

Universidad Autónoma Metropolitana – Azcapotzalco
División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación

**Automatización de procesos para generar reportes de
requerimientos en formato PDF.**

Modalidad: Proyecto Tecnológico

Trimestre 2017 Primavera

Nombre del alumno: Juan Manuel De Lucio Ramírez

Matrícula: 2122001523

Correo: al2122001523@alumnos.azc.uam.mx

Asesor: José Alejandro Reyes Ortiz

Nivel académico: Doctor en Ciencias de la Computación

Categoría: Titular

Departamento de Sistemas

Correo: jaro@correo.azc.uam.mx

Coasesor: Leonardo Daniel Sánchez Martínez

Nivel académico: Doctor en Ciencias y Tecnologías de la Información

Categoría: Asociado

Departamento de Sistemas

Correo: lds@correo.azc.uam.mx

20 de Julio de 2017

Declaratoria

Yo, José Alejandro Reyes Ortiz, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Dr. José Alejandro Reyes Ortiz

Yo, Leonardo Daniel Sánchez Martínez, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Dr. Leonardo Daniel Sánchez Martínez

Yo, Juan Manuel De Lucio Ramírez, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Juan Manuel De Lucio Ramírez

Resumen

Los sistemas de información (SI) son herramientas que nos brindan apoyo para hacer consultas, desarrollar una investigación o tareas, cuando necesitamos documentación de los programas, de alguna empresa, de trabajos, escuela, periódicos, centros comerciales, etc.

Estos sistemas evolucionan a la par de las nuevas tecnologías de comunicación, aunado a esto los sistemas con los que se trabajan se están haciendo cada vez más rigurosos a la hora de solicitar información de los usuarios.

Los SI nos brindan una cantidad de información muy importante, así como también muy extensa, ya que cada sistema creado tiene información replicada con uno o varios SI ya desarrollados y en estos momentos se nos hace difícil confiar en la veracidad de los datos.

Es por eso que es preferible insertar los datos, guardarlos en un medio persistente de almacenamiento y después acceder a ellos de manera remota y generar un recurso para tenerlo de manera física y almacenar solamente la información esencial para no tener información replicada, ya que con esto facilitamos el manejo de la información y se tiene un mejor control de ella, así como también nuestro sistema se incrementa en velocidad y almacenamiento.

Se desarrolló un sistema de información con el patrón de diseño modelo vista controlador, ya que facilita separar el sistema en capas para acceder a ellas cuando lo necesitemos de manera independiente. La división del sistema en capas facilita la modificación de cada una de estas y no afecta al resto de ellas, lo que permitió agilizar el desarrollo del mismo.

La implementación de la tecnología J2EE fue necesaria para la creación de este sistema ya que nos permitió separar en capas y que los procesos sigan un camino recto siempre y cuando cada uno de ellos se comunique con el siguiente o con el anterior más próximo, con esta implementación es más fácil que los procesos no se estanquen y que fluyan con mayor libertad.

Para gestionar la Base de Datos fue necesario usar MySQL Workbench y es el que nos permitió conectarnos desde el entorno de desarrollo NetBeans IDE 8.1 para poder acceder al guardar, buscar y recuperar información una vez insertada por el usuario y con esto no se estuviera replicando los datos de los usuarios y poniendo en peligro la integridad de la información de los mismos, esto quiere decir que una vez guardados los datos se acceden a ella desde los módulos creados para su creación, modificación y eliminación de los recursos PDF que necesitamos.

Al implementarse en el servidor se logró reducir el tiempo de creado de los oficios, generándolos con una alta efectividad y con esto reduciendo el tiempo de espera que se tiene en estos momentos.

Contenido

Introducción	1
Antecedentes	3
Proyectos Terminales	3
Manual.....	3
Tesis	4
Justificación	5
Objetivos	6
Objetivo general.....	6
Objetivos específicos	6
Marco Teórico	7
Sistemas de información (SI).....	7
Base de Datos (BD) y Sistema de Gestión de Base de datos (SGBD)	10
Modelo Cliente-Servidor	11
Arquitectura Modelo Vista Controlador	12
Tecnología J2EE	12
1. Servlet	13
2. Java Server Pages (JSP)	13
3. Java DataBase Connectivity(JDBC).....	14
Desarrollo del Proyecto	15
Diagrama de Módulos.....	15
Diagrama de Casos de Uso.....	18
Diagrama de Secuencia	23
Diagrama de Clases Conceptuales	24
Diagrama de Robustez	25
Diagrama de la Base de Datos	26
Resultados	27
Casos de uso 1).....	27
Casos de uso 2).....	32
Casos de uso 4).....	37
Casos de uso 3).....	41
Resultados de los oficios Credos.....	45
Análisis y Discusión de resultados	48

Conclusiones	49
Referencias bibliográficas	50
Anexos	52
Anexo1	52
Anexo2	53
Anexo3	54
Anexo4	55
Anexo5	57
Anexo6	59
Anexo7	62
Anexo8	63
Anexo9	64
Anexo10	65
Anexo11	66

Introducción

El manejo de información es un tema bastante delicado dentro de cualquier institución. La alta tasa de generación de datos, exige mecanismos adecuados para la administración de la misma, como el uso de nuevas tecnologías digitales, así como la implementación de aplicaciones.

En el caso de la Universidad Autónoma Metropolitana unidad Azcapotzalco (UAM-A) el uso que se le da a las nuevas tecnologías es escaso y carece de una implementación para el trabajo que se realiza día a día en la institución. Con el propósito de llevar una mejor administración de los reportes solicitados y disminuir la replicación de la información que existe en la UAM-A, así como para brindar apoyo al Área de Investigación de Sistemas de Información Inteligentes (AISII), se desarrollará un proyecto para la elaboración de reportes administrativos.

Dicho sistema considera un servidor web para que el usuario pueda llenar con datos los distintos campos programados en él. Dichos datos llegarán a una aplicación para su almacenamiento y generación de reportes en el formato solicitado. Posteriormente, se regresará el formato creado a la aplicación y esta se encargará de guardarlo en la base de datos. El reporte generado se enviará al usuario para la revisión, verificación e impresión del mismo.

La base de datos (BD) contendrá la información de cada reporte generado, así como los datos brindados por el usuario para que pueda modificar los documentos ya existentes por cualquier error que se haya cometido. En este sentido, el sistema aquí propuesto siempre estará disponible y permitirá almacenar y recuperar información relacionada a los reportes en la base de datos.

Para el desarrollo del Sistema de Información se ocupó el modelo vista controlador ya que la facilidad de separar estas tres capas y acceder a ellas cuando lo necesitemos esto agilizará la forma de consultar, modificar y eliminar la información. Se controla el manejo de los accesos y esto hace que el sistema se vuelva dinámico al gestionar esos procesos y responder en milésimas de segundo y así evitar que el sistema se estanque.

La vista y los controladores están relacionados directamente, al controlar los eventos directamente producidos por la interfaz gráfica (vista). Con esta separación de aspectos la aplicación da muchas características como transparencia, escalabilidad, flexibilidad y seguridad.

El sistema cuenta con la propiedad de ser transparente al hacer pensar al usuario que todos los recursos del sistema son gestionados por una sola máquina.

Es escalable al adaptarse al sistema sin perder rendimiento y al estar en crecimiento continuo de manera fluida sin perder la estabilidad de los servicios ofrecidos.

Es flexible al tener muchas opciones para ampliarlo con nuevas funcionalidades de manera sencilla y es seguro al manejar y gestionar la información de los usuarios que acceden al sistema, con la implementación de un inicio de sesión para poder generar sus documentos.

También se implementó la arquitectura cliente-servidor con la cual tiene como gestor a un cliente haciendo peticiones al servidor y este le responde con servicios que este contiene. En nuestro sistema el cliente accede al este cuando inicia su sesión y al querer realizar un oficio y el servidor le envía un formulario con los datos y con cajas de texto para que el usuario las llene.

La implementación de la tecnología J2EE fue necesaria para la creación de este sistema ya que nos permitió separar en capas y que los procesos sigan un camino recto siempre y cuando cada uno de ellos se comuniquen con el siguiente o con el anterior más próximo, con esta implementación es más fácil que los procesos no se estanquen y que fluyan con mayor libertad.

Con este tipo de procesos, la intención que se tiene es crear nuevos documentos a partir de la información existente, generar reportes a la demanda y capturar o modificar información para usos posteriores.

Antecedentes

Proyectos Terminales

1. Replicador de configuración y estado de un manejador de base de datos relacional [1].

Se utiliza una base de datos y con ella se crean PDF's para visualizar la información recopilada de la base de datos. La diferencia consiste en el uso que se les va a dar a la base de datos, en la propuesta se maneja una base de datos que va a estar relacionada con los distintos módulos creados, en la BD se guardarán los PDF's generados, de igual manera las cadenas que el usuario brindo al servidor web para generar su reporte de requerimientos.

2. Servicios Web para la gestión de conocimientos usado un modelo ontológico [2].

En esta propuesta se presentas distintos módulos que son parecidos al proyecto como son el módulo de búsqueda y el módulo de invocación de servicio web, la diferencia encontrada es el uso de una base de datos donde se guardarán los reportes, así como también la información brindada por el usuario, donde en esta propuesta uso una colección de datos en un modelo ontológico.

3. Gestión de calificaciones de cursos mediante servicios Web [3].

En este proyecto se plantea realizar un servicio web que tenga diversas funciones para realizar las tareas y se le facilite el uso y creación de "instrumentos de evaluación" esta parte es similar a la propuesta, la diferencia está en la unión del servicio web con una aplicación que va a gestionar las conexiones con una base de datos y con un programa externo para generarme un recurso PDF y que al final lo podamos observar en el servicio web.

Manual

4. Creación de Ficheros LATEX con GNU Emacs [4].

En este documento se plasma la idea de crear un entorno de desarrollo lo que es unir varias herramientas en una sola para mejorar el rendimiento del programa esto es similar a la propuesta presentada ya que en los diferentes módulos se tienen partes de un entorno de desarrollo, la diferencia se encuentra en que en la propuesta ya se tiene un formato solicitado que es el que se le envía al programa LATEX y nada más se modifican pequeñas partes.

Tesis

5. Implementación de un Modelo de Base de Datos para el Almacenamiento y Consulta de Electrocardiogramas Digitales [5].

Se crean bases de datos para guardar información como también consultarla en plataforma dentro de un lugar (hospital, institución, etc.), la diferencia que se pudo encontrar es el formato en el que se va a guardar la información presente en la propuesta se plantea guardar archivos PDF como también cadenas de caracteres para su uso posterior.

6. Implementación de controles de seguridad en arquitecturas orientadas a servicios (SOA) para servicios web [6].

Se implementa un sistema para solucionar problemas relacionados con la seguridad en los servicios web al intercambiar información entre dos o más aplicaciones es algo en común con la propuesta, la diferencia se encuentra en los protocolos que la tesis utilizó para mantener la seguridad de la información. En la propuesta se manejará un módulo de seguridad con identificadores que se va a relacionar con una base de datos interna de la institución para poder acceder al servicio web.

Justificación

La cantidad de datos que se manejan en los sistemas de información, bases de datos, y servidores es inmensurable. Esto genera problemas de replicación de información innecesaria y problemas de filtraciones de datos a terceros. Esto obliga al uso de tecnologías cada vez más costosas para poder satisfacer las necesidades de los usuarios, lo que no resulta rentable ni concebible para muchas instituciones.

Actualmente en la UAM-A, los profesores y el personal administrativo requieren material para satisfacer el trabajo que se realiza en la institución. Para esto se requiere del llenado de formas, comúnmente a mano, lo que implica un desperdicio de tiempo y recursos por parte del profesor y la UAM-A. Esto quiere decir que un profesor que llena un reporte con N-Datos, está desperdiciando su tiempo en función de la cantidad de datos a llenar. Aunado a esto, la mayoría de los reportes requieren casi siempre de la misma información, lo que quiere decir que si los N campos se requieren en M-Reportes, un profesor está llenando $M \times N$ campos innecesariamente. Esto genera un desperdicio de recursos además de excesiva redundancia de datos.

Con este sistema se implementará una arquitectura en capas, con la que mantendremos separadas la capa de almacenamiento de datos, la lógica de negocio y la de presentación, brindando escalabilidad, flexibilidad, fiabilidad y seguridad, con esto la intención que se tiene es la reducción del tiempo al generar estos reportes y automatizar la forma de hacerlos.

Contaremos con versatilidad en la presentación de los datos por que se tienen múltiples vistas para los mismos datos además de los distintos tipos de acceso.

Objetivos

Objetivo general

Desarrollar una aplicación para generar los reportes de requerimientos utilizados por las diversas áreas del departamento de sistemas de la UAM-A.

Objetivos específicos

1. Diseñar la BD que brinde soporte al sistema.
2. Desarrollar e implementar el módulo para la administración de la BD.
3. Desarrollar e implementar el módulo para la administración de reportes de requerimientos.
4. Evaluar el sistema mediante la comparación de tiempos entre el sistema anterior y el propuesto.

Marco Teórico

Sistemas de información (SI)

En nuestros días los sistemas de información son una parte importante de la vida cotidiana de los seres humanos, al consultar diversas plataformas como son la de la escuela, la oficina o las empresas, los centros comerciales, periódicos, librerías, etc. La razón de esta facilidad radica en el crecimiento de la tecnología, pues basta con tener una computadora, un Tablet o un Smartphone con acceso a Internet a la mano para poder acceder a las herramientas que estos sistemas nos ofrecen.

El principal motivo de estos sistemas es el recibir y procesar los datos para transfórmalos en información y usarla con un sistema que pueda contenerla información y que todo el mundo esté al tanto de esta.

Tomando como referencia “Los sistemas de información: evolución y desarrollo, lo cual dice que los sistemas de información; utilizan como materia prima los datos, los cuales procesan y transmiten para obtener como resultado final información, la cual será suministrada a los diferentes usuarios del sistema, existiendo además un proceso de retroalimentación en la cual se ha de valorar si la información obtenida se adecua a lo esperado” [7]. Con esta definición podemos observar que la replicación de la información se presentará en cualquier sistema de información creado.

Estos sistemas cada vez son más utilizados por las personas, ya que gracias a esta tecnología logran hacer un negocio y logran perfeccionarlo a tal punto de volverse importantes en este ámbito. ¿Pero a qué se debe su rotundo éxito?. Tomando en cuenta el periódico el Universal [8], con las cifras oficiales de Internet Society el 94% de la población mundial tiene acceso a un teléfono móvil, y una gran parte de la población cuenta con servicio de Internet en su Smartphone, esto hace que la consulta de información sea habitual y en gran parte se necesaria para completar diversas tareas que nos proponemos realizar o que estamos realizando.

De los trabajos más reconocidos en nuestros días sobre la evolución de los sistemas de información se encuentra el realizado por Gibson & Nolan en 1974, (77-87).

Etapa	Características
Iniciación	<ul style="list-style-type: none"> ➤ Adquiere las primeras computadoras. Se ponen sistemas transaccionales. ➤ El departamento de Sistemas depende casi siempre del área de contabilidad. ➤ No hay administración informática y la que hay se hace por personal no experto. ➤ El personal es máximo de un operador y/o un programador y su régimen de contratación es por horas. ➤ Se pone en funcionamiento el primer SI después de varios intentos.
Contagio o Expansión	<ul style="list-style-type: none"> ➤ Se pone en producción exitosamente el primer SI y se entra a desarrollar otros SI, generalmente operando cada sistema independientemente y de tipo TSP. ➤ Se independiza el departamento de Sistemas y adquiere mayor categoría aunque depende de otra dependencia. ➤ Se contratan personal especialista preparación académica en el área de sistemas tales como analista de sistemas, analista-programador, programador de sistemas, jefe de desarrollo, jefe de soporte técnico, etc. ➤ La inversión en SI comienza a ser importante con lo que se vuelve imperativos racionalización de los recursos Informáticos.
Control o Formalización	<ul style="list-style-type: none"> ➤ Se formaliza la racionalización del recurso informático para hacer más eficiente lo invertido. ➤ El departamento de SI generalmente alcanza el nivel gerencial, dependiendo organizacionalmente de la dirección administrativa o de la de finanzas. ➤ Empieza a crecer la demanda de nuevos SI y la administración se orienta a control administrativo y a la justificación económica. ➤ Comienza la estandarización en el desarrollo, con lo que se inicia el desarrollo de interfaces para la integración y automatización de procesos. Se requiera de personal para administrar el SI. ➤ Aparece la función de la planeación de SI.
Integración	<ul style="list-style-type: none"> ➤ Se integran los SI en la organización. ➤ Los nuevos desarrollos tecnológicos permite que los usuarios sean más autónomos en su trabajo. ➤ Disminuyen los costos de los recursos informáticos. ➤ Los mismos usuarios pueden comenzar a desarrollar sus propios sistemas con lo que se descentralizada el departamento de SI.
Administración de Datos	<ul style="list-style-type: none"> ➤ Se comienza reconocer la información como un bien muy valioso. ➤ El compartir la información es de vital importancia. ➤ El usuario es el responsabilidad de la integridad de la Información y debe haber acceso diferenciador a los datos.
Madurez	<ul style="list-style-type: none"> ➤ La Informática dentro de la organización es una función básica y se debe gerencia a más alto nivel de dirección. ➤ Los SI se vuelven Sistemas Basados en el Conocimiento y Sistemas Expertos, se logra la integración por redes de datos. ➤ Se pulen muchos de los controles implementados en las etapas anteriores, siendo menos rígidos en la aplicación de los mismos. ➤ Planeación rigurosa con horizontes menos a 5 años.

Tabla 1. Etapas de la evolución de los sistemas de información (Gibson & Nolan, 1974, 77-87) [9].

Todo sistema de información tiene un porque. Esta es una de las cuestiones que todos queremos saber, por qué se realizó o por qué lo hicieron así, la gran mayoría depende del correcto desarrollo y funcionamiento del sistema y por ello resulta fundamental las etapas del desarrollo de los SI y este consta de siete etapas definidas por Hernández Trasobares Alejandro [7].

1. **Definición del proyecto:** en esta etapa se define un proyecto en el cual se satisface un problema con el uso de sistemas de información.
2. **Análisis del sistema:** en esta parte se cuantifica que tan fácil será desarrollar el sistema para darle una rápida solución a los problemas planteados.
3. **Diseño del sistema:** en esta etapa se eligieron las soluciones del problema y a la hora de diseñar se eligen los componentes del sistema hardware y software y la relación que existe entre ellos.
4. **Programación:** en esta etapa se desarrolla el proyecto a nivel software.
5. **Fases de prueba:** en esta parte se evalúa el sistema haciendo que cumpla con las especificaciones funcionales definidas en los anteriores pasos, así como también probarlo en diversas condiciones y dar un veredicto. En esta etapa hay tres tipos de pruebas.
 - i) Pruebas de programa.
 - ii) Pruebas de sistema.
 - iii) Pruebas de aceptación.
6. **Conversión:** ya cumplido el paso anterior y se haya aceptado el sistema se lleva a cabo este paso y se implementa el sistema o se reemplaza.
7. **Producción y mantenimiento:** ya cumplido el paso anterior y estando el sistema en línea se continuará con una serie de evaluaciones por parte del personal administrativo.

Con estas siete etapas del desarrollo de los sistemas de información nos damos una idea de cómo deben de ser y con esto brindarle confianza a la empresa o el trabajo que vayamos a desarrollar para ellos.

Base de Datos (BD) y Sistema de Gestión de Base de datos (SGBD)

Una BD es una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos, tal y como se muestra en la Figura 1 [10].

