de navegación.
- ✚ Podemos desarrollar aplicaciones con cualquiera de los dos Frameworks pero JSF es más flexible y extensible que Struts y Tapestry; lo que representa una ventaja sobre Struts.
- ✚ JSF tiene un modelo de programación flexible lo que representa una ventaja ante Tapestry.
- ✚ El ciclo de vida de JSF es más fácil de entender. Mientras que el ciclo de vida de Tapestry se adapta a una tarea en especial lo que permite obtener una solución más adecuada para algunos problemas.
- ✚ Con respecto a la forma de manejar la navegación entre las páginas JSF tiene la ventaja ya que lo hace en un archivo de configuración y no lo mezcla con el código de la aplicación.
- ✚ Struts permiten un manejo más directo de los datos que se procesan, mientras que Frameworks más actuales como JSF y Tapestry buscan abstraer al

desarrollador del protocolo HTTP, crear modelos de componentes fácilmente extensibles y orientados a eventos.

- ✚ En una ventaja conocer varios Frameworks Web, y saber que no existe uno que sea la solución a todos los problemas y que cada uno ha sido desarrollado con objetivos diferentes.
- ✚ Al momento de seleccionar un Framework debemos escoger el que esté más acorde con los objetivos del proyecto a desarrollar y evaluar ventajas y desventajas de cada uno.

3. DESARROLLO DEL APLICATIVO

TEMA:

SISTEMA DE ADMINISTRACIÓN ESTUDIANTIL PARA LOS COLEGIOS FISCALES DE LA PROVINCIA DE IMBABURA CON HERRAMIENTAS DE SOFTWARE LIBRE.

3.1. FRAMEWORKS UTILIZADOS

Otro aspecto importante en el desarrollo de un proyecto es utilizar las mejores tecnologías disponibles en la actualidad.

Para la capa de la Vista y Controlador se utilizó el Framework Java Server Faces ya que proporciona mayor facilidad para: la creación de Interfaces de Usuario, mapeo entre los formularios y sus clases en el servidor, validación y conversión de datos, gestión de errores, internacionalización y la posibilidad de crear nuestros propios componentes UI, validadores y conversores personalizados de forma sencilla y fácil de mantener. Este Framework según los desarrolladores Java es el Framework de presentación web del futuro.

Para la persistencia de datos se utilizó Hibernate; que es un motor de persistencia de código abierto. Hibernate es una herramienta potente y madura para realizar el mapeo objeto – relacional. Cuenta con una amplia documentación tanto en libros publicados como en tutoriales gratuitos en su página web oficial.

3.2. ENTORNO DE DESARROLLO

Al ser JSF un estándar los IDEs más populares permiten desarrollar aplicaciones con este Framework. Entre los IDEs gratuitos más conocidos están: NetBeans, JDeveloper, Eclipse entre otros. Para el desarrollo de la aplicación en este caso seleccioné JDeveloper 10.1.3.3 de Oracle que es gratuito y se lo puede descargar de Internet, además de que cuenta con excelente documentación, es un IDE estable y no necesita mayores recursos de hardware.

3.3. SERVIDOR DE APLICACIONES

El servidor de aplicaciones seleccionado para poner en producción la aplicación es GlassFish Server ya que es un servidor de código abierto que implementa Java EE 5 que incluye de las últimas versiones de tecnologías como: JavaServer Pages, JavaServer Faces, Enterprise JavaBeans entre otros. [www 25]

Además este servidor es sencillo de instalar y no es necesario realizar configuraciones adicionales para poner una aplicación en producción como se indica en el anexo 1.






3.3.1. REQUISITOS DE HARDWARE Y SOFTWARE NECESARIOS

SISTEMA OPERATIVO	MEMORIA MINIMA	MEMORIA RECOMENDADA	ESPACIO MINIMO EN DISCO	ESPACIO EN DISCO RECOMENDADO	EN JVM
Suse Linux Enterprise Server	512 MB	1GB	250MB	500MB	J2SE 5.0 Java SE 6
Ubuntu Linux	512 MB	1GB	250MB	500MB	J2SE 5.0 Java SE 6
Windows Server 2000, Server 2003, XP Pro, Vista	1GB	2GB	500MB	1GB	J2SE 5.0 Java SE 6

Tabla 5.1.- Requisitos para instalación de GlassFish Server V2.1

3.3.2. BASE DATOS QUE CUMPLEN LOS REQUISITOS DE COMPATIBILIDAD DE JAVA EE

Entre las principales base de datos tenemos las siguientes:

-  Derby 10.2
-  Oracle 10g
-  Oracle 9i
-  Sybase
-  MS SQL 2005

✚ DB2 8.2 y 9.1

✚ MySQL 5.0

✚ PostgreSQL 8.2

3.4. BASE DE DATOS

PostgreSQL es un sistema de base de datos relacional de software libre que destaca por su robustez, escalabilidad y cumplimiento de los estándares SQL. Cuenta con versiones para una amplia gama de sistemas operativos, entre ellos: Linux, Windows, Mac OS X, Solaris y otros más.

PostgreSQL soporta la realización de transacciones seguras; vistas, uniones, claves foráneas, procedimientos almacenados, triggers, etcétera.

Incluye una variedad de tipos de datos como son int, boolean, varchar, smallint, Date, numérico, moneda entre otros. [www 26]

3.5. PERSISTENCIA CON HIBERNATE

Con la creación de la capa de persistencia se consigue que los desarrolladores no necesiten conocer nada acerca del esquema utilizado en la BBDD, más bien conocerán el interface proporcionado por el motor de persistencia. De esta manera conseguimos separar de manera clara y definida, la lógica de negocios de la aplicación con el diseño de la BBDD. [www 27]

Como capa de persistencia desarrollada tan solo tenemos que adaptarla a nuestra arquitectura.

Hibernate es una herramienta de Mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación, mediante archivos declarativos XML que permiten establecer estas relaciones. Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL.

Como todas las herramientas de su tipo, Hibernate busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora orientación a objetos y el usado en las bases de datos

modelo relacional. Para lograr Hibernate permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen. Con esta información Hibernate le permite a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la Programación Orienta a Objetos. Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL. Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

Hibernate está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente.

Hibernate ofrece también un lenguaje de consulta de datos llamado HQL, al mismo tiempo que una API para construir consultas programáticamente conocidas como *criteria*.

Hibernate para Java puede ser utilizado en aplicaciones Java independientes o en aplicaciones Java EE, mediante el componente Hibernate Annotations que implementa el estándar JPA, que es parte de esta plataforma. [www 28]

3.6. DEFINICIÓN DEL PROBLEMA

Hoy en día, los sistemas informáticos están presentes en la mayoría de las actividades laborales que desempeña el ser humano, facilitando dichas actividades al usuario del sistema, permitiéndole hacer su trabajo en menor tiempo y de manera eficiente.

Existen sistemas informáticos para entidades bancarias, para llevar la contabilidad de una empresa, para comercio electrónico, y para todas las áreas que podamos imaginar las cuales han automatizado sus procesos, sin ser una excepción el área educativa.

En nuestro país el estado asigna a la educación un bajo porcentaje del presupuesto general, lo que no ha permitido a todos los colegios adquirir un sistema informático para realizar las actividades propias de estas instituciones como son matriculación de estudiantes, ingreso de notas, registro de asistencia de docentes y estudiantes entre otros. Y las instituciones que lo han logrado adquirir en muchos casos lo han hecho por autogestión.

En nuestra provincia son pocas las instituciones educativas que tienen automatizados los procesos antes mencionados y que para ello cuentan con un sistema informático. Mientras que la gran mayoría de instituciones educativas no cuentan con un sistema informático por lo cual aún realizan su trabajo manualmente lo que conlleva muchos problemas como son:

- ✚ Tienen toda la información de los estudiantes muchas veces solo en papel, o en programas como Excel o Word, sin contar con una base de datos relacional, por lo que si se llegara a dañar el o los archivos que contiene información valiosa de los estudiantes como son notas, o el computador que contiene estos archivos se perdería toda la información, ocasionando graves problemas para la institución y los estudiantes.
- ✚ La elaboración de reportes tanto internos a nivel de la institución, como los reportes que se entregan anualmente a la Dirección de Educación, les lleva mucho tiempo realizarlos y éstos son entregados con retraso y muchas veces con errores debido a la gran cantidad de información que tienen que manejar.

3.7. OBJETIVO GENERAL

Automatizar los procesos que se llevan a cabo en un Colegio Fiscal, con el fin de facilitar la labor del personal administrativo y docente de estas instituciones educativas y a la vez brindar un servicio eficiente a los estudiantes.

3.8. DEFINICIÓN DE REQUERIMIENTOS

- ✚ El sistema contará con diferentes tipos de usuarios, los cuales tendrán distintos permisos de acceso a las distintas funcionalidades del sistema de acuerdo a su perfil.

Los tipos de usuario del sistema son:

- **Administrador**
- **Secretaria**
- **Inspector**
- **Docente**

- ✚ Cualquier usuario independientemente del perfil que tenga puede cambiar su clave de acceso al sistema.
- ✚ Al ingresar al sistema el usuario debe autenticarse con un login y password, y de acuerdo al tipo de usuario al que pertenezca se le presenta el menú con las diferentes opciones a las que puede acceder.
- ✚ Todos los usuarios deben tener una opción que les permita salir del sistema.
- ✚ El administrador del sistema debe tener las siguientes opciones:
 - Puede crear nuevos usuarios o eliminar los que ya existen.
 - Puede crear y cerrar un nuevo periodo escolar.
 - Puede ingresar el horario de clases de cada curso validando que no haya cruces de horas.
 - Puede crear nuevas especialidades, materias y cursos.
 - Puede ingresar los miembros del Consejo Directivo.
- ✚ Los usuarios que pertenezcan al tipo docente deben tener las siguientes opciones:
 - Imprimir su horario personal
 - Ingreso de notas parciales validando que éstas estén entre el rango permitido (1 a 20 y que puedan ser con decimales por Ej.: 17.5) mismas que serán utilizadas para calcular el promedio trimestral y modificarlas en caso de que sea necesario.
 - Ingresar notas de supletorios validando que estén en el rango permitido (1 a 20).
 - Ingresar notas de exámenes de grado validando que estén en el rango permitido (1 a 20).
 - Imprimir la nómina de estudiantes de cada una de las materias que dicta.
 - Imprimir un reporte de calificaciones por trimestre de las diferentes materias que imparte.
 - Crear e imprimir un sílabo de las materias que tiene a su cargo.
 - Registrar el avance de las materias lo que permite a su vez controlar la asistencia de los docentes.

✚ Para el usuario que sea de tipo secretaria el sistema debe tener las siguientes funcionalidades:

- Ingresar e imprimir el distributivo docente para lo cual el sistema debe validar que una materia de un curso no sea asignada a dos o más docentes.
- Ingresar datos de nuevos estudiantes realizando las validaciones que sean necesarias.
- Imprimir el horario por cursos.
- Imprimir la nómina de estudiantes de cada curso.
- Imprimir la lista de estudiantes que se quedan a supletorios por cursos.
- Imprimir la lista de estudiantes que pierden el año de cada curso.
- Imprimir los promedios generales de los estudiantes de un curso.
- Imprimir certificados de matrícula.
- Matriculación de estudiantes, verificando si el estudiante a aprobado o no el año lectivo anterior.
- Debe permitir el cambio de curso de un estudiante, verificando que el cambio sea a un curso del mismo nivel por Ej.: Cambiar de 8vo año paralelo "A" al paralelo "B".
- Permitir editar la matrícula de un estudiante en el caso de que éste decida retirarse de la institución.
- Registrar nuevo personal, realizando las validaciones que sean necesarias, como que los dos primeros dígitos del número de cédula empiece por códigos de provincias entre otras.
- Registrar a los estudiantes que conforman el gobierno estudiantil.
- Generar las estructuras que serán enviadas cada fin de año lectivo a la dirección de educación.
- De cada estudiante puede imprimir la libreta de calificaciones.
- Al fin del año lectivo tiene la opción de imprimir los certificados de promoción, indicando a que curso es promovido el estudiante.
- También puede actualizar los datos informativos de los estudiantes, con la finalidad de contar con información real.

- Permite completar los promedios obtenidos por los estudiantes que no han estado desde 8vo año en la misma institución, con la finalidad de poder obtener una nota de graduación y por consecuencia generar el acta y copia de acta de grado.
- Puede imprimir el acta y copia de acta de los estudiantes que están en tercer año de bachillerato.
- ✚ Un usuario que sea del tipo inspector debe tener las siguientes opciones:
 - Puede registrar a los estudiantes que no han asistido a clases.
 - Imprimir un reporte de todos los estudiantes que han faltado en un periodo de tiempo dado.
 - Calificar la conducta de los estudiantes de los cursos que se encuentran a su cargo.
 - Registrar los docentes que no han asistido a clases.
 - Obtener un reporte de las faltas de los docentes en un periodo de tiempo dado.

3.9. DISEÑO MEDIANTE UML

Lenguaje Unificado de Modelado, es un lenguaje gráfico que permite visualizar, construir y documentar un sistema.

Los diagramas UML son un conjunto de herramientas que permiten modelar sistemas orientados a objetos. [www 29]

Los diagramas utilizados para modelar el sistema son los siguientes:

- ✚ Diagrama de casos de uso
- ✚ Diagrama de secuencias
- ✚ Diagrama de colaboración
- ✚ Diagrama de actividades
- ✚ Diagrama de clases

3.9.1. CASOS DE USO

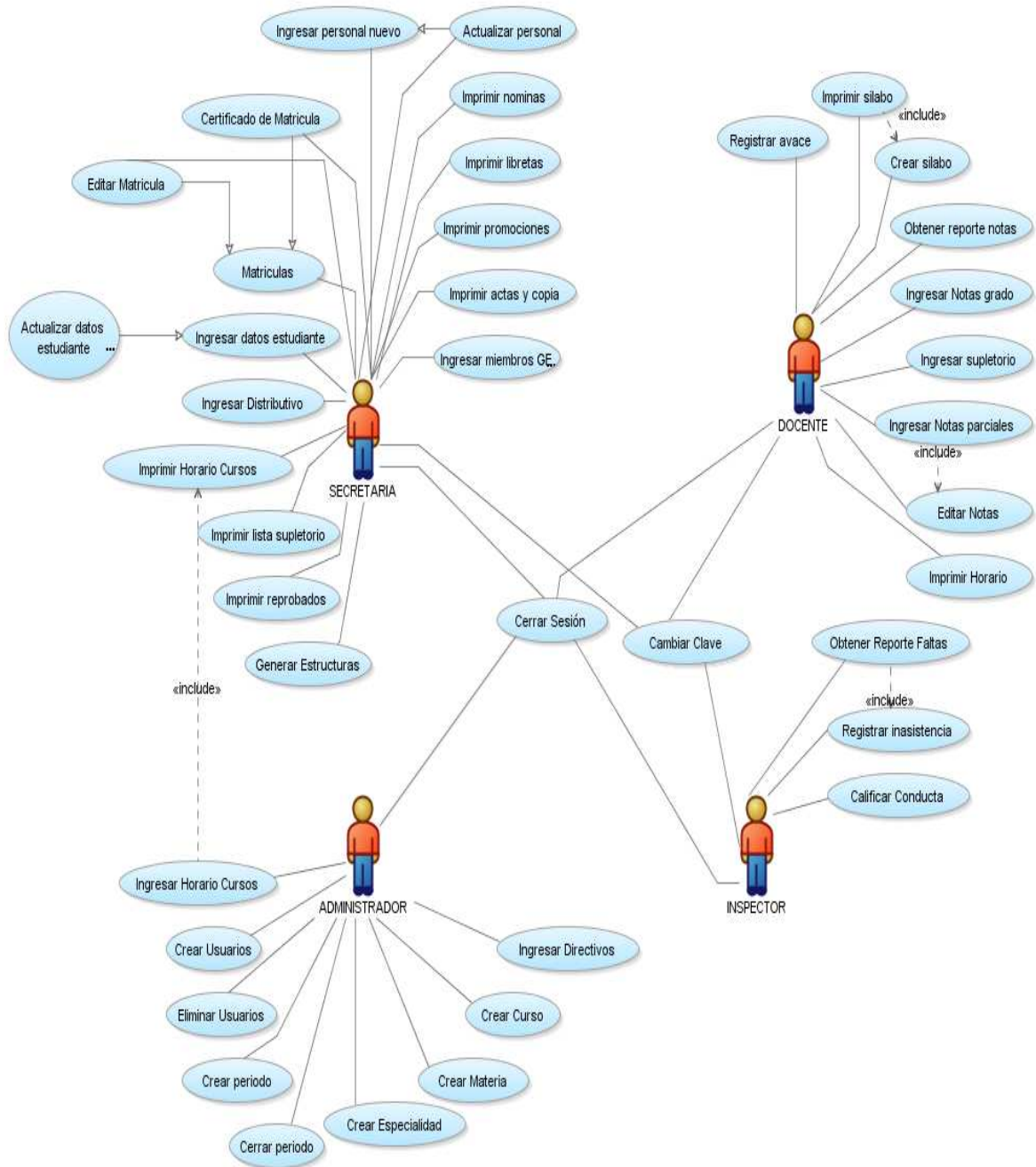


Figura 5.1.- Diagrama de Casos de Uso de la Aplicación

3.9.2. DIAGRAMA DE CLASES

Este diagrama describe la estructura de un sistema mostrando sus clases, atributos y las relaciones que existe entre ellos, este diagrama se utiliza durante el proceso de análisis y diseño.

3.9.3. DIAGRAMAS DE SECUENCIA

Este diagrama muestra el intercambio de mensajes en un momento dado, este diagrama pone énfasis en el orden y momento que se envían mensajes a los objetos. Contribuye a describir la dinámica del sistema en términos de la interacción entre sus objetos a través de mensajes.

