

**Universidad Autónoma Metropolitana Unidad Azcapotzalco  
División de Ciencias Básicas e Ingeniería**

**Licenciatura en Ingeniería en Computación**

**Análisis, diseño e implementación de sistemas de información, para departamento de publicaciones y reportes técnicos del Centro de Investigación en Computación (CIC).**

**Modalidad:** Experiencia profesional.

**Nombre del Alumno:**

José Carlos Cardoso Flores

Matricula: 205205110

**Datos de le empresa:**

Centro de Investigación en Computación, Instituto Politécnico Nacional, Departamento de publicaciones y reportes técnicos.

**Datos de la jefa:**

M. en B. Blanca Miranda Valencia, Jefa del departamento de publicaciones y reportes técnicos.

Trimestre 14 –I

## **Declaratoria**

*Yo, M. en B. Blanca Miranda Valencia, 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.*

---

*Yo, Cardoso Flores José Carlos, 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. Que deberá entregar al Coordinador de Estudios a más tardar el día programado para la evaluación global o de recuperación, según sea el caso, para su firma y sello.*

---

## Tabla de contenidos

<b>1. Resumen ejecutivo</b>	2
<b>2. Descripción de la empresa o institución</b>	3
<b>3. Descripción del departamento o sección</b>	3
<b>4. Descripción técnica de las actividades asociadas al puesto</b>	3-5
<b>5. Implementación y evaluación</b>	5
<b>6. Proyecto 1: Implementación de Open Journal System</b>	5-10
<b>7. Proyecto 2: Implementación de SIABUC9</b>	10-13
<b>8. Proyecto 3: Desarrollo de un sistema WEB para el proyecto FONCICYT 93829</b>	13-18
<b>Conclusiones</b>	18
<b>Bibliografía</b>	18
<b>Apéndice</b>	18-22

---

# 1. Resumen ejecutivo

El conjunto de actividades que analistas y diseñadores de sistemas realizan para desarrollar e implementar un sistema de información se conoce en la literatura como *método de ciclo de vida para el desarrollo de sistemas*. Este método consta de 6 fases: 1) investigación preliminar, 2) determinación de los requerimientos del sistema, 3) diseño del sistema, 4) desarrollo del software, 5) prueba de sistemas, y finalmente 6) la implementación y evaluación que incluye liberación al usuario.

## 1.1. Descripción de proyectos

**Primero y segundo proyecto;** la tarea principal era; de análisis y diseño de sistemas, de aquí que mi trabajo comprendiera cinco de las seis fases del método de ciclo de vida para el desarrollo de sistemas. Los sistemas *Open Journal System* está dedicado a la gestión y administración de revistas electrónicas. Por otro lado SIABUC9 es un sistema encargado de gestionar centros de información o bibliotecas. Ambos requirieron análisis y diseño de sistemas. La investigación preliminar, que abarca la primera fase del método de ciclo de vida para el desarrollo de sistemas, siempre requiere una petición de una persona, la cual fue implementar dichos sistemas de tal manera que pudieran utilizarse como herramientas que incrementaran el desempeño de *Departamento de Publicaciones y Reportes Técnicos y de la biblioteca del CIC*.

La determinación de los requerimientos del sistema, aspecto fundamental del análisis de sistemas es parte del proceso de especificación y se refiere a comprender todas las facetas importantes de la parte de la institución que esté íntimamente ligada al sistema, en este caso el personal del *Departamento de Publicaciones y Reportes Técnicos y de la Biblioteca del CIC*, fueron los objetos de estudio de este análisis. En el caso de estudio era de primordial interés estudiar los procesos de una organización, que en este caso estén ligados a los sistemas que se implementaron, con el fin de potenciar el funcionamiento de los sistemas en esta institución, este análisis debe dar respuesta a las siguientes preguntas clave:

1. ¿Qué es lo que hace?
2. ¿Cómo se hace?
3. ¿Con que frecuencia se presenta?
4. ¿Qué tan grande es el volumen de transacciones o decisiones?
5. ¿Cuál es el grado de eficiencia con el que se efectúan las tareas?
6. ¿Existe algún problema? ¿Qué tan serio es? ¿Cuál es la causa que lo origina?