Esquema General de Uso de una Base de Datos

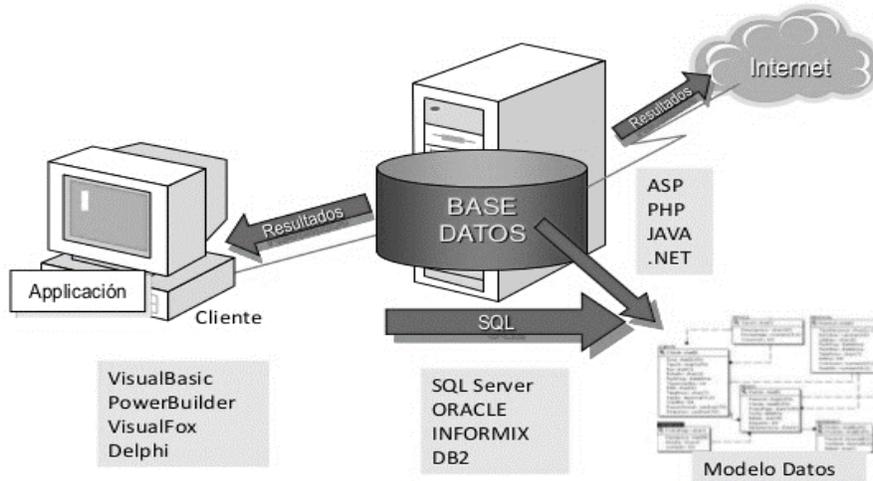


Figura 1. Sistema de una Base de datos

Conforme aumenta el almacenamiento de los Datos guardados en la BD su complejidad para consultar esos datos también aumenta, por esta razón es más factible consultar los datos de una manera diferente.

Un Sistema de gestión de base de datos resuelve este problema al gestionar los datos con una aplicación que se interpone entre el usuario y los archivos al hacerse cargo de la explotación, mantenimiento y comprobación de los datos, tal y como se muestra en la Figura 2 [11].

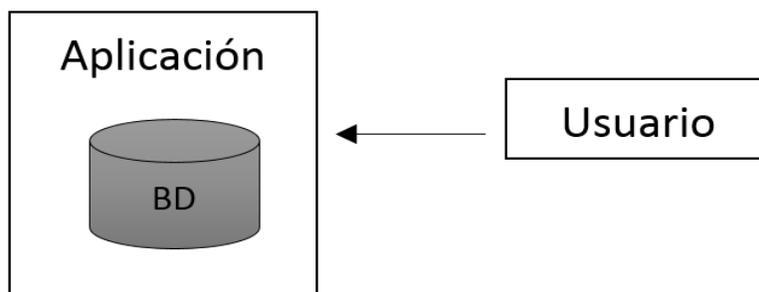


Figura 2. Sistema de Gestión de Base de Datos

Modelo Cliente-Servidor

El modelo cliente-servidor se base en uno o varios clientes extendiendo peticiones a uno o varios servidores con el fin de acceder a un recurso.

El cliente es un proceso que solicita un servicio a un servidor, enviándole una petición y subsecuentemente esperando la respuesta del servidor.

El servidor es una aplicación (Software) en ejecución que está a la espera de peticiones y envía respuestas a uno o varios clientes.

El socket es aquel con el cual dos programas pueden intercambiar cualquier flujo de datos generalmente de manera fiable y ordenada a través de una red de datos como internet [12].

Existen dos tipos de servidores los cuales son:

- El servidor Secuencial
 - i) Procesan una sola solicitud cada vez.
 - ii) Las peticiones que llegan se van almacenando en una cola y se atiende en cuanto termine con la que está procesando.
- El servidor Concurrente
 - i) Un servidor concurrente atiende a varios clientes al mismo tiempo.
 - ii) Más aún, mientras está atendiendo sigue escuchando
 - iii) El problema es que todo cliente tiene que esperar su turno para ser atendido.
 - iv) Si uno de ellos pide un archivo muy grande los demás tienen que esperar.
 - v) La mayor parte de la espera es debido a operaciones de IO, hay capacidad de CPU ociosa.
 - vi) Se trata de crear un nuevo proceso o línea de ejecución cada vez que un cliente “llega” a pedir un servicio.

Implementación en tres capas los componentes están organizados en forma de capas, en la que un componente en una determinada capa puede llamar a componentes en la capa inmediata inferior. Una observación clave es que el control generalmente fluye de capa en capa: las peticiones van de arriba abajo y los resultados de abajo a arriba.

Hay tres capas:

- 1) Capa de interfaz del usuario es la contiene todo lo necesario para establecer una interfaz directa con el usuario, tal como la administración del despliegue de la información.
- 2) Capa de procesamiento es la que contiene las aplicaciones de lado del servidor.
- 3) Capa de datos es la que administra los datos sobre los cuales se está trabajando.

Arquitectura Modelo Vista Controlador

La implementación de este modelo es sumamente importante, ya que con esto separas las aplicaciones con las que trabajas y se hace eficiente el sistema.

El modelo es la parte del sistema que contiene los datos que se encuentran almacenados en la base de datos o en el XML.

La vista es la que gestiona el cómo se van a mostrar esos datos, hace referencia al diseño de las paginas HTML, JSP o PHP.

El controlador son las funcionalidades o reglas con las que vas a trabajar para transformar los datos, así como también determina que modificaciones se tienen que hacer a los datos del modelo cuando se interrelaciona con la vista, también es el que gestiona las vistas del sistema.

Ventajas

- Es posible tener diferentes vistas para el mismo modelo.
- Es posible construir nuevas vistas sin necesidad de modificar el modelo subyacente.
- Proporciona un mecanismo de configuración para componentes complejos mucho más tratables que el puramente basado en eventos.

Tecnología J2EE

Hace referencia a Java Platform, Java Enterprise Edition o a Java EE y es originario del lenguaje Java, este permite arquitecturas de N capas distribuidas y se apoya ampliamente en módulos ejecutándose sobre un servidor de aplicaciones.

El lenguaje Java hace posible que los programadores y desarrolladores puedan escribir, crear y desarrollar aplicaciones todo en un mismo entorno y con la posibilidad de que sea compatible con los equipos y los sistemas operativos donde se está trabajando.

Con el paso del tiempo las empresas pedían tecnologías compatibles con sus sistemas y que hasta ese momento no se tenía, se hizo necesario la creación de herramientas y medios que permitieran el desarrollo de software específico para el sector empresarial, fue así como nació Java 2 Enterprise Edition (J2EE) lenguaje de programación con el potencial suficiente para aportar las técnicas y herramientas que permitan implementar productos de software, es decir, aplicaciones, basadas en unas especificaciones técnicas determinadas.

J2EE incluye diferentes tecnologías como son Enterprise JavaBeans (EJB), Java Servlet, JavaServer Page (JSP), JavaServer Pages Standard Tag Library (JSTL), JavaServer Faces (JSF), Java Message Service (JMS), Java Transaction API (JTA), JavaMail API y JavaBeans Activation Framework (JAF).

1. Servlet

Es una clase escrita en lenguaje Java, las cuales se utilizan para ampliar las capacidades de un servidor. Aunque los servlets pueden responder a cualquier tipo de solicitudes, éstos son utilizados comúnmente para extender las aplicaciones alojadas por servidores web, de tal manera que pueden ser vistos como applets de Java que se ejecutan en servidores en vez de navegadores web. Este tipo de servlets son la contraparte Java de otras tecnologías de contenido dinámico Web, como PHP y ASP.NET.

La palabra servlet deriva de otra anterior, applet, que se refiere a pequeños programas que se ejecutan en el contexto de otro programa.

El uso más común de los servlets es generar páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web [13].

Beneficios de usar Servlet:

- Utiliza menos recursos y son más eficientes, utilización de hilos esto quiere decir que por cada petición crea un proceso ligero para atenderlo, lo cual reduce el uso de la memoria del servidor y del tiempo de respuesta.
- Tiene persistencia esto quiere decir que preserva la información de un objeto de forma permanente, por lo que siguen vivos los procesos una vez terminada la petición.

2. Java Server Pages (JSP)

Es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML y XML, entre otros tipos de documentos. JSP es similar a PHP, pero usa el lenguaje de programación Java.

El rendimiento de una página JSP es el mismo que tendría el servlet equivalente, ya que el código es compilado como cualquier otra clase Java. A su vez, la máquina virtual compilará dinámicamente a código de máquina las partes de la aplicación que lo requieran. Esto hace que JSP tenga un buen desempeño y sea más eficiente que otras tecnologías web que ejecutan el código de una manera puramente interpretada.

Ventajas de usar JSP:

- Es apto para crear clases que manejan lógica de negocios y acceso a datos de una manera detenida y con alto grado de detalle, esto permite separa en niveles las aplicaciones web.
- Herencia de la portabilidad de Java y es posible ejecutar las aplicaciones de diferentes plataformas sin cambios.

Los servlets y Java Server Pages (JSPs) son dos métodos de creación de páginas web dinámicas en servidor usando el lenguaje Java. En ese sentido son similares a otros métodos o lenguajes tales como el PHP, ASP o los CGIs, programas que generan páginas web en el servidor. Sin embargo, se diferencian de ellos en otras cosas.

Para empezar, los JSPs y servlets se ejecutan en una máquina virtual Java, lo cual permite que, en principio, se puedan usar en cualquier tipo de ordenador, siempre que exista una máquina virtual Java para él. Cada servlet (o JSP, a partir de ahora lo usaremos de forma indistinta) se ejecuta en su propio hilo, es decir, en su propio contexto; pero no se comienza a ejecutar cada vez que recibe una petición, sino que persiste de una petición a la siguiente, de forma que no se pierde tiempo en invocarlo (cargar programa + intérprete). Su persistencia le permite también hacer una serie de cosas de forma más eficiente: conexión a bases de datos y manejo de sesiones, por ejemplo [14].

3. Java DataBase Connectivity(JDBC)

Es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

El API JDBC se presenta como una colección de interfaces Java y métodos de gestión de manejadores de conexión hacia cada modelo específico de base de datos. Un manejador de conexiones hacia un modelo de base de datos en particular es un conjunto de clases que implementan las interfaces Java y que utilizan los métodos de registro para declarar los tipos de localizadores a base de datos (URL) que pueden manejar. Para utilizar una base de datos particular, el usuario ejecuta su programa junto con la biblioteca de conexión apropiada al modelo de su base de datos, y accede a ella estableciendo una conexión; para ello provee el localizador a la base de datos y los parámetros de conexión específicos. A partir de allí puede realizar cualquier tipo de tarea con la base de datos a la que tenga permiso: consulta, actualización, creación, modificación y borrado de tablas, ejecución de procedimientos almacenados en la base de datos, etc [15].

Desarrollo del Proyecto

Diagrama de Módulos

Siete etapas para el desarrollo del proyecto:

1. La definición del proyecto se desarrolló en la propuesta del sistema donde se planteó un problema y se propone una solución con la cual se satisface de una manera eficaz.
2. El análisis del sistema a desarrollar fue evaluado con respecto a la eficiencia de creado de los reportes y al tiempo que esto tardaría generando un mejor rendimiento con respecto al sistema anterior.
3. Diseño del sistema se planifico cada una de las interfaces con el diagrama de módulos presentado en la siguiente sección.
4. La programación del sistema a desarrollar se realizó en el entorno grafico de Netbeans y con un gestor de base de datos MySQL, para cada una de las vistas se utilizó la tecnología JSP, para los controladores se utilizaron clases en Java y para los controladores se usó la tecnología servlet.
5. Para cada parte del sistema se probó diferente, para el inicio de sesión se insertaron 5 profesores en la base de datos los cuales en el sistema se probaron exitosamente, para la parte de creado de oficios se probó insertando la información para 10 reportes los cuales también se modificaron y se eliminaron exitosamente.
6. El siguiente paso es ponerlo en línea en este parte se probó con el Servidor Web Apache con el cual su función es simular un servidor externo y gestionar las peticiones de los profesores que acceden al sistema.
7. Para el último paso se siguió evaluando el sistema con una serie de pruebas de seguridad y flexibilidad.

El proyecto cuenta con distintos módulos como se puede observar en la Figura 3.

- **El módulo del servidor web:** es aquel que tiene la función de capturar la página, así como también de establecer conexión con el servidor y desplegarle el inicio de sesión al usuario.
- **El módulo de inicio de sesión:** es el encargado de capturar la información de inicio de sesión (número económico y password asignado). Citados en el anexo 1 donde se puede observar con detenimiento.
- **El módulo de validación inicio de sesión:** es el encargado de guardar los datos, compararlos en la base de datos y dar una respuesta al sistema. Citados en el anexo 2 donde se puede observar con detenimiento.
- **El módulo de administración de la base de datos** es el encargado de hacer las modificaciones directas en la BD. Citados en el anexo 2 donde se puede observar con detenimiento.

- **El módulo de validación de menú** tiene la función de generar un menú para el usuario en donde pueda crear, modificar, mostrar y eliminar formularios, así como también dirigir al usuario a los formularios. Citados en el anexo 3 donde se puede observar con detenimiento.
- **El módulo de creación de oficios** es el encargado de recabar la información necesaria para la creación de un oficio, así como también es el encargado de guardar datos y enviárselos a módulo de validación de información para su revisión. Citados en el anexo 4 donde se puede observar con detenimiento.
- **El módulo de presentación de oficios** tiene la función de recibir un parámetro el cual se lo va a envía al módulo de búsqueda de oficio, recibir un formulario para su revisión con los datos insertados por el usuario sin posibilidad de modificar y mostrar el recurso PDF en una nueva pantalla. Citados en el anexo 5 donde se puede observar con detenimiento.
- **El módulo de modificación de oficios** tiene la funcionalidad recibir un formulario para su modificación y posterior almacenamiento, el formulario contendrá los parámetros a modificar y el número de reporte, los cuales serán enviados al módulo de validación de información y la base de datos. Citados en el anexo 6 donde se puede observar con detenimiento.
- **El módulo de eliminación de oficios** es el encargado de gestionar la entrada de un parámetro el cual será enviado al módulo de búsqueda de oficio, el cual al hallarlo lo va a eliminar de la base de datos, después va actualizar la lista de los oficios existentes. Citados en el anexo 7 donde se puede observar con detenimiento.
- **El módulo de validación de información** tiene la funcionalidad de validar los datos previamente escritos en los campos diseñados y mediante un algoritmo va a aceptar o rechazar la información presentada. Citados en el anexo 8 donde se puede observar con detenimiento.
- **El módulo de búsqueda de oficios** es el encargado de recibir uno o varios parámetros los cuales va a buscarlos en la base de datos y completar la información del formulario previamente escrito. Citados en el anexo 9 donde se puede observar con detenimiento.
- **El módulo de almacenamiento** es el encargado de guardar la información del usuario y de extraer información de la base de datos. Citados en el anexo 10 donde se puede observar con detenimiento.

- **El módulo convertidor de formatos** es el encargado de recabar el código y compilarlo para poder tener el recurso PDF. Además, es el encargado de enviárselo al módulo de generación y recuperación de formatos. Citados en el anexo 11 donde se puede observar con detenimiento.

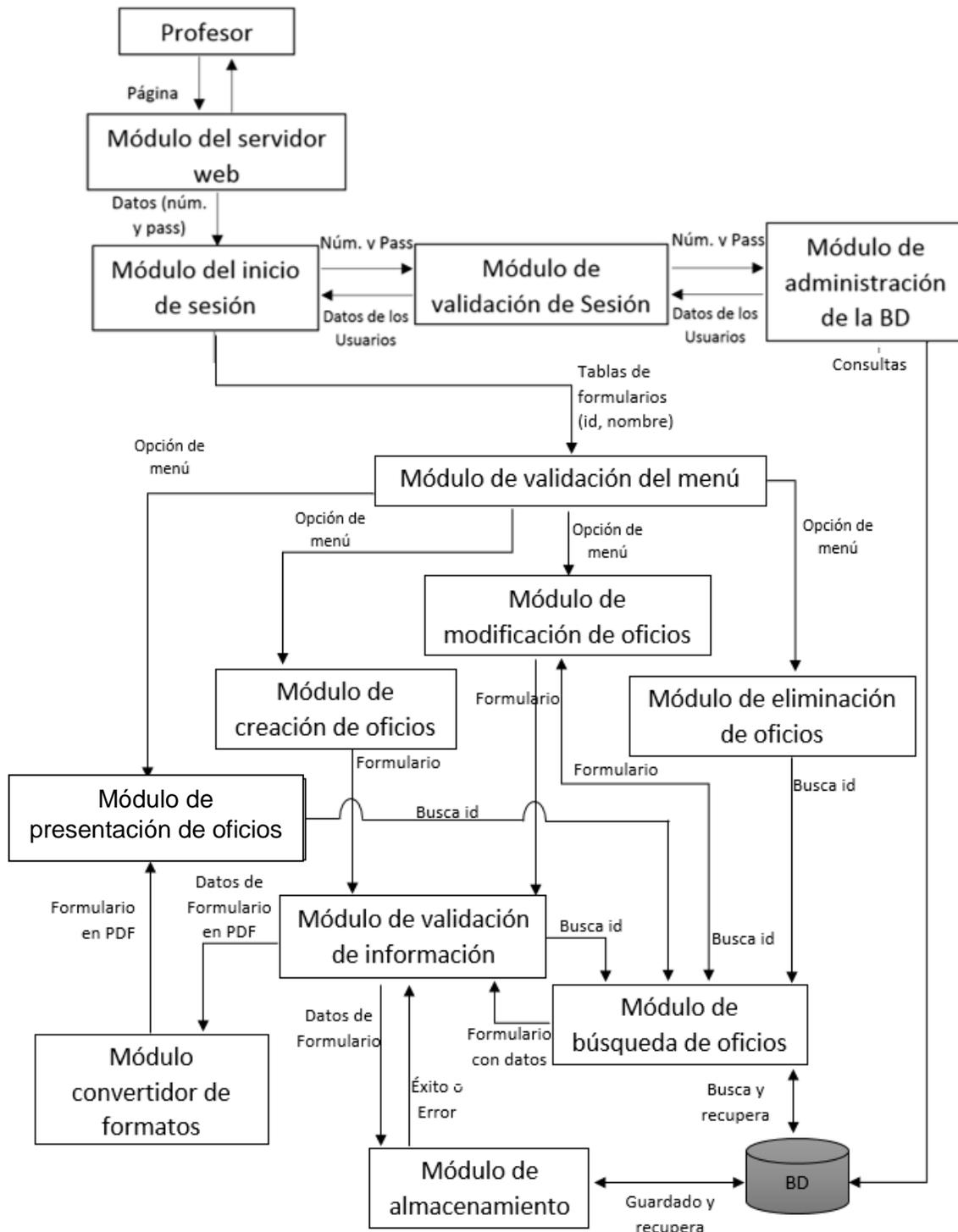


Figura 3. Diagrama del administrador de procesos.

Diagrama de Casos de Uso

Por otro lado, en la Figura 4 se muestran los diagramas de casos de uso (CU) del sistema específico para generar reportes de requerimientos, de los muchos CU que existen en esta área.

1.1) ID: CU1

1.2) Nombre: Guardado de los datos para crear el oficio.

1.3) Actor principal: Administrador o Secretaria.

1.4) Precondiciones: El administrador o la secretaria, debió (debieron) de solicitar la información que se le va a suministrar al sistema, además el usuario debe de estar logueado en el sistema.

1.5) Post-condiciones: Se registra la información del usuario en una base de datos y con la información presentada se generará un reporte de requerimientos en el formato que se solicite.