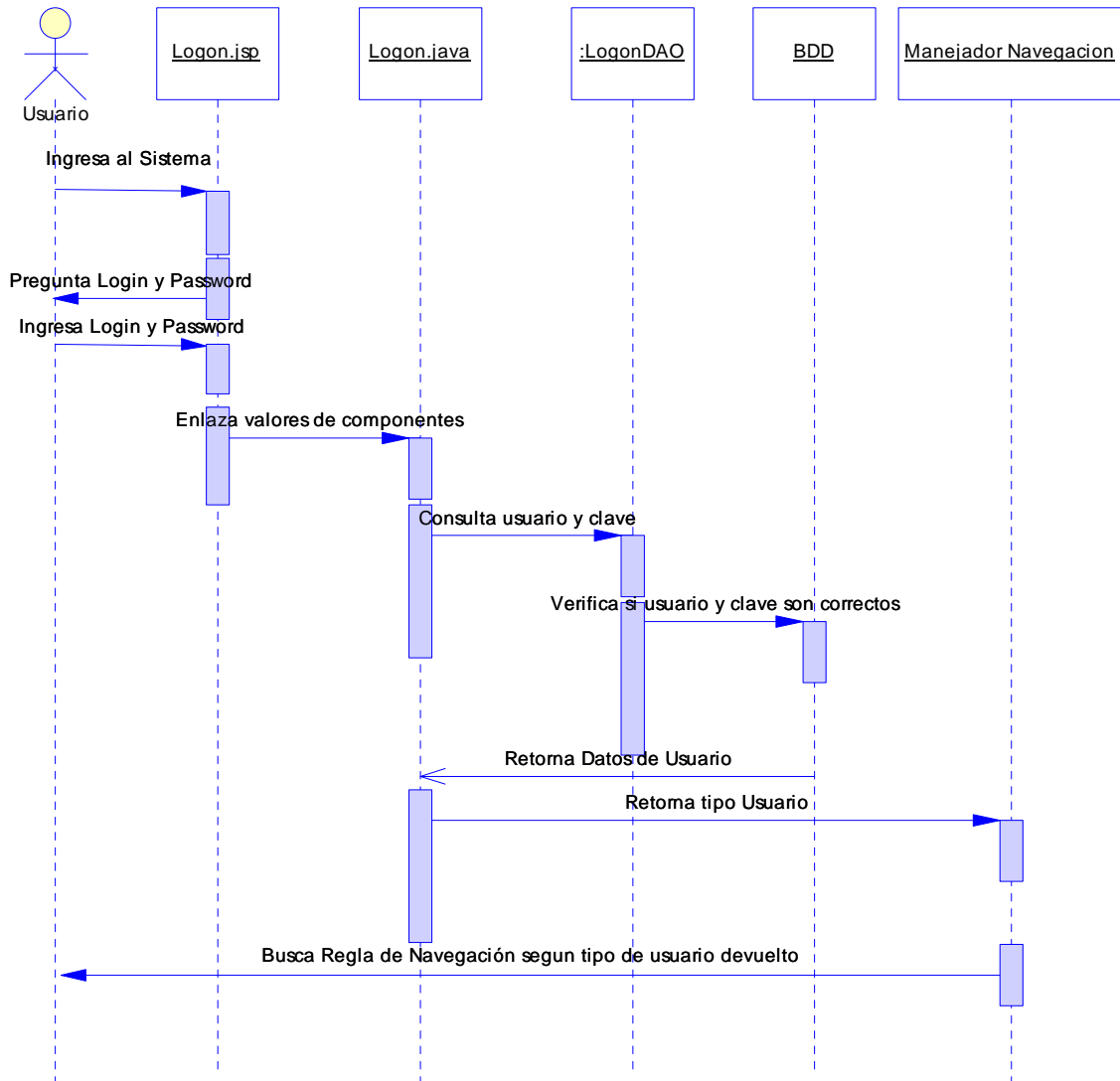


Figura 5.3.- Diagrama de Secuencia de Login

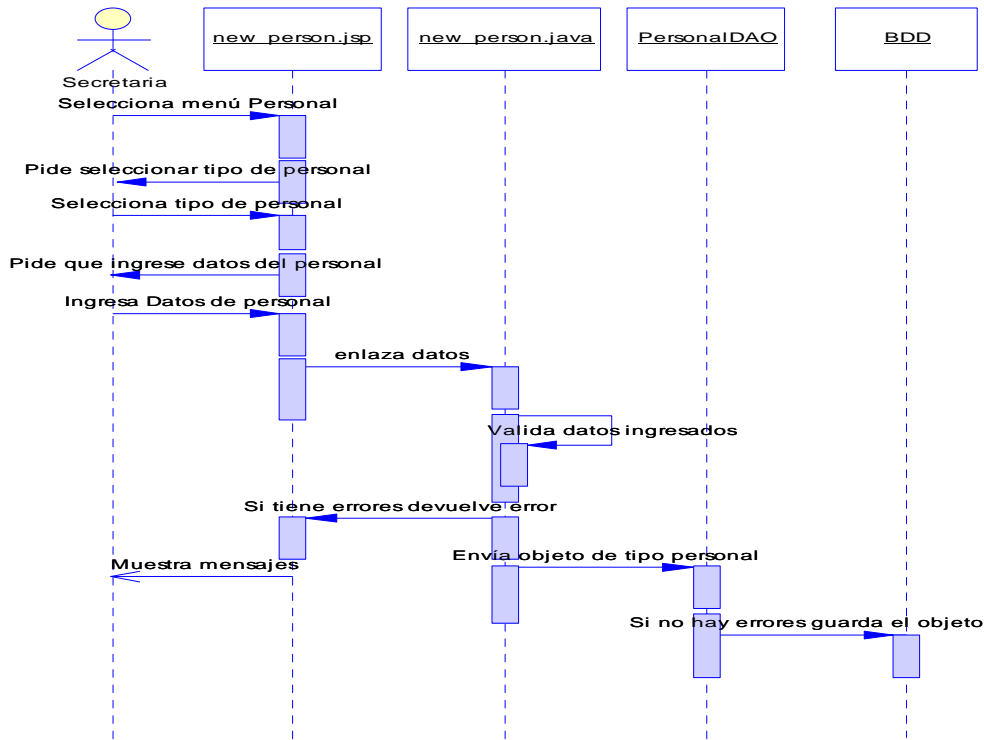


Figura 5.4.- Diagrama de Secuencia para Registrar Personal

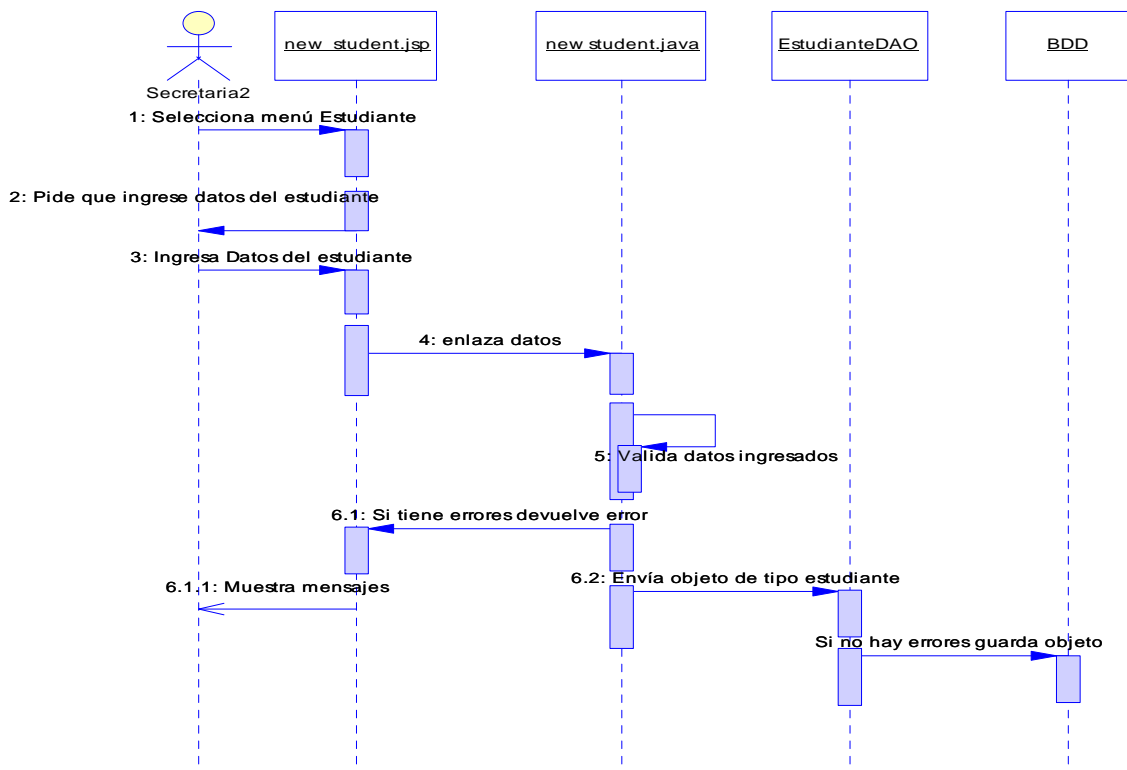


Figura 5.5.- Diagrama de Secuencia para Registrar un Estudiante

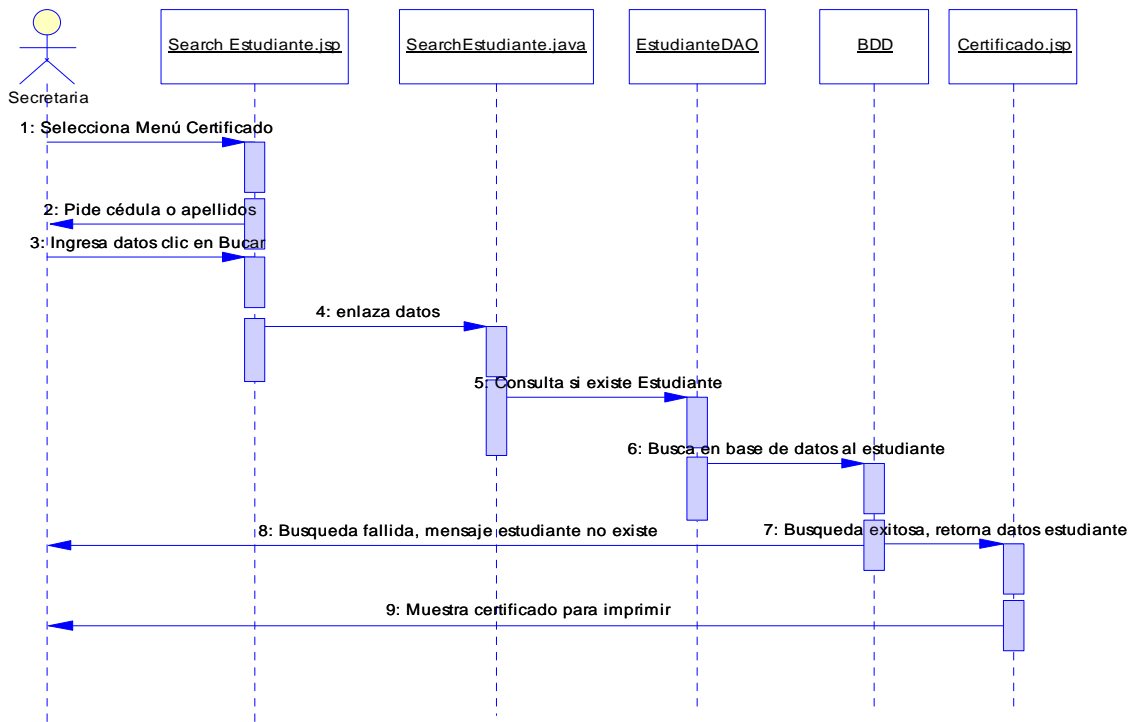


Figura 5.6.- Diagrama de Secuencia para Obtener Certificado de Matrícula

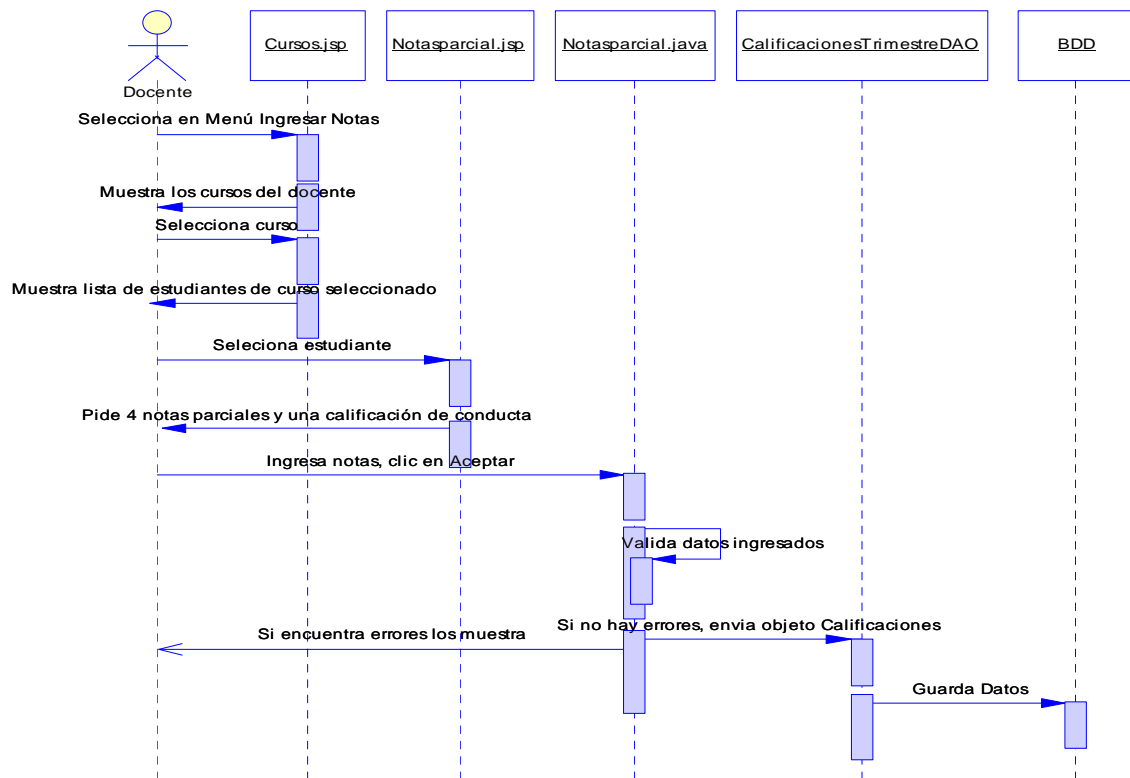


Figura 5.7.- Diagrama de Secuencia para Ingreso de Notas Parciales por Trimestre

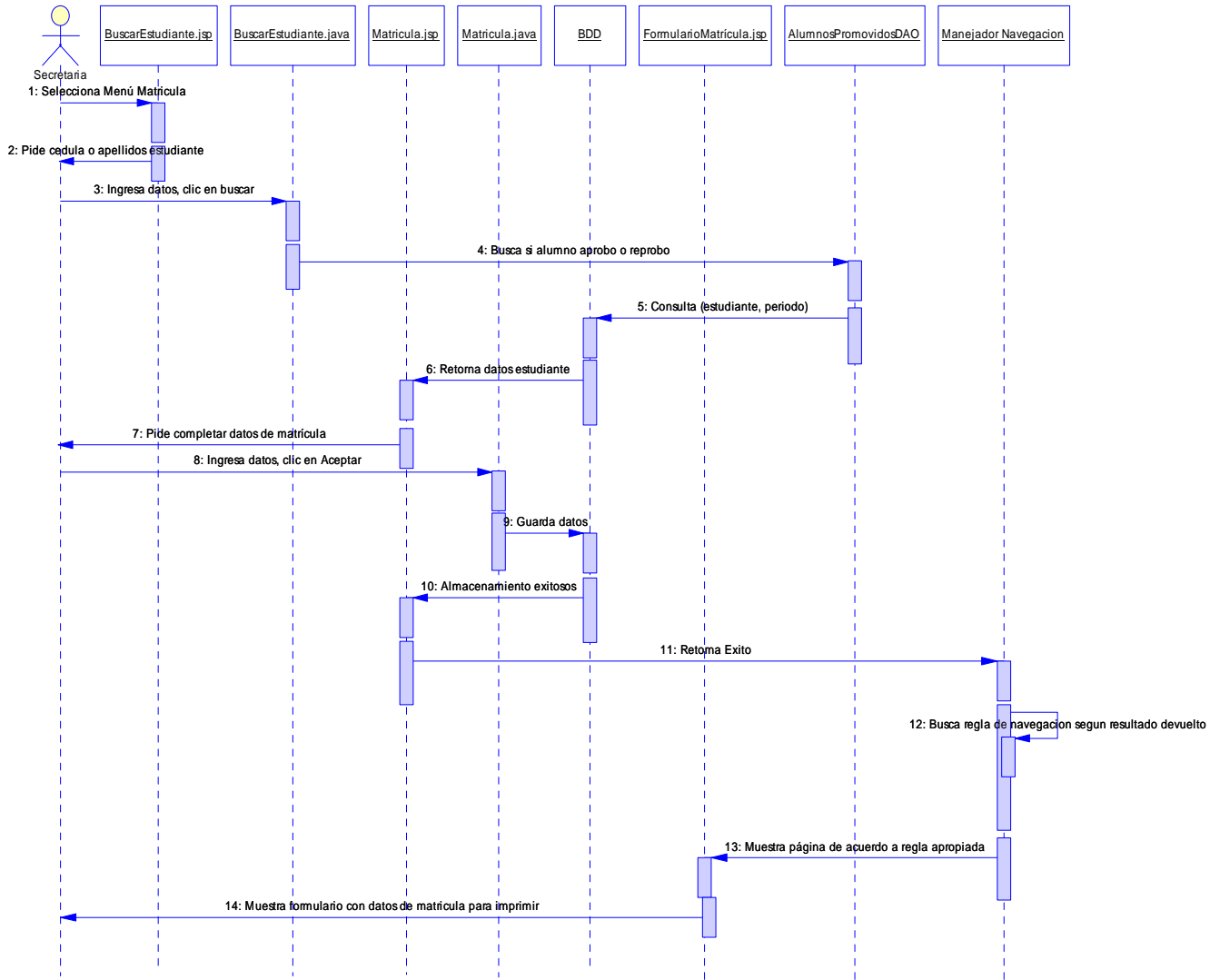


Figura 5.8.- Diagrama de Secuencia para realizar una Matrícula

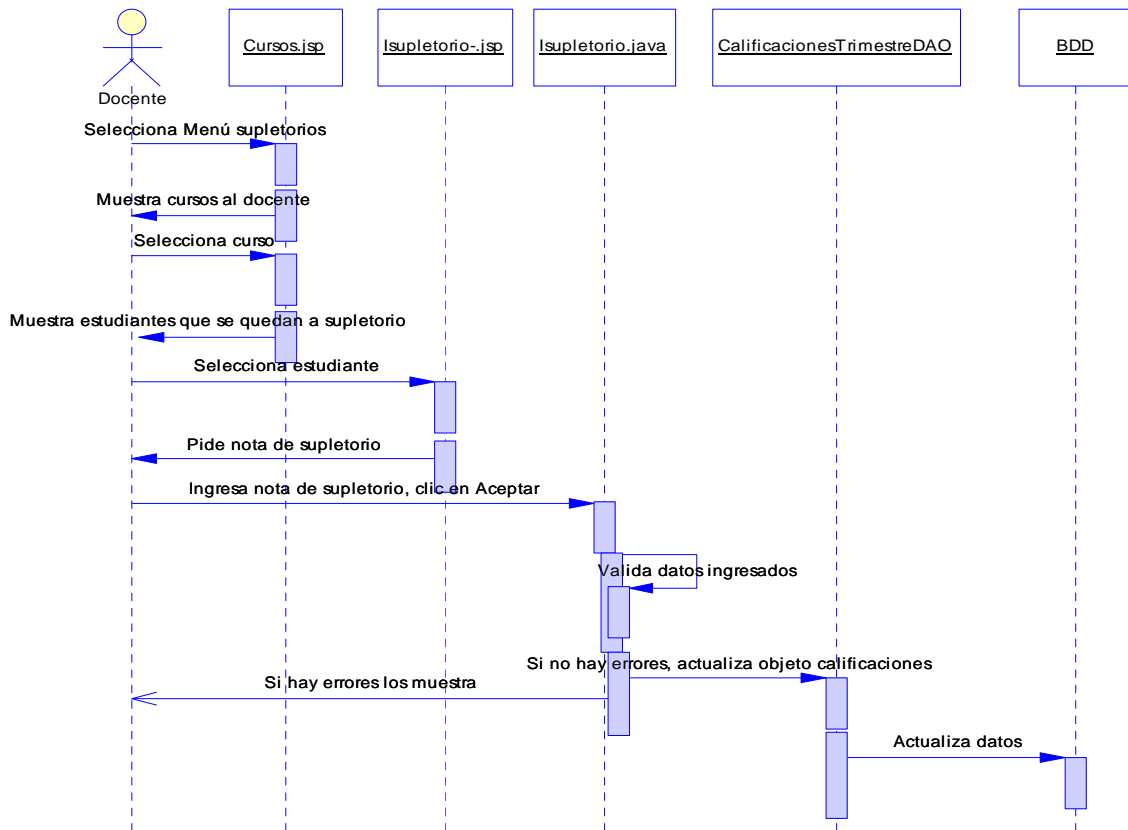


Figura 5.9.- Diagrama de Secuencia para Ingresar Notas de Supletorio

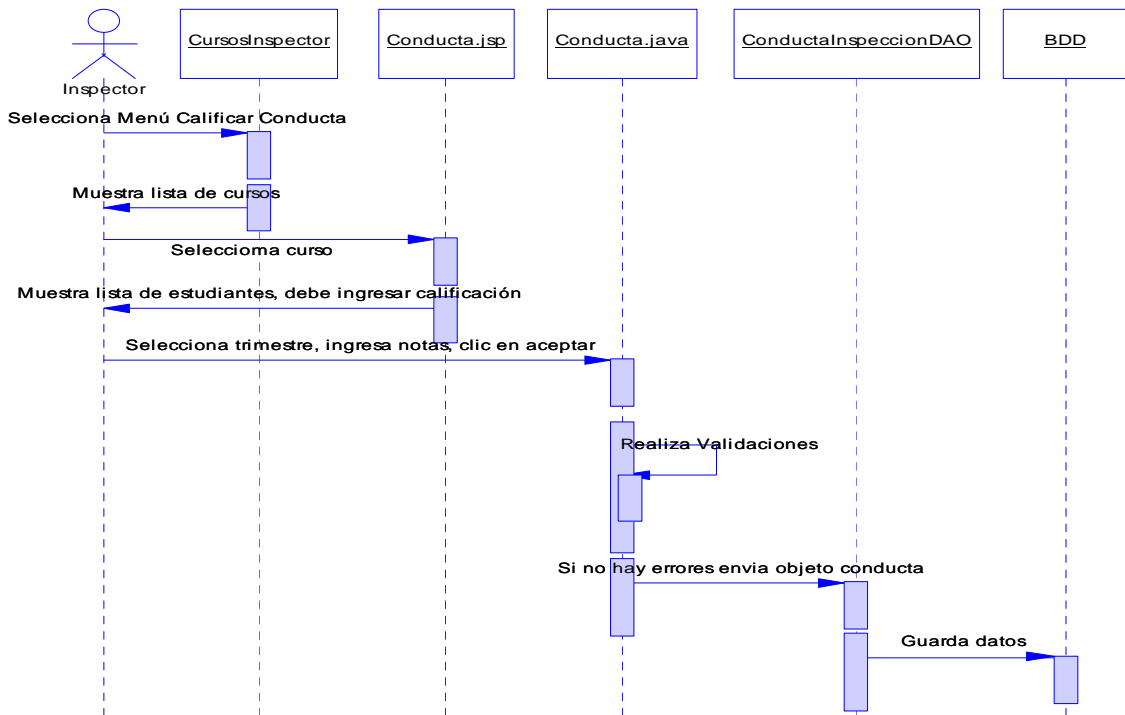


Figura 5.10.- Diagrama de Secuencia para calificar la Conducta

3.9.4. DIAGRAMA DE ACTIVIDADES

Este diagrama describe la secuencia de acciones de un sistema. Los diagramas de actividades son diagramas de flujo, con algunos elementos adicionales

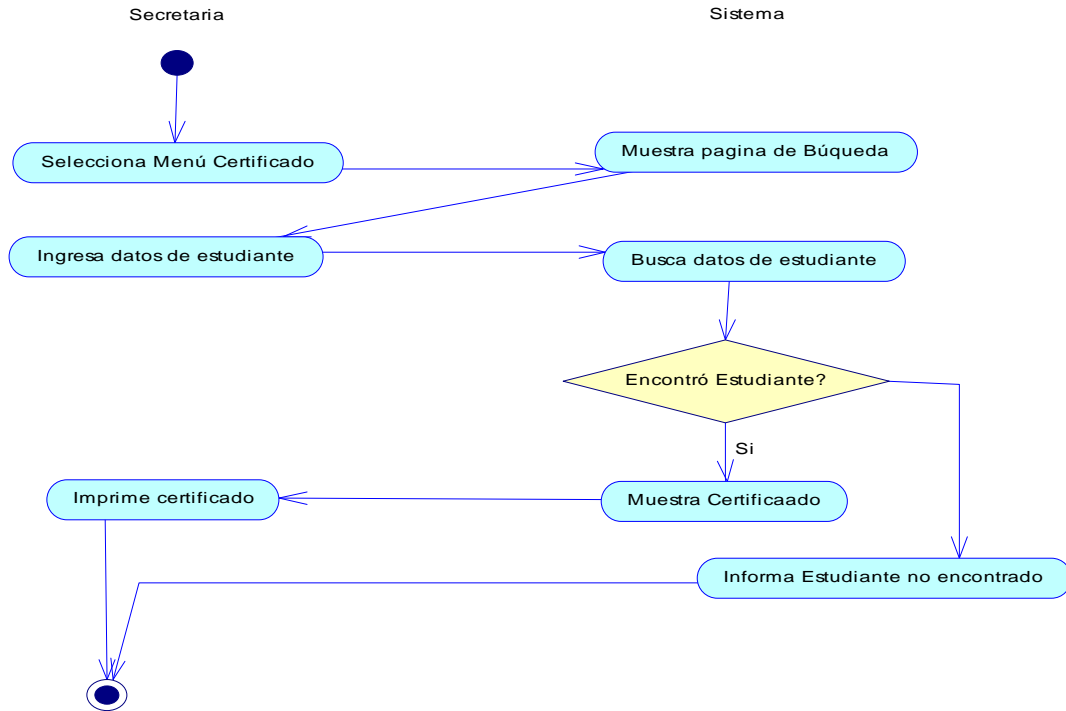


Figura 5.11.- Diagrama de Actividades para Imprimir Certificado de Matrícula

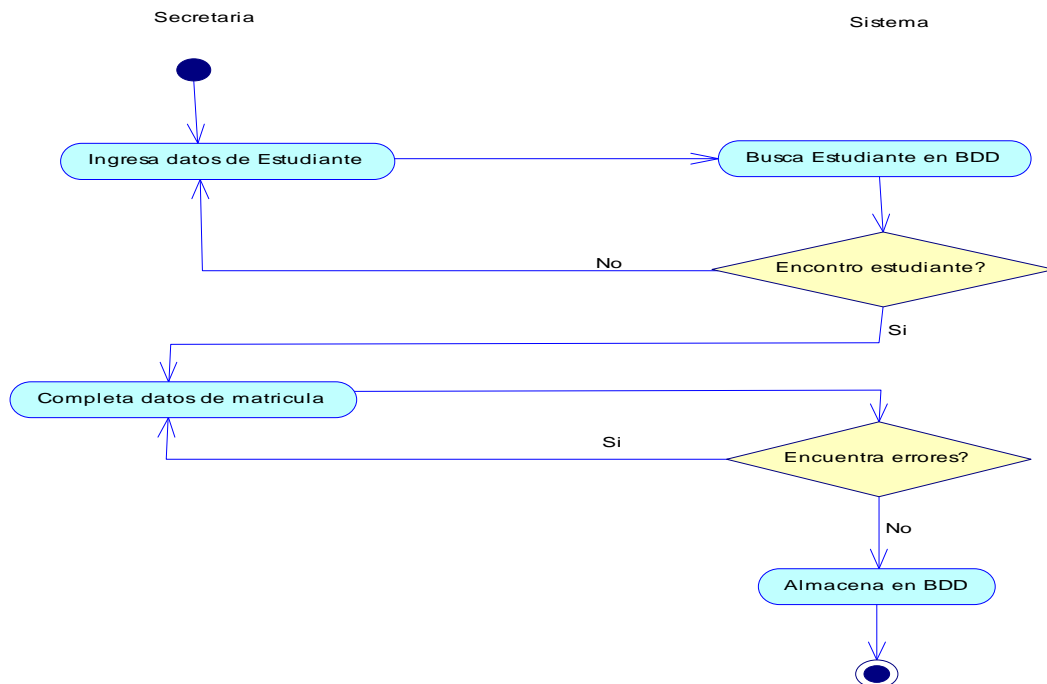


Figura 5.12.- Diagrama de Secuencia para Matricular un estudiante

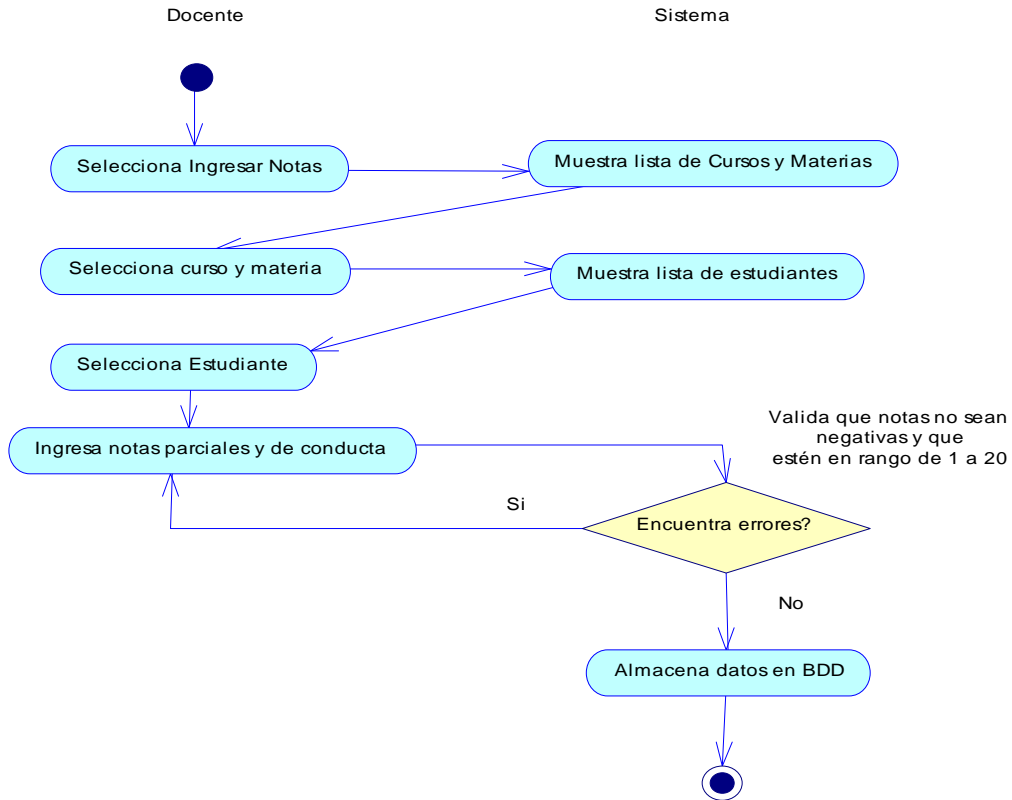


Figura 5.13.- Diagrama de Secuencia para Ingreso de Notas Parciales

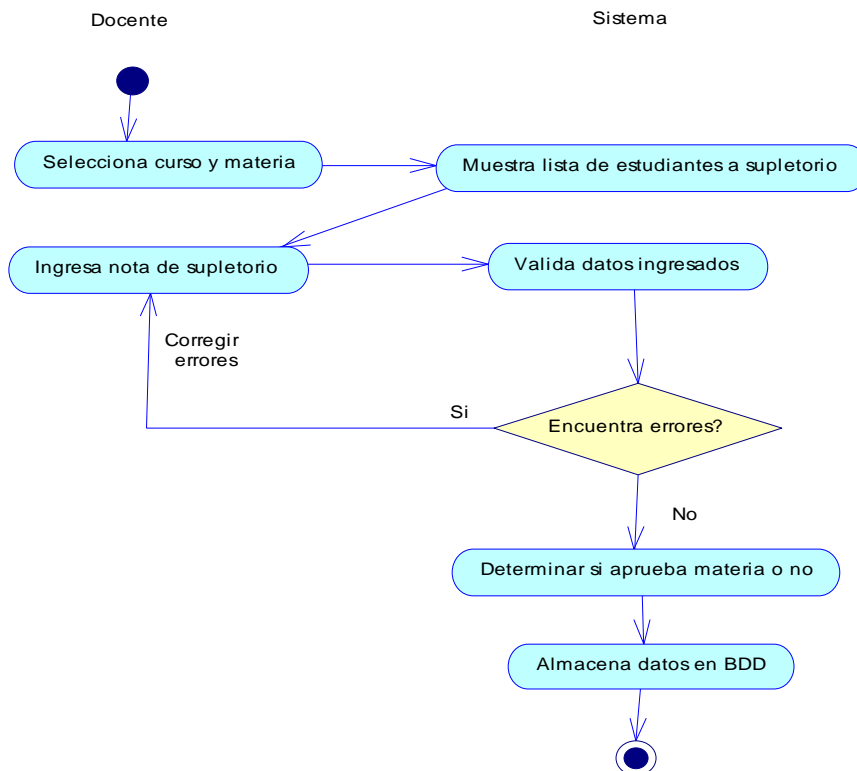


Figura 5.14.- Diagrama de Secuencia para Ingreso de Notas de supletorio

3.9.5. DIAGRAMAS DE COLABORACIÓN

Muestran las interacciones que ocurren entre objetos que participan en determinada situación. Destacan la forma en que las operaciones se producen en el tiempo.