Para dar respuesta a estas preguntas existen diversas herramientas que los analistas de sistemas utilizan frecuentemente. El levantamiento de requerimientos para los dos sistemas, se hizo utilizando el enfoque de entrevistas con el personal involucrado y redactando un documento estándar UML de requerimientos. Este último especifica un historial de revisiones, requerimientos funcionales y no funcionales, información de dominios de interés, y la descripción de lógica de negocio.

En el caso de estudio la parte de diseño de sistemas de información, determinó los detalles que establecen la forma en que el sistema cumple con los requerimientos identificados durante la fase de análisis. Es importante mencionar que el diseño y la fase de prueba de sistemas se ejecutó de forma paralela, durante estas fases los sistemas se emplearon de manera experimental con el fin de asegurar que el software se ajustara a las necesidades requeridas, es decir, que funciona de acuerdo con las especificaciones y en la forma en que los usuarios esperaban que lo hiciera. Se alimentaron los sistemas con un conjunto de datos de prueba para su procesamiento y después se examinaron los resultados.

La fase final fue la implementación y evaluación la cual se refiere al proceso de: 1) verificar e instalar el nuevo equipo, 2) capacitar a los usuarios, 3) instalar la aplicación y construir todos los archivos de datos necesarios para utilizarla. Una vez instaladas, las aplicaciones se emplean durante un cierto periodo y se le proporciona mantenimiento de ser necesario.

***En un tercer proyecto*** se repitieron las mismas etapas del *ciclo de vida del software*. En ese caso se me encomendó desarrollar un sistema WEB, que permitiera a los investigadores que participaron en el proyecto FONCICYT 93829: a) *compartir información referente a este proyecto*, y b) garantizar que el acceso a esta información fuera restringida. Con base en lo anterior decidí desarrollar el sistema WEB con *Java Struts Action Framework, Spring Framework y Hibernate*. Mi elección se basó en las siguientes premisas: 1) Java es una plataforma para el desarrollo de aplicaciones portables, basado en los principios del desarrollo orientado a objetos, y 2) UML (Lenguaje de Modelado Unificado), es una notación que permite el desarrollo de métodos orientados a objetos, lo que facilita trabajar siguiendo una metodología estándar que en el ámbito laboral es ampliamente usada.

En el caso de los Frameworks utilizados, es importante saber que estos son infraestructuras que constituyen la base para el desarrollo de sistemas de un dominio en particular, que permiten la velocidad de desarrollo debido al nivel de reúso de código.

## **2. Descripción de la empresa o institución**

El Centro de Investigación en Computación (CIC) del Instituto Politécnico Nacional, es una institución que realiza investigación científica de vanguardia orientada a: 1) enseñanza a nivel posgrado, 2) investigación básica y aplicada y 3) desarrollo tecnológico.

## **3. Descripción del departamento o sección**

El *Departamento de Publicaciones y Reportes Técnicos* se encarga de darle difusión a las investigaciones, tesis, artículos etc. Mediante la publicación de la revista *Computación y Sistemas*. Este último se encuentra la biblioteca del *Centro de Investigación en Computación*, y además se encarga de proporcionar el servicio de préstamo de libros para el personal académico y alumnos de los posgrados de este centro.

## 4. Descripción técnica de las actividades asociadas al puesto

Como ya lo mencioné mi trabajo fue de analista y diseñador de sistemas, para lograr un mayor detalle a continuación se hace una descripción de las fases previamente descritas.

*4.1. Investigación preliminar: levantamiento de entrevistas orientada al software.*

### 4.2. Determinación de los requerimientos del sistema

Levantamiento de entrevistas orientada a requerimientos. De acuerdo a los requerimientos del documento estándar UML basados en el formato IEEE-830.