1.6) Funcionalidades

- i. Guardar datos para crear el oficio: Es el encargado de obtener los datos del usuario para guardarlos en la BD.

1.7) Escenario Principal

- i. El profesor accede a la página con la finalidad de generar su reporte administrativo.
- ii. El sistema muestra la página de inicio de sesión en la cual le pedirá su número económico y password asignado.
- iii. El profesor ingresará satisfactoriamente al sistema con los datos correctos.
- iv. El sistema le muestra la página principal con datos personales y con un menú desplegable con las opciones para crear y mostrar oficios.
- v. El profesor selecciona la parte del menú que dice CREAR OFICIO.
- vi. El sistema le muestra otra página la cual contiene un formulario en donde el profesor tiene que llenar todos los campos.
- vii. El profesor llena todos los campos del formulario y presiona el botón Guardar.
- viii. El sistema muestra otra página con una liga que dice ÉXITO DE GUARDADO.

- 2.1) **ID:** CU2
- 2.2) **Nombre:** Modificar datos del oficio.
- 2.3) **Actor principal:** Administrador o Secretaria.
- 2.4) **Precondiciones:** Debe existir registro de la creación del oficio presentado en el CU1.
- 2.5) **Post-condiciones:** El usuario modificará los datos del oficio ya creado.

2.6) Funcionalidades

- i. Modificar datos del oficio: Es el encargado de modificar los datos del oficio previamente guardados para usos posteriores.

2.7) Escenario Principal

- i. El profesor accede a la página con la finalidad de generar su reporte administrativo.
- ii. El sistema muestra la página de inicio de sesión en la cual le pedirá su número económico y password asignado.
- iii. El profesor ingresará satisfactoriamente al sistema con los datos correctos.
- iv. El sistema le muestra la página principal con datos personales y con un menú desplegable con las opciones para crear y mostrar oficios.
- v. El profesor selecciona la parte del menú MOSTRA OFICIO.
- vi. El sistema le muestra una página con una tabla que contiene el ID del oficio, NOMBRE del oficio, un botón para MOSTRAR PDF, un botón para MODIFICAR OFICIO y un botón para ELIMINAR OFICIO.
- vii. El profesor presiona el botón MODIFICAR OFICIO.
- viii. El sistema le muestra el formulario del CU1 pero con todos los campos llenados con los datos del profesor.
- ix. El profesor modifica algún dato y presiona el botón MODIFICAR.
- x. El sistema valida los datos introducidos por el profesor y lo envía a otra página que dice ÉXITO DE GUARDADO.

- 3.1) **ID:** CU3
- 3.2) **Nombre:** Eliminar datos del oficio.
- 3.3) **Actor principal:** Administrador o Secretaria.
- 3.4) **Precondiciones:** Debe existir registro de la creación del oficio presentado en el CU1 o modificación de algún registro presentado en el CU2.
- 3.5) **Post-condiciones:** El usuario eliminará datos de los oficios creados.

3.6) **Funcionalidades**

- i. Eliminar datos del oficio: Es el encargado de eliminar los datos de los oficios creados en el CU1 o modificados en el CU2 por cualquier cuestión.

3.7) **Escenario Principal**

- i. El profesor accede a la página con la finalidad de generar su reporte administrativo.
- ii. El sistema muestra la página de inicio de sesión en la cual le pedirá su número económico y password asignado.
- iii. El profesor ingresará satisfactoriamente al sistema con los datos correctos
- iv. El sistema le muestra la página principal con datos personales y con un menú desplegable con las opciones para crear y mostrar oficios.
- v. El profesor selecciona la parte del menú MOSTRA OFICIO.
- vi. El sistema le muestra una página con una tabla que contiene el ID del oficio, NOMBRE del oficio, un botón para MOSTRAR PDF, un botón para MODIFICAR OFICIO y un botón para ELIMINAR OFICIO.
- vii. El profesor presiona el botón de ELIMINAR OFICIO.
- viii. El sistema carga la misma página del inciso vi, sin el archivo.

- 4.1) **ID:** CU4
- 4.2) **Nombre:** Generar Reporte PDF.
- 4.3) **Actor principal:** Administrador o Profesor.
- 4.4) **Precondiciones:** Debe existir registro de la creación del oficio presentado en el CU1.
- 4.5) **Post-condiciones:** El usuario generará PDF's de sus oficios creados en el CU1.

4.6) **Funcionalidades**

- i. **Generar Reporte PDF:** Es el encargado de recabar los datos previamente guardados los cuales son presentados en un formato PDF para su posterior uso e impresión.

4.7) **Escenario Principal**

- i. El profesor accede a la página con la finalidad de generar su reporte administrativo.
- ii. El sistema muestra la página de inicio de sesión en la cual le pedirá su número económico y password asignado.
- iii. El profesor ingresará satisfactoriamente al sistema con los datos correctos.
- iv. El sistema le muestra la página principal con datos personales y con un menú desplegable con las opciones para crear y mostrar oficios.
- v. El profesor selecciona la parte del menú MOSTRA OFICIO.
- vi. El sistema le muestra una página con una tabla que contiene el ID del oficio, NOMBRE del oficio, un botón para MOSTRAR PDF, un botón para MODIFICAR OFICIO y un botón para ELIMINAR OFICIO.
- vii. El profesor presiona el botón MOSTRAR PDF.
- viii. El sistema muestra una página con la fecha de creación del oficio, el ID del oficio asignado, con el oficio llenado con los datos que el profesor ingreso y con un botón para GENERAR PDF.
- ix. El profesor muestra GENERAR PDF.
- x. El sistema le muestra el PDF del oficio.

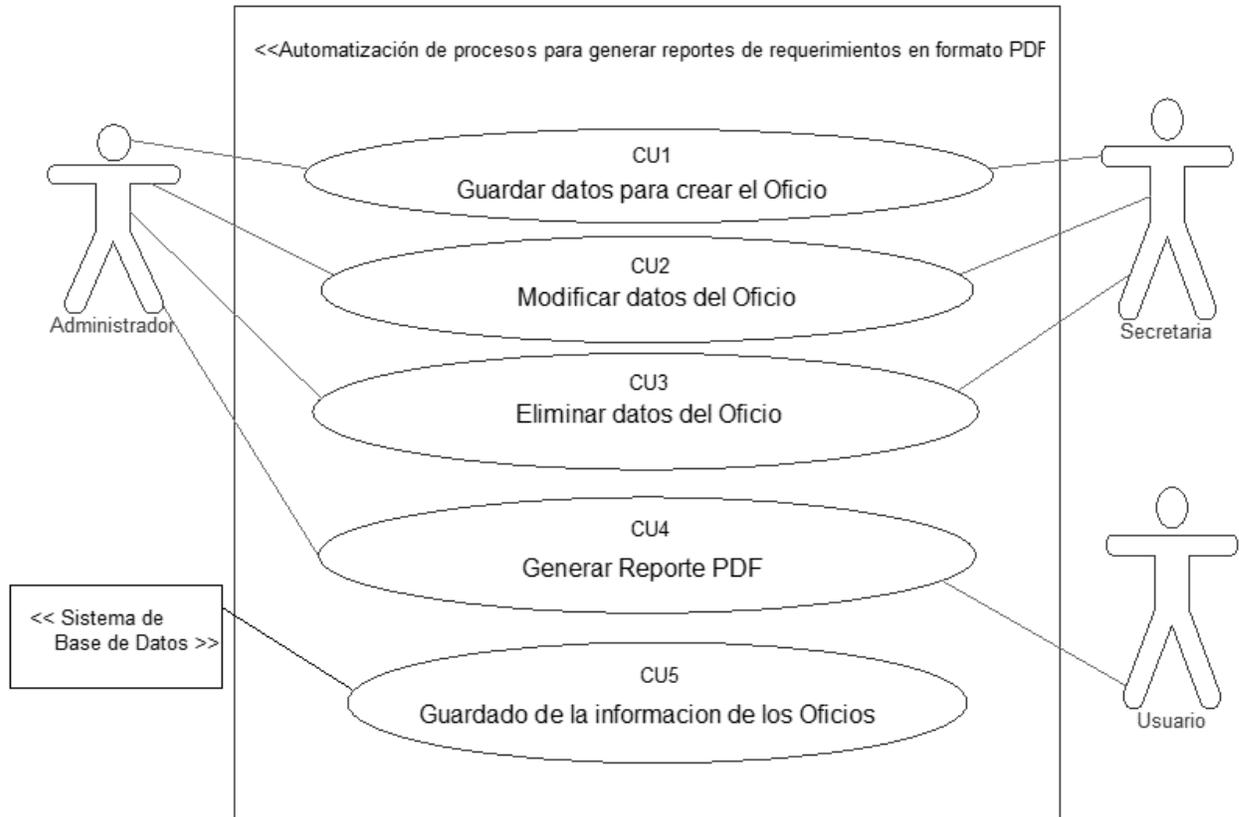


Figura 4. Diagrama de casos de uso del sistema.

Diagrama de Secuencia

Posteriormente en la Figura 5 se muestra el diagrama de secuencia del sistema, se observa la forma correcta de cómo se crearía un oficio de requerimientos.

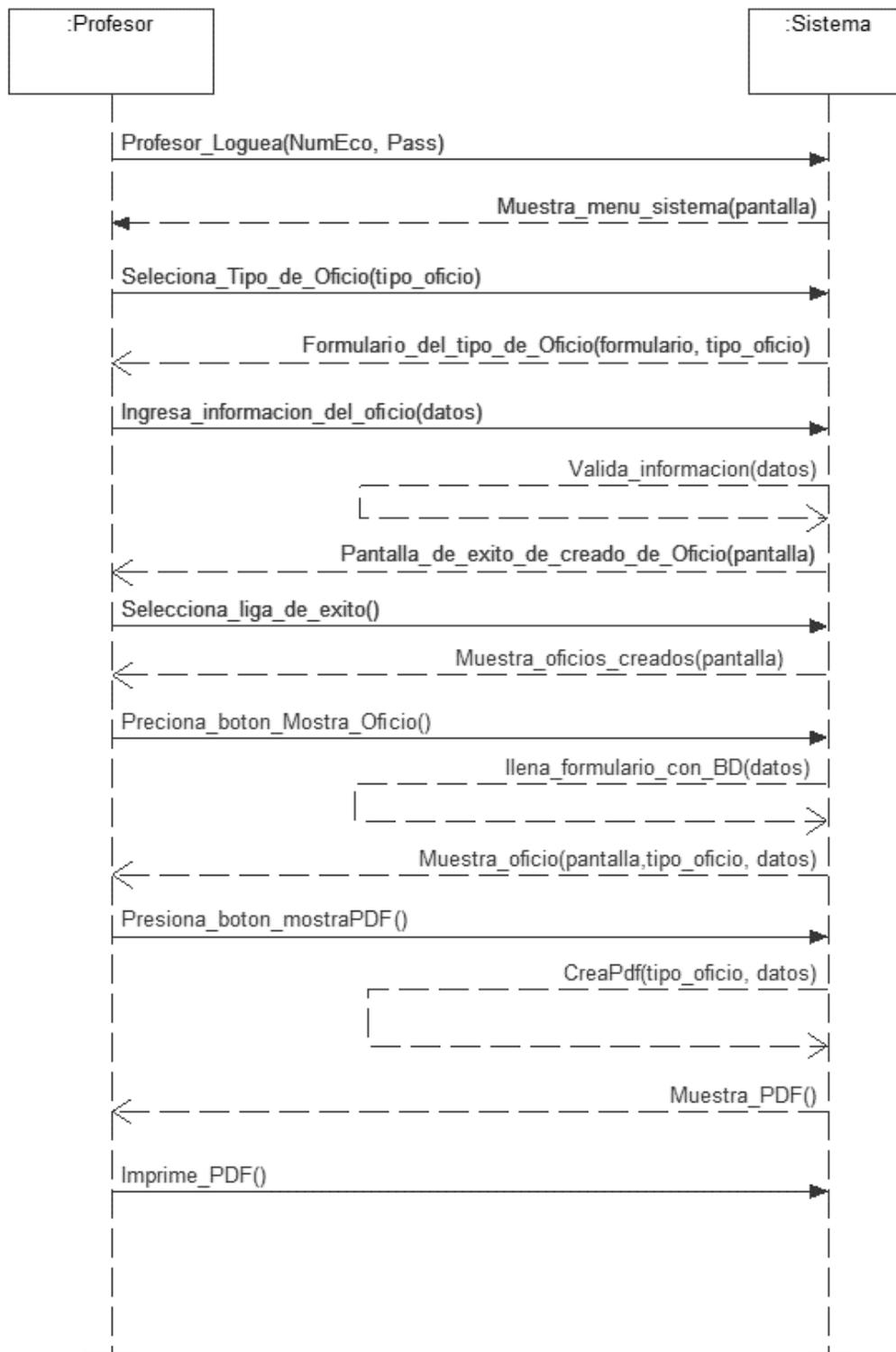


Figura 5. Diagrama de Secuencia del CU1.

Diagrama de Clases Conceptuales

En la Figura 6 se observa el diagrama de clases conceptuales del proyecto Automatizador de procesos administrativos con el cual se puede observar los diferentes objetos de dominio, las asociaciones y los atributos de cada uno.

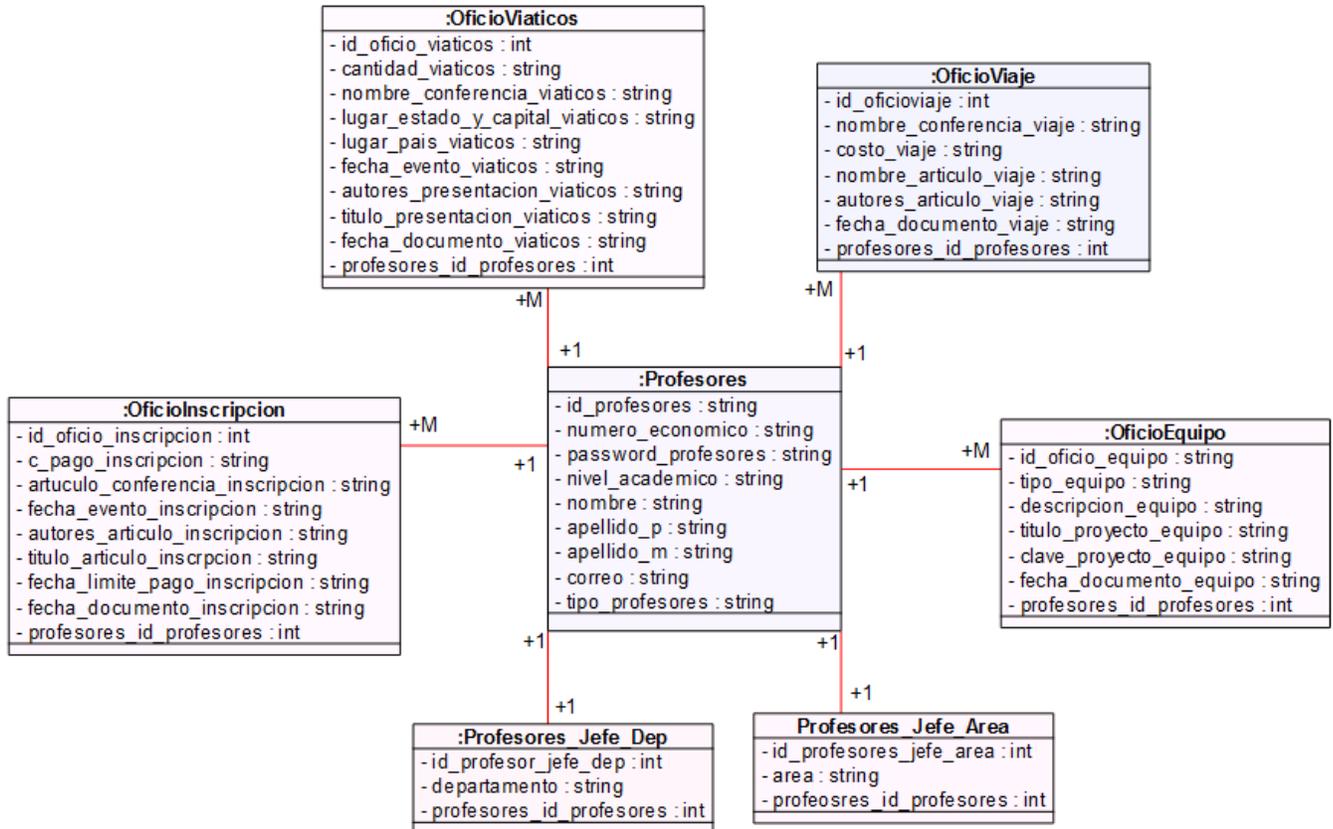


Figura 6. Diagrama de clases conceptuales del proyecto.

Diagrama de Robustez

En la siguiente Figura 7 se muestra el diagrama de robustez de todo el sistema donde se puede observar aplicación de los CU previamente definidos.

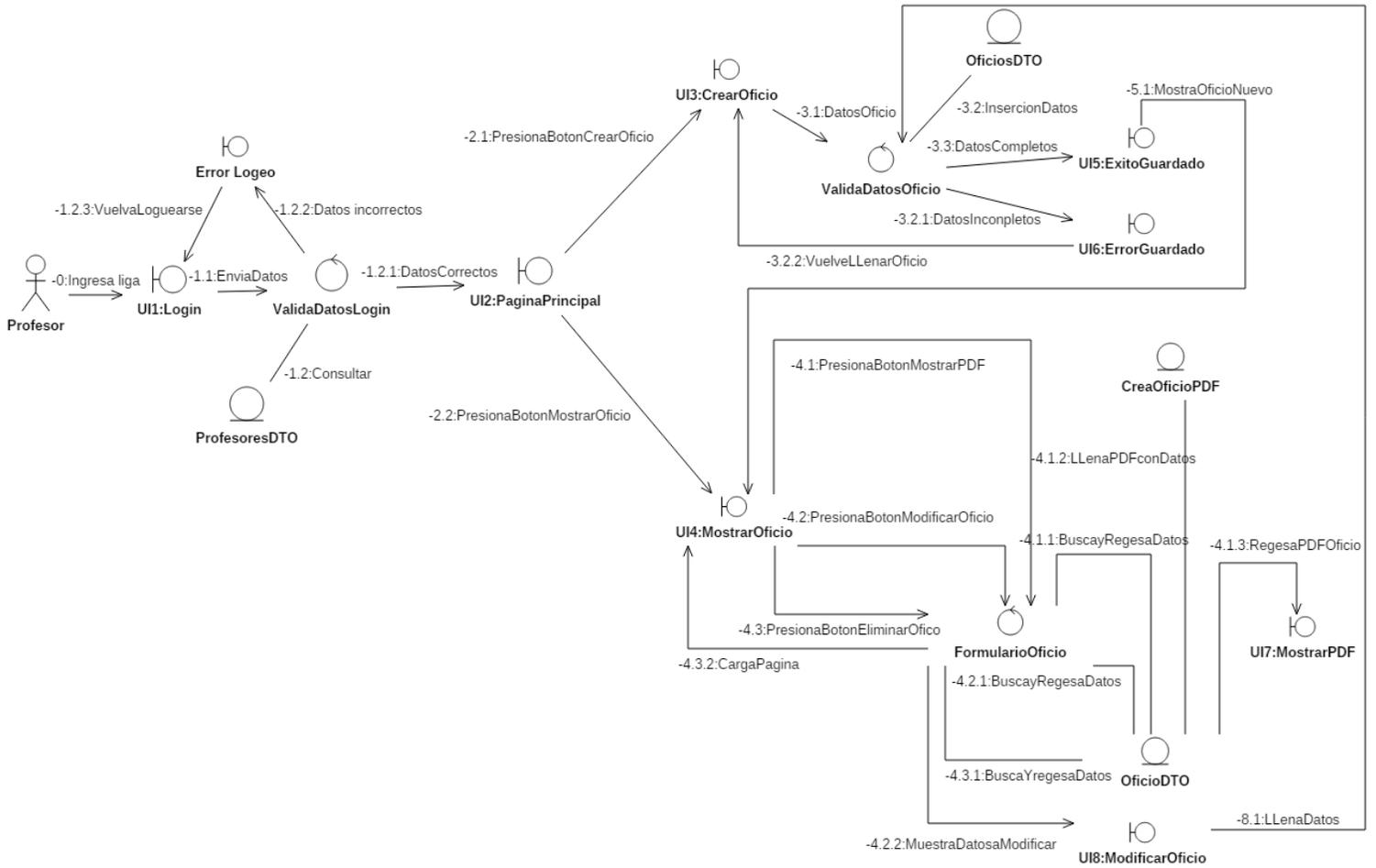


Figura 7. Diagrama de Robustez del Sistema.