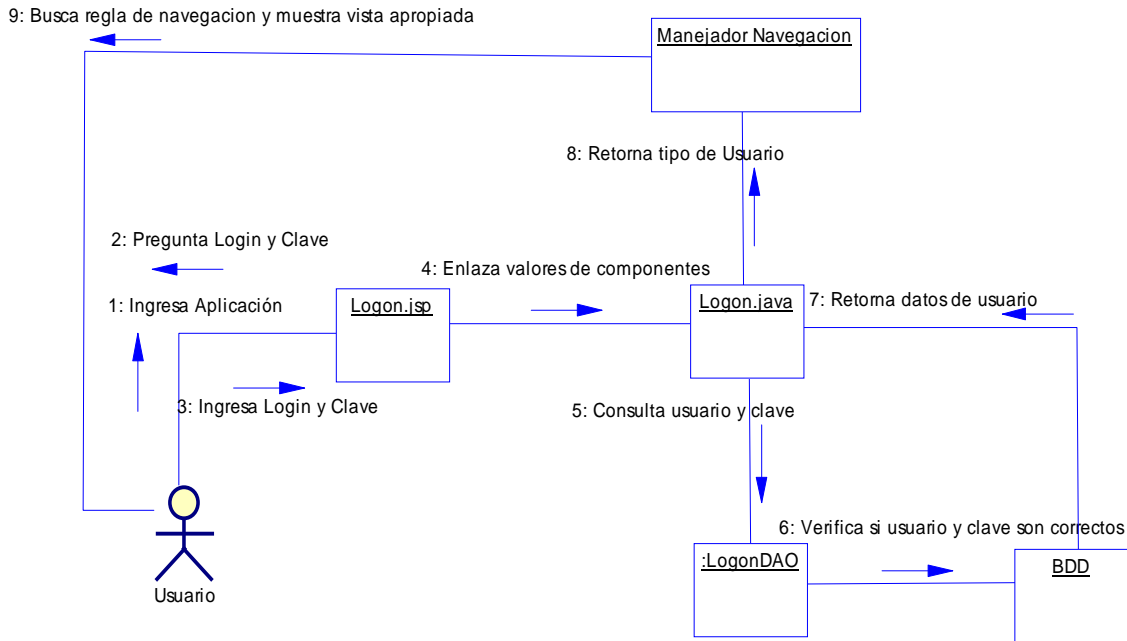


Figura 5.15 Diagrama de Colaboración para Ingresar al sistema

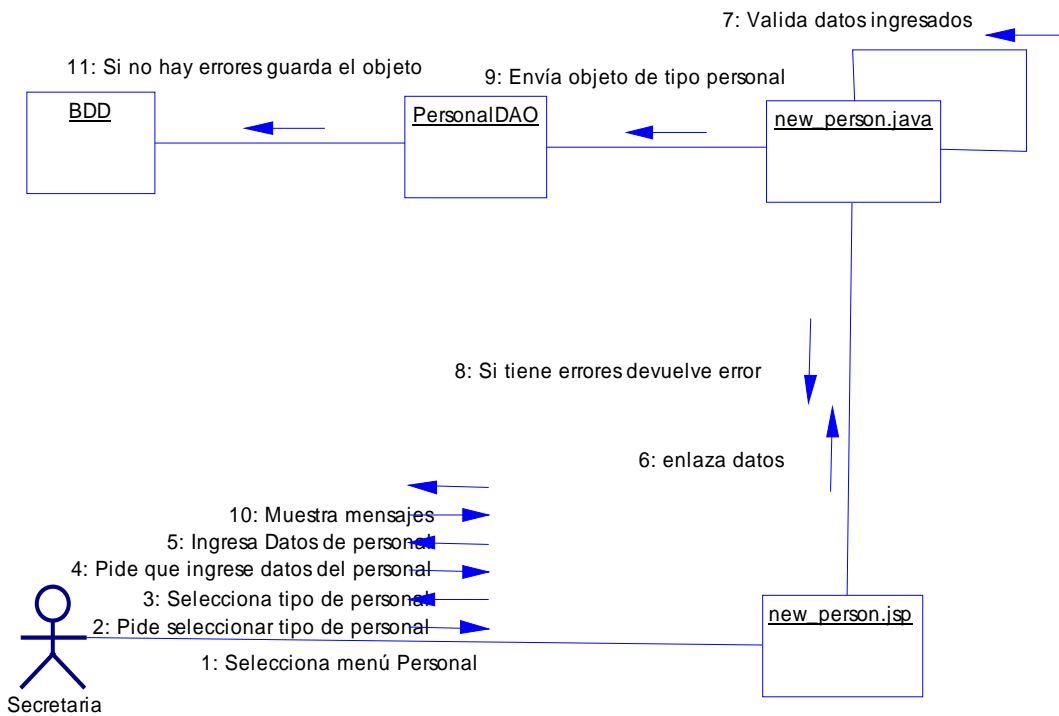


Figura 5.16 Diagrama de Colaboración para registrar datos de personal

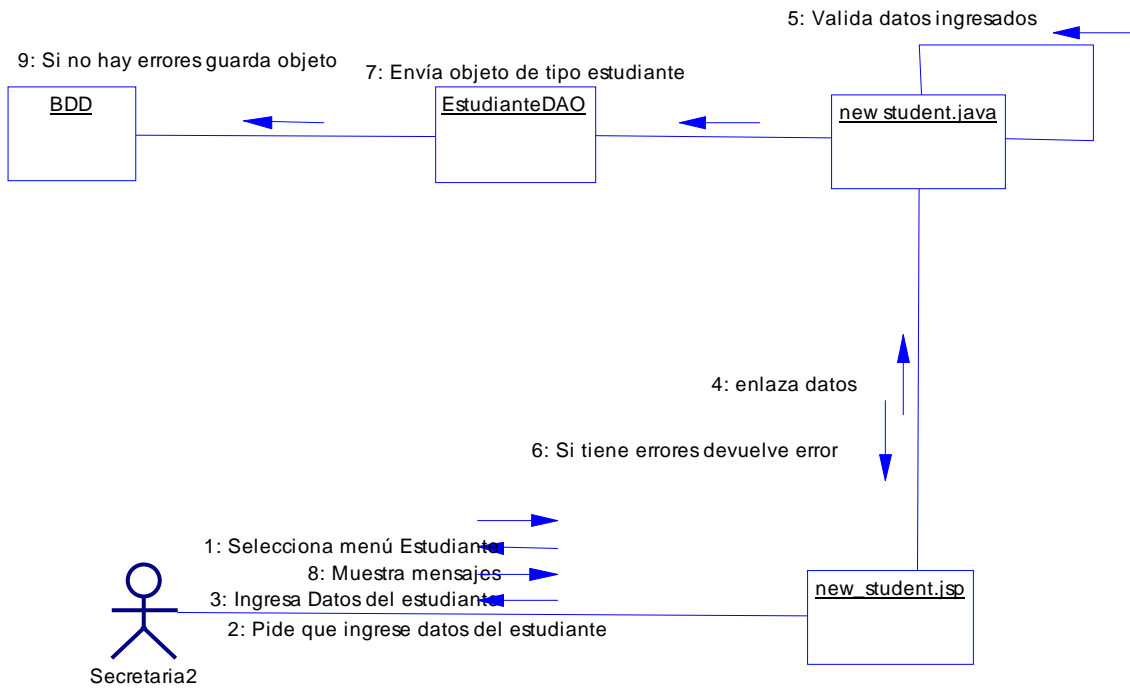


Figura 5.17 Diagrama de Colaboración para registrar un estudiante

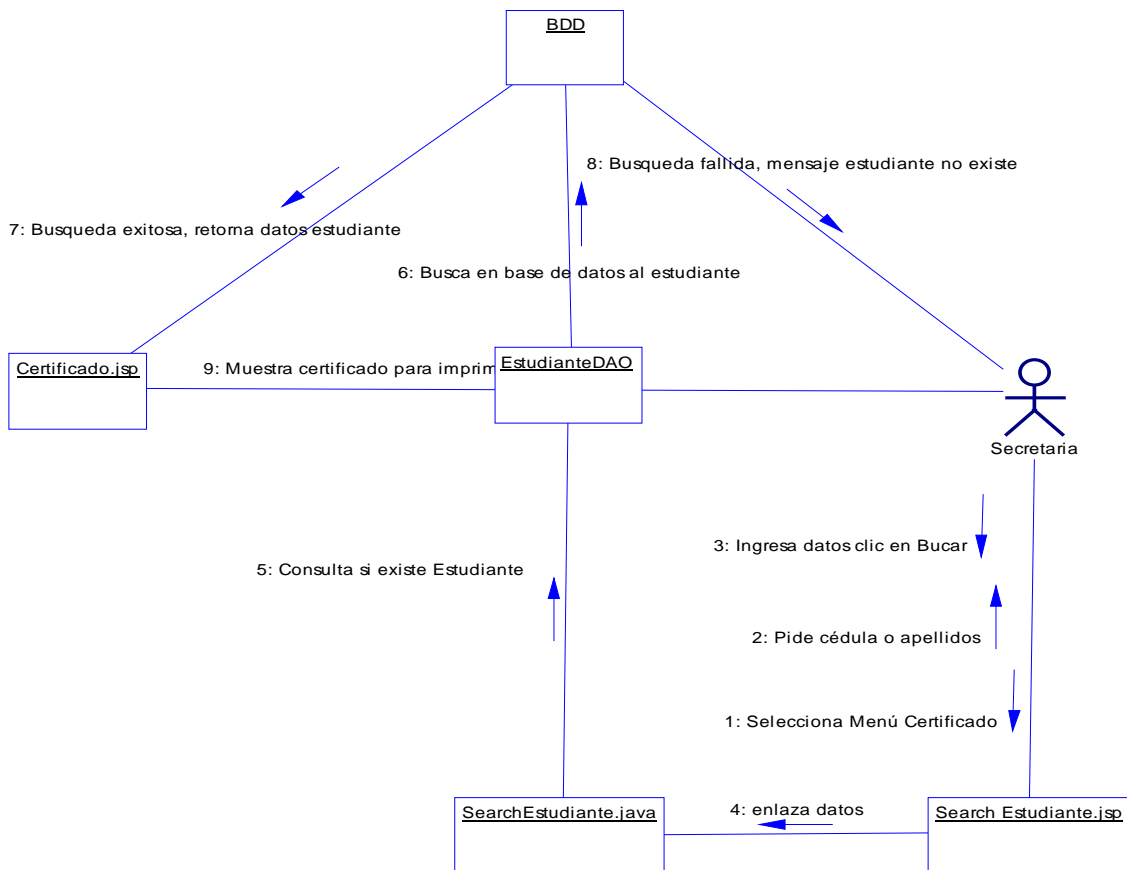


Figura 5.18 Diagrama de Colaboración para obtener certificado de matrícula

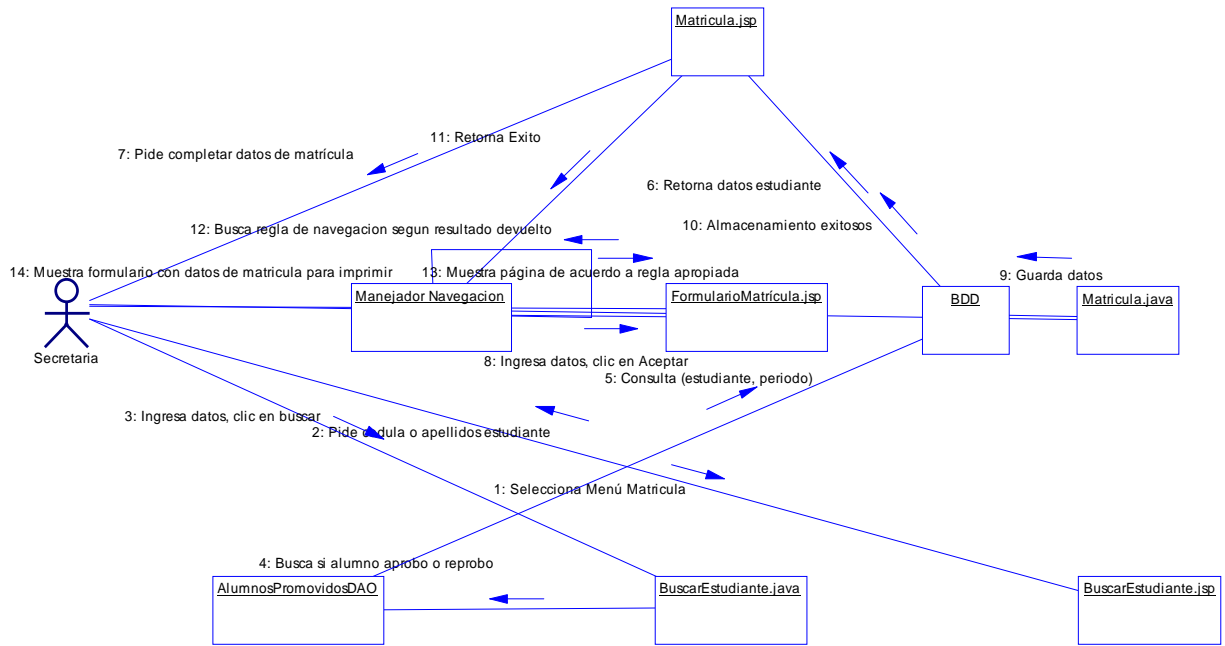


Figura 5.19 Diagrama de Colaboración para matricular un estudiante

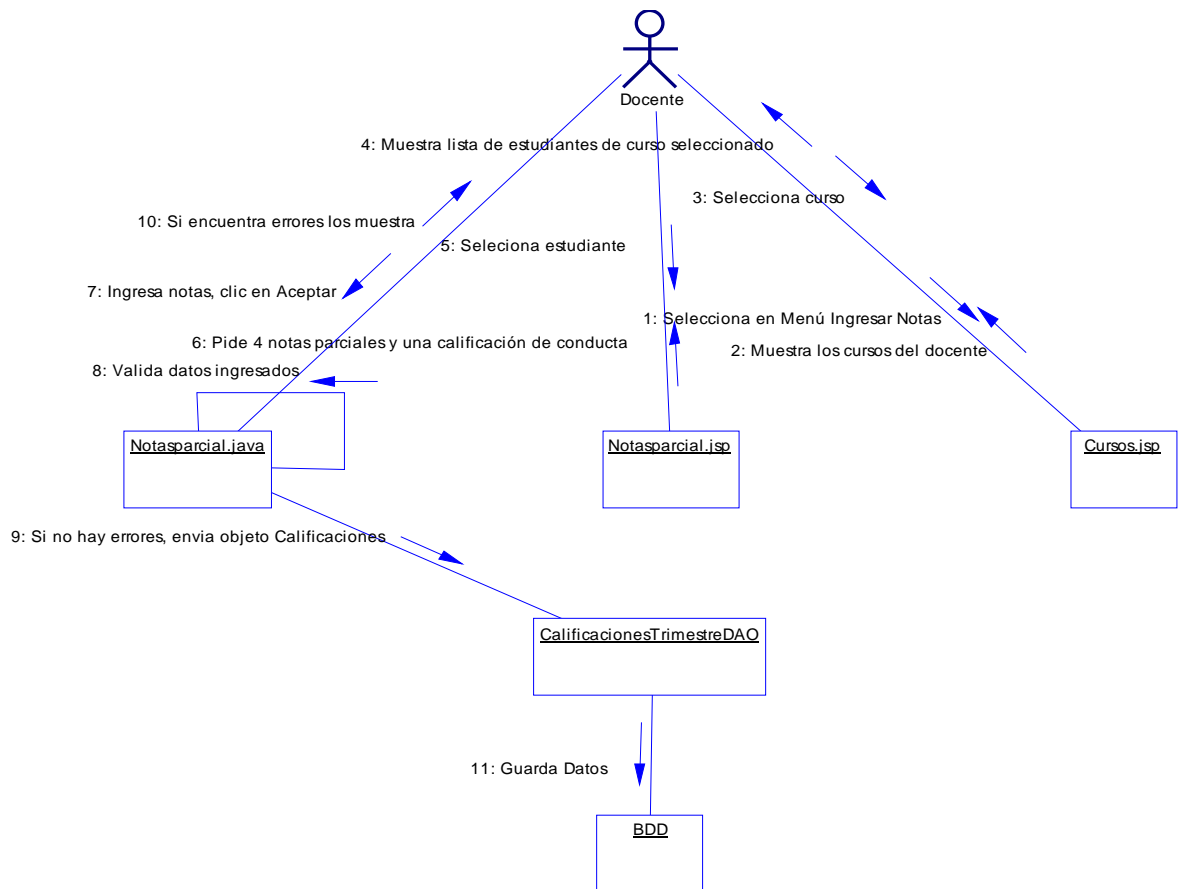


Figura 5.20 Diagrama de Colaboración para ingresar notas parciales

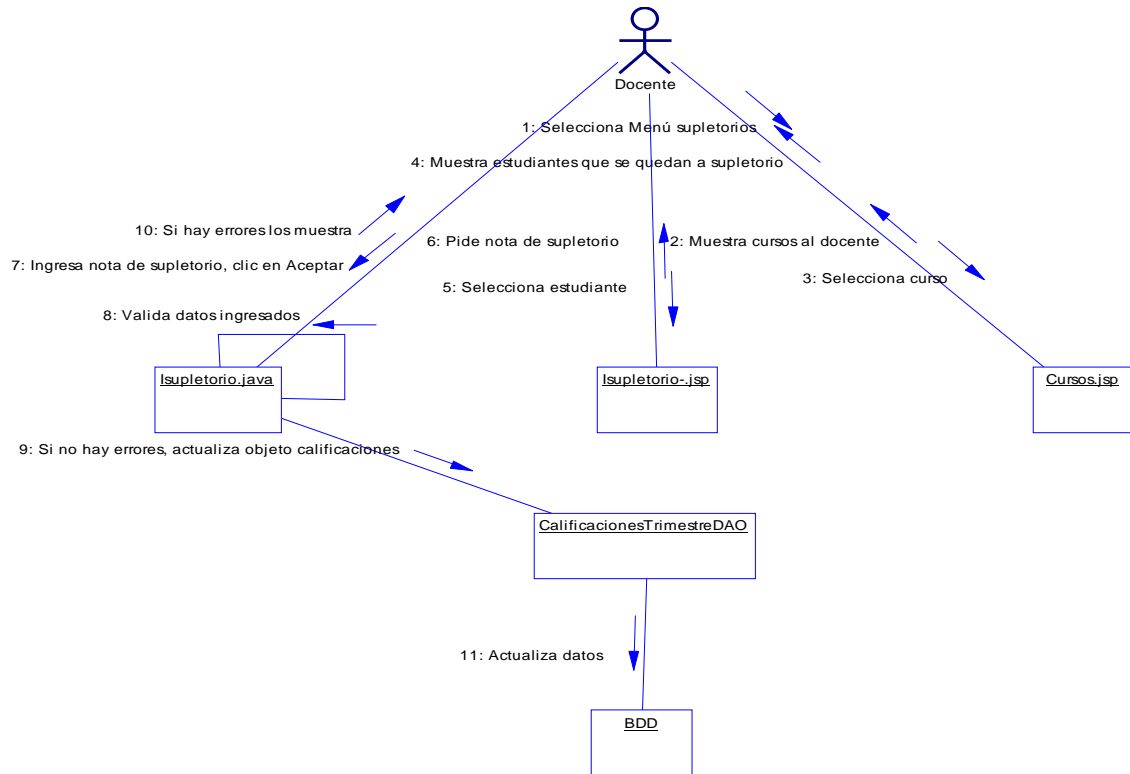


Figura 5.21 Diagrama de Colaboración para ingresar notas de supletorio

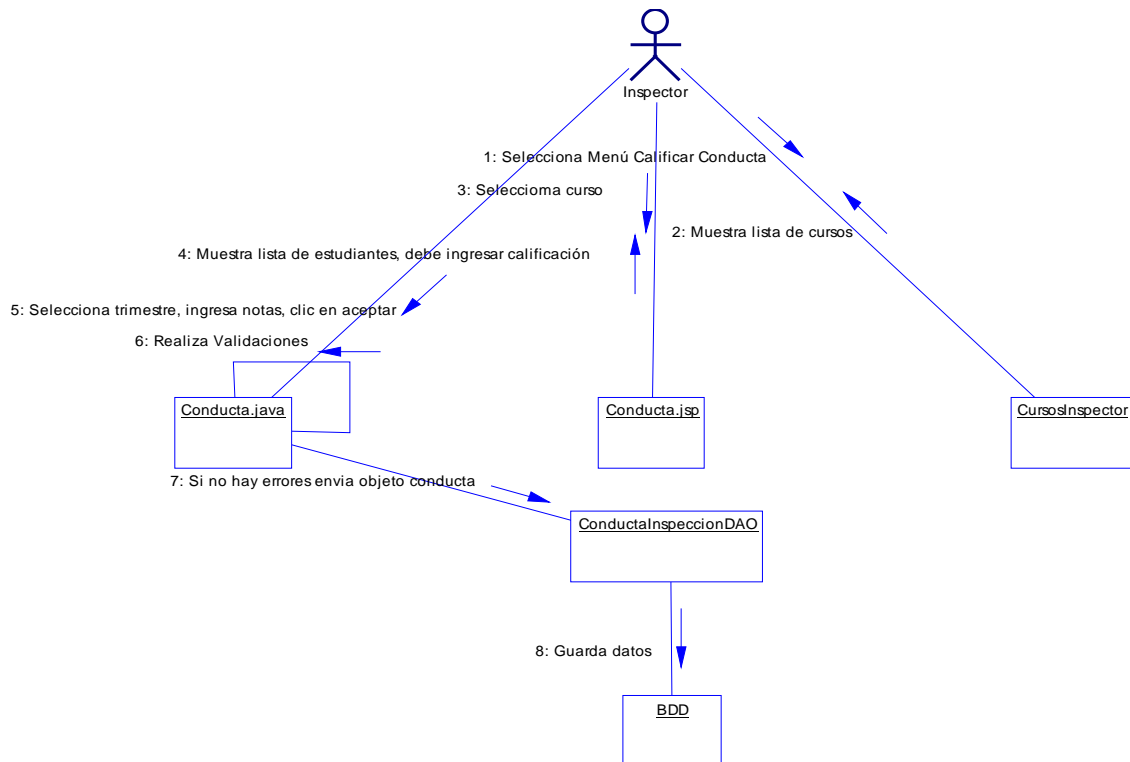


Figura 5.22 Diagrama de Colaboración para calificar conducta

3.9.6. MODELO DE DATOS

3.9.6.1. DESCRIPCIÓN DE LAS TABLAS

TABLA	DESCRIPCION
ALUMNOS_PROMOVIDOS	Esta tabla contiene todos los estudiantes que han aprobado o reprobado un año lectivo.
ASISTDOCENTE	Esta tabla contiene las inasistencias de los docentes.
ASISTENCIA	Esta tabla contiene las faltas de los estudiantes.
AVANCE_MATERIA	En esta tabla se almacena un registro del avance de clase en cada materia.
CALIFICACIONES_TRIMESTRE	Esta tabla contiene los promedios de cada trimestre, notas de supletorios y calificación de conducta.
CAPITULOS_UNIDADES	En esta tabla se almacenas los capítulos registrados en el sílabo de cada materia.
CONDUCTA_INSPECCION	Esta tabla guarda la calificación que el inspector asigna a cada estudiante.
CONTENIDO_UNIDADES	Almacena las unidades que tiene cada capítulo de una materia.
CURSO	Esta tabla contiene todos los cursos que hay en la institución.
DOCENTE	Contiene todos os datos del personal docente de una institución.
ESPECIALIDADES	Contiene todas las especialidades de la institución.
ESTUDIANTE	Contiene los datos informativos de todos los estudiantes de la institución.
GOBIERNO_ESTUDIANTIL	Contiene el registro de los estudiantes que conforman el gobierno estudiantil en cada año lectivo.
GRADUCACION_NOTAS	Esta tabla contiene las notas de grado obtenidas por un estudiante.
HORARIO	Contiene el horario de clases de cada curso en un año lectivo.
MAT_CURSO_DOCENTE1	Contiene el distributivo de las materias y docentes de la institución.
MATERIA	Contiene los datos de todas las materias.
MATRICULA	Contiene un registro de todas las matrículas de un año lectivo.
NOTAS_GRADO	Almacena las notas de los exámenes de grado de cada estudiante.
NOTAS PARCIALES	Almacena las notas parciales de cada trimestre
OBJ_ESPECIFICO	Contiene los objetivos específicos del sílabo de cada materia.
PARAMETROS_COLEGIO	Esta tabla tiene los datos informativos de la institución
PERIODO_ESCO	Almacena los años lectivos y su estado
PERSONAL	Contiene los datos de todo el personal docente, administrativo y de servicios de la institución.

PRIVILEGIOS_ROLES	Contiene los tipos de usuarios para el sistema.
SILABO	Contiene los datos del sílabo de cada materia.
USUARIOS	Contiene los datos de todos los usuarios del sistema.
VIGINGRESONOTAS	Esta tabla contiene las fechas en las cuales se puede ingresar notas trimestrales, de supletorios y de grado.

Tabla 5.1.- Descripción de las principales tablas del sistema

3.9.7. ARQUITECTURA DE LA APLICACIÓN

El objetivo principal de la arquitectura MVC es separar de la forma más limpia posible, las distintas capas de desarrollo, que permita facilidad de mantenimiento y evolución de las aplicaciones. Con el patrón MVC se puede cumplir con el objetivo, ya que permite una separación limpia entre las distintas capas de una aplicación; que según este patrón son Modelo, Vista y Controlador.

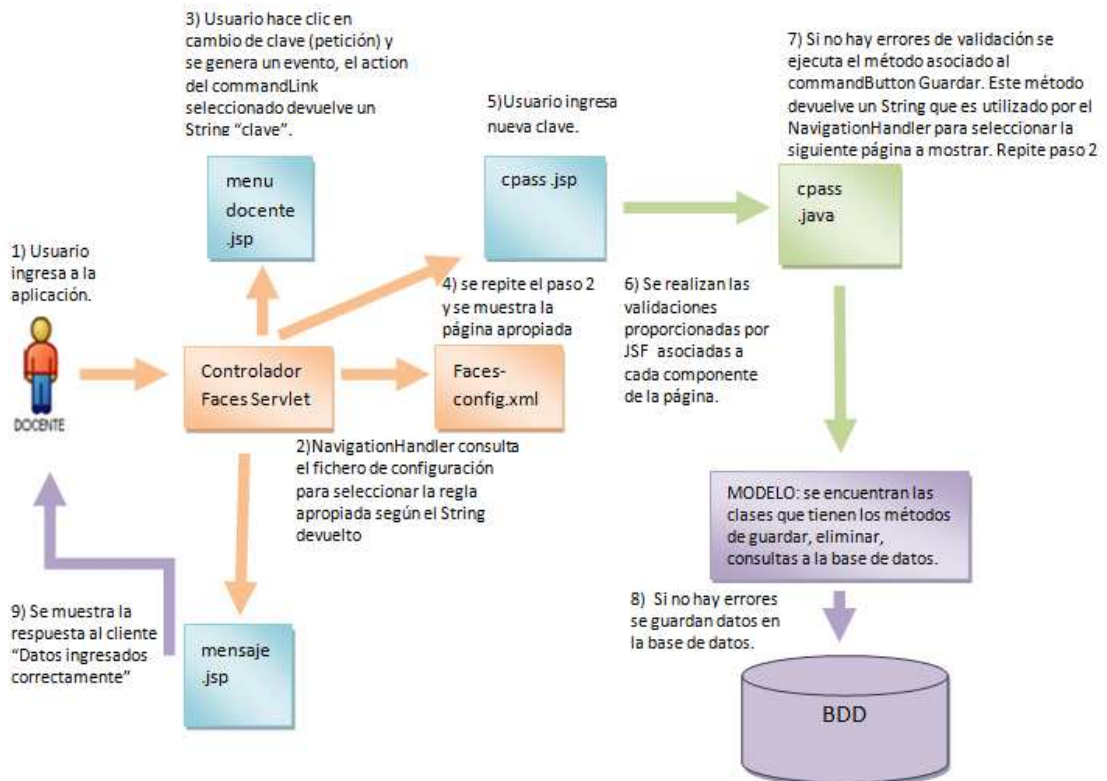


Figura 5. Arquitectura de una Aplicación con el Framework JSF

3.10. PRUEBAS Y PUESTA EN MARCHA

Se ingresó datos reales al sistema con la finalidad de probar el correcto funcionamiento del mismo. Se ingresaron datos de estudiantes, personal docente y administrativo, horario de clases, notas de los tres trimestres y supletorios de un curso, distributivo docente.

Luego de comprobar la integridad del sistema se procedió a instalar el software necesario para que la aplicación funcione en el Colegio Nacional Mixto “28 de Septiembre” ubicado en Priorato.

En el equipo que hace de servidor se encuentra instalado el siguiente software:

- Servidor de aplicaciones open source GlasFish.
- Base de Datos PostgreSQL.
- Máquina Virtual de Java.

Los equipos clientes solo necesitan un navegador web como Internet Explorer o Mozilla Firefox que es libre.

Los manuales técnico y de usuario se encuentran en la sección de anexos.

4. CONCLUSIONES Y RECOMENDACIONES

4.1. VERIFICACIÓN DE LA HIPÓTESIS

Hipótesis:

El Framework Java Server Faces proporciona una rica arquitectura para manejar el estado de los componentes, procesar los datos, validar la entrada del usuario, manejar eventos en una aplicación Web, separar la presentación del comportamiento de dicha aplicación lo que facilita su desarrollo y mantenimiento y a la vez permite obtener una aplicación económica y de calidad que será de gran utilidad a los colegios fiscales de la provincia.

- ✚ JSF proporciona una serie de validadores estándar para realizar validaciones a datos ingresados por el usuario y además permite crear nuestros propios validadores que serán muy útiles para casos generales como por Ej.: crear un validador para mails.
- ✚ Para JSF cada componente UI como botones, cajas de texto, tablas son objetos que podemos manejarlos desde clases java y no son simples etiquetas.
- ✚ Con JSF podemos manejar eventos individualmente de los componentes lo que nos permite tener varios eventos en un sola página lo que no se puede hacer con otros Frameworks como Struts.
- ✚ JSF nos permite obtener una aplicación que separa la lógica de programación y la presentación ya que este Framework se basa en el patrón MVC que separa una aplicación en Modelo, Vista y Controlador.
- ✚ Desarrollar aplicaciones con JSF resulta económico ya que podemos encontrar herramientas gratuitas que soportan este Framework.
- ✚ La aplicación desarrollada es de gran utilidad para los colegios fiscales ya que usando la aplicación tienen su información de una manera ordenada y que está disponible en cualquier momento, además de facilitar la tarea de cálculos de promedios que muchas veces como humanos podemos cometer errores al realizar estos cálculos.

4.2. CONCLUSIONES

- ✚ La evolución de Frameworks Web marca una tendencia en la abstracción del protocolo HTTP, para beneficiarse de los modelos basados en controles y componentes como en la programación de aplicaciones de escritorio.
- ✚ Todos los Frameworks se centran en resolver los mismos problemas: navegación, internacionalización, manejo de errores, validación y conversión de datos ingresados.
- ✚ JSF es mucho más flexible y extensible que otros Frameworks como Struts que se vio en un capítulo anterior.
- ✚ JSF se acerca más a la definición del patrón MVC; ya que a pesar de trabajar sobre el protocolo sin estado HTTP mantiene el estado de los componentes de UI en el servidor, lo que hace posible la creación de aplicaciones web orientadas a eventos.
- ✚ El tiempo de desarrollo de aplicaciones con JSF se reduce debido a los múltiples beneficios y ventajas que este Framework.
- ✚ Una ventaja importante es que JSF es el primer Framework J2EE con una especificación incluida en la última versión de J2EE y es el primer estándar del mercado; razón por la cual todos los servidores J2EE deben soportar JSF y los principales IDEs permiten crear aplicaciones con JSF.
- ✚ Al ser JavaServer Faces un estándar podemos encontrar implementaciones de JSF tanto libres como comerciales lo que nos permite escoger la que más nos convenga.
- ✚ Usar herramientas gratuitas para desarrollar aplicaciones no permite reducir costos en el desarrollo de aplicaciones.
- ✚ Trabajar bajo plataforma Java nos proporciona portabilidad en una aplicación ya que no dependemos de uno u otro sistema operativo.
- ✚ Manejar la navegación mediante reglas que se configuran en un archivo XML facilita el mantenimiento de una aplicación.