### 4.3. Diseño del sistema

**El diseño del sistema se configura de los siguientes desarrollos:**

- Diagramas de proceso, estos diagramas describen de manera detallada que actividades intervienen en un proceso no necesariamente computacional, ejemplo: solicitudes de material de activo fijo o de consumo, las actividades abarcan desde el llenado del formato hasta la firma de autorización, ver Apéndice **Diagrama A1**.
- Diagramas de actividad, estos diagramas describen de manera detallada que actividades intervienen en un proceso computacional. Como ejemplo: retiro de un cajero, ver Apéndice **Diagrama A2**.
- Diagramas de casos de uso, estos diagramas permiten modelar el comportamiento de un sistema, un subsistema o una clase, muestra la interacción de los actores y sus relaciones.
- Diagramas de clases, estos diagramas describen la estructura de un sistema mostrando sus clases, orientados a objetos.
- Desarrollo de diseño del modelo entidad relación de la base de datos, esta es una representación del modelo de la base de datos utilizando un esquema gráfico empleando objetos o entidades principales que se identifican en el problema a resolver, estos se distinguen de otros por sus características particulares denominadas atributos, el enlace que rige la unión de las entidades está representada por la relación del modelo ver Apéndice **Diagrama A3**.
- Desarrollo de diccionario de datos de la base de datos, es un documento en el cual se plasman todas las entidades identificadas describiendo el tipo de datos de sus atributos, las restricciones, formato, el nombre etc.
- Diagrama de navegación para cada módulo web del sistema, en este diagrama se describe de manera grafica el esquema de modelación de la navegación del sistema web, donde se especifica los posibles caminos de navegación para el usuario, funciones que ejecutará en los documentos y enlaces.

#### **4.4. Desarrollo del software**

En esta fase a partir de los resultados obtenidos en la sección anterior se programó el modelo orientado objetos del sistema WEB para el proyecto FONCICYT 93829, en la plataforma Java.

#### **4.5. Prueba de sistemas**

Durante la etapa de pruebas de sistemas, se empleó el sistema de manera experimental para asegurarse de que el software desarrollado no tuviera fallas, y de haberlas corregirlas. Esta etapa solo se llevó a cabo en el caso del proyecto FONCICYT 93829. Para los otros dos proyectos se empleó el sistema de manera experimental, para determinar la funcionalidad generar la documentación UML necesaria y desarrollar manuales de usuarios o roles en el caso del sistema *OJS*.

### **5. Implementación y evaluación**

En la implementación de los sistemas *Open Journal System* y *SIABUC9*, fueron necesarias algunas configuraciones de servicios, de correo electrónico, de manejadores de bases de datos y de hoja de estilos de sitio. La liberación de estos dos sistemas fue exitosa brindando al *Departamento de Publicaciones y Reportes Técnicos* y a la *Biblioteca del CIC* herramientas útiles para mejorar su desempeño. El sistema para el proyecto FONCICYT también fue exitosamente liberado y usado por los investigadores que participaron en este proyecto.

En la implementación del sistema *Open Journal System*, la capacitación para los usuarios en el uso de este sistema constituía un aspecto esencial por ello los manuales desarrollados fueron un aspecto fundamental que permitieron que esta fase se llevará de forma ágil (se anexa parte de ellos).

### **Descripción detallada del o de los proyectos en los que se participó.**

#### **6. Proyecto 1: Implementación de Open Journal System**

El objetivo primordial para el Departamento de Publicaciones y Reportes Técnicos era que, mediante el uso del sistema *OJS*, aumentar el número de lectores de la revista, y aumentar la difusión entre la comunidad de investigadores, nacionales e internacionales.

#### ***Descripción Técnica***

*Open Journal System* es un sistema muy completo de gestión y administración de revistas electrónicas, basado en una estructura *cliente servidor*; la comunicación entre usuarios se realiza por medio del correo electrónico; siendo una parte fundamental del sistema *OJS*.

Es un software libre desarrollado por *Public Knowledge Project* y la universidad de California y ampliamente utilizado por instituciones nacionales e internacionales.

### *Características técnicas OJS*

Requerimientos recomendados del servidor:

1. **PHP** >= 4.2.x (incluyendo PHP 5.x); Microsoft IIS requiere PHP 5.x
2. **MySQL** >= 3.23.23 (incluyendo MySQL 4.x) o' **PostgreSQL** >= 7.1 (incluyendo PostgreSQL 8.x)
3. **Apache** >= 1.3.2x o' >= 2.0.4x o' Microsoft IIS 6
4. Sistema Operativo: Linux, BSD, Solaris, Mac OS X, Windows.