Diagrama de la Base de Datos

En la Figura 8 se observa el diagrama de la base de datos con sus respectivos tipos de datos que tiene, así como también su conexión con la tabla principal.

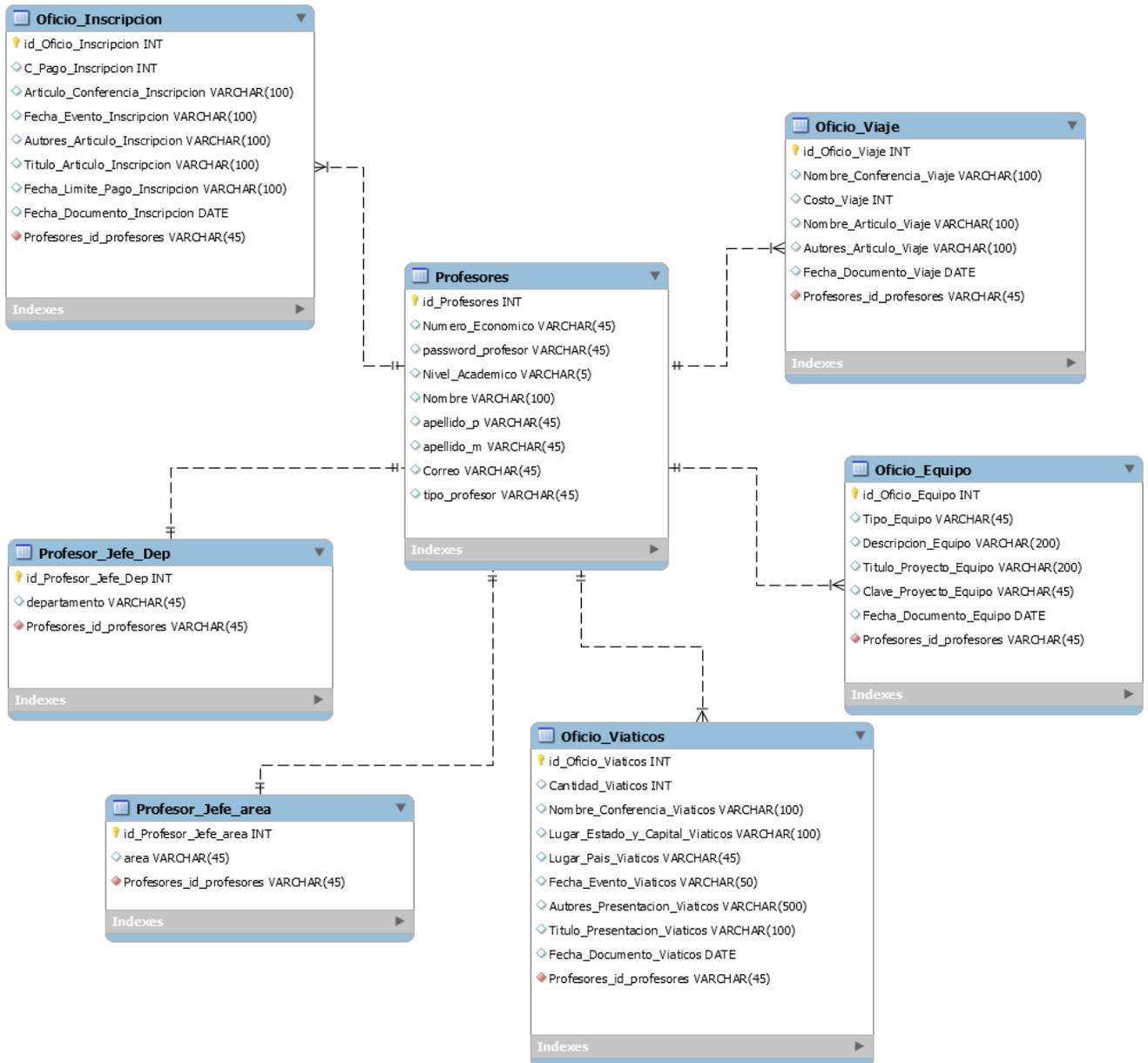


Figura 8. Diagrama de la Base de Datos del Proyecto.

Resultados

Casos de uso 1)

Crear un oficio de requerimientos con interfaz desde el inicio de sesión.



Figura 9. Inicio de sesion en la pagina correspondiente al proyecto



Figura 10. Acceso a la sesión

Se coloca el Numero Económico correspondiente al profesor y su password asignado por el sistema para poder acceder y presiona el botón Entrar.

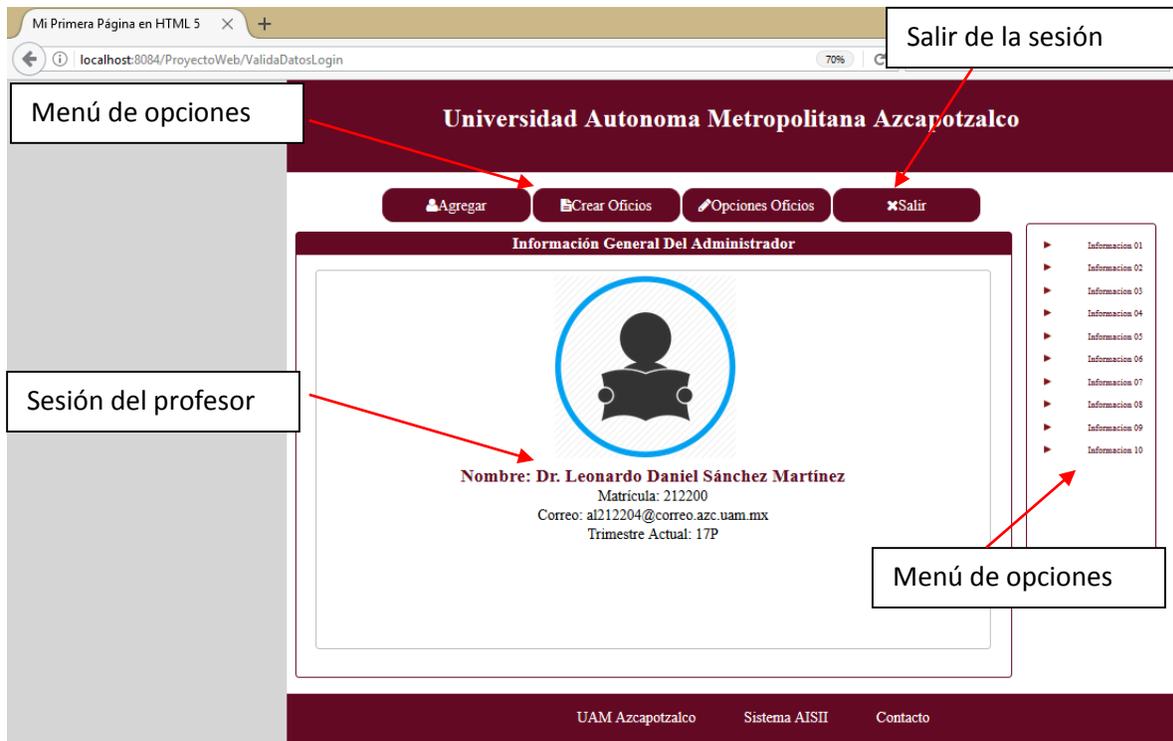


Figura 11. Muro de sesión

Esta sería el muro de información del profesor asignado, tiene un menú donde puede crear y opciones de los oficios, así como también salir se la sesión.

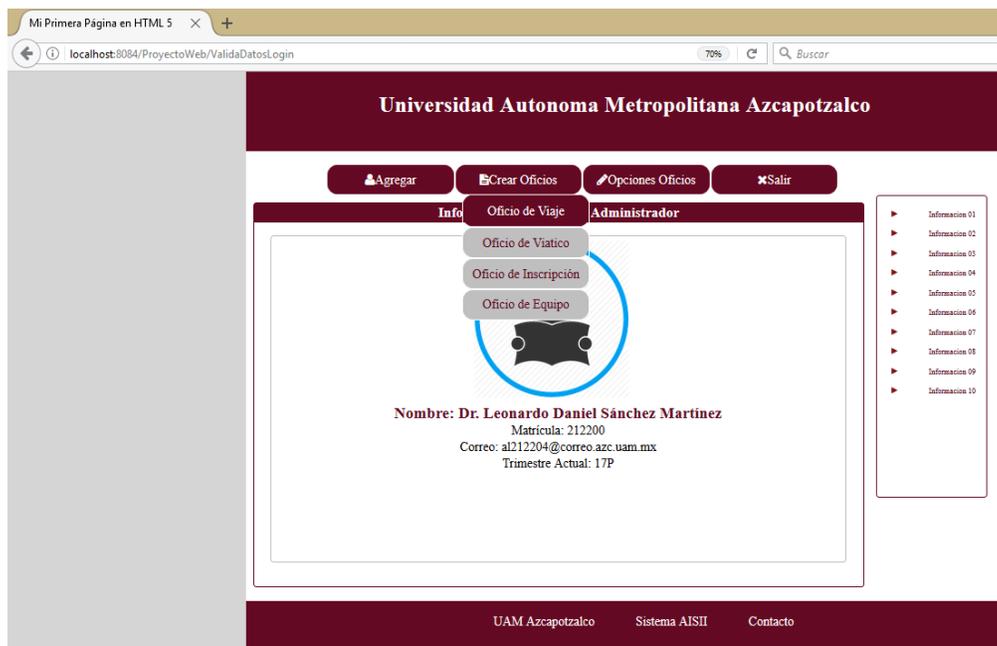


Figura 12. El profesor presiona la opción del menú Crear Oficio le aparecen cuatro opciones selecciona la que el necesita adecuándose al tipo de oficio que quiere.

Universidad Autónoma Metropolitana Azcapotzalco

Oficio de Viaje

Jueves , 6 de Julio de 2017

Dr. Jesus Isidro Gonzales Trejo
Jefe del Departamento de Sistemas

Por medio del presente se solicita su apoyo para realizar las gestiones necesarias para que se lleve a cabo el reembolso al **Juan Manuel** por concepto del viaje por la asistencia al siguiente evento:

a) Viaje por un monto de con un costo \$ MXN, por haber asistido a la conferencia en el cual se presento el siguiente artículo de investigación:

Autores:

Este monto saldrá de los recursos del Área de Sistemas de Información Inteligentes de la partida 45 "Gastos de viaje".

Anexo copia del gafete, la primera hoja de los artículos de investigación publicados, boleto de avión y factura del cobro.

Mucho le agradeceré la atebción que preste a esta solicitud, ya que esta asistencia sirvió para la difusión de la investigación en conjunto del Área.

Sin mas por el momento, quedo atento a cualquier aclaración.

Dr. José Alejandro Reyes Ortoz
Jefe del Área de sistemas de Información Inteligentes

Guardar Limpia

UAM Azcapotzalco Sistema VU Derechos Reservados 2015

Figura 13. Crear oficio

Al seleccionar el oficio requerido dentro del menú, le aparecerá esta página con el formulario del oficio para que el usuario llene los campos solicitados.

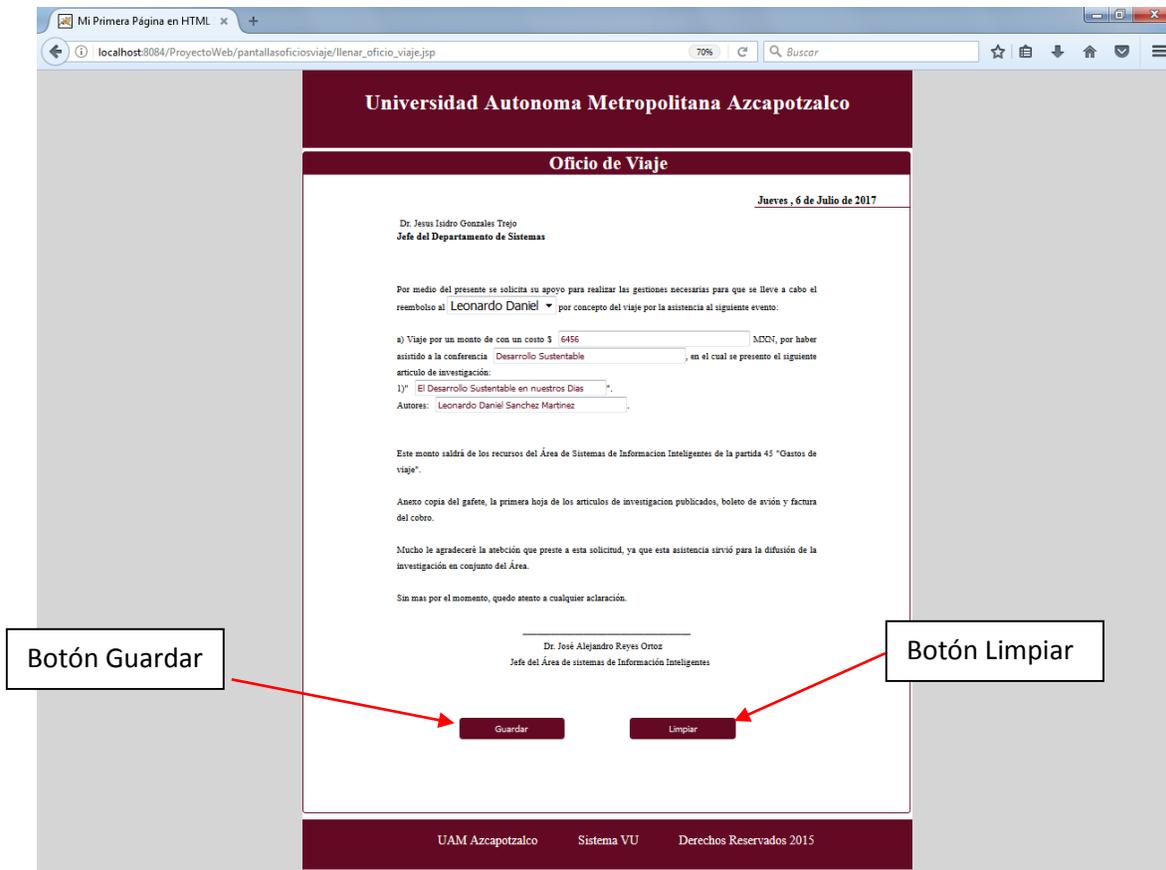


Figura 14. El profesor tiene que presionar el botón Guardar o el botón limpiar que sirve para limpiar todos los campos ya llenos.

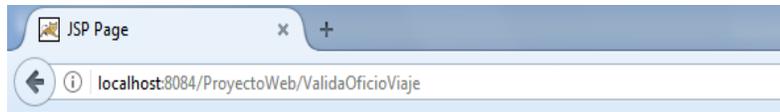


Error de oficio de Viaje

Campo del Costo Viaje Incompleto de Modificar

Figura 15. Página de error

Al presionar el botón Guardar el sistema comprueba que todos los campos estén llenados si el profesor no lleno uno le aparecerá la siguiente página indicándole que campo le faltó llenar y regresándolo a la Figura 13, en caso contrario se va a la Figura 16.



Exito de oficio de Viaje

Figura 16. Página de éxito.

El sistema ha revisado los datos y todos fueron correctos lo enviará a esta página, la cual es una liga para enviarlo a la Figura 17.

Universidad Autonoma Metropolitana Azcapotzalco

Tus Oficio de Viaje

ID	Nombre Presentacion	Mostrar	Modificar	Eliminar
1	Web Services Ontology Population_1	MostrarReporte	ModificarReporte	EliminarReporte
2	Web Services Ontology Population_2	MostrarReporte	ModificarReporte	EliminarReporte
3	Web Services Ontology Population_3	MostrarReporte	ModificarReporte	EliminarReporte
4	Web Services Ontology Population_4	MostrarReporte	ModificarReporte	EliminarReporte
5	Web Services Ontology Population_5	MostrarReporte	ModificarReporte	EliminarReporte
6	Web Services Ontology Population_6	MostrarReporte	ModificarReporte	EliminarReporte
7	El Desarrollo Sustentable en nuestros Dias	MostrarReporte	ModificarReporte	EliminarReporte

UAM Azcapotzalco Sistema VU Derechos Reservados 2015

Figura 17. Oficios creados por el profesor

Al presionar la liga anterior lo enviara a esta página para que el profesor pueda observar su documento creado, así como también podrá observar los documentos creados con anterioridad y le presentara tres opciones que son Mostrar Reporte, Modificar Reporte y Eliminar Reporte, estos son utilizados en los próximos casos de uso.

Casos de uso 2)

Modificar un oficio de requerimientos con interfaz desde el inicio de sesión.

Universidad Autonoma Metropolitana Azcapotzalco

Iniciar Sesión

Numero Economico:

Contraseña:

UAM Azcapotzalco Sistema VU Derechos Reservados 2015

Figura 18. Inicio de sesion en la pagina correspondiente al proyecto.

Universidad Autonoma Metropolitana Azcapotzalco

Iniciar Sesión

Numero Economico:

Contraseña:

UAM Azcapotzalco Sistema VU Derechos Reservados 2015

Figura 19. Acceso a la sesión.

Se coloca el Numero Económico correspondiente al profesor y su password asignado por el sistema para poder acceder y presiona el botón Entrar.