- ✚ Para una institución educativa tener un sistema que automatice sus procesos es de gran ayuda para brindar un mejor y eficaz servicio a sus estudiantes, además de evitar errores, y tener toda su valiosa información en una base de datos.
- ✚ Los usuarios después de recibir la capacitación reconocieron los beneficios que les traería utilizar el sistema, como obtener un reporte de promedios de cada curso sin necesidad de tener que realizar ningún cálculo.
- ✚ Antes de seleccionar a GlassFish como servidor de aplicaciones se probó otros servidores como OC4J de Oracle, JBoos, Apache Tomcat que necesitaban más recursos de hardware y configuraciones adicionales para desplegar la aplicación. GlassFish es libre, fácil de instalar y para desplegar la aplicación es muy sencillo, como se indica en el Anexo A.
- ✚ Exportar reportes a formato PDF en la fase de desarrollo no fue sencillo pero al momento de poner la aplicación en producción dio problemas ya que no se mostraban los reportes o no se generaban por lo que opté por exportar a formato HTML aunque no se muestren los gráficos a pesar de incluir las respectivas librerías.
- ✚ Es importante indicar que los requerimientos para el desarrollo del sistema se los tomó de un colegio fiscal al azar y en base a éstos se elaboró el sistema, mismo que podría ser instalado en otros colegios que cumplan con los mismos parámetros bajo los que fue desarrollado.

4.3. RECOMENDACIONES

- ✚ Antes de empezar a desarrollar una aplicación es recomendable hacer una comparación de los Frameworks más conocidos y según las características y ventajas que cada uno tenga y seleccionar el que más convenga.
- ✚ Una vez seleccionado el Framework para desarrollar una aplicación es muy importante investigar si hay suficiente información tanto en libros como en internet ya que esto nos ayudará mucho durante el desarrollo de la aplicación.
- ✚ Podemos seleccionar el IDE de nuestra preferencia ya que al ser JSF un estándar los IDEs más populares lo soportan.

- ✚ Para reducir costos en el desarrollo de una aplicación lo más recomendable es utilizar herramientas de software libre.
- ✚ Instalar el sistema antes de terminar de desarrollarlo permite que los usuarios puedan hacer sugerencias acerca de que funcionalidades se pueden incorporar al sistema. Además permite a los usuarios familiarizarse poco a poco con el sistema y se les facilite utilizarlo una vez que éste listo para ponerlo en funcionamiento.
- ✚ Realizar la capacitación a los usuarios del sistema tratando de no utilizar términos técnicos, hacerlo lo más comprensible para ellos y tomar en cuenta sus opiniones sobre el sistema. Explicándoles que el uso del sistema les facilitará su trabajo y que éste es necesario para la institución.

4.4. POSIBLES TEMAS DE TESIS

- ✚ Desarrollo de una aplicación Web utilizando JavaServer Faces y teléfonos celulares como dispositivos clientes.
- ✚ Evolución de Frameworks Web.
- ✚ Integración de JavaServer Faces con AJAX.

ANEXO A

MANUAL TÉCNICO

INSTALACIÓN DE LA BASE DE DATOS

PostgreSQL es un sistema de gestión de base de datos relacional y es de software libre y a continuación se muestra su instalación.

Ejecutamos el archivo de instalación postgresql-8.2

ACCESO A LA BASE DE DATOS



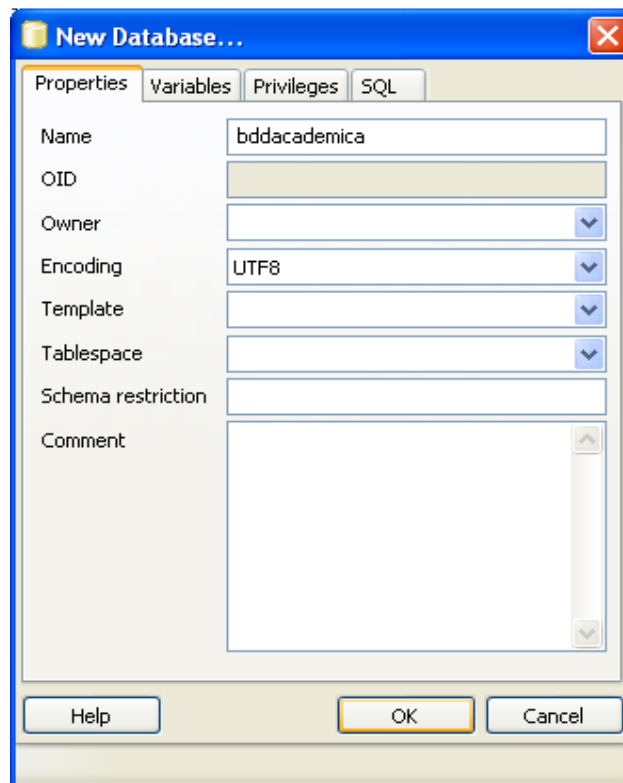
Luego de abrir PgAdmin III para la administración de la base de datos nos pide la contraseña que pusimos en la instalación de postgres



Y creamos una nueva base de datos



Ponemos un nombre a la base de datos en este caso bddacademica y encoding UTF8



Hacemos clic en OK para finalizar la creación de la base de datos.

Para crear las tablas de la base de datos podemos hacerlo mediante un script generado en este caso con Power Designer y hacemos clic en Execute Query.

```

[C:\Documents and Settings\Gaby\Mis documentos\DisenoBDD\script24.sql]
File Edit View Query Eavourites Help
bddacademica on localhost:5432
Output pane
Data Output Explain Messages History
/*=====*/
/* DBMS name: PostgreSQL 7.3 */
/* Created on: 12/01/2009 19:57:56 */
/*=====*/

/*=====*/
/* Table: CARGO */
/*=====*/
create table CARGO (
  COD_CARGO VARCHAR(4) not null,
  NOMBRE_CARGO VARCHAR(40) not null,
  constraint PK_CARGO primary key (COD_CARGO)
);

/*=====*/
/* Table: DEPARTAMENTO */
/*=====*/
create table DEPARTAMENTO (
  COD_DEPTO VARCHAR(2) not null,
  NOMBRE_DEPTO VARCHAR(30) not null,
  constraint PK_DEPARTAMENTO primary key (COD_DEPTO)
);

/*=====*/
/* Table: DIA */
/*=====*/
create table DIA (
  COD_DIA INT2 not null,
  NOM_DIA VARCHAR(15) not null,

```

INSTALACIÓN DE SERVIDOR DE APLICACIONES

Para Poner el aplicativo en producción se utilizó como servidor de aplicaciones Sun GlassFish Enterprise Server V2.1 y su instalación se detalla a continuación.

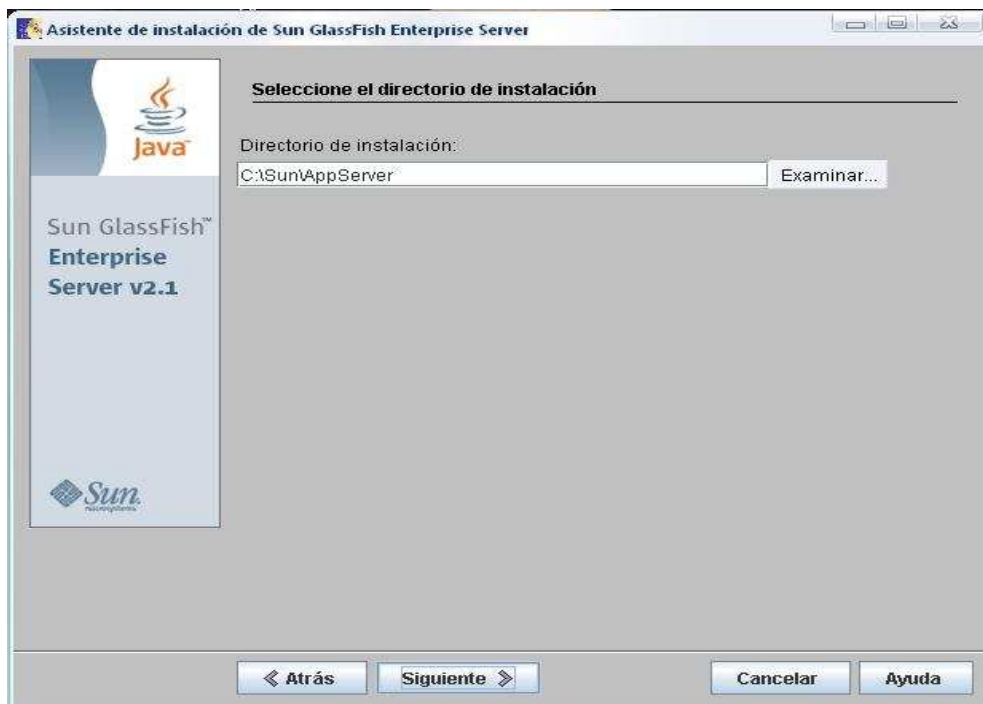
GlassFish es libre por lo cual podemos descargar el instalador de la página web y ejecutamos el archivo de instalación



Hacemos clic en siguiente y nos muestra el acuerdo de licencia para aceptarlo hacemos clic en Sí



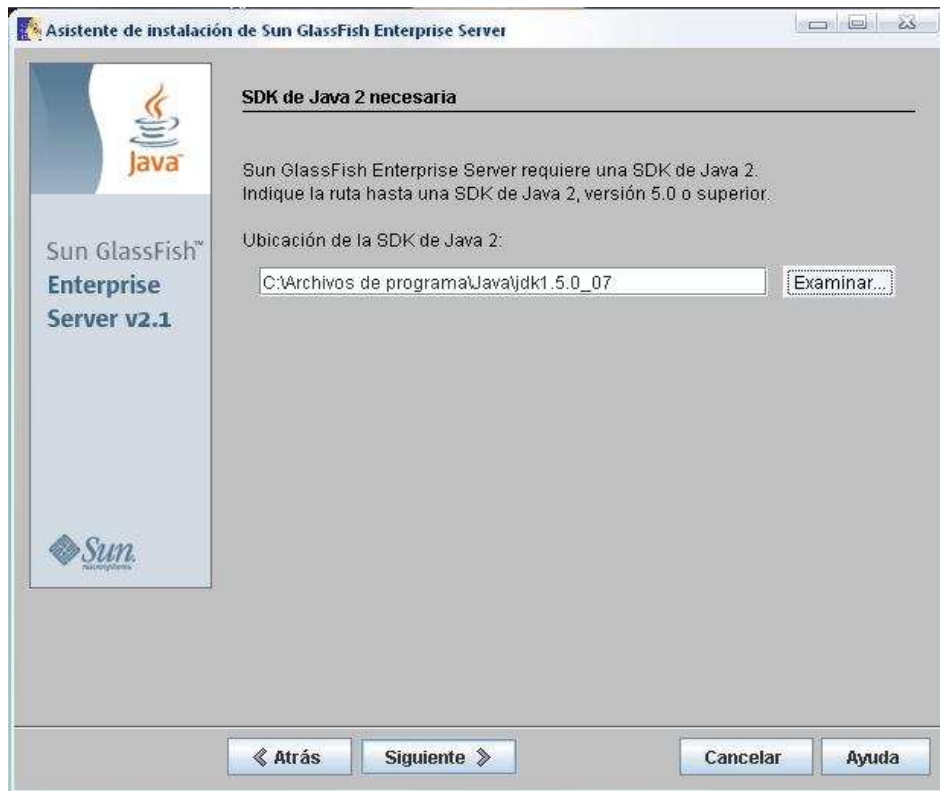
Hacemos clic en siguiente donde debemos seleccionar el directorio de instalación o dejar el directorio por defecto que es C:\Sun\AppServer



Hacemos clic en Siguiente y nos pregunta si deseamos crear el directorio de instalación o seleccionar uno distinto en este paso seleccionamos Crear Directorio



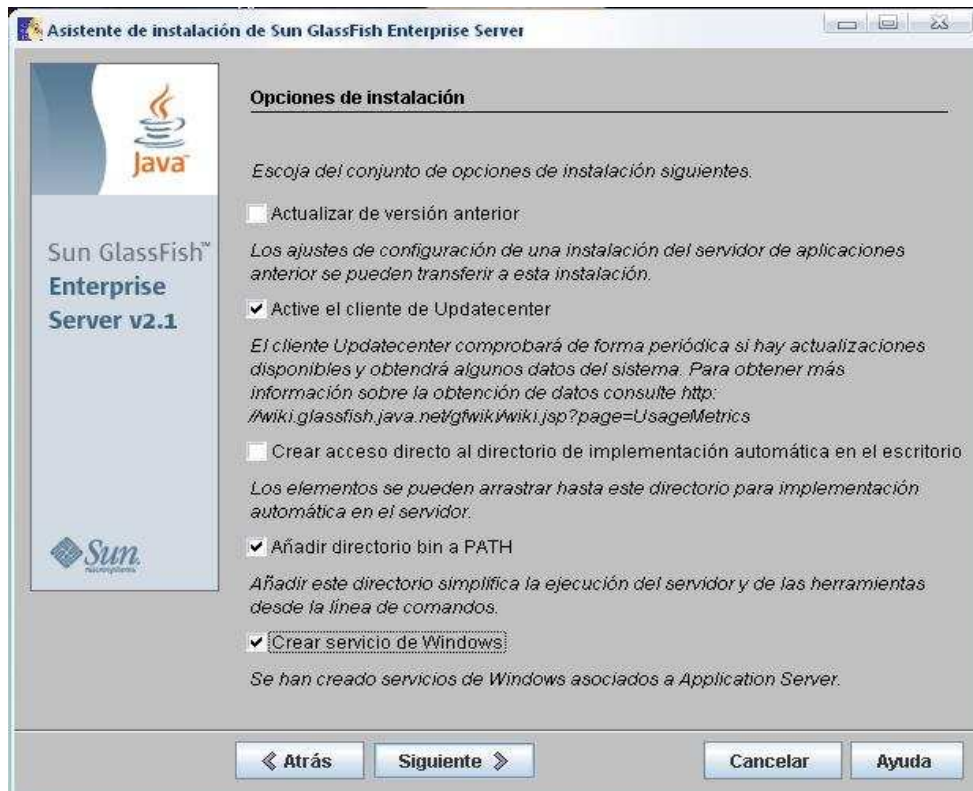
Luego nos pide la ruta del SDK versión 5 o superior



Hacemos clic en Siguiente donde nos pide configurar la administración con los siguientes datos nombre de usuario y contraseña para acceder a la consola de administración



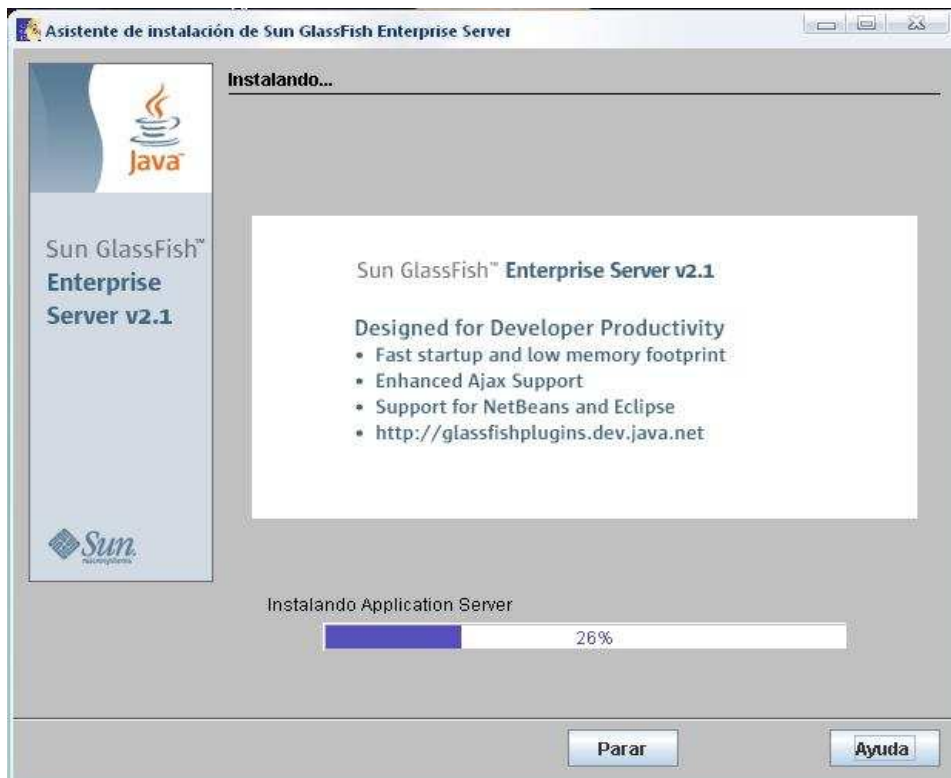
Hacemos clic en siguiente y nos pide seleccionar las opciones de instalación



Hasta aquí se ha terminado de configurar el servidor y esta listo para instalar



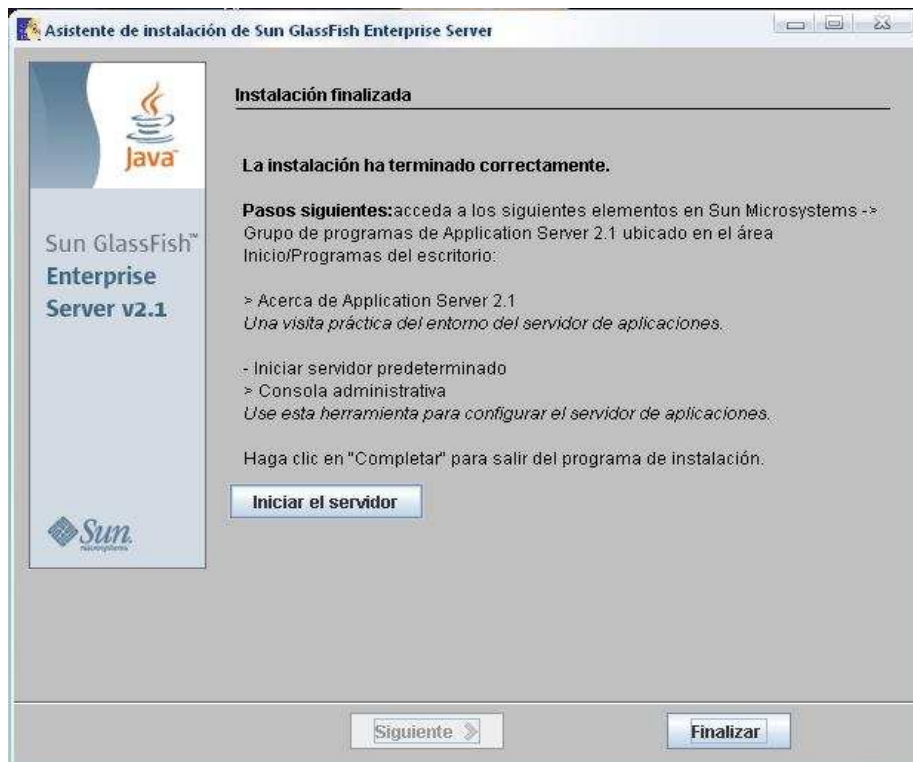
Para comenzar la instalación hacemos clic en Instalar Ahora



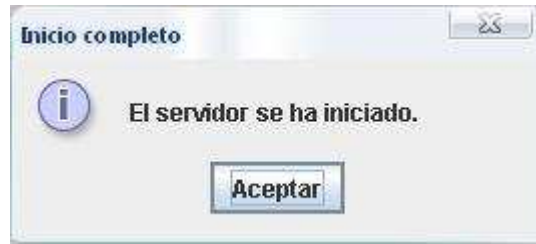
Al completar el 100% de instalación nos pide registrarnos u omitir este paso.



La instalación ha terminado y hacemos clic en iniciar el servidor



Y por último nos muestra un mensaje de que el servidor se ha iniciado.

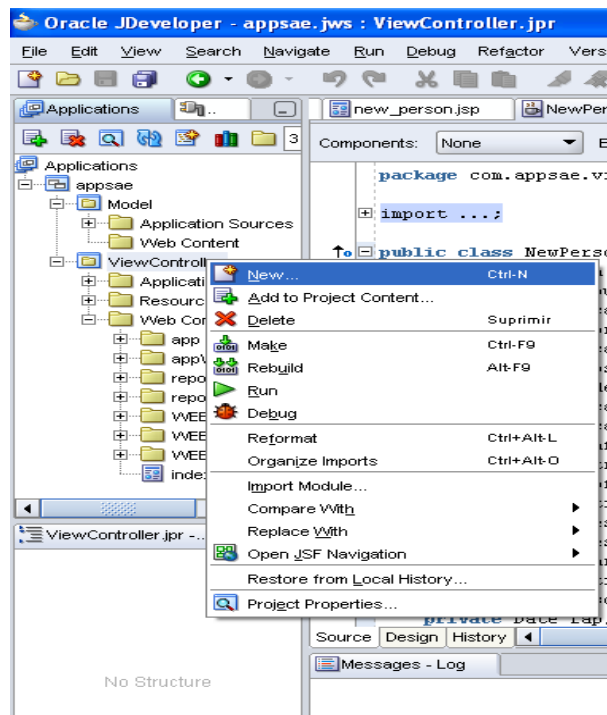


GENERACIÓN DEL ARCHIVO .WAR DE LA APLICACIÓN

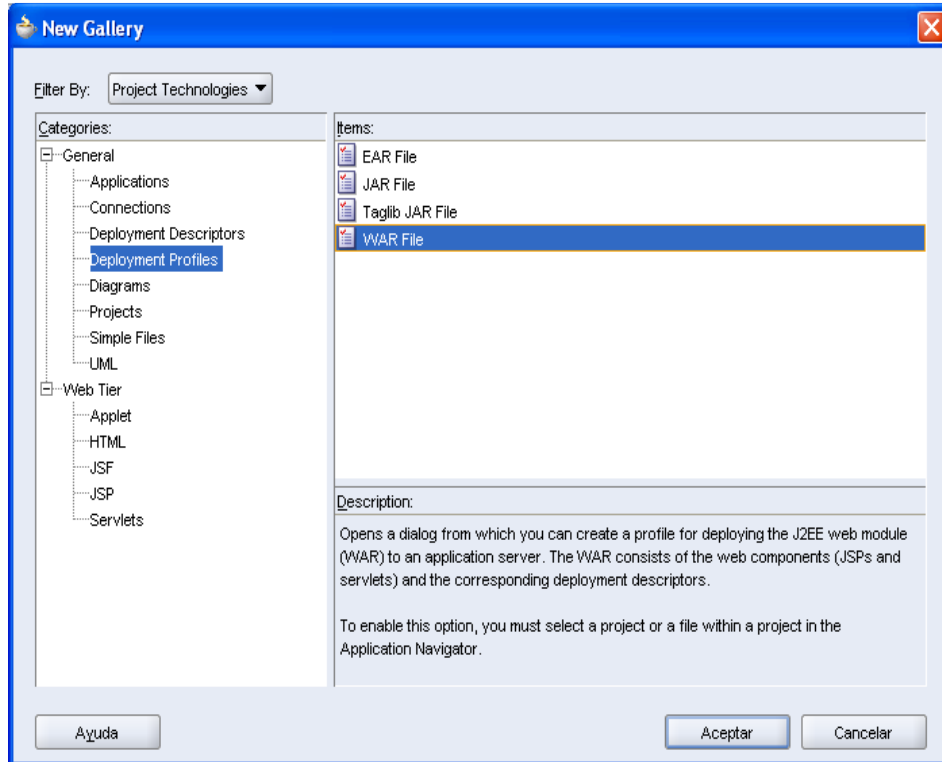
Oracle JDeveloper 10.1.3.3 es Entorno de Desarrollo utilizado para el desarrollo de la aplicación es gratuito y lo podemos descargar de Internet.

Para generar el archivo appsaewar que luego sera desplegado en el Servidor de Aplicaciones abrimos JDeveloper y realizamos los siguientes pasos.

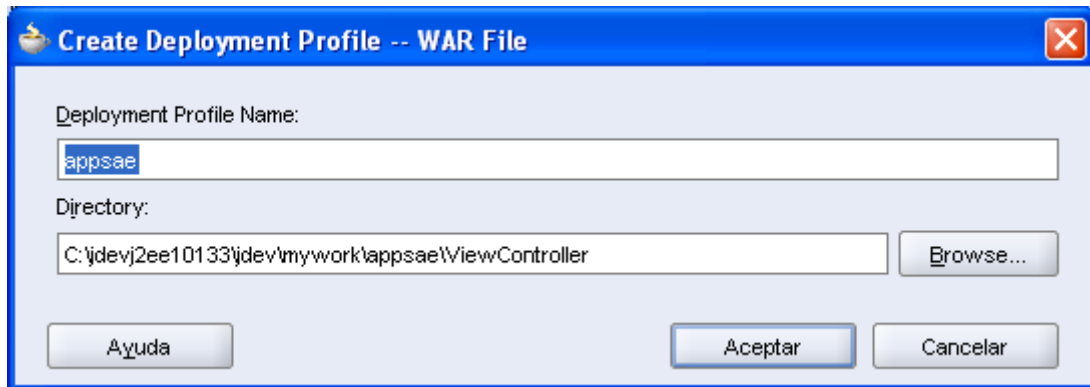
Dentro del proyecto seleccionamos ViewController y haciendo clic derecho sobre este seleccionamos Nuevo



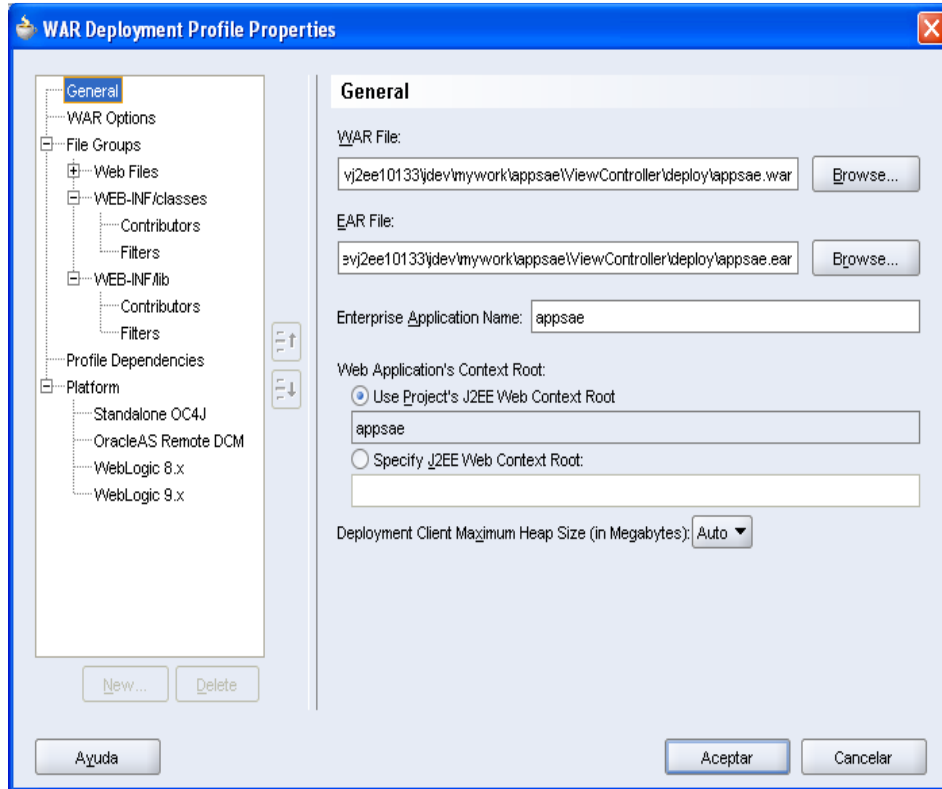
Desplegamos la pestaña general y seleccionamos Deployment Profiles y dentro de esta categoría tenemos varias opciones de las cuales seleccionamos WAR File y hacemos clic en Aceptar