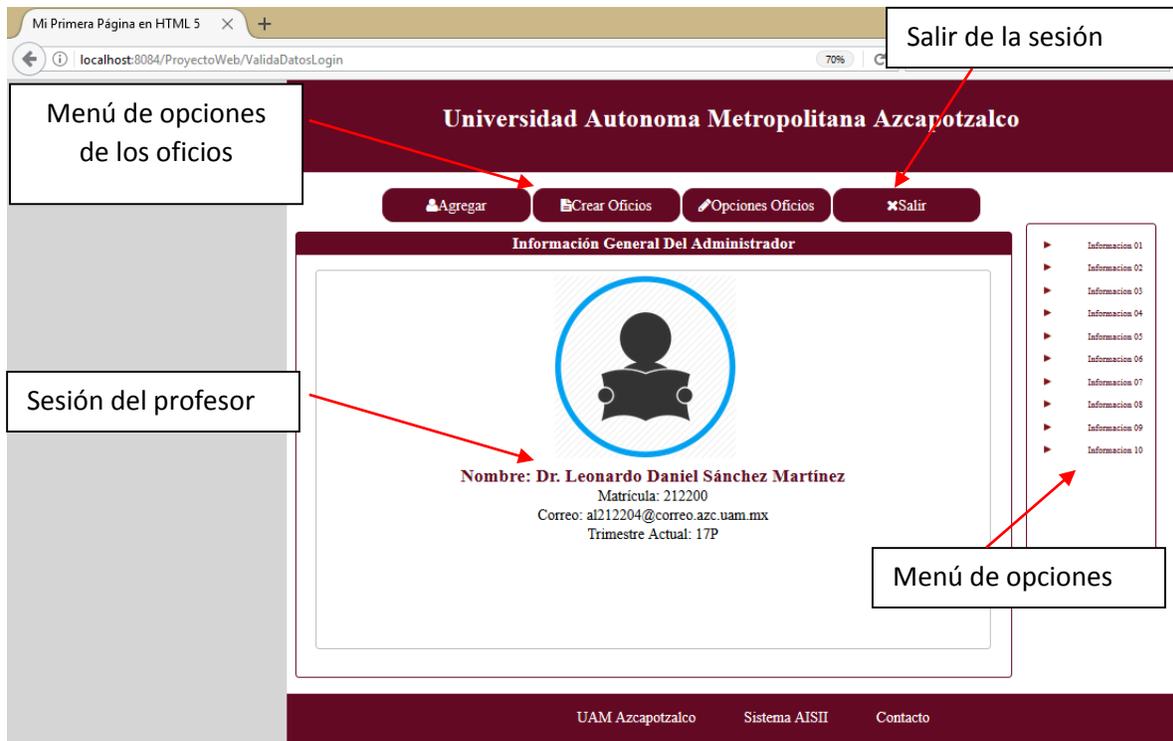


Figura 20. Esta sería el muro de información del profesor asignado, tiene un menú donde puede crear y observar los oficios, así como también salir se la sesión.

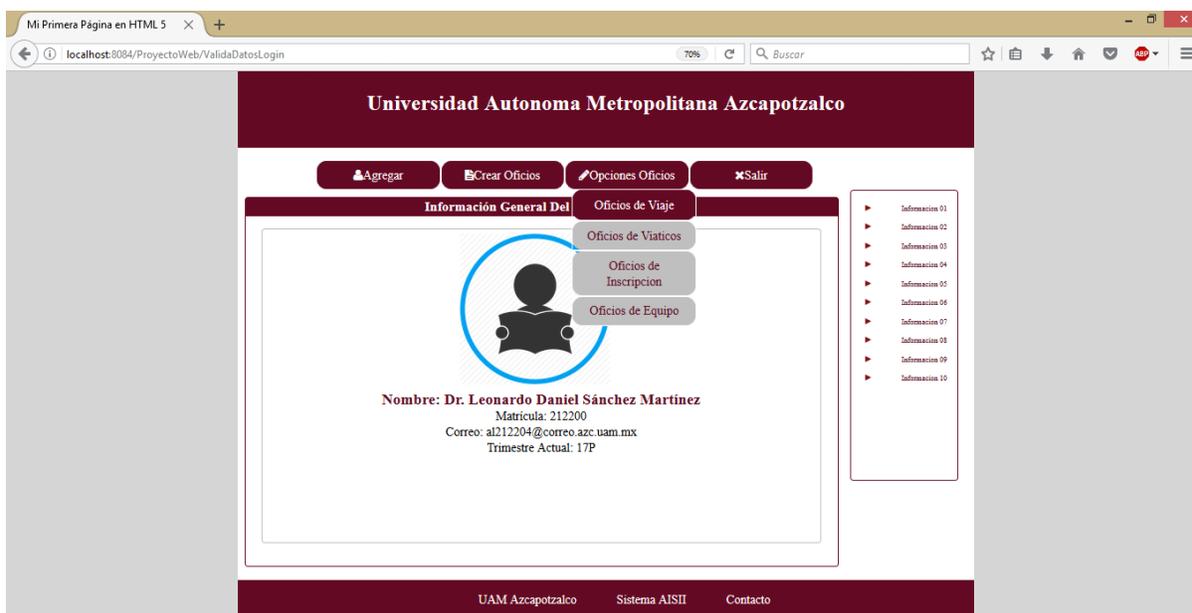


Figura 21. El profesor presiona la opción del menú Opciones Oficios le aparecen cuatro opciones y selecciona la que el necesita.

Mi Primera Página en HTML x +

localhost:8084/ProyectoWeb/pantallasoficiosviaje/formulario_oficio_viaje.jsp

Buscar

Universidad Autonoma Metropolitana Azcapotzalco

Tus Oficio de Viaje

ID	Nombre Presentacion	Mostrar	Modificar	Eliminar
1	Web Services Ontology Population_1	MostrarReporte	ModificarReporte	EliminarReporte
2	Web Services Ontology Population_2	MostrarReporte	ModificarReporte	EliminarReporte
3	Web Services Ontology Population_3	MostrarReporte	ModificarReporte	EliminarReporte
4	Web Services Ontology Population_4	MostrarReporte	ModificarReporte	EliminarReporte
5	Web Services Ontology Population_5	MostrarReporte	ModificarReporte	EliminarReporte
6	Web Services Ontology Population_6	MostrarReporte	ModificarReporte	EliminarReporte
7	El Desarrollo Sustentable en nuestros Dias	MostrarReporte	ModificarReporte	EliminarReporte

UAM Azcapotzalco Sistema VU Derechos Reservados 2015

Modificar Reporte

Figura 22. Le muestra una tabla con los oficios creados por el usuario y con varias opciones, el profesor selecciona la opción de Modificar Reporte.

Universidad Autónoma Metropolitana Azcapotzalco

Oficio de Viaje

Fecha de creación → Fecha de Creacion del Oficio: 2017-07-06 11:25:25

Numero de Reporte → Numero de Reporte: 7

Nombre del profesor → Por medio del presente se solicita su apoyo para realizar las gestiones necesarias para que se lleve a cabo el reembolso al Juan Manuel por concepto del viaje por la asistencia al siguiente evento:

Costo del viaje → a) Viaje por un monto de con un costo \$ 6456 MXN, por haber asistido a la conferencia: Desarrollo Sustentable, en el cual se presento el siguiente artículo de investigación:

Conferencia del viaje → IV El Desarrollo Sustentable en nuestros Dias *

Artículo de investigación → Autores: Leonardo Daniel Sanchez Martinez

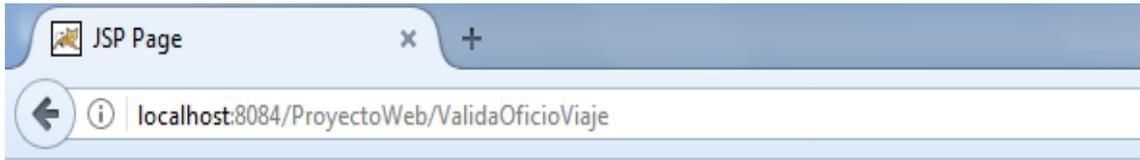
Autores → Este monto saldrá de los recursos del Área de Sistemas de Información Inteligentes de la partida 45 "Gastos de viaje".

Botón Modificar → Modificar

UAM Azcapotzalco Sistema VU Derechos Reservados 2015

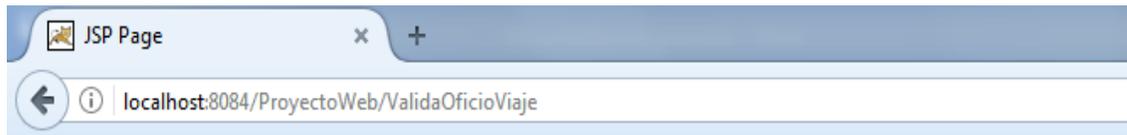
Figura 23. Reporte a modificar.

Al presionar Modificar Reporte le aparecerá esta página con el formulario llenado, solo tiene que borrar el dato que esta incorrecto y escribir los nuevos datos, posteriormente presionar el botón Modificar para que se pueda guardar el reporte Modificado.



Exito de oficio de Viaje

Figura 24. El guardado de la información fue exitoso



Error de oficio de Viaje

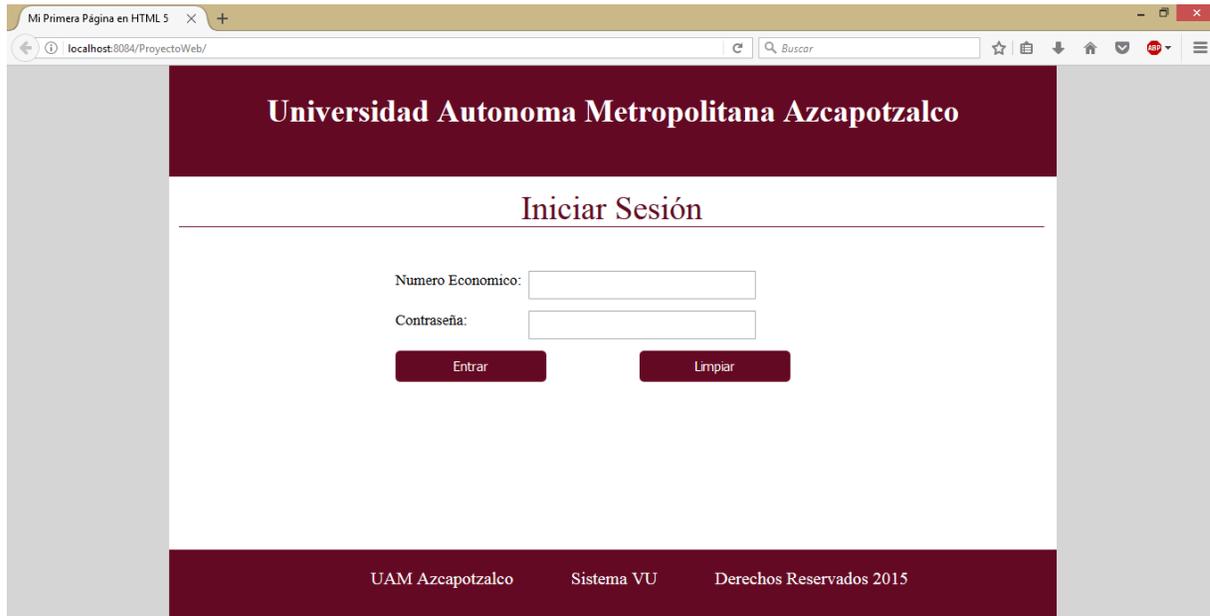
Campo del Costo Viaje Incompleto de Modificar

Figura 25. El guardado de la información fue erróneo.

Al presionar el botón Modificar el sistema verifica la información y si el usuario omitió Datos lo enviara a la figura 25 mostrándole que dato omitió, si fue correcto el proceso lo enviara a la figura 24 esas páginas son ligas que lo enviara a la figura 22 presentada con anterioridad.

Casos de uso 4)

Mostrar un oficio de requerimientos con interfaz desde el inicio de sesión.



Universidad Autonoma Metropolitana Azcapotzalco

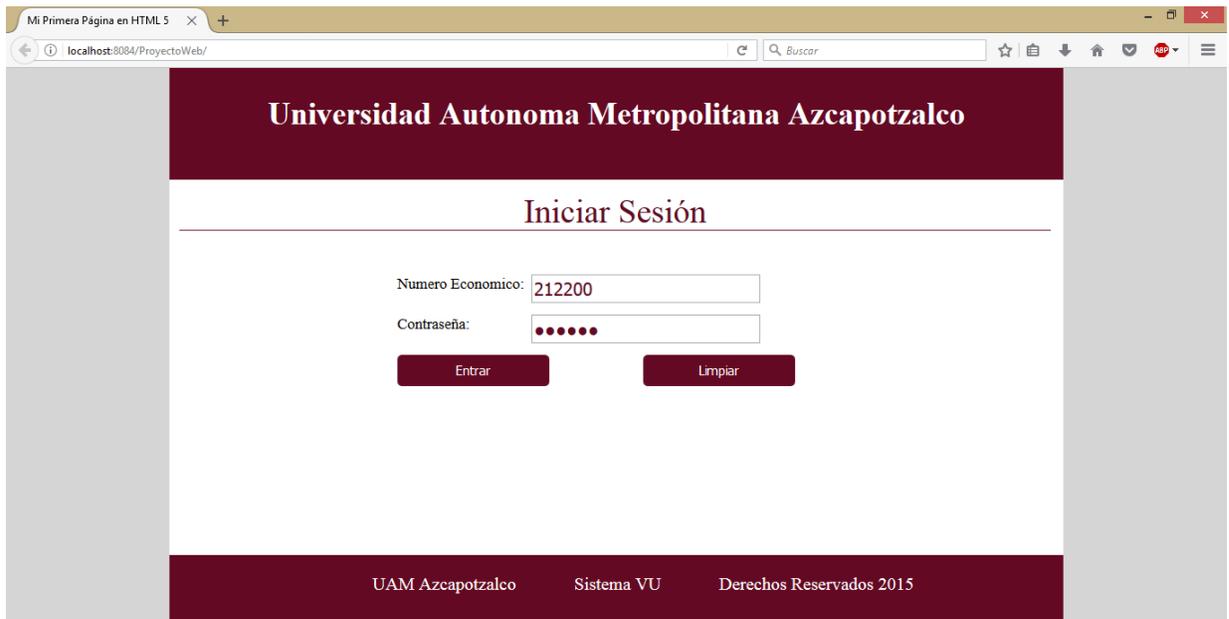
Iniciar Sesión

Numero Economico:

Contraseña:

UAM Azcapotzalco Sistema VU Derechos Reservados 2015

Figura 26. Inicio de sesion en la pagina correspondiente al proyecto.



Universidad Autonoma Metropolitana Azcapotzalco

Iniciar Sesión

Numero Economico:

Contraseña:

UAM Azcapotzalco Sistema VU Derechos Reservados 2015

Figura 27. Acceso a la sesión

Se coloca el Numero Económico correspondiente al profesor y su password asignado por el sistema para poder acceder y presiona el botón Entrar.

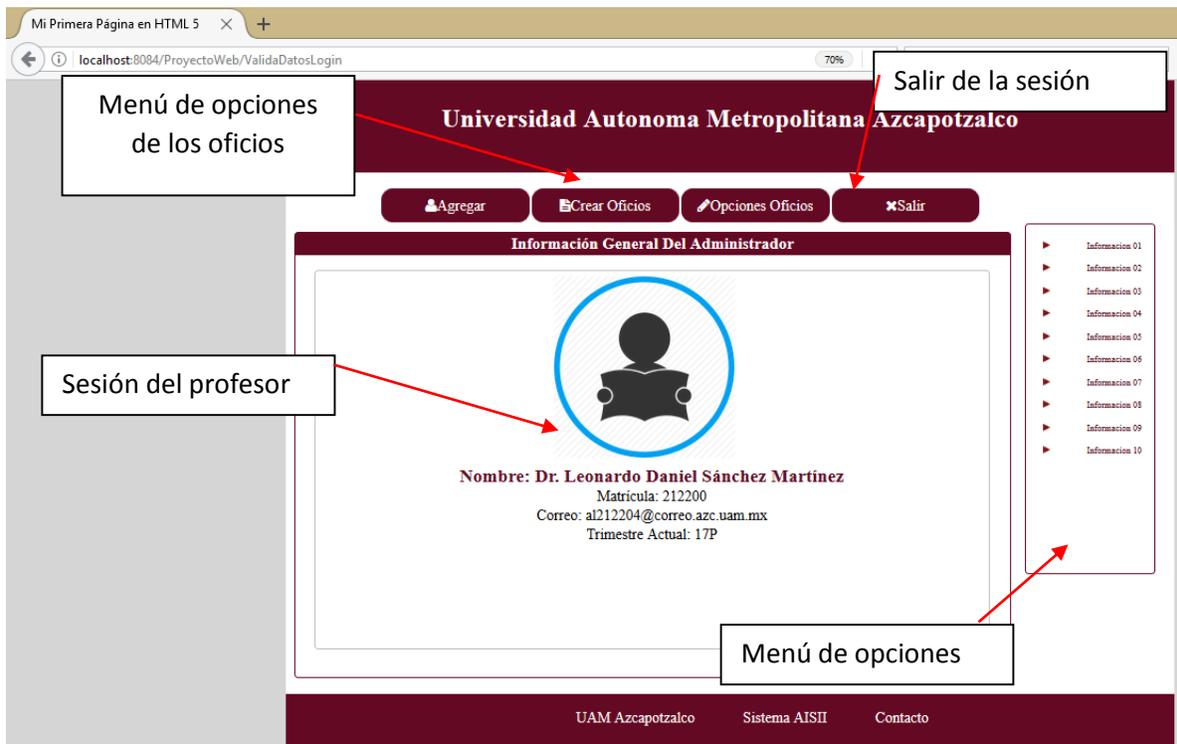


Figura 28. Esta sería el muro de información del profesor asignado, tiene un menú donde puede crear y observar los oficios, así como también salir se la sesión.

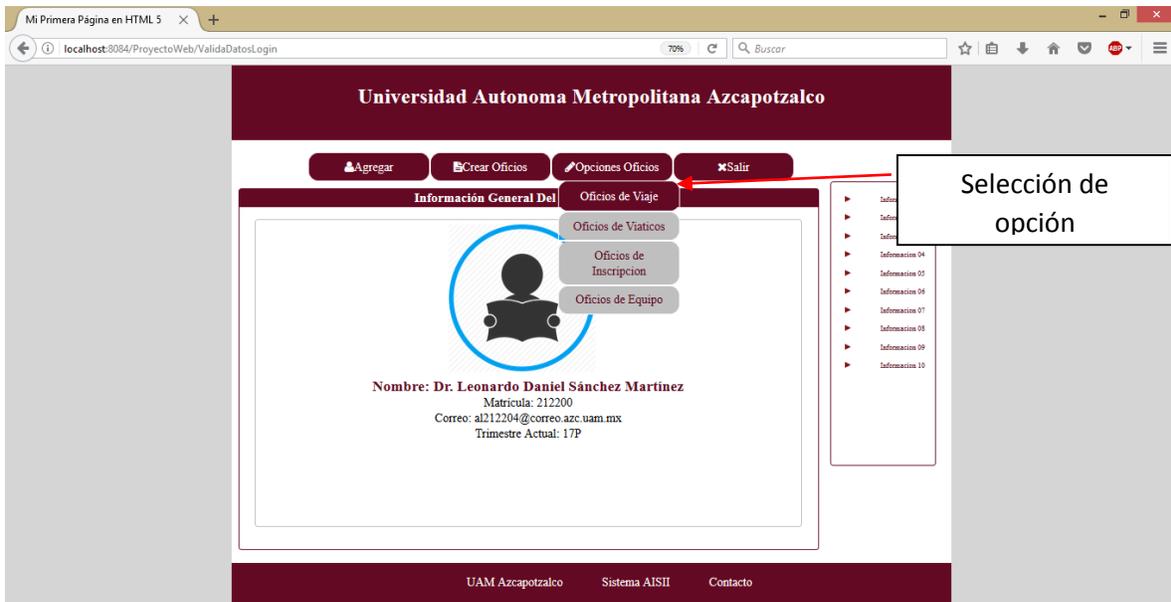


Figura 29. El profesor presiona la opción del menú Mostrar Oficio le aparecen cuatro opciones selecciona la que el necesita.

Universidad Autonoma Metropolitana Azcapotzalco

Tus Oficio de Viaje

ID	Nombre Presentacion	Mostrar	Modificar	Eliminar
1	Web Services Ontology Population_1	MostrarReporte	ModificarReporte	EliminarReporte
2	Web Services Ontology Population_2	MostrarReporte	ModificarReporte	EliminarReporte
3	Web Services Ontology Population_3	MostrarReporte	ModificarReporte	EliminarReporte
4	Web Services Ontology Population_4	MostrarReporte	ModificarReporte	EliminarReporte
5	Web Services Ontology Population_5	MostrarReporte	ModificarReporte	EliminarReporte
6	Web Services Ontology Population_6	MostrarReporte	ModificarReporte	EliminarReporte
7	El Desarrollo Sustentable en nuestros Dias	MostrarReporte	ModificarReporte	EliminarReporte

UAM Azcapotzalco Sistema VU Derechos Reservados 2015

Oficio Creado Previamente

Mostrar Reporte

Figura 30. Le muestra una tabla con los oficios creados por el usuario y con varias opciones el profesor selecciona la opción de Mostrar Reporte y lo enviará a la siguiente página.

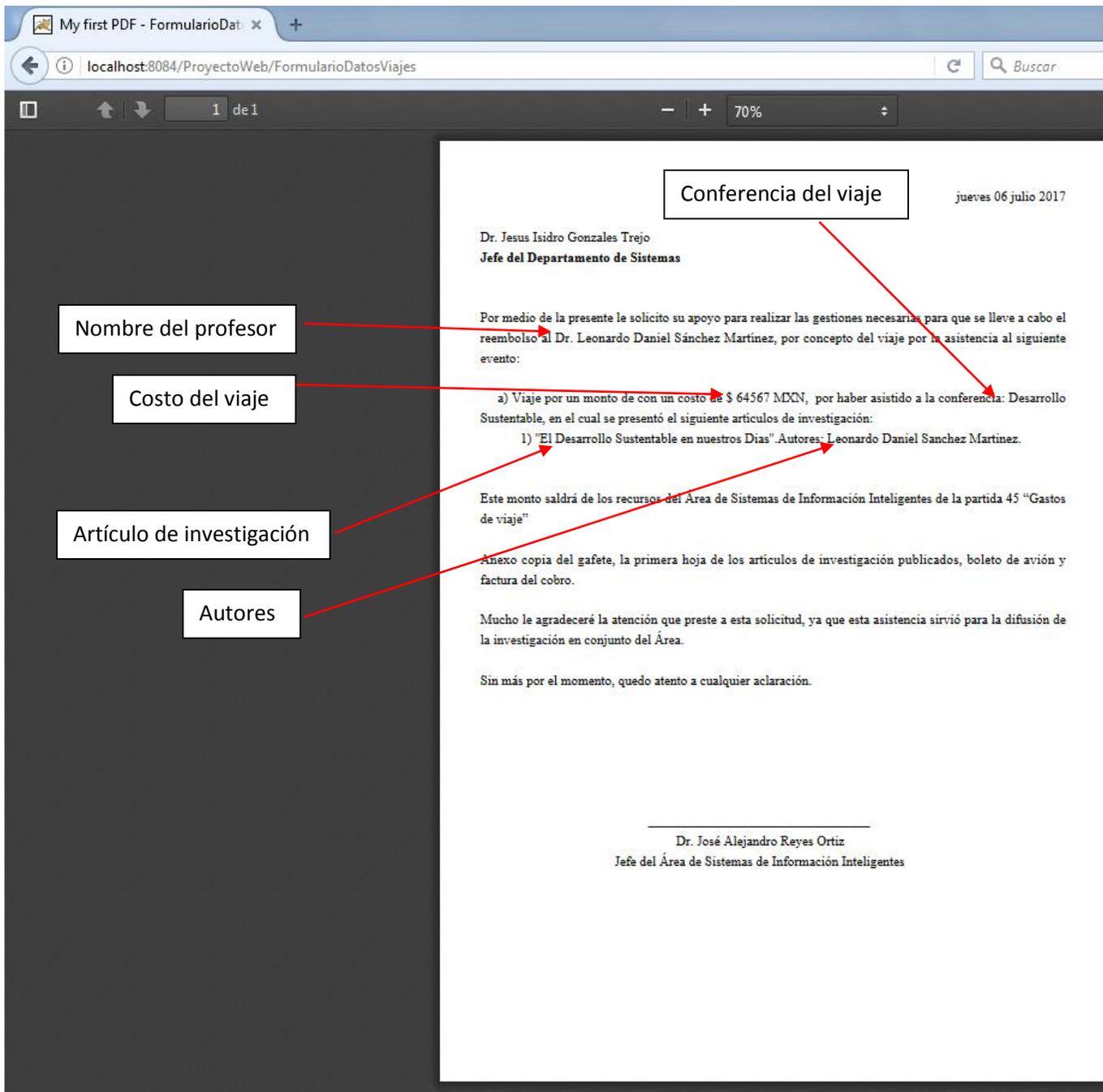


Figura 31. Presentación del Oficio en formato PDF

En esta página se muestra el PDF creado por el profesor con los datos que el insertó en el CU1 y con las facilidades de una página con PDF que son la de imprimir, guardar, y observar el número de páginas que contiene el documento.

Se puso el caso de uso 4 antes que el caso de uso 3 ya que este sería la forma de crear, modificar y mostrar un oficio de requerimientos y en el último caso sería el de eliminar.

Casos de uso 3)

Eliminar un oficio de requerimientos con interfaz desde el inicio de sesión.

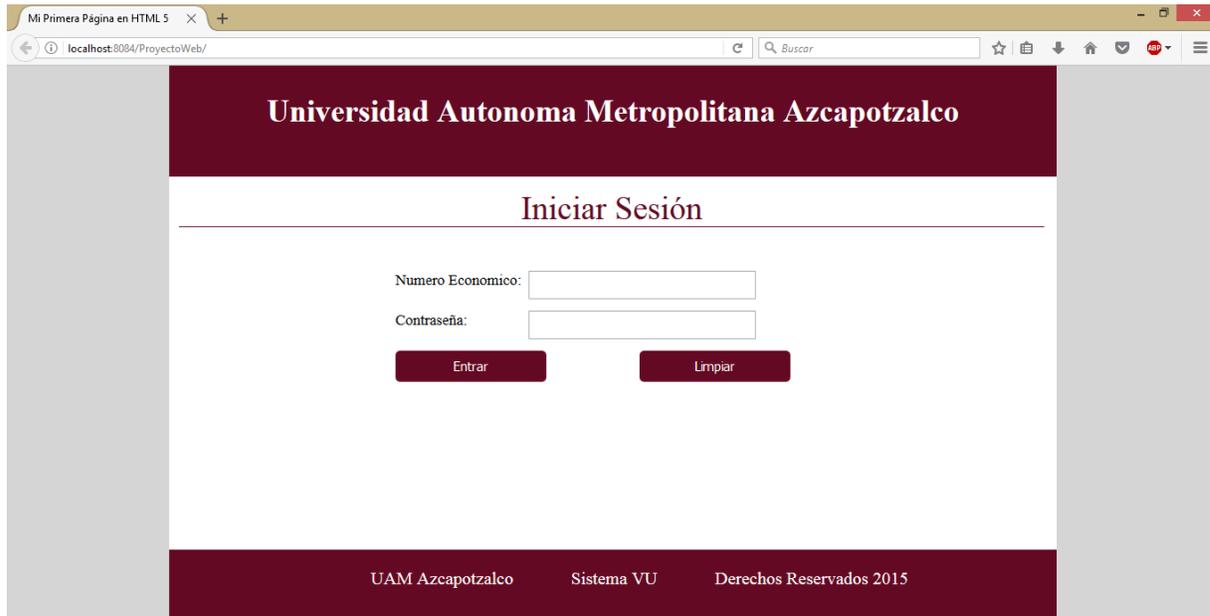


Figura 32. Inicio de sesion en la pagina correspondiente al proyecto.

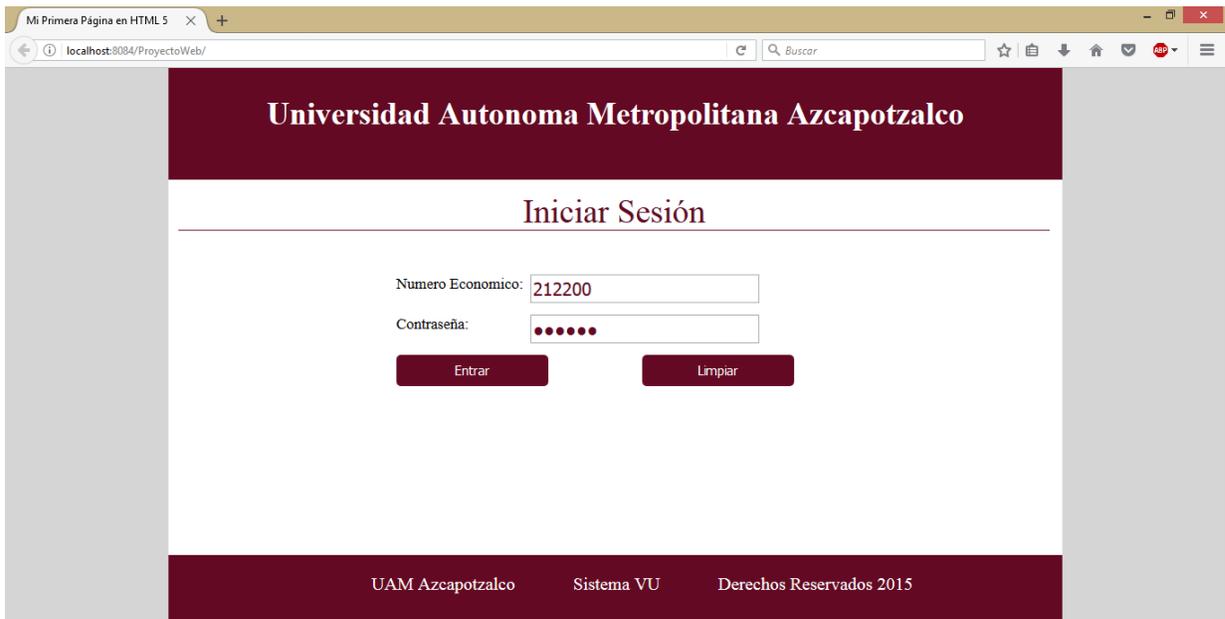


Figura 33. Acceso a la sesión

Se coloca el Numero Económico correspondiente al profesor y su password asignado por el sistema para poder acceder y presiona el botón Entrar.

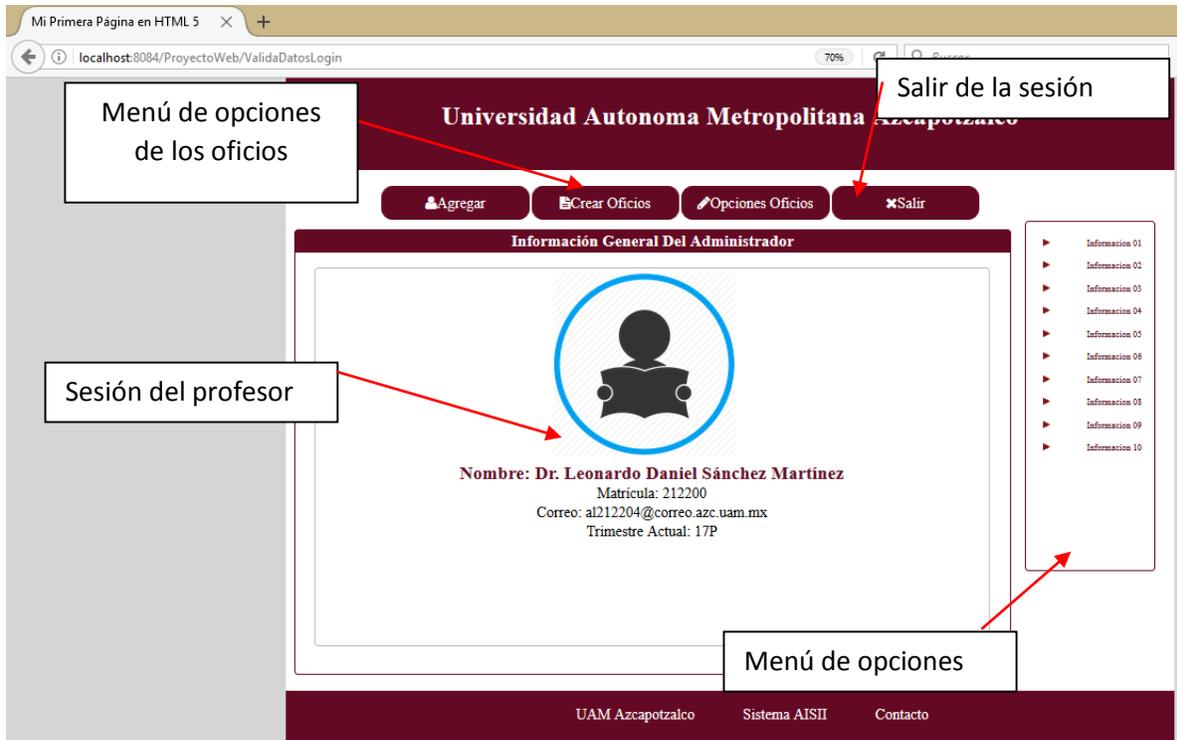


Figura 34. Esta sería el muro de información del profesor asignado, tiene un menú donde puede crear y observar los oficios, así como también salir se la sesión.

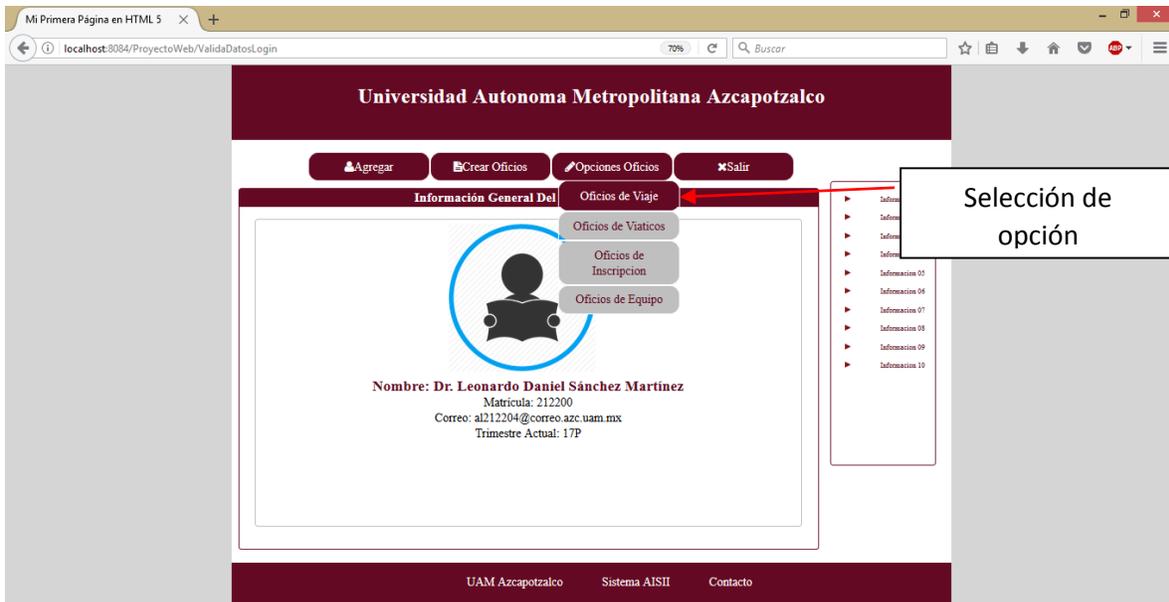


Figura 35. El profesor presiona la opción del menú Mostrar Oficio le aparecen cuatro opciones selecciona la que el necesita.



Figura 36. Le muestra una tabla con los oficios creados por el usuario y con varias opciones el profesor selecciona la opción de Eliminar Reporte.

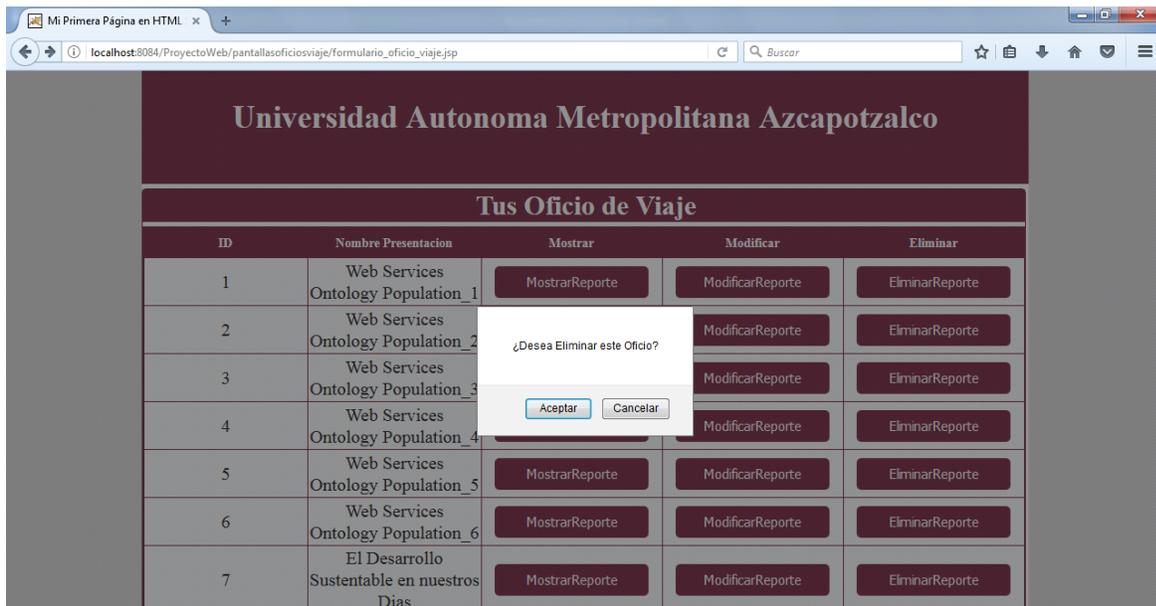


Figura 37. Se le va aparecer un cuadro de confirmación con el cual va a seleccionar la opción de Aceptar si quiere borrar el reporte o Cancelar si no lo quiere hacer.

Universidad Autonoma Metropolitana Azcapotzalco

Tus Oficio de Viaje

ID	Nombre Presentacion	Mostrar	Modificar	Eliminar
1	Web Services Ontology Population_1	MostrarReporte	ModificarReporte	EliminarReporte
2	Web Services Ontology Population_2	MostrarReporte	ModificarReporte	EliminarReporte
3	Web Services Ontology Population_3	MostrarReporte	ModificarReporte	EliminarReporte
4	Web Services Ontology Population_4	MostrarReporte	ModificarReporte	EliminarReporte
5	Web Services Ontology Population_5	MostrarReporte	ModificarReporte	EliminarReporte
6	Web Services Ontology Population_6	MostrarReporte	ModificarReporte	EliminarReporte

UAM Azcapotzalco Sistema VU Derechos Reservados 2015

Figura 38. Una vez selecciona el paso anterior recargara la página con la misma tabla del paso anterior, pero sin el archivo creado en el CU1.