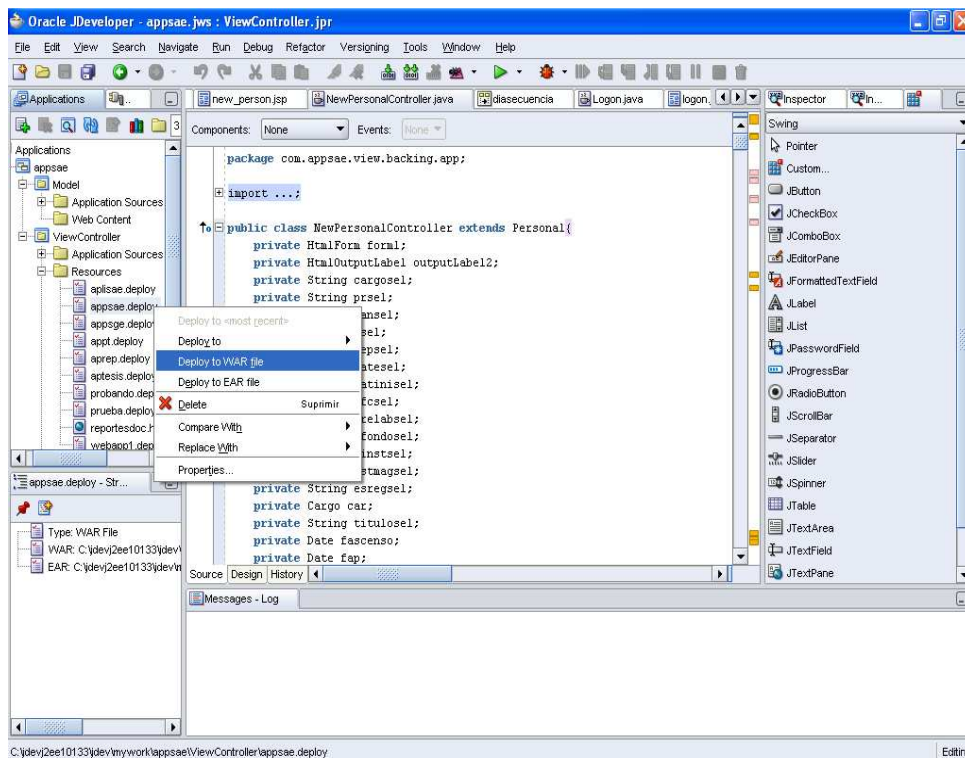
Damos un nombre al archivo a generar en este caso appsae y hacemos clic en aceptar



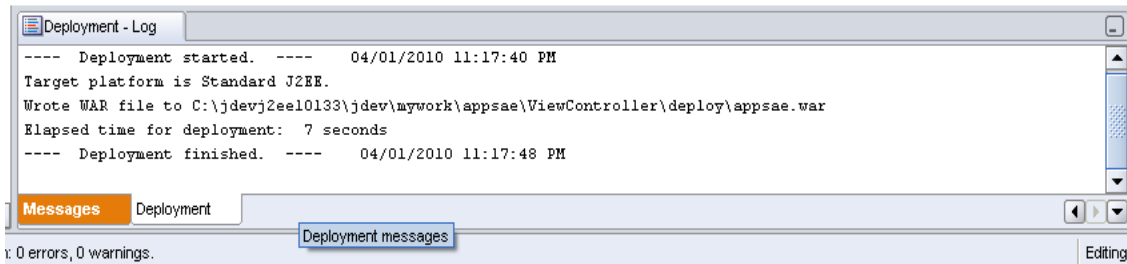
Nos muestra las propiedades del archivo .war que se va a generar como en que directorio se va a localizar este archivo, no es necesario cambiar las configuraciones por defecto; y hacemos clic en aceptar



Dentro de ViewController se crea un directorio Resources donde van a estar todos los .deploy que se han creado. Para terminar con la generación del archivo .war hacemos clic derecho en appsae.deploy y seleccionamos Deploy to WAR File



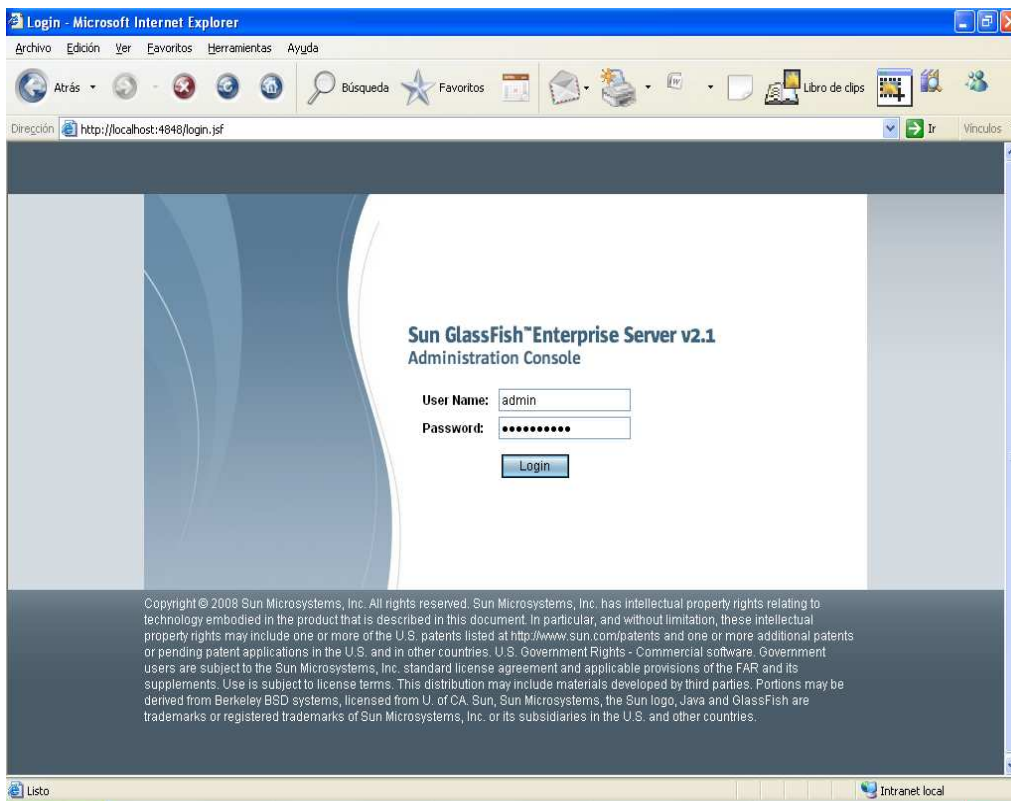
En la ventana de compilación nos muestra si hubo errores o no y en que directorio se generó el archivo .war de la aplicación



SUBIR LA APLICACIÓN AL SERVIDOR DE APLICACIONES

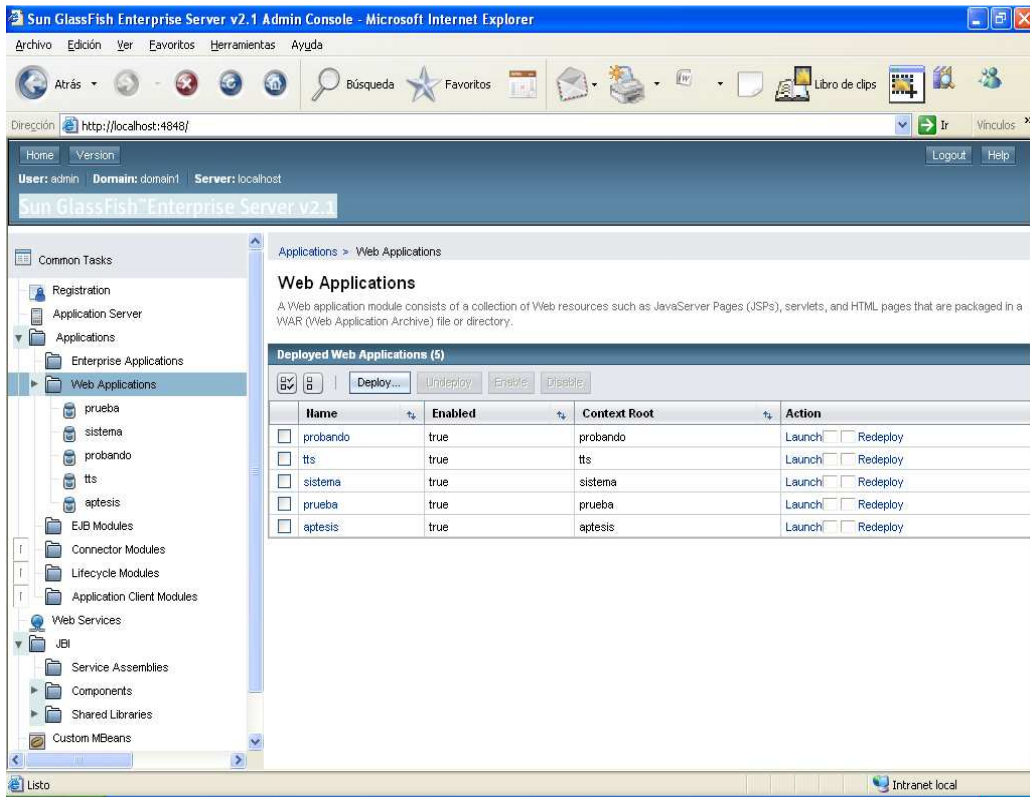
Para poner en producción la aplicación se utiliza el archivo .war generado en el paso anterior y siguiendo estos pasos.

Para abrir la consola de administración de GlassFish Server abrimos un navegador de Internet y digitamos la dirección <http://localhost:4848/>

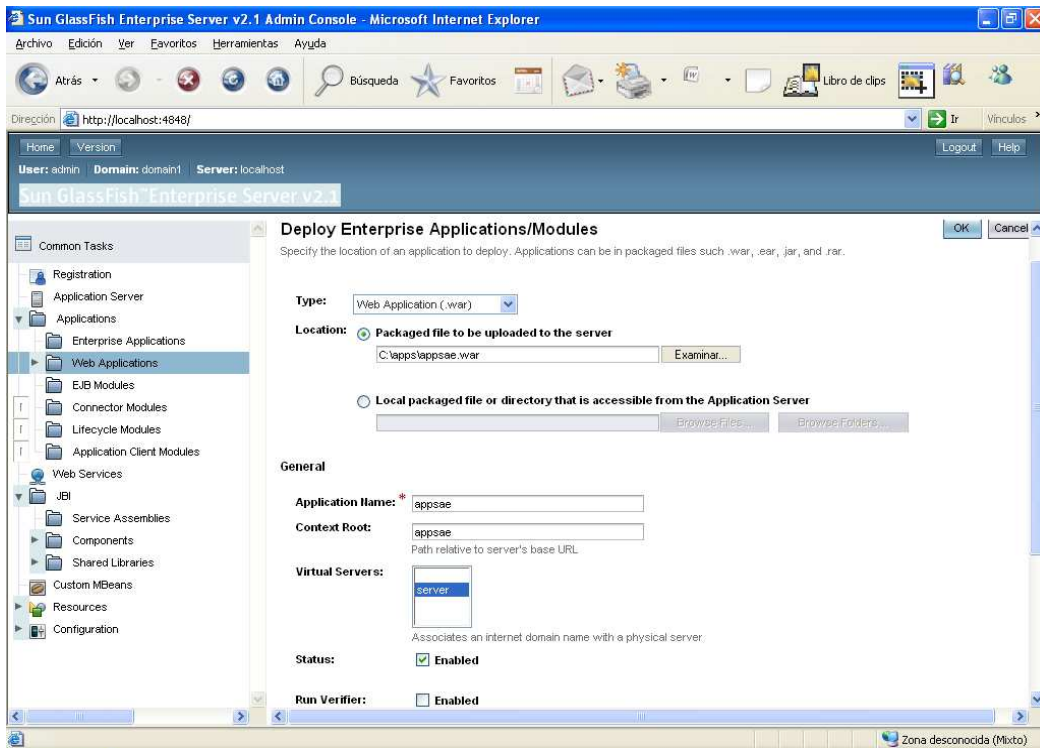


Digitamos nombre de usuario y contraseña para ingresar a la administración del servidor de aplicaciones.

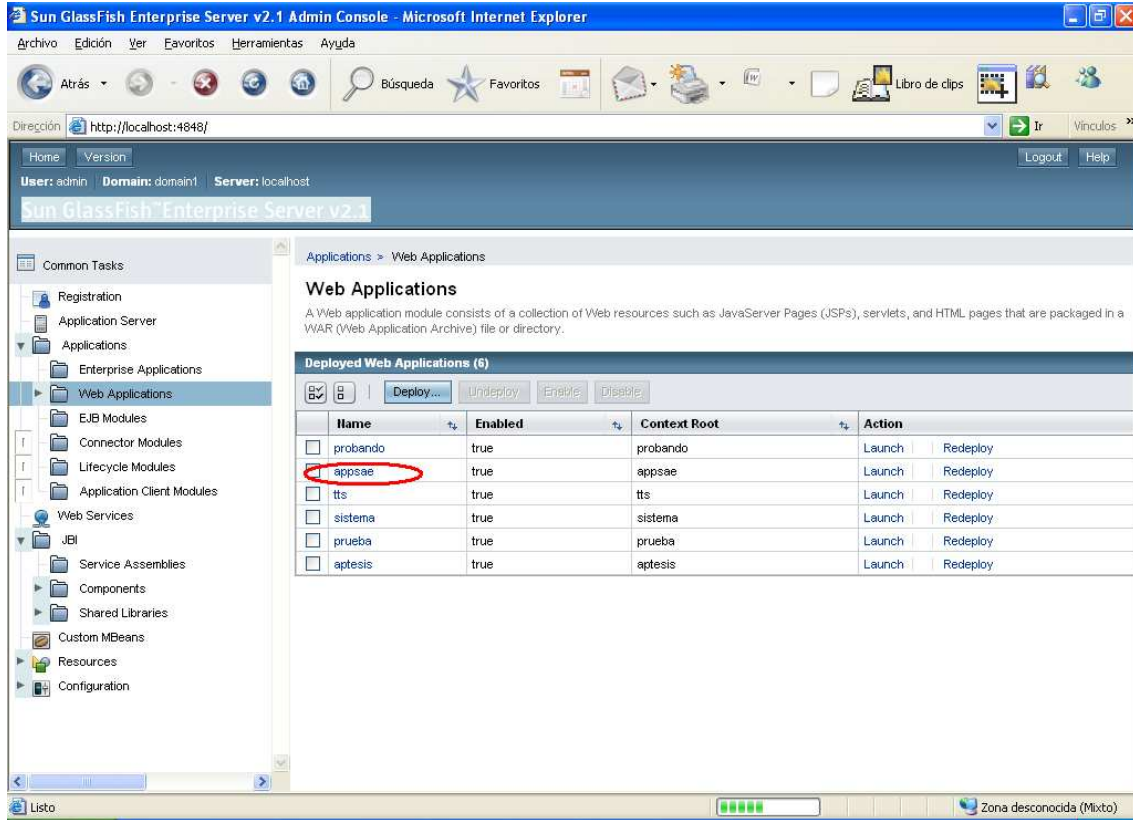
Para desplegar la aplicación seleccionamos en la parte izquierda de la consola Web Applications



Hacemos clic en Deploy



Haciendo clic en examinar damos la ubicación del archivo .war de la aplicación, seleccionamos el servidor en este caso server, seleccionamos Status y Precompile JSPs, y en libraries ponemos todas la librerías que utiliza la aplicación y hacemos clic en OK.



Y vemos que la aplicación desplegada ahora aparece en la lista de aplicaciones que tiene disponibles el servidor de aplicaciones.

Para abrir la aplicación desplegada hacemos clic en Launch o digitando en la barra de dirección del explorador <http://localhost:8080/appsae>.

El puerto 4848 es puerto de administración y el 8080 es utilizado como puerto HTTP para las aplicaciones que se despliegan en el servidor.

ANEXO B

MANUAL DE USUARIO

SISTEMA DE ADMINISTRACION ESTUDIANTIL

INDICACIONES GENERALES

El sistema de administración estudiantil permite la automatización de las tareas de un colegio lo que trae beneficios al personal docente y administrativo del plantel.

TIPOS DE USUARIOS


El sistema cuenta con tres tipos de usuarios y cada uno puede realizar tareas específicas según su tipo.

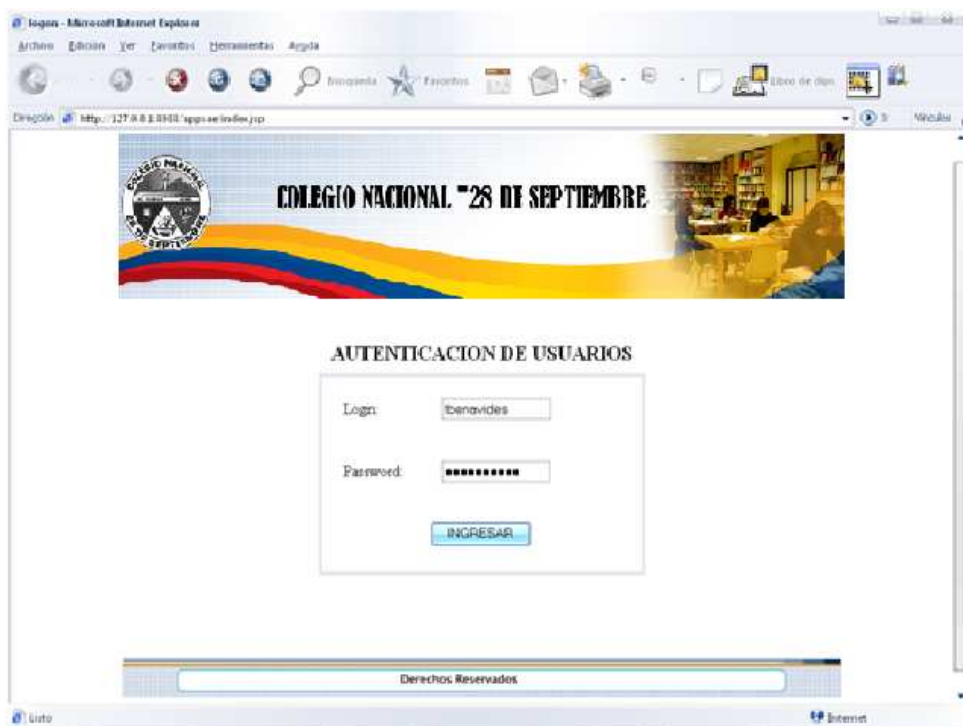
- a) **TIPO ADMINISTRADOR:** este tipo de usuario puede realizar las siguientes tareas: crear y eliminar usuarios del sistema, crear y cerrar un año lectivo, crear un nuevo curso, crear una nueva especialidad, crear nuevas materias, crear el horario de trabajo.
- b) **TIPO SECRETARIA:** este tipo de usuario como su nombre lo indica será asignado a la secretaria del plantel y podrá realizar las siguientes tareas: cambiar su clave de ingreso al sistema, ingresar el distributivo docente, ingresar datos de un nuevo estudiante, imprimir: horarios, nóminas, promedios de curso, listas de estudiantes que se quedan a supletorios, listas de estudiantes que pierden el año, realizar matrículas, imprimir certificados de matrícula entre otras tareas.
- c) **TIPO DOCENTE:** este tipo de usuario corresponde a todos los docentes del plantel, y podrá realizar las siguientes tareas: cambiar su clave de ingreso al sistema, imprimir su horario, ingresar notas parciales, ingresar notas de supletorios, imprimir reporte de notas, crear un sílabo de las diferentes materias que dicta y registrar el avance de las mismas en cada curso.

- d) **TIPO INSPECTOR:** este tipo de usuario corresponde al o los inspectores del plantel; y puede realizar las siguientes tareas: cambiar su clave de ingreso al sistema, calificar la conducta y registrar las faltas de los estudiantes de los cursos que tiene a su cargo, imprimir en reporte de las conductas obtenidas por los estudiantes, obtener un reporte de faltas en un periodo de tiempo determinado.

INGRESO AL SISTEMA

Para ingresar al sistema debe seguir estos pasos:

- a) Ingresar a Internet Explorer haciendo doble clic en el ícono  u otro navegador Web como Mozilla Firefox.
- b) En la barra de direcciones escribir <http://localhost:8080/appsae> y se desplegará la siguiente pantalla.



- c) A continuación ingresar su nombre de usuario y contraseña.
- d) Hacer clic en el botón INGRESAR.
- e) A continuación se mostrará un menú de acuerdo al tipo de usuario.

MANUAL PARA EL TIPO DE USUARIO ADMINISTRADOR

Si usted es el administrador después de ingresar su nombre de usuario y contraseña el sistema le mostrará el siguiente menú.



DESCRIPCIÓN DE LAS OPCIONES DEL MENÚ DE ADMINISTRADOR

- Nuevo Usuario.-** esta opción le permite al administrador crear un nuevo usuario para el sistema. Las personas que pueden tener un usuario para ingresar al sistema son: personal docente, inspectores y la secretaria del establecimiento. Los nombres de usuario serán la inicial del primer nombre seguido del primer apellido en letras minúsculas por Ej.: tbenavides, este nombre debe ser único no se puede repetir.



A continuación se explica detalladamente la información que se debe ingresar para crear un nuevo usuario.

CAMPO	DETALLE
Login	Es el nombre con que el usuario ingresara al sistema. Este nombre debe ser único y debe tener una longitud de 3 a 15 caracteres Ej.: tcarrera
Password	Es la contraseña que con la que cada usuario podrá ingresar al sistema. Pueden ser letras o números mínimo 3 y máximo 15.
Cédula	Aquí debe ingresar el número de cédula del nuevo usuario. Debe tener 10 números por Ej.: 1001152521
Tipo de Usuario	De la lista debe seleccionar el tipo de usuario según el cargo que desempeñe, puede ser docente o inspector.

Errores que se pueden presentar y sus soluciones:

ERROR	SOLUCIÓN
Login ya existe	Cambiar el nombre de usuario o login.
Ya tiene una cuenta de Usuario	Cada persona solo puede tener un usuario para ingresar al sistema excepto las personan que son docentes y a la vez inspector los cuales podrán tener dos usuarios diferentes.
Cédula incorrecta	Ingresar correctamente su número de cédula.
Tipo de Usuario	De la lista debe seleccionar el tipo de usuario según el cargo que desempeñe, puede ser docente o inspector.

- Eliminar Usuario.-** esta opción permite eliminar un usuario del sistema simplemente haciendo clic sobre la palabra Eliminar que se encuentra a la derecha del usuario que va a ser eliminado como se indica en la siguiente figura.



[Menú](#)

ELIMINAR CUENTAS DE USUARIO



Cuentas de Usuario

Usuario	Privilegio	Eliminar
tbenavides	SECRETARIA	Eliminar
vbenavides	ADMINISTRADOR	Eliminar
nrevelo	DOCENTE	Eliminar
cvarela	DOCENTE	Eliminar
eburbano	DOCENTE	Eliminar
jpuga	DOCENTE	Eliminar

- **Crear Periodo.-** Esta opción permite crear un nuevo año lectivo, esto solo se debe hacer al terminar el año lectivo anterior al que se va a crear o antes de empezar las matrículas del nuevo año lectivo, en la siguiente figura se muestra un ejemplo:



A continuación se explica detalladamente la información que se debe ingresar para crear un nuevo año lectivo:

CAMPO	DETALLE
Año Lectivo	Se debe ingresar el año lectivo en el formato año-año por Ej.: 2009-2010.
Fecha Inicio	Ingresar la fecha en que inicia el año lectivo en el formato día/mes/año. Este dato es obligatorio.
Fecha Finalización	Ingresar la fecha en que finaliza el año lectivo en el formato día/mes/año. Este dato no es obligatorio.

- **Cerrar Periodo.-** esta opción permite dejar inactivo un año lectivo, esto se debe hacer al finalizar cada año lectivo, cuando ya se hayan ingresado notas de supletorios y se hayan obtenido las promociones de los estudiantes que han aprobado el año lectivo. Se debe ingresar la fecha que termina el año lectivo en formato día/mes/año y hacer clic en el botón Desactivar Periodo, como se indica en la siguiente figura.



[Menú](#)

CERRAR AÑO LECTIVO

Año Lectivo:	2008-2009
Fecha Cierre:	30/06/2009

Cerrar Periodo

Derechos Reservados

- **Asignar Materias.-** esta opción permite asignar a los docentes las materias que estarán a su cargo; esto facilita el ingreso del distributivo docente.

ASIGNAR MATERIAS A DOCENTES

Docente:	TCNLG. ARTEAGA OMAR REMIGIO
Materias:	COMPUTACION

Agregar

Docente	Materia
DR. MAIRENE ARCINIEGA	CIENCIAS NATURALES
DR. BERTHA ESPERANZA NUÑEZ	LEGISLACION LABORAL
DR. BERTHA ESPERANZA NUÑEZ	SOCIO ECONOMIA
TCNLG. OMAR REMIGIO ARTEAGA	OPTATIVA
TCNLG. OMAR REMIGIO ARTEAGA	COMPUTACION

- **Especialidad.-** esta opción le permite al administrador crear una nueva especialidad, en caso de que la institución cree una especialidad, como se indica en la siguiente figura.



[Menú](#)

CREAR NUEVA ESPECIALIDAD

Código Especialidad:	11
Sección:	Diurna
Descripción:	FISICO MATEMATICO
Título que Otorga:	BACHILLER EN CIENCIAS
Especialización:	FISICO MATEMATICO
Código DEC:	10845
Estado Registro:	INSERCIÓN NUEVO

Guardar

Derechos Reservados

A continuación se explica detalladamente los campos que debe ingresar el administrador.

CAMPO	DESCRIPCIÓN	OBLIGATORIO
Código especialidad	Este código es numérico y está establecido por la dirección de educación.	Si
Sección	En caso de que el plantel tenga varias secciones escoger una.	Si
Descripción	En este campo debe ingresar la especialización por Ej.: Sociales	Si
Título que otorga	Se ingresa el título que otorga la especialidad por Ej.: Bachiller en Ciencias	Si
Código DEC	Es el código de la resolución de crear una nueva especialidad. Es de tipo numérico.	Si
Estado de Registro	Seleccionar Nuevo	Si

- **Curso.-** esta opción permite crear un nuevo curso por ejemplo un 8vo año de educación básica paralelo B. para crear el curso debe ingresar algunos datos como se indica en la siguiente figura.



[Menú](#)

CREAR NUEVO CURSO

Código Curso:	BASCD0802
Especialidad:	NINGUNO
Sección:	Diurna
Nº Curso:	1
Paralelo:	B
Año de Grado:	<input type="radio"/> Si <input checked="" type="radio"/> No
Nivel de Educación:	EDUCACION BASICA
Fecha de Creación:	01/09/2000
Código Apertura:	MEMO 01-R
<input type="button" value="Guardar"/>	

A continuación se explica detalladamente los campos que se deben completar para crear un nuevo curso.

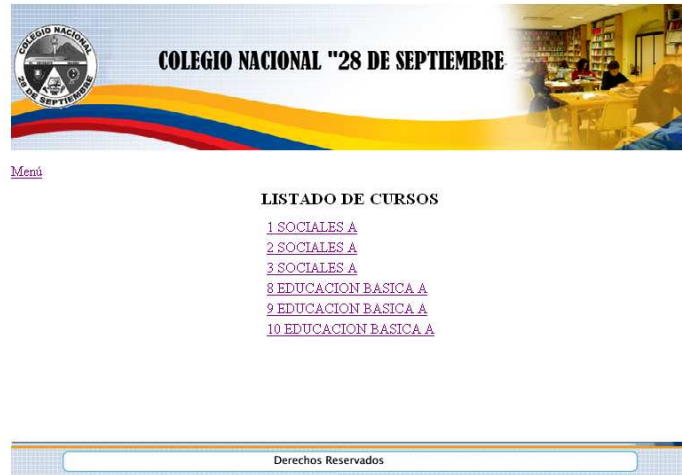
CAMPO	DESCRIPCION	OBLIGATORIO
Código curso	Este código se lo asigna de acuerdo a un formato establecido por la dirección de Educación y tiene la siguiente forma: EEEECCPP E = especialidad Ej.: FFMM S = sección, por Ej.: si es diurna es D C = es número de curso 8,9,10 de educación básica y 1,2,3 para años de bachillerato. P = paralelo por Ej.: paralelo A=01, B=02 y así sucesivamente según la letra que corresponde. Formando el código con los ejemplos dado queda: FFMMD0201 que será segundo año de bachillerato paralelo A.	Si
Especialidad	Seleccionar una especialidad de la lista, en caso de ser un curso de educación básica escoger especialidad Ninguna.	Si
Sección	En caso de que el colegio cuente con más de una sección escoger una de la lista.	Si
Número de Curso	Escoger un número de la lista para educación básica son 8,9 y 10 y para cursos de bachillerato son 1,2 y 3 que deben seleccionar según el curso que desee crear.	Si
Paralelo	Seleccionar un paralelo de lista son caracteres del alfabeto que están entre la A y Z	Si
Año de Grado	Seleccionar Si en caso de que sea un curso del tercer año de bachillerato, es decir si en este año se entrega un título, caso contrario seleccionar no.	Si
Nivel de educación	Seleccionar Educación básica o Bachillerato según corresponda al curso a crear.	Si
Fecha de Creación	Ingresar la fecha en que fue creado el curso en el formato día/mes/año.	Si
Fecha de cierre	En caso de que este paralelo sea eliminado ingresar la fecha de eliminación en formato día/mes/año.	No
Código apertura	Es el código de la resolución con la que se crea el curso.	No
Registro	Seleccionar Nuevo	Si

- Materia.-** Esta opción permite crear una nueva materia, en caso de ser necesario, para lo cual debe completar cierta información como se indica en la siguiente figura.