Resultados de los oficios Credos

En las siguientes Figuras se muestran la comparación de los oficios creados por el usuario y por el sistema.

jueves 22 junio 2017

Dr. Jesus Isidro Gonzales Trejo
Jefe del Departamento de Sistemas

Por medio de la presente le solicito su apoyo para realizar las gestiones necesarias para que se lleve a cabo el reembolso al Dr. ~~Jose Alejandro Reyes Ortiz~~ por concepto del viaje por la asistencia al siguiente evento:

a) Viaje por un monto de con un costo de \$ 1,000.00 MXN, por haber asistido a la conferencia: FesCSIS 2016, en el cual se presentó el siguiente artículos de investigación:

1) "Web Services Ontology Population_3".Autores: Jose S. Reyes Ortiz, Maricela Claudia.

Este monto saldrá de los recursos del Área de Sistemas de Información Inteligentes de la partida 45 "Gastos de viaje"

Anexo copia del gafete, la primera hoja de los artículos de investigación publicados, boleto de avión y factura del cobro.

Mucho le agradeceré la atención que preste a esta solicitud, ya que esta asistencia sirvió para la difusión de la investigación en conjunto del Área.

Sin más por el momento, quedo atento a cualquier aclaración.

Dr. José Alejandro Reyes Ortiz
Jefe del Área de Sistemas de Información Inteligentes

Figura 39. Oficio creado por los profesores antes del sistema.

Lunes, 19 de septiembre de 2016

Dr. Jesús Isidro González Trejo
Jefe del Departamento de Sistemas

Por medio de la presente le solicito su apoyo para realizar las gestiones necesarias para que se lleve a cabo el reembolso al Dr. **José Alejandro Reyes Ortiz**, por concepto del viaje por la asistencia al siguiente evento:

- a) Viaje por un monto de con un costo de \$ **4000.00** MXN, por haber asistido a la conferencia FedCSIS 2016, en el cual se presentó el siguiente artículos de investigación:
 - 1. Web Services Ontology Population through Text Classification". Autores: **José A. Reyes-Ortiz**, Maricela Bravo y Hugo Pablo.

Este monto saldrá de los recursos del Área de Sistemas de Información Inteligentes de la partida 45 "Gastos de viaje"

Anexo copia del gafete, la primera hoja de los artículos de investigación publicados, boleto de avión y factura del cobro.

Mucho le agradeceré la atención que preste a esta solicitud, ya que esta asistencia sirvió para la difusión de la investigación en conjunto del Área.

Sin más por el momento, quedo atento a cualquier aclaración.

Dr. José Alejandro Reyes Ortiz
Jefe del Área de Sistemas de Información Inteligentes

Figura 40. Oficio creado por el sistema.

Tomando como referencia la página TestDeMecanografía [16] una persona suele escribir de 31 a 41 palabras por minuto, en la Tabla 2 se muestra el tiempo que tardaría una persona en tan solo escribir el oficio requerido, esto implicaría un tiempo extra para editarlo y darle la presentación adecuada y otro tiempo si el usuario se equivocó, esto conllevará una pérdida de tiempo innecesaria para el creador del oficio.

		PPM (31)
Oficio Viaje	191 palabras	6.16 minutos
Oficio Viáticos	216 palabras	6.9 minutos
Oficio Inscripción	239 palabras	7.7 minutos
Oficio equipo	207 palabras	6.6 minutos

Tabla 2. Tiempo de escritura por palabra

	Oficio Viaje		Oficio Viatico	
	Tiempo con el sistema	Tiempo sin el sistema	Tiempo con el sistema	Tiempo sin el sistema
Profesor 1	2 min.	15 min.	2 min.	17 min.
Profesor 2	3 min.	20 min.	3 min.	16 min.
Profesor 3	2 min.	14min.	4 min.	22 min.
Profesor 4	2 min.	18 min.	2 min.	14 min.

Tabla 3.1. Tabla de tiempos de creado de oficios de requerimientos por el sistema.

	Oficio inscripción		Oficio equipo	
	Tiempo con el sistema	Tiempo sin el sistema	Tiempo con el sistema	Tiempo sin el sistema
Profesor 1	1 min.	19 min.	2 min.	15 min.
Profesor 2	3 min.	22 min.	1 min.	14 min.
Profesor 3	2 min.	15min.	3 min.	16min.
Profesor 4	4 min.	14 min.	3 min.	18 min.

Tabla 3.2. Tabla de tiempos de creado de oficios de requerimientos por el sistema.

Con esto nos damos una idea del tiempo que tarda en realizarse un oficio de requerimientos con el sistema desarrollado y el sistema actual.

Análisis y Discusión de resultados

Se obtuvo en la sección anterior un total de 16 pruebas (4 pruebas de cada tipo de oficio creado), todos estos dando una media de 2.5 minutos aproximadamente.

Los errores que el sistema presento fue por que el profesor no inserto los datos correctos (incompatible) esto genero un retraso con el sistema dando los picos de 4 minutos.

También se pudo observar otro caso donde el profesor inserto muchas palabras en el campo de texto esto genero los picos de 4 minutos.

En los picos más bajos donde el profesor no insertó mucha información y no presento ningún error.

Con esto nos damos una idea del tiempo que nos va a llevar realizar un oficio de requerimientos con el sistema desarrollado.

Conclusiones

Con la creación del sistema para generar reportes de requerimientos se puede observar un cambio radical en el tiempo de creado, ya que el sistema se vuelve meramente dinámico al procesar los datos de entrada y de salida con esto se acelera el proceso y la demanda de creado de los Oficios.

La tecnología J2E nos permitió separar las tres capas y esto facilitó el trabajo que se tiene al acceder a cada una de ellas. Cada capa está definida y no puede ser accesada sin los datos correctos, es decir, no se puede brincar capas ya que el sistema está definido para seriarse con la próxima capa.

Con la creación de cada módulo se puede observar los procesos que desarrolla el sistema de forma dinámica con el único fin de agilizar el sistema, eliminar la replicación que pudiese existir, con la información presente en un solo lugar.

Se utilizó el modelo vista controlador ya que es el más completo al separar todas las partes del código y ocupándolas en el momento que se necesitan, con la finalidad de ahorrar memoria y que no haya procesos repetidos en el mismo instante.

El proyecto Automatización de Procesos para generar reportes de requerimientos en formato PDF es un sistema que pretende actualizar la forma de generar los reportes de requerimientos, las dimensiones y las funcionalidades del proyecto tiene la capacidad de flexibilidad ya que se puede incrementar en software y hardware y adaptarse sin perder el rendimiento que tiene, así como también se puede adecuar a nuevas tecnologías de J2EE que nos permita.

Con el paso del tiempo se planea que este sistema no se quede así, sino que siga creciendo y dotando a el área que lo solicito de nuevas formas de generar trabajos, optimizando las que se encuentran en este momento.

Referencias bibliográficas

- [1] G. Becerril Velázquez and I. Gutiérrez Limón, "Replicador de configuración y estado de un manejador de base de datos relacional", Proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Unidad Azcapotzalco, México, 2013.
- [2] M. González Mondragón, "Servicios Web para la gestión de conocimiento usando un modelo ontológico", Proyecto terminal, Universidad Autónoma Metropolitana Unidad Azcapotzalco, México, 2015.
- [3] S. Avendaño Méndez, "Gestión de calificaciones de cursos mediante servicios Web", Proyecto terminal, Universidad Autónoma Metropolitana Unidad Azcapotzalco, México, 2012.
- [4] *Creación de Ficheros LATEX con GNU Emacsx*, GNU Free Documentation License, 2004, <ftp://ftp.dante.de/tex-archive/info/spanish/guia-atx/guia-atx.pdf>.
- [5] D. López Barrón, "Implementación de un Modelo de Base de Datos para el Almacenamiento y Consulta de Electrocardiogramas Digitales", Licenciatura, Universidad Autónoma de Sinaloa, 2011.
- [6] E. Anaya López, "IMPLEMENTACIÓN DE CONTROLES DE SEGURIDAD EN ARQUITECTURAS ORIENTADAS A SERVICIOS (SOA) PARA SERVICIOS WEB", Maestro en Ciencias, Instituto Politécnico Nacional, 2011.
- [7] Hernández Trasobares, A. (2003). Los sistemas de información evolución y desarrollo. *Proyecto social: Revista de relaciones laborales*, (ISSN 1133-3189), pp.149-165.
- [8] El Universal. (2017). *Con acceso a telefonía móvil 94% de la población mundial*. [online] Available at: <http://www.eluniversal.com.mx/articulo/techbit/2015/08/10/con-acceso-telefonía-móvil-94-de-la-población-mundial> [Accessed 18 Jul. 2017].
- [9] Trujillo Hernández, L. (2012). *Metodología para determinar la evolución de los sistemas informáticos de organizaciones regionales*. 1st ed. [ebook] Colombia: Universidad de Manizales, pp.152-166. Available at: <http://revistasum.umanizales.edu.co/ojs/index.php/ventanainformatica/article/viewFile/188/236> [Accessed 4 Jul. 2017].
- [10] Gómez Fuentes, M. (2013). *Notas del Curso Bases de Datos*. 1st ed. México D.F.: Universidad Autónoma Metropolitana, pp.1-15.
- [11] Quintas Ripoll, L. (2017). *SISTEMAS DE GESTIÓN DE BASES DE DATOS*. 1st ed. [ebook] EOI Escuela de Negocios, pp.3-8. Available at: <https://www.eoi.es/es/file/18427/download?token=p2ULNDJr> [Accessed 5 Jul. 2017].
- [12] Ecured.cu. (2017). *Socket - EcuRed*. [online] Available at: <https://www.ecured.cu/Socket> [Accessed 19 Jul. 2017].
- [13] JTagua. (2017). *Tutorial de Servlet (1): Introducción, Ciclo de Vida y Ejemplo básico*. [online] Available at: <https://jtagua.wordpress.com/2010/10/31/tutorial-de-servlet-1-introduccion-ciclo-de-vida-y-ejemplo-basico/> [Accessed 19 Jul. 2017].
- [14] Merelo Guervos, J. (2017). *Tutorial de JSPs, Java Server Pages*. [online] Geneura.ugr.es. Available at: <http://geneura.ugr.es/~jmerelo/JSP/> [Accessed 19 Jul. 2017].

[15] Adictosaltrabajo.com. (2017). *Introducción a JDBC | adictosaltrabajo*. [online] Available at: <https://www.adictosaltrabajo.com/tutoriales/introjdb/> [Accessed 19 Jul. 2017].

[16] Testdemecanografia.com. (2017). Promedio de Palabras por Minuto y cómo mejorarlo. [online] Available at: <http://www.testdemecanografia.com/promedio-ppm/> [Accessed 20 Jul. 2017].

Anexos

Anexo1

```
<form method="post" action="<%=request.getContextPath()%>/ValidaDatosLogin"
class="cajas_texto">

    <p>Numero Economico: </p><input type="text" name="numero_economico"
class="caja_texto_1">

    <div class="estira"></div>
    <br>
    <script LANGUAGE="JavaScript">function Entrar(password)
{window.location=password+".htm"}
</script>

    <p>Contraseña:</p><input type="password" name="password_profesor"
class="caja_texto_1" >

    <div class="estira"></div>
    <br>
    <input type="reset" value="Limpiar" id="boton_limpiar">
    <input type="submit" value="Entrar" id="boton_entrar" >
</form>

//Al regresarnos una respuesta satisfactoria nos enviara a la página principal del
//profesor que también forma parte del módulo de inicio sesión.

<section id="cuerpo_principal">
    <article class ="info">
        <h1>
            Información General
        </h1>
        <%
            String numero_economico=(request.getAttribute("numero_economico")).toString();
            String nivel_academico =(request.getAttribute
("nivel_academico")).toString();
            String nombre = (request.getAttribute("nombre")).toString();
            String apellido_p = (request.getAttribute("apellido_p")).toString();
            String apellido_m = (request.getAttribute("apellido_m")).toString();
            String correo = (request.getAttribute("correo")).toString();
            %>
        <p> Nombre: <%=nivel_academico + " " + nombre + " " + apellido_p + " " +
apellido_m%> </p>
        <p>Matrícula: <%=numero_economico %></p>
        <p>Correo: <%= correo %></p>
        <p>Trimestre Actual: 17P</p>
    </article>
</section>
```

Anexo2

//Se puede observar el código del Servlet en el cual recibimos dos parámetros y los //cuales corresponden al módulo de validación de sesión.

```
@WebServlet(name = "ValidaDatosLogin", urlPatterns = {"/ValidaDatosLogin"})
public class ValidaDatosLogin extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        PrintWriter out = response.getWriter();

        String numero_economico = request.getParameter("numero_economico");
        String password_profesor = request.getParameter("password_profesor");
```

Se puede observar el código del Servlet el cual corresponde al módulo de administración de la base de datos.

```
        Profesores pac= ProfesoresDTO.BuscarProfesor(numero_economico,
password_profesor);
        if(pac.getNombre()!="")
        {
            request.setAttribute("tipo_profesor",pac.getTipo_profesor());
            if( request.getAttribute("tipo_profesor").equals("Administrador"))
            {
                request.setAttribute("numero_economico", pac.getNumero_economico());
                request.setAttribute("nivel_academico", pac.getNivel_academico());
                request.setAttribute("nombre", pac.getNombre());
                request.setAttribute("apellido_p", pac.getApellido_p());
                request.setAttribute("apellido_m", pac.getApellido_m());
                request.setAttribute("correo", pac.getCorreo());
                request.getRequestDispatcher("/pagina_principalAdm.jsp").forward(request,
response);
            }
            else
            {
                request.setAttribute("numero_economico", pac.getNumero_economico());
                request.setAttribute("nivel_academico", pac.getNivel_academico());
                request.setAttribute("nombre", pac.getNombre());
                request.setAttribute("apellido_p", pac.getApellido_p());
                request.setAttribute("apellido_m", pac.getApellido_m());
                request.setAttribute("correo", pac.getCorreo());
                request.getRequestDispatcher("/pagina_principal.jsp").forward(request,
response);
            }
        }
        else
        {
            request.getRequestDispatcher("/errorLogin.jsp").forward(request,
response);
        }
        out.close();
    }
}
```

Anexo3

//Esta parte son una serie de ligas que nos envían a las diferentes páginas del //sistema.

```
<section class="menu_horizontal">
  <nav>
    <ul class="menu_h_varios">
      <li>
        <a id="titulo" href="#">Crear Oficios</a>
        <ul>
          <li><a id="menu_titulo"
href="pantallasoficiosviaje/llenar_oficio_viaje.jsp">Oficio de Viaje</a></li>
          <li><a id="menu_titulo"
href="pantallasoficiosviaticos/llenar_oficio_viaticos.jsp">Oficio de Viatico</a></li>
          <li><a id="menu_titulo"
href="pantallasoficiosinscripcion/llenar_oficio_inscripcion.jsp">Oficio de
Inscripción</a></li>
          <li><a id="menu_titulo"
href="pantallasoficiosequipo/llenar_oficio_equipo.jsp">Oficio de Equipo</a></li>
        </ul>
      </li>
      <li>
        <a id="titulo" href="#">Mostrar Oficios</a>
        <ul>
          <li><a id="menu_titulo"
href="pantallasoficiosviaje/formulario_oficio_viaje.jsp">Oficios de Viaje</a></li>
          <li><a id="menu_titulo"
href="pantallasoficiosviaticos/formulario_oficio_viaticos.jsp">Oficios de
Viaticos</a></li>
          <li><a id="menu_titulo"
href="pantallasoficiosinscripcion/formulario_oficio_inscripcion.jsp">Oficios de
Inscripcion</a></li>
          <li><a id="menu_titulo"
href="pantallasoficiosequipo/formulario_oficio_equipo.jsp">Oficios de Equipo</a></li>
        </ul>
      </li>
      <li>
        <a id="titulo" href="index.jsp">Salir</a>
      </li>
    </ul>
  </nav>
</section>
```

Anexo4

```
//Plantilla para crear un oficio

<form method="post" action="<%=request.getContextPath()%>/ValidaOficioViaje"
class="cajas_texto">
  <label>
    Por medio del presente se solicita su apoyo para realizar las gestiones
necesarias para que se lleve a
    cabo el reembolso al
    <select name="profesor_id_profesor" SIZE=1 <!--
onChange="javascript:alert('prueba');-->"
    <%
      ArrayList<Profesores> profes= new ArrayList<Profesores>();
      ProfesoresDTO p= new ProfesoresDTO();
      profes= p.MostrarTodosProfesores("");
      for(int i=0;i<profes.size();i++)
      {
        %>
        <option
value="<%=profes.get(i).getId_profesores()%>"><%=profes.get(i).getNombre()%></option>
        <%
      }
    %>
  </select>
  por concepto del viaje por la asistencia al siguiente evento:
  <br>
  <br>
  a) Viaje por un monto de con un costo $<input type="text" name="costo_viaje"
class="caja_texto_1">
  MXN, por haber asistido a la conferencia<input type="text"
name="nombre_conferencia_viaje" class="caja_texto_1">,
  en el cual se presento el siguiente articulo de investigación:
  <br>
  1)"<input type="text" name="nombre_presentacion_viaje"
class="caja_texto_1">".
  <br>
  Autores:<input type="text" name="autores_presentacion_viaje"
class="caja_texto_1">.
  <br><br><br>
  Este monto saldrá de los recursos del Área de Sistemas de Informacion
Inteligentes de la partida 45 "Gastos de
  viaje".
  <br><br>
  Anexo copia del gafete, la primera hoja de los articulos de investigacion
publicados, boleto de avión y factura del
  cobro.
  <br><br>
  Mucho le agradeceré la atebción que preste a esta solicitud, ya que esta
asistencia sirvió para la difusión de la
  investigación en conjunto del Área.
  <br><br>
  Sin mas por el momento, quedo atento a cualquier aclaración.
</label>
<br><br>
<label style="margin-left:30%;">_____</label>
<br>
<label style="margin-left:35%;"> Dr. José Alejandro Reyes Ortoz </label>
<br>
<label style="margin-left:27%;">Jefe del Área de sistemas de Información
Inteligentes</label>
```

```

        <br><br><br><br>
        <input type="submit" name="guard" value="Guardar" id="boton_guarar" >
        <input type="reset" value="Limpiar" id="boton_limpiar">
</form>
//Valores de los botones que van a ser enviados al servlet

//Parte del servlet ValidaOficioViaje

if("Guardar".equals(request.getParameter("guard")))
    {
        OficioViaje pac = new OficioViaje();

if(request.getParameter("costo_viaje").equals("")||request.getParameter("nombre_conferencia_viaje").equals("")||request.getParameter("nombre_presentacion_viaje").equals("")||request.getParameter("autores_presentacion_viaje").equals(""))
    { //en esta parte verifica que los campos estén llenos

if(request.getParameter("costo_viaje").equals("")&&request.getParameter("nombre_conferencia_viaje").equals("")&&request.getParameter("nombre_presentacion_viaje").equals("")&&request.getParameter("autores_presentacion_viaje").equals(""))
//en esta parte verifica que los campos estén llenos
        tipo="Todos los campos Incompletos";
        else if(request.getParameter("costo_viaje").equals(""))
            tipo="Campo del Costo Viaje Incompleto";
        else if(request.getParameter("nombre_conferencia_viaje").equals(""))
            tipo="Campo del Nombre de la conferencia Incompleto";
        else if(request.getParameter("nombre_presentacion_viaje").equals(""))
            tipo="Campo del Nombre de la Presentacion Incompleto";
        else
            tipo="Campo del Autores de la Presentacion Incompleto";
        request.getSession().setAttribute("error", tipo);

request.getRequestDispatcher("/pantallasoficiosviaje/error_oficios.jsp").forward(request, response);
    }
}

```

Anexo5

```
//plantilla del formulario para revisar los datos esten correctos

<form method="post" action="<%=request.getContextPath()%>/FormularioDatosViajes"
class="cajas_texto">
  <label>
    Dr. Jesus Isidro Gonzales Trejo
  <br>
  <b>Jefe del Departamento de Sistemas</b>
  <br> <br>
  Por medio del presente se solicita su apoyo para realizar las gestiones
necesarias para que se lleve a cabo el
  reembolso al <%= nivel_academico+ " "+nombre + " "+apellido_p+ " "+apellido_m%>
  por concepto del viaje por la asistencia al siguiente evento:
  <br><br>
  a) Viaje por un monto de con un costo $<%= " "+costo_viaje%> MXN, por haber
asistido
  a la conferencia <%= " "+nombre_conferencia_viaje%>, en el cual se presento el
siguiente articulo de investigación:
  <br>
  1) "<%= " "+nombre_presentacion_viaje%>".Autores: <%= "
"+autores_presesntacion_viaje%>
  <br><br><br>
  Este monto saldrá de los recursos del Área de Sistemas de Informacion
  Inteligentes de la partida 45 "Gastos de viaje".
  <br><br>
  Anexo copia del gafete, la primera hoja de los articulos de investigacion
  publicados, boleto de avión y factura del cobro.
  <br><br>
  Mucho le agradeceré la atebción que preste a esta solicitud, ya que esta
  asistencia sirvió para la difusión de la
  investigación en conjunto del Área.
  <br><br>
  Sin mas por el momento, quedo atento a cualquier aclaración.
  </label>
  <br><br><br><br><br><br>
  <label style="margin-left:30%;">_____</label>
  <br>
  <label style="margin-left:35%;"> Dr. José Alejandro Reyes Ortoz </label>
  <br>
  <label style="margin-left:27%;">Jefe del Área de sistemas de Información
  Inteligentes</label>
  <br><br><br><br>
  <input type="submit" name="Genofiviaje" value="GenerarPDF"
  id="boton_generarpdf" >
  </form>

//parte del servlet que me envía a la clase Java
else if("GenerarPDF".equals(request.getParameter("Genofiviaje")))
{
  request.getRequestDispatcher("/pantallasoficiosviaje/PDFviaje.jsp").forward(request,
  response);
}

//código de la clase Java, me recupera la ruta y el nombre del oficio y lo abre en el
//visor PDF de la computadora
<%
  String FILE = "F:\\PROYECTO\\ProyectoWeb\\pdf\\OficiosViajes\\";
  String nombre = request.getAttribute("NombreArchivoViaje").toString();
```

```
System.out.println(nombre);
FILE=FILE+nombre;
System.out.println(FILE);

File archivo = new File(FILE);
FileInputStream ficheroInput = new FileInputStream(archivo);
InputStream in = new FileInputStream(FILE);
int tam= (int) archivo.length();

int tamañoInput = ficheroInput.available();
byte[] datosPDF = new byte[tamañoInput];
ficheroInput.read( datosPDF, 0, tamañoInput);

response.reset();
response.setHeader("Content-disposition","inline; filename=archivo.pdf" );
response.setContentType("application/pdf");
response.setContentLength(tamañoInput);
response.getOutputStream().write(datosPDF);

ficheroInput.close();
%>
```

Anexo6

//plantilla para modificar el oficio creado

```
<form method="post" action="<%=request.getContextPath()%>/ValidaOficioViaje"
class="cajas_texto">
  <label>
  <br>
  <h2>Numero de Reporte: <input type="text" name="id_oficio_viaje"
value="<%=id_oficio_viaje%" class="caja_texto_1" readonly></h2>
  <br>
```

Por medio del presente se solicita su apoyo para realizar las gestiones necesarias para que se lleve a cabo el

reembolso al

```
<select name="profesores_id_profesores" SIZE=1 <!--
onChange="javascript:alert('prueba');-->">
```

```
<%
```

```
ArrayList<Profesores> profes= new ArrayList<Profesores>();
```

```
ProfesoresDTO p= new ProfesoresDTO();
```

```
profes= p.MostrarTodosProfesores("");
```

```
for(int i=0;i<profes.size();i++)
```

```
{
```

```
%>
```

```
<option
```

```
value="<%=profes.get(i).getId_profesores()%>"><%=profes.get(i).getNombre()%></option>
```

```
<%
```

```
}
```

```
%>
```

```
</select>
```

por concepto del viaje por la asistencia al siguiente evento:

a) Viaje por un monto de con un costo \$<input type="text" name="costo_viaje" class="caja_texto_1" value="<%=costo_viaje%">

MXN, por haber asistido a la conferencia<input type="text" name="nombre_conferencia_viaje" class="caja_texto_1" value="<%=nombre_conferencia_viaje%">,

en el cual se presento el siguiente articulo de investigación:

<p></p>1)"<input type="text" name="nombre_presentacion_viaje" class="caja_texto_1" value="<%=nombre_presentacion_viaje%">".

<p></p> Autores: <input type="text" name="autores_presentacion_viaje" class="caja_texto_1" value="<%=autores_presesntacion_viaje%">".

Este monto saldrá de los recursos del Área de Sistemas de Informacion Inteligentes de la partida 45 "Gastos de viaje".

Anexo copia del gafete, la primera hoja de los articulos de investigacion publicados, boleto de avión y factura del

cobro.

Mucho le agradeceré la atebción que preste a esta solicitud, ya que esta asistencia sirvió para la difusión de la

investigación en conjunto del Área.

Sin mas por el momento, quedo atento a cualquier aclaración.

```
</label>
```

```
<br><br>
```

```
<br><br>
```

```
<label style="margin-left:30%;">_____</label>
```

```

        <br>
        <label style="margin-left:35%;"> Dr. José Alejandro Reyes Ortoz </label>
        <br>
        <label style="margin-left:27%;">Jefe del Área de sistemas de Información
        Inteligentes</label>
        <br><br><br><br>
        <input type="submit" name="modi" value="Modificar" id="boton_modificar" >
        <input type="reset" value="Limpiar" id="boton_limpiar">
//Valores de los botones que van a ser enviados al servlet
</form>

//Parte del Servlet toma el Value y Name del voton y lo compara
else if("Modificar".equals(request.getParameter("modi")))
{
    OficioViaje pac = new OficioViaje();

    if(request.getParameter("costo_viaje").equals("")||request.getParameter("nombre_conferencia_viaje").equals("")||request.getParameter("nombre_presentacion_viaje").equals("")||request.getParameter("autores_presentacion_viaje").equals("")||request.getParameter("id_oficio_viaje").equals(""))
    {

    if(request.getParameter("costo_viaje").equals("")&&request.getParameter("nombre_conferencia_viaje").equals("")&&request.getParameter("nombre_presentacion_viaje").equals("")&&request.getParameter("autores_presentacion_viaje").equals("")&&request.getParameter("id_oficio_viaje").equals(""))
        tipo="Todos los campos Incompletos";
    else if(request.getParameter("costo_viaje").equals(""))
        tipo="Campo del Costo Viaje Incompleto de Modificar";
    else if(request.getParameter("nombre_conferencia_viaje").equals(""))
        tipo="Campo del Nombre de la conferencia Incompleto";
    else if(request.getParameter("nombre_presentacion_viaje").equals(""))
        tipo="Campo del Nombre de la Presentacion Incompleto";
    else if(request.getParameter("autores_presentacion_viaje").equals(""))
        tipo="Campo del Autores de la Presentacion Incompleto";
    else
        tipo="Campo del Id del Reporte";
        request.getSession().setAttribute("error", tipo);
//Compara que los cuadros de texto este llenados.
request.getRequestDispatcher("/pantallasoficiosviaje/error_oficios.jsp").forward(request, response);
    }
    else{
        String costo_viaje=request.getParameter("costo_viaje");
        String
nombre_conferencia_viaje=request.getParameter("nombre_conferencia_viaje");
        String
nombre_presentacion_viaje=request.getParameter("nombre_presentacion_viaje");
        String
autores_presesntacion_viaje=request.getParameter("autores_presentacion_viaje");
//        String fecha_documento_viaje=request.getParameter("fecha_documento_viaje");
        String
profesores_id_profesores=request.getParameter("profesores_id_profesores");
        String id_oficio_viaje=request.getParameter("id_oficio_viaje");

        pac.setCosto_viaje(costo_viaje);
        pac.setNombre_conferencia_viaje(nombre_conferencia_viaje);
        pac.setNombre_presentacion_viaje(nombre_presentacion_viaje);
        pac.setAutores_presesntacion_viaje(autores_presesntacion_viaje);
//        pac.setFecha_documento_viaje(fecha_documento_viaje);
        pac.setProfesores_id_profesores(profesores_id_profesores);

```

```
        pac.setId_oficio_viaje(id_oficio_viaje);  
        String resp = OficioViajeDTO.ModificarOficioViaje(pac);  
        request.getRequestDispatcher("/pantallasoficiosviaje/oficios_exito.jsp").forward(request,  
        response);  
    }  
}
```

Anexo7

```
//Parte del servlet que me elimina un oficio
else if ("EliminarReporte".equals(request.getParameter("eli")))
{
    String valor_eli =
OficioViajeDTO.EliminarOficioViaje(request.getParameter("id_reporte").toString());

request.getRequestDispatcher("/pantallasoficiosviaje/formulario_oficio_viaje.jsp").forward(request, response);
    valor = "Eliminar Reporte";
}
```

Anexo8

```
//Esta parte validamos la informacion si esta complete la enviamos con exito
if(request.getParameter("costo_viaje").equals("")||request.getParameter("nombre_conferencia_viaje").equals("")||request.getParameter("nombre_presentacion_viaje").equals("")||request.getParameter("autores_presentacion_viaje").equals("")/*||request.getParameter("fecha_documento_viaje").equals("")*/)
{

if(request.getParameter("costo_viaje").equals("")&&request.getParameter("nombre_conferencia_viaje").equals("")&&request.getParameter("nombre_presentacion_viaje").equals("")&&request.getParameter("autores_presentacion_viaje").equals("")/*&&request.getParameter("fecha_documento_viaje").equals("")*/)
    tipo="Todos los campos Incompletos";
    else if(request.getParameter("costo_viaje").equals(""))
        tipo="Campo del Costo Viaje Incompleto";
    else if(request.getParameter("nombre_conferencia_viaje").equals(""))
        tipo="Campo del Nombre de la conferencia Incompleto";
    else if(request.getParameter("nombre_presentacion_viaje").equals(""))
        tipo="Campo del Nombre de la Presentacion Incompleto";
    else
        tipo="Campo del Autores de la Presentacion Incompleto";
    request.getSession().setAttribute("error", tipo);

request.getRequestDispatcher("/pantallasoficiosviaje/error_oficios.jsp").forward(request, response);
}
else{
    String costo_viaje=request.getParameter("costo_viaje");
    String
nombre_conferencia_viaje=request.getParameter("nombre_conferencia_viaje");
    String
nombre_presentacion_viaje=request.getParameter("nombre_presentacion_viaje");
    String
autores_presesntacion_viaje=request.getParameter("autores_presentacion_viaje");
//    String fecha_documento_viaje=request.getParameter("fecha_documento_viaje");
    String profesores_id_profesores=request.getParameter("profesor_id_profesor");
    pac.setCosto_viaje(costo_viaje);
    pac.setNombre_conferencia_viaje(nombre_conferencia_viaje);
    pac.setNombre_presentacion_viaje(nombre_presentacion_viaje);
    pac.setAutores_presesntacion_viaje(autores_presesntacion_viaje);
//    pac.setFecha_documento_viaje(fecha_documento_viaje);
    pac.setProfesores_id_profesores(profesores_id_profesores);

    String resp = OficioViajeDTO.RegistraOficioViaje(pac);
    if(!resp.equalsIgnoreCase("")){

request.getRequestDispatcher("/pantallasoficiosviaje/oficios_exito.jsp").forward(request, response);
    }
}
}
```

Anexo9

```
//Búsqueda parcial del número de reporte

OficioViaje pac =
OficioViajeDTO.BuscarOficioViaje(request.getParameter("id_reporte").toString());
    if(pac.getId_oficio_viaje()!="")
    {
        request.setAttribute("id_oficio_viaje",pac.getId_oficio_viaje());
        request.setAttribute("costo_viaje", pac.getCosto_viaje());
        request.setAttribute("nombre_conferencia_viaje",
pac.getNombre_conferencia_viaje());
        request.setAttribute("nombre_presentacion_viaje",
pac.getNombre_presentacion_viaje());

request.setAttribute("autores_presesntacion_viaje",pac.getAutores_presesntacion_viaje());

request.setAttribute("fecha_documento_viaje",pac.getFecha_documento_viaje());
        request.setAttribute("profesores_id_profesores",
pac.getProfesores_id_profesores());
        Profesores profe= new Profesores();
        if(pac.getProfesores_id_profesores()!="" &&
pac.getId_oficio_viaje()!="")
        {
            profe = OficioViajeDTO.BuscarunProfesores((String)
request.getAttribute("profesores_id_profesores"),(String)
request.getAttribute("id_oficio_viaje"));
            request.setAttribute("nivel_academico", profe.getNivel_academico());
            request.setAttribute("nombre", profe.getNombre());
            request.setAttribute("apellido_p", profe.getApellido_p());
            request.setAttribute("apellido_m", profe.getApellido_m());
        }

request.getRequestDispatcher("/pantallasoficiosviaje/PDFviaje.jsp").forward(request,
response);
//
request.getRequestDispatcher("/pantallasoficiosviaje/mostrar_oficio_viaje.jsp").forward(r
equest, response);
        valor = "Mostrar Reporte";
    }
}
```

Anexo10

```
//Ya recuperado los valores los enpaquetamos y se los enviamos para guardado
else{
    String costo_viaje=request.getParameter("costo_viaje");
    String
nombre_conferencia_viaje=request.getParameter("nombre_conferencia_viaje");
    String
nombre_presentacion_viaje=request.getParameter("nombre_presentacion_viaje");
    String
autores_presentacion_viaje=request.getParameter("autores_presentacion_viaje");
//    String fecha_documento_viaje=request.getParameter("fecha_documento_viaje");
    String profesores_id_profesores=request.getParameter("profesor_id_profesor");
    pac.setCosto_viaje(costo_viaje);
    pac.setNombre_conferencia_viaje(nombre_conferencia_viaje);
    pac.setNombre_presentacion_viaje(nombre_presentacion_viaje);
    pac.setAutores_presentacion_viaje(autores_presentacion_viaje);
//    pac.setFecha_documento_viaje(fecha_documento_viaje);
    pac.setProfesores_id_profesores(profesores_id_profesores);

    String resp = OficioViajeDTO.RegistraOficioViaje(pac);
    if(!resp.equalsIgnoreCase("")){

request.getRequestDispatcher("/pantallasoficiosviaje/oficios_exito.jsp").forward(request,
response);
    }
}
```

Anexo11

```
//código que me genera un PDF con los datos que el profesor guardo

String FILE = "E:\\Descargas\\PROYECTO\\ProyectoWeb\\pdf\\OficiosViaticos\\";
nombre=elementos.get("nombre").replace(" ", "");
nombre="OficioViaticos"+nombre+id_oficio;
FILE= FILE+nombre+".pdf";

Document document = new Document();
PdfWriter.getInstance(document, new FileOutputStream(FILE));
document.open();
addMetaData(document);
addTitlePage(document);
document.close();

Calendar cal = Calendar.getInstance();
Date fecha = new Date( cal.getTimeInMillis());
SimpleDateFormat formato = new SimpleDateFormat("EEEEEEEE dd MMMMMMMM yyyy");

Paragraph preface = new Paragraph();
Paragraph preface1 = new Paragraph();
Paragraph preface2 = new Paragraph();
Paragraph preface3 = new Paragraph();
Paragraph preface4 = new Paragraph();
Paragraph preface5 = new Paragraph();
Paragraph preface6 = new Paragraph();
Paragraph preface7 = new Paragraph();
Paragraph preface8 = new Paragraph();
Paragraph preface9 = new Paragraph();
Paragraph preface10 = new Paragraph();
Paragraph preface11 = new Paragraph();
Paragraph preface12 = new Paragraph();

//      preface = new Paragraph(elementos.get("fecha_documento_viaticos"),blackFont);
preface = new Paragraph(formato.format(fecha),blackFont);
preface.setAlignment(Element.ALIGN_RIGHT);
document.add(preface);

addEmptyLine(preface1, 1);
preface1.add(new Paragraph("Dr. Jesus Isidro Gonzales Trejo",blackFont));
preface1.add(new Paragraph("Jefe del Departamento de Sistemas",smallBold));
addEmptyLine(preface1, 2);
document.add(preface1);

preface2 = new Paragraph(" Por medio de la presente le solicito su apoyo para
realizar las gestiones necesarias para que se lleve a cabo el pago de viáticos por la
cantidad de $ "+elementos.get("cantidad_viaticos")+ " MXN, a cargo del "
+ "Área de Sistemas de Información Inteligentes, para que el
"+elementos.get("nivel_academico")+ " "+elementos.get("nombre")+
"+elementos.get("apellido_p")+ " "+elementos.get("apellido_m")+ ","
+ " asista a la conferencia
"+elementos.get("nombre_conferencia_viaticos")+ "a realizarse en
"+elementos.get("lugar_estado_capital_viaticos")+ ",
"+elementos.get("lugar_pais_viaticos")+ ", los días
"+elementos.get("fecha_evento_viaticos")+ ", en el cual "
+ "presentará el siguiente artículo:", blackFont);
preface2.setAlignment(Element.ALIGN_JUSTIFIED);
addEmptyLine(preface2, 1);
document.add(preface2);
```

```

        preface3 = new Paragraph("          a)
Authors:"+elementos.get("autores_presesntacion_viaticos")+", Título:
"+elementos.get("titulo_presentacion_viaticos")+".",blackFont);
        preface3.setAlignment(Element.ALIGN_JUSTIFIED);
        addEmptyLine(preface3, 2);
        document.add(preface3);

        preface4 = new Paragraph("Este monto serán cubierto con recursos del Área de
Sistemas de Información Inteligentes de la partida 46 “Viáticos”. Se solicita el cargo
por comprobar para el "+elementos.get("nivel_academico")+ " "+elementos.get("nombre")+
"+elementos.get("apellido_p")+ " "+elementos.get("apellido_m")+ " "
+ "lo más pronto que sea posible.",blackFont);
        preface4.setAlignment(Element.ALIGN_JUSTIFIED);
        addEmptyLine(preface4, 1);
        document.add(preface4);

        preface5.add(new Paragraph("Anexo la carta de aceptación y primera hoja del
trabajo.",blackFont));
        addEmptyLine(preface5, 1);
        preface5.add(new Paragraph("Mucho le agradeceré la atención que preste a esta
solicitud, ya que este pago de los artículos servirá para la difusión de la investigación
derivada del trabajo en conjunto del Área.",blackFont));
        addEmptyLine(preface5, 2);
        preface5.add(new Paragraph("Sin más por el momento, quedo atento a cualquier
aclaración.",blackFont));
        addEmptyLine(preface5, 6);
        document.add(preface5);

        preface6.add(new Paragraph("_____ ",blackFont));
        preface6.setIndentationLeft(150);
        document.add(preface6);
        preface7.add(new Paragraph("Dr. José Alejandro Reyes Ortiz",blackFont));
        preface7.setIndentationLeft(175);
        document.add(preface7);
        preface8.add(new Paragraph("Jefe del Área de Sistemas de Información
Inteligentes",blackFont));
        preface8.setIndentationLeft(120);
        document.add(preface8);
        document.newPage();
}

```