A continuación se explica detalladamente la información que se debe completar.

CAMPO	DESCRIPCION	OBLIGATORIO
Código Materia	Este código está establecido por la dirección de educación. Es de tipo numérico y se debe consultar el archivo Tablas donde se encuentran todos los códigos y nombres de materias.	Si
Nombre	Es el nombre de la materia. Ej.: EDUCACION PARA LA DEMOCRACIA.	Si
Fecha de Creación	Fecha en que fue creada la materia, en formato día/mes/año	Si
Resolución	Es el código de la resolución de crear la materia.	No
Fecha de Resolución	Fecha en que se resuelve crear la materia en formato día/mes/año.	No
Registro	Seleccionar Nuevo	Si

- **Crear Horario.-** esta opción permite ingresar el horario de trabajo para un año lectivo. El sistema muestra una lista de todos los cursos que hay en el plantel, entonces se debe seleccionar el curso del que se va a ingresar el horario como se muestra en la siguiente figura.



Una vez que se ha seleccionado el curso aparece la siguiente pantalla que permite ingresar el horario siguiendo estos pasos:

- Escoger un día de la lista por Ej.: Lunes
- De la lista de materias disponibles para ese curso seleccionar en orden según la hora empezando por la primera y hacer clic en el botón agregar, entonces la materia pasará a otra lista con la hora asignada como se indica en la siguiente figura.

The screenshot shows the 'INGRESO DE HORARIOS' form. At the top, there are links for 'Atrás' and 'Menú'. The form has two input fields: 'Curso:' with the value '2 SOCIALES A' and 'Seleccione Día:' with a dropdown menu showing 'Lunes'. Below these fields is a table with two columns: 'Materias Disponibles' and 'Hora Materia'. The 'Materias Disponibles' column lists various subjects, with 'COMPUTACION' highlighted. An 'Agregar' button is located between the two columns. The 'Hora Materia' column shows '1' in the 'Hora' row and 'COMPUTACION' in the 'Materia' row. At the bottom of the form, there is a 'Guardar' button.

- c) Se debe repetir el procedimiento el paso b hasta completar las horas del día en este caso 7 como se indica en la siguiente figura.

[Atrás](#)

[Menú](#)

INGRESO DE HORARIOS

Curso:	2 SOCIALES A
Seleccione Dia:	Lunes ▾

Materias Disponibles			Horario	
LEGISLACION LABORAL	<input type="button" value="Agregar"/>	1	FILOSOFIA	
ADMINISTRACION		2	FILOSOFIA	
COMPUTACION		1	COMPUTACION	
FILOSOFIA		2	COMPUTACION	
INVESTIGACION		3	MATEMATICA	
PSICOLOGIA		4	MATEMATICA	
LITERATURA		5	ESTUDIOS SOCIALES	
CULTURA FISICA		6	ESTUDIOS SOCIALES	
INGLES		7	FILOSOFIA	
MATEMATICA				
ESTUDIOS SOCIALES				
SOCIO ECONOMIA				
ED. AMOR Y SEXUALIDAD				
FUND. DE CONTABILIDAD				

- d) Una vez terminado de asignar las materias debe hacer clic en el botón Guardar.
- e) Se debe seguir el mismo procedimiento para el resto de días, y cursos; para seleccionar otro curso hacer clic en Atrás y si desea volver al menú principal hacer clic en Menú.
- **Fechas ingreso notas.-** con esta opción el administrador del sistema puede parametrizar el sistema para que el docente solo pueda ingresar notas en un rango de fechas establecidas para el efecto, el usuario debe seleccionar el tipo de nota e ingresar las fechas el sistema permitirá el ingreso de estas notas.



[Menú](#)

FECHAS DE VIGENCIA PARA INGRESO DE NOTAS

Tipo de Nota:	Notas Supletorios ▾
Trimestre:	Selecciones ▾
Fecha Inicio:	12/06/2010
Fecha Finalización:	20/06/2010

Guardar

Derechos Reservados

Si el docente hace clic por ejemplo en ingreso de notas de supletorio y las fechas de ingreso no están vigentes el sistema mostrará un mensaje indicando que el sistema no está habilitado para el ingreso de notas como se indica en la siguiente figura.

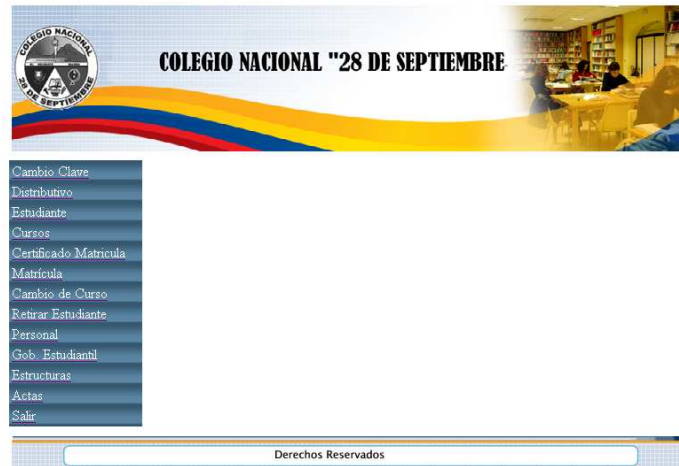
Sistema no está habilitado para ingreso de notas...

[Regresar...](#)

- **Salir.-** esta opción le permite salir del sistema. Al hacer clic en esta opción regresará a la pantalla principal donde le pide nombre de usuario y contraseña; en esta pantalla usted puede hacer clic en el botón cerrar del Internet Explorer. Como se indica en la siguiente figura.

MANUAL PARA EL TIPO DE USUARIO SECRETARIA

Después de que usted ha ingresado su nombre de usuario y contraseña se mostrará el siguiente menú.



DESCRIPCIÓN DE LAS OPCIONES DEL MENU

- **Cambio Clave.-** haciendo clic en esta opción el usuario puede cambiar la contraseña de ingreso al sistema. El usuario debe ingresar la nueva clave y en el casillero de confirmación de clave volverla a escribir y hacer clic en el botón guardar, si no hay ningún error la clave será cambiada y regresará al menú principal.

En caso de que la nueva clave y la confirmación de clave no sean iguales el sistema dará el siguiente error: Falló confirmación de contraseña.

Menú

COLEGIO NACIONAL "28 DE SEPTIEMBRE"

CAMBIO DE CLAVE DE USUARIO

Usuario:

Clave Actual:

Nueva Clave:

Confirme Clave:

• Falló confirmación de contraseña, no son iguales...

Derechos Reservados

Este error se soluciona ingresando correctamente la clave.

- **Distributivo.-** esta opción se refiere al distributivo de materias y docentes que se realiza al inicio del año lectivo; es decir permite asignar las materias que dicta cada docente en los diferentes cursos.

Se debe seleccionar un curso de la lista, una materia y se mostrarán solo los docentes que dicten la materia seleccionada, e ingresar el número de horas semanales que se dictará dicha materia y continuación hacer clic en el botón Guardar.

Menú

DISTRIBUTIVO DOCENTE

Año Lectivo: 2008-2009

Curso	Materia	Docente	Nº Horas
1 SOCIALES A	COMPUTACION	TCNLG. OMAR REMIGIO ARTEAGA	2

Aquí también tenemos la opción del botón Imprimir que nos permite obtener un impreso del distributivo con la siguiente información: el nombre del docente, el curso, la materia y el número de horas.

[Menú](#)

COLEGIO NACIONAL MIXTO 28 DE SEPTIEMBRE
DISTRIBUTIVO DOCENTE PERIODO 2008-2009

Docente	Curso	Materia	Horas
DR. MAIRENE ARCINIEGA MONTALVO	3 SOCIALES A	ED. AMOR Y SEXUALIDAD	2
DR. MAIRENE ARCINIEGA MONTALVO	2 SOCIALES A	ED. AMOR Y SEXUALIDAD	2
TCNLG. OMAR REMIGIO ARTEAGA ROSERO	3 SOCIALES A	COMPUTACION	2
TCNLG. OMAR REMIGIO ARTEAGA ROSERO	2 SOCIALES A	COMPUTACION	2
TCNLG. OMAR REMIGIO ARTEAGA ROSERO	1 SOCIALES A	MATEMATICA	2
TCNLG. OMAR REMIGIO ARTEAGA ROSERO	1 SOCIALES A	COMPUTACION	2
LIC. AURA ELISA BENAVIDES IMBAQUINGO	1 SOCIALES A	FILOSOFIA	4
LIC. AURA ELISA BENAVIDES IMBAQUINGO	2 SOCIALES A	FILOSOFIA	3
LIC. AURA ELISA BENAVIDES IMBAQUINGO	3 SOCIALES A	FILOSOFIA	3
LIC. AURA ELISA BENAVIDES IMBAQUINGO	3 SOCIALES A	ESTUDIOS SOCIALES	8
LIC. AURA ELISA BENAVIDES IMBAQUINGO	1 SOCIALES A	LOGICA Y ETICA	2
LIC. EDUARDO VICENTE BURBANO PORTILLA	3 SOCIALES A	PSICOLOGIA	2
LIC. EDUARDO VICENTE BURBANO PORTILLA	2 SOCIALES A	INVESTIGACION	2
LIC. EDUARDO VICENTE BURBANO PORTILLA	1 SOCIALES A	INVESTIGACION	2
LIC. EDUARDO VICENTE BURBANO PORTILLA	1 SOCIALES A	EDUACION DEMOCRACIA	1
LIC. EDUARDO VICENTE BURBANO PORTILLA	2 SOCIALES A	PSICOLOGIA	2
LIC. JAIME DAMIAN CALDERON LOPEZ	1 SOCIALES A	ADMINISTRACION	2
LIC. JAIME DAMIAN CALDERON LOPEZ	3 SOCIALES A	ADMINISTRACION	2
LIC. JAIME DAMIAN CALDERON LOPEZ	2 SOCIALES A	ADMINISTRACION	2
LIC. TERESA ELIZABETH CARRERA MARQUEZ	1 SOCIALES A	INGLES	5

ⓧ Listo
Intranet local

- **Estudiante.-** esta opción permite ingresar los datos personales de un nuevo estudiante. Se debe llenar la información que pide el sistema como se muestra a continuación

[Menú](#)

REGISTRAR NUEVO ESTUDIANTE
INFORMACIÓN DEL ESTUDIANTE

Cédula:	<input type="text"/>
Apellido Paterno:	<input type="text" value="CAMACAS"/>
Apellido Materno:	<input type="text" value="CARRERA"/>
Nombres:	<input type="text" value="SERGIO LUIS"/>
Dirección:	<input type="text" value="CARRERA PIMAN"/>
Fecha Nacimiento:	<input type="text" value="05/08/1994"/>
Estado Civil:	<input type="radio"/> Casado <input checked="" type="radio"/> Soltero
Género:	<input type="radio"/> Femenino <input checked="" type="radio"/> Masculino
Libreta Militar:	<input type="text"/>
Nacionalidad:	<input type="text" value="Ecuatorina"/>
Provincia:	<input type="text" value="IMBABURA"/>
Cantón:	<input type="text" value="IBARRA"/>
Parroquia:	<input type="text" value="DOLOROSA DEL PRIORATO"/>
Establecimiento Anterior:	<input type="text" value="Escuela Modelo Velasco Ibarra"/>
Registro:	<input type="text" value="NUEVO"/>

La Información de la imagen anterior es obligatoria es decir que no se puede dejar sin llenar ya que el sistema no guardará la información si no está completa. A continuación se explica detalladamente la información que se debe ingresar:

CAMPO	DESCRIPCIÓN	OBLIGATORIO
Cédula	Se debe ingresar el número de cédula en caso de que el estudiante la tenga, debe	No

	tener 10 dígitos y sin raya por Ej.: 1001563250 caso contrario el sistema nos dará un error de cédula	
Apellido Paterno	Ingresar el apellido paterno del estudiante.	No
Apellido Materno	Ingresar el apellido materno del estudiante.	Si
Nombres	Ingresar los nombres del estudiante.	Si
Dirección	Ingresar la dirección del domicilio del estudiante.	Si
Fecha de Nacimiento	Ingresar la fecha de nacimiento del estudiante, con el formato dd/mm/aaaa (dia/mes/año), por Ej.: 17/06/1995	Si
Género	Seleccionar masculino o femenino	Si
Libreta Militar	Ingresar el número de la libreta militar del estudiante en caso de que la tenga.	No
Nacionalidad	Seleccionar de la lista a que nacionalidad pertenece el estudiante.	Si
Provincia	Seleccionar de la lista en que provincia reside el estudiante.	Si
Cantón	Seleccionar de la lista en que cantón reside el estudiante.	Si
Parroquia	Seleccionar de la lista en que parroquia reside el estudiante.	Si
Establecimiento anterior	Ingresar el nombre del establecimiento en el que estudió anteriormente el estudiante.	No
Registro	Seleccionar nuevo de la lista ya que solo se ingresa datos de nuevos estudiantes.	Si

Esta es información correspondiente al estudiante, pero también se puede llenar información adicional como es información de los padres del estudiante, e información socioeconómica que no es obligatorio llenar.

INFORMACIÓN DE LOS PADRES

Cédula Padre:	<input type="text"/>	Nombre Padre:	<input type="text"/>
Instrucción:	Seleccione Instrucción ▾	Actividad:	Seleccione Actividad ▾
Profesión:	Seleccione Profesión ▾	Vive Padre:	<input type="radio"/> Si <input type="radio"/> No
Nacionalidad:	Seleccione Nacionalidad ▾		

Cédula Madre:	1001895624	Nombre Madre:	FLOR CARREA
Instrucción:	Secundaria ▾	Actividad:	AMA DE CASA ▾
Profesión:	Seleccione Profesión ▾	Vive Madre:	<input checked="" type="radio"/> Si <input type="radio"/> No
Nacionalidad:	Ecuatoriana ▾		

INFORMACIÓN ADICIONAL

Hogar Social:	Seleccione Tipo ▾	Hogar Económico:	Seleccione Tipo ▾
N° de Hijos:	<input type="text"/>	Cuantos Estudian:	<input type="text"/>
Con quien vive:	Seleccione Tipo ▾	Trabaja estudiante:	<input type="radio"/> Si <input type="radio"/> No

Guardar

Intranet I

CAMPO	DESCRIPCIÓN	OBLIGATORIO
Cédula Madre	Ingresar el número de cédula de la madre del estudiante la tenga, debe tener 10 dígitos y sin raya por Ej.: 1001563250 caso contrario el sistema nos dará un error de cédula	No
Nombre Madre	Ingresar el nombre completo de la madre del estudiante.	Si
Instrucción	Seleccionar de la lista el grado de instrucción que tiene la madre del estudiante.	No
Actividad	Seleccionar de la lista la actividad a la que se dedica la madre del estudiante.	No
Profesión	Seleccionar de la lista la profesión de la madre del estudiante	No
Vive Madre	Seleccionar si vive o no la madre del	No

	estudiante.	
Nacionalidad	Seleccionar de la lista la nacionalidad de la madre del estudiante.	No

Para la información del padre del estudiante tiene los mismos campos que para la madre.

En información adicional ningún campo es obligatorio si desea puede completar esta información.

Al terminar de ingresar la información haga clic en el botón Guardar, si no hay ningún error el sistema guarda la información del estudiante y regresa al menú principal, caso contrario primero deberá corregir los errores. El error más común en este formulario es no ingresar datos que son obligatorios.

- **Cursos.-** haciendo clic en Cursos nos despliega una lista de todos los cursos que hay en el plantel y otras opciones como se indica en la siguiente figura:



- **Horario.-** nos permite visualizar el horario de clases de un curso, que también se lo puede obtener impreso haciendo clic en la opción imprimir que se encuentre en la parte inferior, como se indica en la siguiente figura.

COLEGIO NACIONAL 28 DE SEPTIEMBRE

HORARIO DE CLASES						
CURSO:	null	AÑO LECTIVO: 2008-2009				
HORA	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	
07:15:00 - 08:00:00	COMPUTACION	ESTUDIOS SOCIALES	SOCIOECONOMIA	LEGELACION LABORAL	ESTUDIOS SOCIALES	
08:00:00 - 08:45:00	COMPUTACION	ESTUDIOS SOCIALES	SOCIOECONOMIA	FILOSOFIA	ESTUDIOS SOCIALES	
08:45:00 - 09:30:00	MATEMATICA	INVESTIGACION	FUND. DE CONTABILIDAD	PSICOLOGIA	LITERATURA	
09:30:00 - 10:15:00	MATEMATICA	INVESTIGACION	FUND. DE CONTABILIDAD	PSICOLOGIA	LITERATURA	
10:45:00 - 11:30:00	ESTUDIOS SOCIALES	LITERATURA	CULTURA FISICA	INGLES	INGLES	
11:30:00 - 12:15:00	ESTUDIOS SOCIALES	LITERATURA	CULTURA FISICA	INGLES	INGLES	
12:15:00 - 13:00:00	FILOSOFIA	LEGELACION LABORAL	ADMINISTRACION	ESTUDIOS SOCIALES	ED. AMOR Y SEXUALIDAD	
13:00:00 - 13:45:00	FILOSOFIA	LEGELACION LABORAL	ADMINISTRACION	ESTUDIOS SOCIALES	ED. AMOR Y SEXUALIDAD	

Si desea obtener la impresión en sentido horizontal primero debe configurar la página siguiendo estos pasos.

- a) Hacer clic en Archivo y luego en configuración de página y seleccionar sentido horizontal como se indica a continuación.



- o **Nómina.-** esta opción nos permite visualizar la nómina de estudiantes del un curso, también se lo puede obtener impreso haciendo clic en la opción imprimir.

[Atrás](#)

COLEGIO NACIONAL MIXTO 28 DE SEPTIEMBRE
NOMINA DE ESTUDIANTES

2008-2009

Curso: 2 SOCIALES
Paralelo: A

Nº	Apellidos y Nombres
1	ANDRANGO SIMBAÑA MIGUEL EDUARDO
2	ANRANGO SANDOVAL EVELIN GUISELDA
3	CAMPANA MERINO CAROLINA EVANGELINA
4	CARAGOLLA QUILO CARLOS JAVIER
5	CERVANTES POZO GUADALUPE INES
6	CLERQUE MORA FAUSTO RAMIRO
7	CORAL MARTINEZ NORMA PATRICIA
8	CRIOLLO ARROYO MARCELA ALEJANDRA
9	CUASPUD CANGAS JESSICA MARICELA
10	GONZALES LOPEZ ERIKA PAULINA
11	GUAJAN PANTOJA GABRIELA ESTEFANIA

- **Promedios:** esta opción permite un reporte del promedio trimestral de los estudiantes de un curso ordenado descendientemente es decir desde el promedio más alto al más bajo, para esto se debe seleccionar un trimestre de la lista y se obtendrá el reporte que se lo puede imprimir haciendo clic en el botón respectivo.

[Atrás](#) [Menú](#)

PROMEDIO GENERAL DE CADA ESTUDIANTE

Curso: 2 SOCIALES A
Trimestre: 1

Puesto	Apellidos y Nombres	Promedio
1	CAMPANA MERINO CAROLINA EVANGELINA	18,57
2	GUAJAN PANTOJA GABRIELA ESTEFANIA	18,43
3	MEDIAVILLA LEON SHEYLA ESTEFANIA	17,14
4	CUASPUD CANGAS JESSICA MARICELA	17,00
5	GONZALES LOPEZ ERIKA PAULINA	16,79
6	IBUJES POTOSI PAMELA ALEJANDRA	16,64
7	TAYAN CHICAIZA JAIRO ESTEBAN	16,36
8	IPIALES PERUGACHI ANGELITA VIVIANA	16,29
9	CERVANTES POZO GUADALUPE INES	15,93
10	CORAL MARTINEZ NORMA PATRICIA	15,57
11	CARAGOLLA QUILO CARLOS JAVIER	14,64

- **Supletorio.-** Haciendo clic en esta opción se puede obtener una lista de los estudiantes de un curso que se quedan a supletorio indicando en que materias y con qué puntaje se quedaron, se la puede imprimir haciendo clic en el botón respectivo.

COLEGIO NACIONAL 28 DE SEPTIEMBRE

HORARIO DE CLASES						
CURSO:	null	AÑO LECTIVO: 2008-2009				
HORA	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	
07:15:00	08:00:00	COMPUTACION	ESTUDIOS SOCIALES	SOCIOECONOMIA	LEGISLACION LABORAL	ESTUDIOS SOCIALES
08:00:00	08:45:00	COMPUTACION	ESTUDIOS SOCIALES	SOCIOECONOMIA	FLOSOFIA	ESTUDIOS SOCIALES
08:45:00	09:30:00	MATEMATICA	INVESTIGACION	FUND. DE CONTABILIDAD	PSICOLOGIA	LITERATURA
09:30:00	10:15:00	MATEMATICA	INVESTIGACION	FUND. DE CONTABILIDAD	PSICOLOGIA	LITERATURA
10:45:00	11:30:00	ESTUDIOS SOCIALES	LITERATURA	CULTURA FISICA	INGLES	INGLES
11:30:00	12:15:00	ESTUDIOS SOCIALES	LITERATURA	CULTURA FISICA	INGLES	INGLES
12:15:00	13:00:00	FLOSOFIA	LEGISLACION LABORAL	ADMINISTRACION	ESTUDIOS SOCIALES	ED. AMOR Y SEXUALIDAD
13:00:00	13:45:00	FLOSOFIA	LEGISLACION LABORAL	ADMINISTRACION	ESTUDIOS SOCIALES	ED. AMOR Y SEXUALIDAD

- **Perdidos Año.-** haciendo clic en esta opción se puede imprimir una lista de los estudiantes de un curso que pierden el año.
- **Estudiante.-** haciendo clic en esta opción obtenemos una lista de los estudiantes del curso seleccionado y tiene dos opciones que son libreta y promoción.



COLEGIO NACIONAL "28 DE SEPTIEMBRE"

[Atrás](#) [Menú](#)

Curso: 2 SOCIALES A

Apellidos y Nombres	Libretas	Promoción
ANDRANGO SIMBAÑA MIGUEL EDUARDO	Libreta	Promoción
ANRANGO SANDOVAL EVELIN GUISSOLA	Libreta	Promoción
CAMPANA MERINO CAROLINA EVANGELINA	Libreta	Promoción
CARAGOLLA QUILLO CARLOS JAVIER	Libreta	Promoción
CERVANTES POZO GUADALUPE INES	Libreta	Promoción
CLERQUE MORA FAUSTO RAMIRO	Libreta	Promoción
CORAL MARTINEZ NORMA PATRICIA	Libreta	Promoción
CRIOLO ARROYO MARCELA ALEJANDRA	Libreta	Promoción
CUASPUD CANGAS JESSICA MARICELA	Libreta	Promoción
GONZALES LOPEZ ERIKA PAULINA	Libreta	Promoción
GUAJAN PANTOJA GABRIELA ESTEFANIA	Libreta	Promoción
IBUJES POTOSI PAMELA ALEJANDRA	Libreta	Promoción
IPALES PERUGACHI ANGELITA VIVIANA	Libreta	Promoción
LECHON ANDRANGO JOHN WILLIAM	Libreta	Promoción
MEDIAVILLA LEON SHEYLA ESTEFANIA	Libreta	Promoción
TAYAN CHICAIZA JAIRO ESTEBAN	Libreta	Promoción
VALDIVIEZO FLORES GLENDY PATRICIA	Libreta	Promoción

Derechos Reservados

- **Libreta.-** esta opción nos permite obtener la libreta de calificaciones del estudiante seleccionado, con la lista de materias y su respectivo promedio, promedio general de aprovechamiento y conducta. Se puede imprimir la libreta haciendo clic en el botón correspondiente.

COLEGIO NACIONAL MIXTO 28 DE SEPTIEMBRE

Ibarra - Ecuador

Estudiante: CAMPANA MERINO CAROLINA EVANGELINA
Curso: 2 SOCIALES A

CALIFICACIONES TRIMESTRALES

Materia	Trim 1	Trim 2	Trim 3
FILOSOFIA	20	19	20
MATEMATICA	19	15	16
LEGISLACION LABORAL	18	19	17
INGLES	19	20	18
SOCIO ECONOMIA	18	19	19
INVESTIGACION	18	18	17
PSICOLOGIA	18	18	17
ESTUDIOS SOCIALES	19	18	18
LITERATURA	20	18	19
CULTURA FISICA	15	18	16
ED. AMOR Y SEXUALIDAD	20	20	18
ADMINISTRACION	17	19	18
COMPUTACION	19	18	19

- **Promoción.-** esta opción permite imprimir el certificado anual de promoción que obtienen los estudiantes que aprueba el año, la promoción solo se puede imprimir al finalizar el año lectivo cuando ya se hayan ingresado las notas del tercer trimestre y supletorios.

CERTIFICADO ANUAL DE PROMOCIÓN

Año Lectivo:2008-2009

El Colegio, conforme a la ley de educación, confiere el presente certificado de PROMOCION al estudiante GONZALES LOPEZ ERIKA PAULINA del 2 SOCIALES del ciclo Bachillerato, luego de haberse presentado a exámenes y obtener las siguientes calificaciones:

COMPUTO DE CALIFICACIONES

Materias	Trim 1	Trim 2	Trim 3	Total	Promedio	Supletorio	Aprueba
FILOSOFIA	19	19	20	58	19,33		S
MATEMATICA	12	14	15	41	13,67		S
LEGISLACION LABORAL	15	17	18	50	16,67		S
INGLES	16	20	19	55	18,33		S
SOCIO ECONOMIA	17	19	18	54	18		S
INVESTIGACION	17	18	17	52	17,33		S
PSICOLOGIA	17	19	17	53	17,67		S
ESTUDIOS SOCIALES	17	18	19	54	18		S
LITERATURA	19	17	19	55	18,33		S
CULTURA FISICA	16	18	13	47	15,67		S
ED. AMOR Y SEXUALIDAD	20	20	19	59	19,67		S
ADMINISTRACION	17	19	18	54	18		S

Intranet

- **Certificado de Matrícula.-** al hacer clic en esta opción se desplegará la siguiente pantalla que permite obtener un certificado de matrícula de un estudiante siguiendo estos pasos:

- Se debe escoger un tipo de búsqueda sea por cédula o apellidos del estudiante, e ingresar los datos respectivos y hacer clic en el botón Buscar.



[Menú](#)

Buscar Estudiante por: Cédula Apellidos

Apellidos:



- Se mostrará la siguiente pantalla, con una lista de estudiantes que tengan los apellidos con los que se realizó la búsqueda.



[Atrás](#) [Menú](#)

Estudiantes	Curso
CLERQUE MORA FAUSTO RAMIRO	2 SOCIALES A



- Aquí se debe seleccionar el estudiante que estamos buscando y se obtiene un certificado de matrícula listo para imprimir como se indica en la siguiente figura.

COLEGIO NACIONAL MIXTO 28 DE SEPTIEMBRE

IBARRA 21-jun-2009

CERTIFICO:QUE EL SR. / SRTA. CLERQUE MORA FAUSTO RAMIRO PREVIO LOS REQUISITOS LEGALES SE SE MATRICULO COMO ALUMNO DEL SEGUNDO AÑO DEL CICLO DIVERSIFICADO DE ESTE COLEGIO PARA EL PERIODO LECTIVO 2008-2009 LA PARTIDA CORRESPONDIENTE CONSTA EN EL FOLIO N° 141 DEL RESPECTIVO LIBRO DE MATRICULAS.

SRA. TANIA VERONICA BENAVIDES

SECRETARIA

- **Matrícula.-** esta opción permite matricular a estudiantes en los respectivos cursos siguiendo estos pasos.
 - Al hacer clic en Matrícula se muestra la siguiente pantalla, donde tenemos que escoger el año lectivo anterior para los estudiantes que son del colegio y seleccionar nuevo estudiante en caso de que sea un estudiante que ingresa al 8vo año de básica o que venga de otro colegio, y además ingresar los apellidos del estudiante y hacer clic en el botón Buscar para que nos muestre la información del estudiante que se va a matricular.



[Menú](#)

BUSQUEDA DE ESTUDIANTES PARA MATRICULA

Año Lectivo Anterior:	2008-2009
Apellidos:	CAMPANA

- En caso de que exista varios estudiantes con los mismos apellidos como puede ser el caso de que sean hermanos aparece una lista de los

estudiantes que concuerdan con los apellidos de la búsqueda y qué curso aprobaron el año lectivo anterior.

Menú

Apellidos y Nombres	
Estudiantes	Curso
CAMPANA MERINO LENIN VINICIO	10 EDUCACION BASICA A
CAMPANA MERINO CINTHIA VIVIANA	1 SOCIALES A
CAMPANA MERINO CAROLINA EVANGELINA	2 SOCIALES A

Derechos Reservados

- De la lista de estudiantes que muestra se debe seleccionar el que estaba buscando y se mostrará una pantalla que debe completar la siguiente información.

Menú

MATRICULA

Fecha:	18/03/2010
Estudiante:	CAMPANA MERINO LENIN VINICIO
Año Lectivo:	2008-2009
Curso:	1 SOCIALES A
Número de Matricula:	1
Representante:	SRA FLORE MERINO

Aceptar

Derechos Reservados

CAMPO	DETALLE	OBLIGATORIO
Curso	Se debe seleccionar un curso de los de la lista, en caso de que el estudiante haya aprobado el año anterior en el colegio, en la lista de cursos solo estará disponible el curso al que fue promovido, y si es un estudiante nuevo en la lista se mostrarán todos los cursos que hay en el establecimiento.	Si

Representante	En este campo se debe escribir el nombre de la persona que va a ser representante del estudiante en la institución.	Si
---------------	---	----

- Una vez que se haya completado la información se debe hacer clic en el botón Aceptar, para imprimir el formulario de matrícula que será firmado por el representante y la secretaria, como se indica en la siguiente figura.

COLEGIO NACIONAL MIXTO 28 DE SEPTIEMBRE

Código Matrícula: 5
 Folio: 5
 N° Matrícula: 1
 Estudiante: CHILUIZA BENITEZ DIEGO ARMANDO

El 22/06/09 , en la ciudad de IBARRA, CHILUIZA BENITEZ DIEGO ARMANDO de nacionalidad Ecuatoriana, hijo (a) de: y de SRA. BENITEZ , se matricula como alumno del 2 SOCIALES del ciclo Bachillerato , en el año lectivo 2009-2010

IBARRA , 22-jun-2009

SRA. BENITEZ

Sra. TANIA VERONICA BENAVIDES

REPRESENTANTE

SECRETARIA

[Menú](#)

- Hacer clic en imprimir permite obtener un impreso del formulario de matrícula; y si desea regresar al menú principal hacer clic en Menú.
- **Retirar estudiante.-** esta opción permite retirar a un estudiante del establecimiento, con lo que el estudiante ya no constará en la nómina del curso siguiendo estos pasos:
 - Al hacer clic en la opción Retirar estudiante aparece la siguiente pantalla donde se debe seleccionar el tipo de búsqueda sea por cédula o por apellidos, e ingresar la información de acuerdo a lo seleccionado.

COLEGIO NACIONAL "28 DE SEPTIEMBRE"

[Menú](#)

RETIRAR ESTUDIANTE

Buscar Estudiante por: Cédula Apellidos

Apellidos:

Derechos Reservados

- En caso de que en el año lectivo actual haya varios estudiantes con los mismos apellidos, mostrará una lista con los posibles estudiantes de los cuales se seleccionará el estudiante que se estaba buscando como se ha indicado en casos anteriores. Y en caso de que solo exista un estudiante con dichos apellidos se mostrará la siguiente pantalla en donde se debe seleccionar un motivo por el cual el estudiante abandona la institución y a continuación hacer clic en el botón Retirar y se regresará al menú principal, como se muestra en la siguiente figura.

COLEGIO NACIONAL "28 DE SEPTIEMBRE"

[Atrás](#) [Menú](#)

DATOS DEL ESTUDIANTE

Apellidos:	CHILUIZA BENTEZ
Nombres:	DIEGO ARMANDO
Motivo Salida:	CAMBIO DE INSTITUCION
Código Matrícula:	108
Curso:	1 CIENCIAS
Paralelo:	A

Derechos Reservados

- **Personal.-** esta opción permite registrar un nuevo integrante del personal de la institución como se muestra a continuación.

REGISTRO DE NUEVO PERSONAL

Tipo Personal: Administrativo

Cédula:	1000314656
Apellido Paterno:	
Apellido Materno:	ARROYO
Nombres:	ROSA ELENE
Dirección:	NAZACOTA PUENTO Y EMPERADOF
Título:	NINGUNO
Cargo:	TECNICO A
Departamento:	SERVICIOS
Fecha Ingreso Magisterio:	30/05/1988
Fecha Nacimiento:	28/08/1941
Estado Civil:	<input type="radio"/> Casado <input type="radio"/> Soltero <input checked="" type="radio"/> Viudo <input type="radio"/> Divorciado <input type="radio"/> Unión Libre
Género:	<input checked="" type="radio"/> Femenino <input type="radio"/> Masculino
Fecha Último Ascenso:	
Libreta Militar:	
N° Afiliación al IESS:	
Categoría Nominal Actual:	0
Partida Presupuestaria:	

Una vez ingresados todos los datos requeridos hacer clic en el botón Guardar, si no hubo ningún error en los datos ingresados se regresará al menú principal. A continuación se indica detalladamente los campos necesarios para el ingreso de nuevo personal.

- **Gobierno Estudiantil.-** esta opción permite registrar a los estudiantes que conforman el gobierno estudiantil siguiendo estos pasos.
 - Al hacer clic en gobierno estudiantil aparece una pantalla en la que se escoge el tipo de búsqueda y se ingresa los apellidos o cédula del estudiante y luego se hace clic en el botón Buscar.



COLEGIO NACIONAL "28 DE SEPTIEMBRE"

[Menú](#)

ESTUDIANTES DEL GOBIERNO ESTUDIANTIL

Buscar Estudiante por: Cédula Apellidos

Apellidos: CAMPANA

Buscar

- Si hay coincidencia de apellidos mostrará la siguiente pantalla de donde se debe escoger el estudiante que buscamos.



- Una vez seleccionado el estudiante nos muestra la siguiente pantalla donde se debe completar la siguiente información: seleccionar de la lista el cargo que ocupa el estudiante y además ingresar la fecha en que inicia sus funciones en formato día/mes/año y hacer clic en el botón guardar.



- **Estructuras.-** esta opción permite generar las estructuras que cada institución entrega a la Dirección de Educación en cada fin de año lectivo. Las estructuras que se generan son: identificación de institución, personal, nómina de estudiantes, plan general de cursos, plan general de materias, distributivo docente, especialidades, gobierno estudiantil, resumen de calificaciones anuales, matrículas. Para generar cualquiera de estas estructuras se debe completar la siguiente información: fecha en formato día/mes/año, seleccionar de la lista el

año lectivo al que corresponden las estructuras, ingresar el código DECI que es designado por la Dirección de Educación a cada institución educativa, y seleccionar de la lista la estructura que desea y hacer clic en el botón Generar.

COLEGIO NACIONAL "28 DE SEPTIEMBRE"

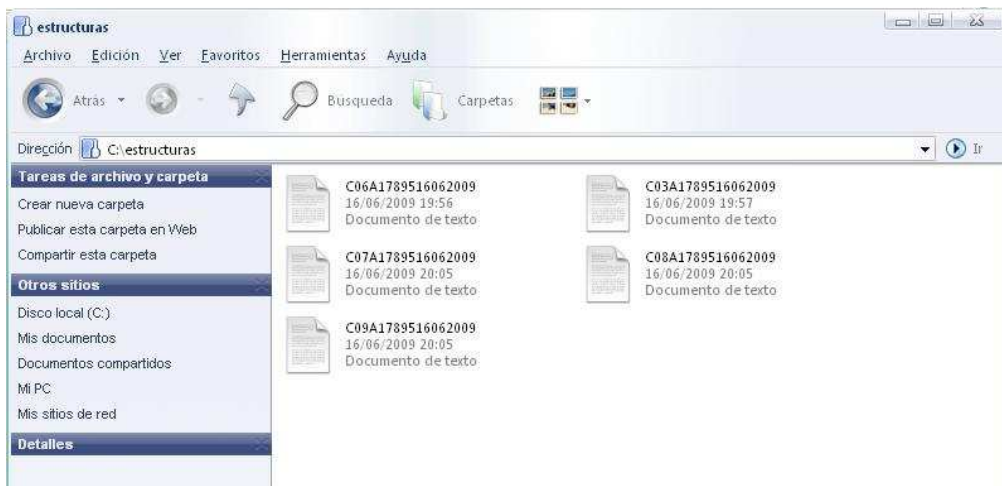
[Menú](#)

GENERACION DE ESTRUCTURAS

Fecha:	17/03/2010
Año Lectivo:	2008-2009
Código DECI:	1841
Estructura:	Personal

Derechos Reservados

Para ver las estructuras generadas debe ir al disco C y abrir la carpeta llamada estructuras, ahí se encuentran todas las estructuras que ha generado el sistema, como indica la siguiente figura. Si desea regresar al menú principal hacer clic en Menú.



MANUAL PARA EL TIPO DE USUARIO DOCENTE

Luego de que haya ingresado su nombre de usuario y contraseña el sistema mostrará el siguiente menú.



- **Cambio Clave.-** haciendo clic en esta opción el usuario puede cambiar la contraseña de ingreso al sistema. El usuario debe ingresar la nueva clave y en el casillero de confirmación de clave volverla a escribir y hacer clic en el botón guardar, si no hay ningún error la clave será cambiada y regresará al menú principal.

Menú

CAMBIO DE CLAVE DE USUARIO

Usuario:	fbenevides
Clave Actual:	1001862521
Nueva Clave:
Confirme Clave:

Guardar

Derechos Reservados

En caso de que la nueva clave y la confirmación de clave no sean iguales el sistema dará el siguiente error: Falló confirmación de contraseña.



Este error se soluciona ingresando correctamente la clave.

- **Horario.-** esta opción permite imprimir al docente su horario, con las materias que dicta en los respectivos cursos. Para obtener el impreso debe hacer clic en Imprimir. En la siguiente figura se muestra un ejemplo de cómo se visualiza el horario.

COLEGIO NACIONAL 28 DE SEPTIEMBRE
HORARIO DE CLASES POR DOCENTE

DOCENTE:		DR. BERTHA ESPERANZA NUÑEZ			
AÑO LECTIVO:		2008-2009			
DIA	HORA	CURSO		MATERIA	
Lunes	7.15 AM - 8.00 AM	3	SOCIALES	A	LEGISLACION LABORAL
Martes	12.15 PM - 1.00 PM	2	SOCIALES	A	LEGISLACION LABORAL
Martes	1.00 PM - 1.45 PM	2	SOCIALES	A	LEGISLACION LABORAL
Jueves	7.15 AM - 8.00 AM	2	SOCIALES	A	LEGISLACION LABORAL

- **Notas Parciales.-** esta opción le permite al docente ingresar 4 notas parciales para obtener un promedio trimestral siguiendo estos pasos.
 - Al hacer clic en Notas parciales se muestra una lista con los cursos y las materias que dicta en cada uno de ellos.
 - Al seleccionar una materia y su respectivo curso se muestra la lista de estudiantes que pertenecen al curso seleccionado, con dos opciones que son de Ingresar y Editar notas que se encuentran a la derecha del nombre de cada estudiante como se indica en la siguiente figura.

INGRESO Y EDICION DE NOTAS

Curso
2 SOCIALES A LEGISLACION LABORAL
3 SOCIALES A LEGISLACION LABORAL
3 SOCIALES A SOCIO ECONOMIA

Nº	Apellidos y Nombres	Ingresar	Editar
1	AGUILAR VALLEJO MARLON SANTIAGO	Ingresar	Editar
2	CASTRO PERUGACHI AMANDA MARIANELA	Ingresar	Editar
3	CHAVEZ MORENO EDISON JAVIER	Ingresar	Editar
4	ECHEVERRIA CUESTA JHONATAN ALEJANDRO	Ingresar	Editar
5	ESPINOSA FERIGRA EDISON GEOVANY	Ingresar	Editar
6	ILES VALLES KLEVER EMERSON	Ingresar	Editar
7	JUMA NAZATE SABRINA ALEXANDRA	Ingresar	Editar
8	LOMAS SANDOVAL MIRIAN GABRIELA	Ingresar	Editar
9	MAYNAGUEZA QUENAN WILLIAM SANTIAGO	Ingresar	Editar
10	MOROCHO LOPEZ RICARDO SAUL	Ingresar	Editar
11	POZO QUIROZ PABLO JAVIER	Ingresar	Editar
12	QUILO IBADANGO CRISTIAN ANDRES	Ingresar	Editar
13	ROMAN FLORES DENIS VINICIO	Ingresar	Editar
14	SANDOVAL PUPIALES OSCAR ANDERSON	Ingresar	Editar
15	SANDOVAL TOBAR JONATHAN PATRICIO	Ingresar	Editar

- **Ingresar.-** Para ingresar las 4 notas parciales se debe hacer clic en el vínculo Ingresar que está a la derecha del estudiante del cual se va a ingresar las notas y se muestra la siguiente pantalla que contiene la siguiente información: nombre del docente, la materia, el nombre del estudiante y una lista para seleccionar el trimestre al que pertenecen las notas que se van a ingresar, además aquí se ingresa la conducta que tiene cada estudiante en cada materia como se indica en la siguiente figura.

NOMINA PARA INGRESO DE NOTAS

Docente:	DR. MAIRENE ARCONIEGA
Materia:	CIENCIAS NATURALES
Curso:	10 EDUCACION BASICA A
Trimestre:	1

Nº	Estudiante	Nota1	Nota2	Nota3	Nota4
1	CAMPANA MERINO LENIN VINICIO				
2	CASTILLO MARTINEZ ANGELA FERNANDA				
3	CHANDI PASTILLO JUAN DIEGO				
4	CUASPUD CANGAS JOCELYN ALEXANDRA				
5	ENRIQUEZ QUITO IVAN SIMON				
6	ENRIQUEZ VIANA HENRY OSWALDO				
7	ERAZO ICHAU TANIA FERNANDA				
8	JUMA NAZATE IRMA FERNANDA				
9	JUMA VILATUÑA LEYDI LISETH				
10	LOPEZ POZO ALVARO JAVIER				
11	MONTENEGRO CARLOSAMA JUAN GUILLERMO				
12	NIPAS QUITO LUIS FRANCISCO				
13	ORTIZ FIERRO DIANA ELIZABETH				

- a) Se debe seleccionar el trimestre, en caso de que las notas parciales tengan decimales serán ingresados con el signo. por Ej. 17.5 o sin decimales 17.
- b) Se debe ingresar la nota de conducta en el casillero correspondiente y hacer clic en el botón Guardar y regresará a la pantalla anterior donde está la lista de estudiantes para seguir ingresando notas y si desea regresar al menú principal hacer clic en Menú.

NOMINA PARA INGRESO DE NOTAS

Docente:	DR. MAIRENE ARCINIEGA
Materia:	CIENCIAS NATURALES
Curso:	10 EDUCACION BASICA A
Trimestre:	1

Nº	Estudiante	Nota1	Nota2	Nota3	Nota4
1	CAMPANA MERINO LENIN VINICIO	12	11	19	17
2	CASTILLO MARTINEZ ANGELA FERNANDA	17	18	11	12
3	CHANDI PASTILLO JUAN DIEGO	11	12	17	17
4	CUASPUD CANGAS JOCELYN ALEXANDRA	16	17	18	10
5	ENRIQUEZ QUITO IVAN SIMON	10	11	10	17
6	ENRIQUEZ VIANA HENRY OSWALDO	17	16	10	18
7	ERAZO ICHAU TANIA FERNANDA	17	18	7	8
8	JUMA NAZATE IRMA FERNANDA	17	18	11	12
9	JUMA VILATUÑA LEYDI LISETH	11	10	18	17
10	LOPEZ POZO ALVARO JAVIER	17	17	16	11
11	MONTENEGRO CARLOSAMA JUAN GUILLERMO	17	11	17	11
12	NIPAS QUITO LUIS FRANCISCO	17	16	11	12
13	ORTIZ FIERRO DIANA ELIZABETH	18	16	16	10

- o **Editar.-** la opción de edita solo se debe utilizar en caso que se haya cometido errores al digitar las notas, siguiendo estos pasos.
 - a) Hacer clic en editar, al hacer esto se mostrará la siguiente pantalla donde se debe escoger el trimestre del cual se van a editar las notas y el sistema mostrará las calificaciones obtenidas en el trimestre seleccionado.

NOMINA PARA INGRESO DE NOTAS

Docente:	DR. MAIRENE ARCINIEGA
Materia:	CIENCIAS NATURALES
Curso:	10 EDUCACION BASICA A
Trimestre:	1

Nº	Estudiante	Nota1	Nota2	Nota3	Nota4
1	CAMPANA MERINO LENIN VINICIO	12	11	19	17
2	CASTILLO MARTINEZ ANGELA FERNANDA	17	18	11	12
3	CHANDI PASTILLO JUAN DIEGO	11	12	17	17
4	CUASPUD CANGAS JOCELYN ALEXANDRA	16	17	18	10
5	ENRIQUEZ QUITO IVAN SIMON	10	11	10	17
6	ENRIQUEZ VIANA HENRY OSWALDO	17	16	10	18
7	ERAZO ICHAU TANIA FERNANDA	17	18	7	8
8	JUMA NAZATE IRMA FERNANDA	17	18	11	12
9	JUMA VILATUÑA LEYDI LISETH	11	10	18	17
10	LOPEZ POZO ALVARO JAVIER	17	17	16	11
11	MONTENEGRO CARLOSAMA JUAN GUILLERMO	17	11	17	11
12	NIPAS QUITO LUIS FRANCISCO	17	16	11	12
13	ORTIZ FIERRO DIANA ELIZABETH	18	16	16	10

b) Se debe cambiar la nota incorrecta y hacer clic en el botón Modificar Notas y regresará la lista de estudiantes.

- **Notas de supletorio.**- esta opción permite ingresar las notas de supletorio. Para hacer el ingreso seguir estos pasos:
 - Al hacer clic en notas de supletorio se muestra una lista con los cursos y las materias que imparte cada docente; de esta lista el docente debe seleccionar la materia de la cual va a ingresar las notas y se mostrará una lista solo con los estudiantes que se han quedado a supletorio como se muestra en la siguiente figura.

Menú

Cursos

- [2 SOCIALES A LEGISLACION LABORAL](#)
- [3 SOCIALES A LEGISLACION LABORAL](#)
- [3 SOCIALES A SOCIO ECONOMIA](#)

Apellidos y Nombres
ANDRANGO SIMBAÑA MIGUEL EDUARDO
CARAGOLLA QUILO CARLOS JAVIER
CLERQUE MORA FAUSTO RAMIRO
CORAL MARTINEZ NORMA PATRICIA
CRIOLLO ARROYO MARCELA ALEJANDRA
LECHON ANDRANGO JOHN WILLIAM
VALDIVIEZO FLORES GLENDY PATRICIA

- Para ingresar la nota de supletorio debe hacer clic en el nombre del estudiante del cual va a ingresar la nota y se mostrará la siguiente pantalla y regresará a la pantalla anterior para seguir ingresando notas de

supletorio. La pantalla de ingreso de nota de supletorio tiene la siguiente información: el nombre del docente, nombre de la asignatura, nombre del estudiante y un casillero donde debe ingresar la nota del estudiante puede ser de 1 a 20.

[Menú](#)

NOTAS DE SUPLETORIO

Profesor (a):	DR. BERTHA ESPERANZA NUÑEZ ALMEIDA
Materia	LEGISLACION LABORAL
Estudiante:	ANDRANGO SIMBAÑA MIGUEL EDUARDO
Nota Supletorio:	<input type="text" value="16"/>
<input type="button" value="Guardar"/>	

- Notas de Grado.-** esta opción permite ingresar las notas obtenidas en los exámenes de grado de las materias que son de especialidad. Al hacer clic en esta opción le mostrará las materias que imparte el docente en tercer año de bachillerato; si alguna de las materias que imparte es de especialidad debe hacer clic sobre dicha materia y se mostrará la lista de estudiantes de dicho curso, como se muestra en la siguiente figura.

[Menú](#)

Cursos

- [3 SOCIALES A SOCIO ECONOMIA](#)
- [3 SOCIALES A LEGISLACION LABORAL](#)

Apellidos y Nombres
AGUILAR VALLEJO MARLON SANTIAGO
CASTRO PERUGACHI AMANDA MARIANELA
CHAVEZ MORENO EDISON JAVIER
ECHEVERRIA CUESTA JHONATAN ALEJANDRO
ESPINOSA FERIGRA EDISON GEOVANY
LES VALLES KLEVER EMERSON
JUMA NAZATE SABRINA ALEXANDRA
LOMAS SANDOVAL MIRIAN GABRIELA
MAYNAGUEZA QUENAN WILLIAM SANTIAGO
MOROCHO LOPEZ RICARDO SAUL
POZO QUIROZ PABLO JAVIER
QUILO IBADANGO CRISTIAN ANDRES
ROMAN FLORES DENIS VINICIO
SANDOVAL PUPIALES OSCAR ANDERSON
SANDOVAL TOBAR JONATHAN PATRICIO

- Para ingresar la nota de grado de un estudiante debe hacer clic sobre su nombre y se mostrará la siguiente pantalla que tendrá la siguiente información: nombre del docente, nombre de la materia, nombre del estudiante y un casillero donde deberá ingresar la nota obtenida por el estudiante y hacer clic en el botón Guardar, como se indica a continuación.

[Menú](#)

INGRESO NOTAS DE GRADO

Profesor (a):	BERTHA ESPERANZA NUÑEZ ALMEIDA
Materia:	SOCIO ECONOMIA
Estudiante:	AGUILAR VALLEJO MARLON SANTIAGO
Nota Examen:	<input type="text" value="16"/>

- **Reporte de Notas Parciales.-** esta opción le permite al docente obtener un reporte de notas de las diferentes materias que imparte como indica en la siguiente figura.



[Menú](#)

REPORTE DE NOTAS

Trimestre:

Curso: [3 SOCIALES A LEGISLACION LABORAL](#)
[2 SOCIALES A LEGISLACION LABORAL](#)
[3 SOCIALES A SOCIO ECONOMIA](#)

- Para Obtener el reporte primero debe seleccionar un trimestre de la lista y luego hacer clic en la materia de la cual quiere tener un reporte o resumen de notas como se indica en la siguiente figura.

[Atrás](#) [Menú](#)

Profesor:	DR. BERTHA ESPERANZA NUÑEZ ALMEIDA
Curso:	2 SOCIALES
Paralelo:	A
Materia:	LEGISLACION LABORAL
Trimestre:	1

Nº	Apellidos y Nombres	Nota 1	Nota 2	Nota 3	Nota 4	Promedio	Total	Promedio
1	ANDRANGO SIMBAÑA MIGUEL EDUARDO	7.0	1.0	14.0	9.0	7.75	31.0	8
2	ANRANGO SANDOVAL EVELIN GUISSELA	10.0	10.0	15.0	9.0	11.0	44.0	11
3	CAMPANA MERINO CAROLINA EVANGELINA	20.0	19.0	18.0	14.0	17.75	71.0	18
4	CARAGOLLA QUILO CARLOS JAVIER	6.0	10.0	14.0	5.0	8.75	35.0	9
5	CERVANTES POZO GUADALUPE INES	17.0	18.0	19.0	14.0	17.0	68.0	17
6	CLERQUE MORA FAUSTO RAMIRO	9.0	6.0	16.0	11.0	10.5	42.0	9
7	CORAL MARTINEZ NORMA PATRICIA	6.0	10.0	16.0	7.0	9.75	39.0	10
8	CRIOLLO ARROYO MARCELA ALEJANDRA	6.0	8.0	11.0	7.0	8.0	32.0	8
9	CUASPOD CANGAS JESSICA MARICELA	14.0	10.0	18.0	13.0	13.75	55.0	14
10	GONZALEZ LOPEZ BERKA RAJHINA	12.0	16.0	18.0	15.0	15.25	61.0	15

- Para tener un documento impreso del reporte hacer clic en Imprimir.
- Si desea obtener el reporte de otra materia haga clic en Atrás, para regresar al menú principal haga clic en Menú.
- **Reporte de Notas Trimestral.-** esta opción permite obtener un resumen de los promedios trimestrales obtenidos por cada estudiante en una materia, primero debe seleccionar la materia y curso del que desea obtener el reporte como se indica a continuación.



[Menú](#)

REPORTE DE NOTAS POR MATERIA

Curso y Materia

[3 SOCIALES A - LEGISLACION LABORAL](#)

[2 SOCIALES A - LEGISLACION LABORAL](#)

[3 SOCIALES A - SOCIO ECONOMIA](#)

El reporte que se obtiene es el siguiente

COLEGIO NACIONAL 28 DE SEPTIEMBRE

REPORTE DE PROMEDIOS POR MATERIA

CURSO:	2 SOCIALES A	AÑO LECTIVO:	2008-2009
MATERIA:	LEGISLACION	DOCENTE:	1001099462
FECHA:	23/03/10 09:27 PM		

APELLIDOS Y NOMBRES	PROMEDIOS		
	TRIM 1	TRIM 2	TRIM 3
ANDRANGO SIMBAÑA MIGUEL EDUARDO	8	9	11
ANRANGO SANDOVAL EVELIN GUISSELA	11	18	17
CAMPANA MERINO CAROLINA	18	19	17
CARAGOLLA QUILO CARLOS JAVIER	9	14	10
CERVANTES POZO GUADALUPE INES	17	14	16
CLERQUE MORA FAUSTO RAMIRO	9	9	13
CORAL MARTINEZ NORMA PATRICIA	10	14	14
CRIOLO ARROYO MARCELA	8	9	8
CUASPUD CANGAS JESSICA	14	17	15
GONZALES LOPEZ ERIKA PAULINA	15	17	18
GUAJAN PANTOJA GABRIELA	18	19	16
IBUJES POTOSI PAMELA	17	17	17
IPALES PERUGACHI ANGELITA	16	16	17
JUMA VILATUÑA ROBINSON ALEXIS	12	null	null
LECHON ANDRANGO JOHN WILLIAM	13	11	12
MEDIAVILLA LEON SHEYLA	13	15	14

- **Reporte General por Curso.-** con esta opción se puede obtener un reporte de los promedios generales de todas las materias de un curso como se indica en la siguiente figura.

COLEGIO NACIONAL 28 DE SEPTIEMBRE

REPORTE DE PROMEDIO DEL CURSO POR MATERIAS

CURSO:	2 SOCIALES A	AÑO LECTIVO:	2008-2009
DETALLE DE PROMEDIOS			
MATERIAS	PRIMER TRIMESTRE		
ADMINISTRACION	15,16		
COMPUTACION	15,21		
CULTURA FISICA	16,00		
ED. AMOR Y SEXUALIDAD	17,44		
ESTUDIOS SOCIALES	15,84		
FILOSOFIA	16,58		
FUND. DE CONTABILIDAD	15,95		
INGLES	15,74		
INVESTIGACION	16,47		
LEGISLACION LABORAL	12,83		
LITERATURA	16,58		
MATEMATICA	10,89		
PSICOLOGIA	15,56		
SOCIO ECONOMIA	15,11		

Este reporte además indica el promedio general del curso.

PROMEDIO GENERAL DEL CURSO: 15,38

- Sílabo.-** esta opción le permite al docente crear un sílabo de cada materia que imparte. Un sílabo es una especie de catálogo de una materia que tiene varias partes como: introducción, objetivo general, objetivos específicos, las unidades y sus respectivos contenidos; lo que le permitirá al docente registrar el avance de cada materia en su respectivo curso.

Para crear un sílabo debe seguir estos pasos:

- Al hacer clic en sílabo se muestra al docente una lista de las materias que imparte como se indica en la siguiente figura.

Menú

Materias	Curso		
LEGISLACION LABORAL	3 SOCIALES A	Crear	Imprimir
LEGISLACION LABORAL	2 SOCIALES A	Crear	Imprimir
SOCIO ECONOMIA	3 SOCIALES A	Crear	Imprimir

Derechos Reservados

- De la lista de materias que se indican en la pantalla anterior se debe seleccionar una haciendo clic sobre el nombre de la materia y se mostrará una pantalla como la que se indica a continuación, en este formulario de debe ingresar una introducción, y un objetivo general de la materia, luego haga clic en continuar para seguir llenando el sílabo. La introducción y el objetivo son campos obligatorios.

COLEGIO NACIONAL MIXTO 28 DE SEPTIEMBRE
SILABO DE LEGISLACION LABORAL

INFORMACION GENERAL

Asignatura: LEGISLACION LABORAL
Docente: NUÑEZ ALMEIDA BERTHA ESPERANZA

INTRODUCCION

La Legislación Laboral es una materia que permite conocer a los educandos las leyes que rigen en cuanto a lo laboral.

OBJETIVO GENERAL

Conocer sobre la legislación laboral en nuestro país.

- Al hacer clic en continuar nos muestra la siguiente pantalla donde se debe ingresar los objetivos específicos de la materia; se puede tener varios objetivos específicos en una asignatura, cada objetivo debe ser ingresado de forma individual y hacer clic en el botón Guardar, si ya no desea ingresar más objetivos hacer clic en el botón Continuar.

Ingrese Objetivos Especificos:

Conocer la importancia de la asignatura Legislación Laboral.

- Al hacer clic en continuar obtendremos la siguiente pantalla donde se va a ingresar las unidades de la asignatura siguiendo estos pasos:
 - a) **Número de la unidad.**- que se va a ingresar, este campo es obligatorio.
 - b) **Nombre de la Unidad.**- aquí se debe ingresar el nombre del a unidad, este campo es obligatorio.
 - c) **Descripción.**- en este campo de puede ingresar una descripción de la asignatura, este campo es opcional.
 - d) Luego debe hacer clic en Agregar Contenido para poder ingresar los subtemas de cada unidad como se muestra a continuación.

UNIDADES

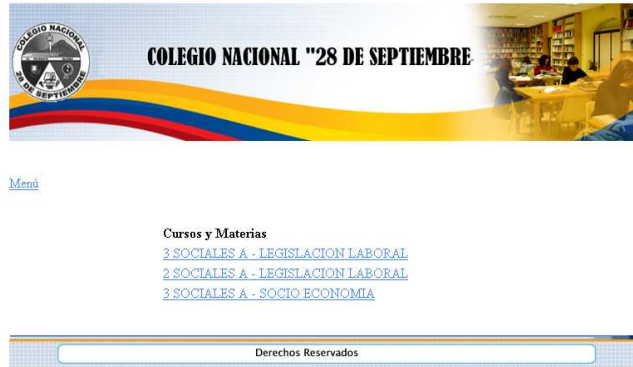
Unidad N°:	1
Nombre Unidad:	Introducción a la Legislación Laboral
Descripción:	Descripción de la definición y la importancia de la asignatura.

- Al hacer clic en Agregar Contenido permite ingresar los subtemas de la unidad que se creó en el paso anterior, como se indica en la siguiente figura.

UNIDADES

Unidad N°:	1
Tema N°:	1
Nombre Tema:	Definición de Legislación Laboral

- Primero se debe ingresar el número del tema y su respectivo nombre estos campos son obligatorios. Cada unidad puede tener varios temas, al hacer clic en Guardar se graba el tema y a la vez permite ingresar otro tema de la misma unidad, este proceso se repite según el número de temas que tenga la unidad. Si ya no desea agregar más temas a la unidad debe hacer clic en Nueva Unidad.
- Solamente el momento que se termina de ingresar las unidades y sus respectivos temas se debe hacer clic en el botón Terminar para concluir con el sílabo y se regresará al menú principal.
- Tome en cuenta que no debe presionar el botón terminar antes de concluir el sílabo ya que no podrá volver a ingresar para terminarlo.
- **Avance Materia.-** esta opción permite a cada docente llevar un registro del avance de los temas o unidades que hay en clase de cada materia. Al hacer clic en esta opción muestra una lista con las materias que imparte el docente. Para registrar el avance debe seleccionar una materia haciendo clic sobre el nombre de la materia.



- Al seleccionar una materia se mostrará la siguiente pantalla, con datos informativos como: nombre del docente, curso, nombre de la materia, horas semanales de la materia, la fecha en la que se registra el avance en día, mes, año; además muestra una lista con las unidades que tiene la materia.

REGISTRO DE AVANCE DE CONTENIDOS

Docente:	NUÑEZ ALMEIDA BERTHA ESPERANZA
Curso:	2 SOCIALES A
Materia:	LEGISLACION LABORAL
Horas Semana:	2
Fecha:	17/11/2009

UNIDADES	CONTENIDO
Introducción a la Legislación Laboral	<input type="radio"/> Definición de Legislación Laboral
Legislación Laboral en el Ecuador	

- Para registrar el avance debe seleccionar una unidad de la lista de unidades haciendo clic sobre el nombre de la unidad y automáticamente se mostrará los temas que tiene la unidad seleccionada, de estos temas debe seleccionar con un clic el tema que ha sido impartido en clases y hacer clic en el botón Guardar y regresará al menú principal.

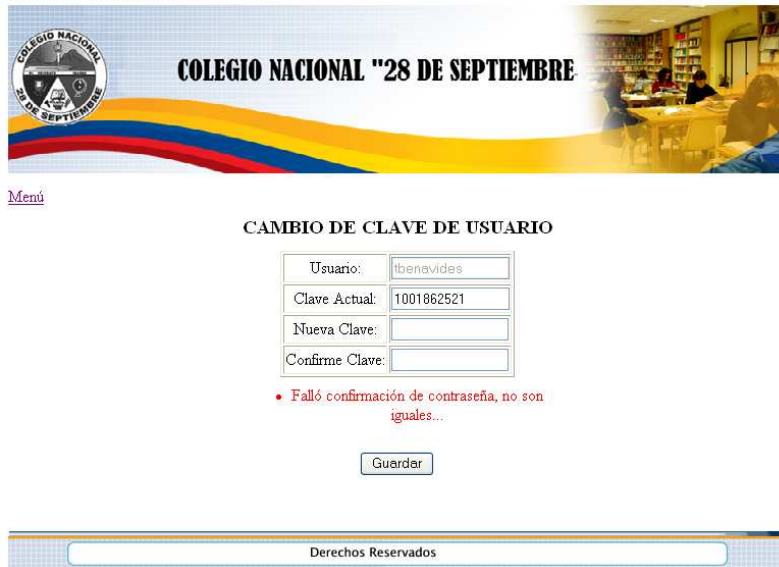
MANUAL DE USUARIO PARA EL TIPO DE USUARIO INSPECTOR



- **Cambio Clave.-** haciendo clic en esta opción el usuario puede cambiar la contraseña de ingreso al sistema. El usuario debe ingresar la nueva clave y en el casillero de confirmación de clave volverla a escribir y hacer clic en el botón guardar, si no hay ningún error la clave será cambiada y regresará al menú principal.



En caso de que la nueva clave y la confirmación de clave no sean iguales el sistema dará el siguiente error: Falló confirmación de contraseña.



Este error se soluciona ingresando correctamente la clave.

- **Cursos.-** al hacer clic en esta opción muestra una lista de los cursos con dos opciones que son: Conducta y Asistencia como se muestra en la siguiente figura.



- **Conducta.-** esta opción permite registrar la calificación que el inspector otorga a los estudiantes de un curso. Para registrar la calificación de conducta debe hacer clic en el vínculo Conducta del curso correspondiente y se mostrará una pantalla con la lista de los estudiantes del curso

seleccionado y un casillero a la derecha de cada nombre para ingresar la nota de conducta; para grabar esta información debe hacer clic en el botón Guardar.

CALIFICACION DE CONDUCTA POR CURSOS

Inspector:	PROF. JULIO MIGUEL PUGA FLORES
Curso:	3 SOCIALES A
Trimestre:	3

Nº	Apellidos y Nombres	Conducta
1	AGUILAR VALLEJO MARLON SANTIAGO	20
2	CASTRO PERUGACHI AMANDA MARIANELA	20
3	CHAVEZ MORENO EDISON JAVIER	20
4	ECHEVERRIA CUESTA JHONATAN ALEJANDRO	20
5	ESPINOSA FERIGRA EDISON GEOVANY	20
6	ILES VALLES KLEVER EMERSON	20
7	JUMA NAZATE SABRINA ALEXANDRA	20
8	LOMAS SANDOVAL MIRIAN GABRIELA	20
9	MAYNAGUEZA QUENAN WILLIAM SANTIAGO	20
10	MOROCHO LOPEZ RICARDO SAUL	20
11	POZO QUIROZ PABLO JAVIER	20
12	QUILO IBADANGO CRISTIAN ANDRES	20

- **Asistencia Estudiantes.-** esta opción permite registrar las faltas de los estudiantes. Para registrar la inasistencia debe seguir estos pasos: Hacer clic en Asistencia del curso correspondiente y obtendrá la lista de estudiantes del curso seleccionado como se indica en la siguiente figura.

REGISTRO DE ASISTENCIA POR CURSO

Curso:	2 SOCIALES A
Fecha:	17/06/09

Apellidos y Nombres	No Asiste
ANDRANGO SIMBAÑA MIGUEL EDUARDO	<input type="checkbox"/>
ANRANGO SANDOVAL EVELIN GUISSOLA	<input type="checkbox"/>
CAMPANA MERINO CAROLINA EVANGELINA	<input type="checkbox"/>
CARAGOLLA QUILO CARLOS JAVIER	<input type="checkbox"/>
CERVANTES POZO GUADALUPE INES	<input type="checkbox"/>
CLERQUE MORA FAUSTO RAMIRO	<input type="checkbox"/>
CORAL MARTINEZ NORMA PATRICIA	<input type="checkbox"/>
CRIOLO ARROYO MARCELA ALEJANDRA	<input type="checkbox"/>
CUASPUD CANGAS JESSICA MARICELA	<input type="checkbox"/>
GONZALES LOPEZ ERIKA PAULINA	<input type="checkbox"/>
GUAJAN PANTOJA GABRIELA ESTEFANIA	<input type="checkbox"/>
IBUJES POTOSI PAMELA ALEJANDRA	<input type="checkbox"/>
IPIALES PERUGACHI ANGELITIA VIVIANA	<input type="checkbox"/>
JUMA VILATUÑA ROBINSON ALEXIS	<input type="checkbox"/>
LECHON ANDRANGO JOHN WILLIAM	<input type="checkbox"/>
MEDIAVILLA LEON SHEYLA ESTEFANIA	<input type="checkbox"/>

Lista Intranet local

Para registrar la inasistencia primero debe ingresar la fecha en el formato día/mes/año y a continuación hacer clic en el cuadro de selección que se encuentra a la derecha de cada estudiante y solo se debe registrar a los estudiantes que faltan a clases, al hacer clic en el cuadro de selección correspondiente al estudiante que ha faltado a clases se dibuja un visto en el cuadro seleccionado como se indica en la siguiente figura y se guardará la falta.

REGISTRO DE ASISTENCIA POR CURSO

Curso:	2 SOCIALES A
Fecha	17/06/09

Apellidos y Nombres	No Asiste
ANDRANGO SIMBAÑA MIGUEL EDUARDO	<input type="checkbox"/>
ANRANGO SANDOVAL EVELIN GUISELLA	<input checked="" type="checkbox"/>
CAMPANA MERINO CAROLINA EVANGELINA	<input type="checkbox"/>
CARAGOLLA QUILO CARLOS JAVIER	<input type="checkbox"/>
CERVANTES POZO GUADALUPE INES	<input type="checkbox"/>
CLERQUE MORA FAUSTO RAMIRO	<input type="checkbox"/>
CORAL MARTINEZ NORMA PATRICIA	<input type="checkbox"/>
CRIOLO ARROYO MARCELA ALEJANDRA	<input type="checkbox"/>
CUASPUD CANGAS JESSICA MARICELA	<input type="checkbox"/>
GONZALES LOPEZ ERIKA PAULINA	<input type="checkbox"/>
GUAJAN PANTOJA GABRIELA ESTEFANIA	<input type="checkbox"/>
IBUES POTOSI PAMELA ALEJANDRA	<input type="checkbox"/>
IPIALES PERUGACHI ANGELITA VIVIANA	<input type="checkbox"/>
JUMA VILATUÑA ROBINSON ALEXIS	<input type="checkbox"/>
LECHON ANDRANGO JOHN WILLIAM	<input type="checkbox"/>
MEDIAVILLA LEON SHEYLA ESTEFANIA	<input type="checkbox"/>

- o **Asistencia Docentes.-** esta opción permite registrar las faltas de los docentes, muestra una lista del personal docente como se indica en la siguiente figura.

[Menú](#)

LISTA DE PERSONAL DOCENTE

Nombres y Apellidos	
LIC. NELLY MARIANA DE JESUS REVELO	Editar Asistencia
LIC. CESAR STALIN VARELA	Editar Asistencia
LIC. EDUARDO VICENTE BURBANO	Editar Asistencia
PROF. JULIO MIGUEL PUGA	Editar Asistencia
LIC. JOSE IGNACIO GAVILANES	Editar Asistencia
LIC. ROSARIO MAGDALENA MORAN	Editar Asistencia
PROF. LUIS ERNESTO FUENTES	Editar Asistencia
LIC. JAIME DAMIAN CALDERON	Editar Asistencia
LIC. AURA ELISA BENAVIDES	Editar Asistencia
DR. BERTHA ESPERANZA NUÑEZ	Editar Asistencia
LIC. MARIA INES FRAGA	Editar Asistencia
DR. EDGAR MARCELO MENDEZ	Editar Asistencia
DR. MAIRENE ARCINIEGA	Editar Asistencia
LIC. TERESA ELIZABETH CARRERA	Editar Asistencia

Al dar clic en Editar asistencia nos muestra el horario del docente; para registrar las faltas se debe dar clic en el día y hora que ha faltado el docente.

[Menú](#) [Atrás](#)

REGISTRO DE ASISTENCIA DOCENTE

Fecha de Registro:	23/03/2010
Docente:	DR. MAIRENE ARCINIEGA

Día	Nº Hora	Hora Inicio	Hora Fin	Materia	Curso	No Asiste
Lunes	6	11:30:00	12:15:00	ED. AMOR Y SEXUALIDAD	9 EDUCACION BASICA	<input type="checkbox"/>
Lunes	1	07:15:00	08:00:00	CIENCIAS NATURALES	8 EDUCACION BASICA	<input type="checkbox"/>
Lunes	2	08:00:00	08:45:00	CIENCIAS NATURALES	8 EDUCACION BASICA	<input type="checkbox"/>
Lunes	3	08:45:00	09:30:00	ED. AMOR Y SEXUALIDAD	9 EDUCACION BASICA	<input type="checkbox"/>
Lunes	4	09:30:00	10:15:00	ED. AMOR Y SEXUALIDAD	9 EDUCACION BASICA	<input type="checkbox"/>
Lunes	7	12:15:00	13:00:00	CIENCIAS NATURALES	9 EDUCACION BASICA	<input type="checkbox"/>
Lunes	8	13:00:00	13:45:00	CIENCIAS NATURALES	9 EDUCACION BASICA	<input type="checkbox"/>
Lunes	5	10:45:00	11:30:00	ED. AMOR Y SEXUALIDAD	9 EDUCACION BASICA	<input type="checkbox"/>
Martes	7	12:15:00	13:00:00	CIENCIAS NATURALES	9 EDUCACION BASICA	<input type="checkbox"/>
Martes	2	08:00:00	08:45:00	ED. AMOR Y SEXUALIDAD	9 EDUCACION BASICA	<input type="checkbox"/>
Martes	8	13:00:00	13:45:00	CIENCIAS NATURALES	9 EDUCACION BASICA	<input type="checkbox"/>
Martes	1	07:15:00	08:00:00	ED. AMOR Y SEXUALIDAD	9 EDUCACION BASICA	<input type="checkbox"/>
Martes	3	08:45:00	09:30:00	CIENCIAS NATURALES	8 EDUCACION BASICA	<input type="checkbox"/>
Martes	4	09:30:00	10:15:00	CIENCIAS NATURALES	8 EDUCACION BASICA	<input type="checkbox"/>

- **Reporte de conducta.-** esta opción le permite obtener un reporte de la conducta de un curso para esto debe seleccionar un curso de la lista y en la pantalla siguiente seleccionar el trimestre del cual quiere obtener el reporte como se muestra a continuación. Este reporte lo puede imprimir haciendo clic en el botón correspondiente.

COLEGIO NACIONAL MIXTO 28 DE SEPTIEMBRE

Inspector:	PROF. JULIO MIGUEL PUGA FLORES
Curso:	2 SOCIALES A
Trimestre:	1

Nº	Apellidos y Nombres	Conducta
1	ANDRANGO SIMBAÑA MIGUEL EDUARDO	17
2	ANRANGO SANDOVAL EVELIN GUISSOLA	17
3	CAMPANA MERINO CAROLINA EVANGELINA	17
4	CARAGOLLA QUILLO CARLOS JAVIER	17
5	CERVANTES POZO GUADALUPE INES	17
6	CLERQUE MORA FAUSTO RAMIRO	17
7	CORAL MARTINEZ NORMA PATRICIA	17
8	CRIOLLO ARROYO MARCELA ALEJANDRA	17
9	CUASPUD CANGAS JESSICA MARICELA	17
10	GONZALES LOPEZ ERIKA PAULINA	17
11	GUAJAN PANTOJA GABRIELA ESTEFANIA	17
12	IBUJES POTOSI PAMELA ALEJANDRA	17
13	IPIALES PERUGACHI ANGELITA VIVIANA	17
14	JUMA VILATUÑA ROBINSON ALEXIS	17
15	LECHON ANDRANGO JOHN WILLIAM	17
16	MEDIAVILLA LEON SHEYLA ESTEFANIA	17
17	TAYAN CHICAIZA JAIRO ESTEBAN	17

- **Justificar faltas de estudiantes.-** esta opción permite al inspector justificar las faltas de los estudiantes. Primero debe buscar al estudiante por apellidos como se indica a continuación.



El sistema le indica todos los estudiantes que tengan el apellido ingresado y que hayan faltado como se indica en la siguiente figura.



Curso	Estudiante	Fecha
2 SOCIALES A	ANRANGO SANDOVAL EVELIN GUISEL	2009-06-17 Justificar
2 SOCIALES A	ANRANGO SANDOVAL EVELIN GUISEL	2009-06-18 Justificar

Seleccionar el estudiante del que se desea justificar la falta; ingresar la información que se solicita.



[Atrás](#) [Menú](#)

JUSTIFICACION DE FALTAS

Estudiante:	ANRANGO SANDOVAL EVELIN GUISSOLA
Curso:	2 SOCIALES A
Fecha:	2009-06-17
Motivo:	Enfermo, adjunta certificado médico
Representante:	SPA. EMILIA SANDOVAL

Luego de completar la información se imprime un certificado de justificación como se indica en la siguiente figura.

COLEGIO NACIONAL "28 DE SEPTIEMBRE"

FORMULARIO DE JUSTIFICACION

El Sr.(a) SRA. EMILIA SANDOVAL, concurió a este establecimiento para justificar la falta del/a estudiante ANRANGO SANDOVAL EVELIN GUISSOLA del 2 SOCIALES A correspondiente al día 17 de junio del 2009, por cuanto el estudiante puede presentar este justificativo para ingresar a clases.

REPRESENTATE

INSPECTOR

[Menú](#)

- **Reporte de Inasistencia Docente.-** esta opción está en el menú principal y le permite obtener un reporte de los estudiantes que han faltado en un tiempo determinado. Para esto debe seguir estos pasos:
 - Ingresar desde que fecha y hasta que fecha desea buscar en los casilleros correspondientes y luego hacer clic en Buscar y le mostrará una lista de los

estudiantes que han faltado ordenado por fecha. Este reporte lo puede imprimir haciendo clic en el botón correspondiente.

REPORTE DE INASISTENCIA DOCENTE


Desde:	01/03/2010
Hasta:	20/03/2010

Fecha	Nº Hora	Curso	Docente	Estado
2010-03-01	1	2 SOCIALES	ARTEAGA OMAR REMIGIO	I
2010-03-01	2	2 SOCIALES	REVELO NELLY MARIANA DE JESUS	I
2010-03-03	5	2 SOCIALES	REVELO NELLY MARIANA DE JESUS	I
2010-03-06	7	2 SOCIALES	ARCINIEGA MAIRENE	I
2010-03-06	8	2 SOCIALES	ARCINIEGA MAIRENE	I
2010-03-09	5	2 SOCIALES	MORAN ROSARIO MAGDALENA	J
2010-03-12	2	9 EDUCACION BASICA	ARCINIEGA MAIRENE	I
2010-03-12	7	2 SOCIALES	ARCINIEGA MAIRENE	I
2010-03-12	8	2 SOCIALES	ARCINIEGA MAIRENE	I
2010-03-13	1	1 SOCIALES	ARTEAGA OMAR REMIGIO	I
2010-03-13	6	2 SOCIALES	REVELO NELLY MARIANA DE JESUS	I
2010-03-13	7	2 SOCIALES	NUÑEZ BERTHA ESPERANZA	I
2010-03-13	8	2 SOCIALES	NUÑEZ BERTHA ESPERANZA	J

ANEXO C

APLICACIÓN CON STRUTS

HERRAMIENTAS UTILIZADAS

 Base de datos: PostgreSQL

 IDE: JDeveloper

DESARROLLO DE LA APLICACIÓN

Para realizar la comparación entre los Frameworks mencionados anteriormente se desarrolló una pequeña aplicación con Struts; que consiste en que un usuario puede registrarse en el sistema y votar por su equipo de fútbol favorito.

Formulario de Registro de usuarios

En este formulario el usuario debe registrarse ingresando sus datos personales como son: email, nombre y apellido, estos tres campos son obligatorios; si uno de estos campos no son ingresados el sistema mostrará un mensaje de error, indicando que estos datos son obligatorios, como se indica en la siguiente figura.

REGISTRO DE USU@RIOS

em@:	dgabyts@hotmail.com
Nombre:	Gabriela
Apellido:	
<input type="button" value="Enviar"/>	

Errores de validación

Por favor verifica los siguientes errores para continuar...

- El campo no debe ser nulo

Si el usuario ha ingresado correctamente los datos podrá acceder a la página de votación y seleccionar su equipo favorito.

AHORA PUEDES VOTAR POR TU EQUIPO FAVORITO!!!

<input type="radio"/> EMELEC
<input type="radio"/> BARCELONA
<input type="radio"/> DEPORTIVO QUITO
<input type="radio"/> NACIONAL
<input type="radio"/> ESPOLI
<input type="radio"/> DEPORTIVO CUENCA
<input type="radio"/> AUCAS
<input checked="" type="radio"/> LDU
<input type="button" value="Aceptar"/>

GLOSARIO

ADSL.- línea asimétrica de suscripción digital. Es una tecnología que permite transmitir información digital con elevado ancho de banda sobre líneas telefónicas. Permite conectarse a internet sin interferir en las llamadas telefónicas de la línea que utiliza.

API.- Interfaz de Programación de Aplicaciones. Representa una interfaz de comunicación entre componentes de software.

Backbone.- es un mecanismo de conectividad primario en un sistema distribuido. Todos los sistemas que tengan conexión al backbone pueden interconectarse entre sí, aunque también puedan hacerlo directamente o mediante redes alternativas.

CERN.- Organización Europea para la investigación nuclear.

CGI.- Common Gateway Interface es una tecnología de la www que permite al navegador web solicitar datos de un programa ejecutado en un servidor web.

Conmutación.- Cambiar una conexión específica o el control de una determinada operación.

DARPA.- Agencia de Investigación de Proyectos Avanzado de Defensa, responsable del desarrollo de nuevas tecnologías para uso militar.

GUI.- Interfaz Gráfica de Usuario, utiliza un conjunto de objetos gráficos para representar información y acciones disponibles en una interfaz.

Hibernate.- es una potente herramienta de Mapeo Objeto – Relacional para plataforma JAVA, que facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación.

Hipermedia.- Término que hace referencia al conjunto de métodos para escribir, diseñar, o componer contenidos que tengan texto, video, audio, mapas, etc. y que poseen interactividad con los usuarios.

Hipertexto.- Documento que reúne imágenes, textos, sonidos o vídeos relacionados entre sí por medio de enlaces. La World Wide Web es una forma de usar la Internet por medio de hipertextos conectados entre sí.

HQL.- Hibernate Query Language o Lenguaje de Consultas Hibernate.

HTML.- Lenguaje de marcas de hipertexto para el desarrollo de páginas Web, describe la estructura y el contenido en forma de texto, así como para complementar el texto con objetos como imágenes.

HTTP.- Protocolo de transferencia de hipertexto.

IP.- serie de números asociados a un computador con el cual es posible identificarlo dentro de una red.

JSP.- JavaServer Pages o Páginas de Servidor Java, es una tecnología Java que permite generar contenido dinámico para la Web.

LGPL.- Licencia Pública General Mínima de GNU, es una licencia de software creada por Free Software Foundation, pretende garantizar la libertad de compartir y modificar el software libre y asegurar que el software sea libre para todos los usuarios.

POO.- Programación Orientada a Objetos, este tipo de programación usa objetos y sus interacciones para diseñar software. Incluye varias técnicas como herencia, polimorfismo, encapsulamiento entre otros. La mayoría de lenguajes de programación soportan este tipo de programación.

Script.- son fragmentos de código Java contenidos en una página JSP.

Servlet.- son componentes del servidor y pueden ser ejecutados en cualquier plataforma. Se cargan de forma dinámica por el entorno de ejecución Java del servidor cuando son invocados. Cuando se recibe una petición del cliente el servidor web inicia el servlet solicitado; este procesa la petición y envía la respuesta de vuelta al servidor web para que sea enviada al cliente.

URI.- Localizador Uniforme de Recursos; es un conjunto de caracteres que siguen un formato y es usado para nombrar recursos, como documentos e imágenes en Internet.

URL.- Localizador Uniforme de Recursos. En Internet estos recursos pueden ser imágenes, documentos de texto, portales Web, sitios FTP, archivos de audio, etc.

Sílabo.- Consiste en un plan de trabajo para un curso concreto, descripción de los objetivos, secuenciación de los contenidos y de las actividades, especificación de la metodología y forma de evaluación. Traslada la filosofía del currículo a un plan detallado de enseñanza.

BIBLIOGRAFÍA

PAGINAS DE INTERNET

- [www 01] http://www.bab-soft.com/es/disenio_desarrollo_aplicaciones_web.php
- [www 02] <http://csharp.ikor.org>
- [www 03] livedocs.adobe.com/.../8_es/using/gs_12_u6.htm
- [www 04] <http://jcp.org/en/home/index>
- [www 05] http://es.wikipedia.org/wiki/M%C3%A1quina_virtual_Java
- [www 06] http://es.wikipedia.org/wiki/Plataforma_Java
- [www 07] <http://java.sun.com/javaee/index.jsp>
- [www 08] http://es.wikipedia.org/wiki/patron_de_diseño
- [www 09] http://es.wikipedia.org/wiki/Modelo_Vista_Controlador
- [www 10] http://www.javahispano.org/contenidos/es/comparativa_de_frameworks_web/
- [www 11] <http://iie.fing.edu.uy/~nacho/publicaciones/clasiliptus.pdf>
- [www 12] <http://150.185.75.30/atiwiki/index.php/STRUTS>
- [www 13] http://jsf_struts
- [www 14] http://www.programacion.net/tutorial/joa_struts
- [www 15] www.slideshare.net/siis/struts-en-java
- [www 16] <http://jakarta.apache.org/tapestry>
- [www 17] <http://seamcity.madeinxpain.com/archives/comparativa-tapestry>
- [www 18] http://www.programacion.net/java/tutorial/jsf_intro
- [www 19] <http://www.oracle.com/technology/products/jdev/101/howtos/myfaces>
- [www 20] <http://theopensource.com/javajsf.htm>
- [www 21] <http://amap.cantabria.es/confluence/display/BASE/JSF>
- [www 22] <http://www.dsc.ufcg.edu.br/~jacques/cursos/daca/html/jsf/jsf.htm>
- [www 23] <http://ame.endesa.es/comfluense/AME>
- [www 24] <http://java.sun.com/j2ee/javaserverfaces/docs/tutorial.html>
- [www 25] <http://docs.sun.com/app/docs/doc/820-7426/abpaj?a=view>
- [www 26] <http://postgresql.uptodown.com/>
- [www 27] http://www.javahispano.org/contenidos/es/manual_hibernate/
- [www 28] <http://es.wikipedia.org/wiki/Hibernate>
- [www 29] <http://www.ingenierosoftware.com/analisisydiseno/uml.php>