*OJS* es un software de código abierto y libre disposición de revistas en todo el mundo con el fin de que el acceso abierto es una opción viable para las revistas.

En este proyecto la tarea principal era hacer un análisis de sistemas, dentro de cinco de las seis fases del método de ciclo de vida para el desarrollo de sistemas.

Se realizó una entrevista para la investigación preliminar con la jefa del Departamento de Publicaciones y Reportes Técnicos, la Maestra Blanca Miranda Valencia. En esta entrevista quedó claro que la problemática principal era la pérdida de información al evaluar los artículos sometidos. Esto último causó pensar en automatizar el problema a través de una automatización y digitalización de los elementos involucrados en dicho proceso.

### **6.1. Fase de determinación de los requerimientos del sistema**

Se procedió al levantamiento de entrevistas; fase orientada a requerimientos con autores de artículos, correctores de estilo y personal que administra la revista *Computación y Sistemas*, se documentó mediante el formato estándar UML de requerimientos.

Para poder realizar la fase de diseño de sistemas y la fase de prueba de sistemas al mismo tiempo, se tuvieron que hacer las siguientes actividades:

- Instalación del sistema OJS versión 2.3.6 en el servidor de pruebas con sistema operativo Linux, instalación de las plataformas PHP 5.2.6 Y MySQL 5.0.6.
- Configuración de los parámetros en el archivo config.inc.php de OJS, que se muestran en la Tabla 1, con un mayor detalle. De acuerdo a la configuración del url de prueba, correo electrónico, base de datos y almacenamiento de archivos.

Tabla 1. Configuración de prueba de parámetros OJS

General		DataBase		Files	
base_url	http://localhost/revistacic	Driver	mysql	files_dir	/var/www/cicfiles
		username	root		
email	On	password	*****		
		Name	revistacicojs		
		persistent	On		

### 6.2. En la fase de diseño del sistema se desarrollo los siguientes diagramas de:

- Proceso de los roles principales que intervienen en el arbitraje de un artículo; autor, editor y corrector de estilo.
- Actividad, de los 7 roles que participan en este sistema *OJS*, autor, editor, editor de Sección, editor de maquetación, revisor, corrector y corrector de pruebas.
- Casos de uso, con las interacciones que los 7 roles tienen con cada subsistema de *OJS*.
- Navegación para cada módulo del sistema, donde se especifico documentos y enlaces que intervienen en cada módulo.

### 6.3. Fase de prueba de sistemas:

En esta fase de desarrollaron *pruebas piloto* creando un número de ejemplo de la revista *Computación y Sistemas*, se enviaron artículos de prueba al sistema *OJS*, para simular el arbitraje de dichos artículos.

#### *Manuales*

Se desarrollaron *siete manuales* correspondientes a los roles que participan en el proceso de arbitraje: 1) autor. **Figura 1**, 2) editor, 3) editor de Sección, 4) editor de maquetación, 5) revisor, 6) corrector, y 7) corrector de pruebas. Así como un *manual general* **Figura 2**, que guía paso a paso el proceso de arbitraje de un artículo, desde el envío hasta su publicación.



Para poder subir un nuevo artículo es necesario dar clic en "Comenzar Nuevo envío"

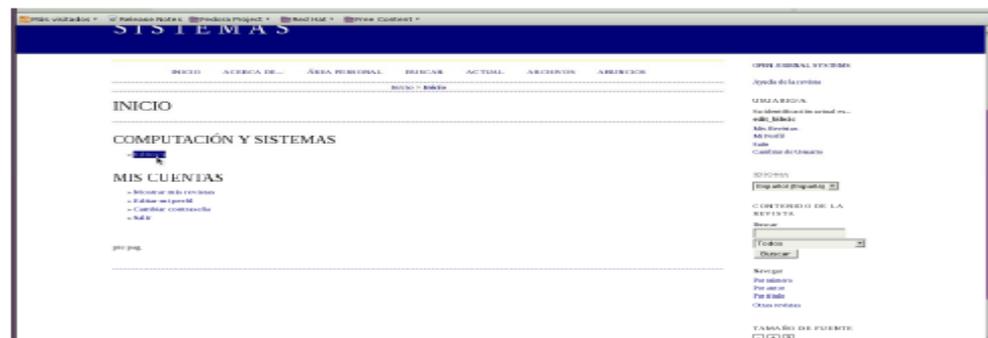


**Figura 1.** Manual de usuario para el rol de autor.

**PROCESO DE CREACION DE UN NÚMERO Y PROCESO DE CALIFICACION DE NUEVO ARTÍCULO**

**INICIO DE PROCESO CON: EDITOR**

Al iniciar se proporciona el login de EDITOR y aparece la siguiente pantalla dar clic en editor:



Para crear un número nuevo de la revista damos clic en "crear número"



**Figura 2.** Manual de usuario para el proceso general.

Configuración de la hoja de estilos de sitio archivo style.css, modificación de la apariencia del sitio web para los propósitos de la revista Computación y Sistemas **Figura3**.



**Figura3.** Apariencia general del sistema.

#### 6.4. Fase de implementación y evaluación:

Dentro de esta fase se procedió a la instalación del sistema OJS versión 2.3.6 en el servidor final con sistema operativo Linux, instalación de las plataformas PHP 5.2.6 Y MySQL 5.0.6.

En la Tabla 2, se muestra la configuración de los parámetros en el archivo config.inc.php de OJS. De acuerdo a la configuración del url final, correo electrónico, base de datos y almacenamiento de archivos.

Tabla 2. Configuración final de parámetros OJS.

General		DataBase		Files	
base_url	http://cys.cic.ipn.mx/ojs/index.php/CyS/	driver	mysql	files_dir	/var/www/cysfiles
		username	root		
email	On	password	*****		
		name	revistacys		
		persistent	On		

## **6.5. Fase de capacitación del Personal**

Se procedió a capacitar al personal del área de publicaciones en el manejo del sistema, administración de usuarios y proceso de arbitraje de un nuevo artículo utilizando el sistema *Open Journal System*.

Personal capacitado: Lic. Roció Soto, Lic. Claudia Andrade, Lic. Lourdes Salazar, M. en B. Blanca Miranda Valencia.

Dentro de esta fase se necesitó la elaboración de una guía de levantamiento de servicios: correo electrónico y de base de datos, previendo el reinicio o apagado del equipo servidor, para evitar pérdida de información y garantizar la comunicación mediante correos electrónicos. Con el fin de proporcionar un servicio de guía para el usuario. Cabe mencionar que la guía fue hecha por instrucción e impresa para ser proporcionada a quien lo requiriese.

### **Beneficios del sistema**

A partir de la liberación de la página de la revista se ha tenido una importante respuesta por parte de la comunidad recibiendo una gran cantidad de artículos en periodos cortos:

- noviembre a diciembre de 2010 se recibieron 80 artículos.
- enero a mayo de 2011 se recibieron 110 artículos.

Se ha casi duplicando el promedio anual recibido en años anteriores, haciendo un total de 190 artículos recibidos desde agosto de 2010 fecha de liberación del sistema.

Cabe señalar que esta revista es de corte internacional por lo que es notorio el hecho de que se hayan recibido artículos de diversos países de Latinoamérica, ampliando fuertemente la difusión de la revista *Computación y Sistemas*, consiguiendo que: la revista sea conocida a nivel internacional; lo anterior debido al uso de nuevas tecnologías para su difusión, aunado la facilidad de uso para someter y arbitrar artículos, en otras palabras lograr la automatización de procesos editoriales que anteriormente se hacían de manera sistemática pero de forma manual. Lo anterior realza los beneficios alcanzados al contar: 1) con un mayor control, y 2) una mejor administración de los artículos enviados.

## **7. Proyecto 2: Implementación de SIABUC9**

El objetivo principal de implementar el sistema SIABUC9 es que debido a las excelentes referencias que de él se tenían. Se requirió implementarlo y documentarlo, para mejorar el rendimiento de la Biblioteca del Centro de Investigación en Computación del IPN.

Se hace notar que SIABUC9 es el nombre del sistema no representa abreviatura alguna, ni son sus siglas en inglés.

## 7.1. Descripción Técnica

SIABUC9 es un sistema muy completo es un software de licencia, auxiliar en las labores cotidianas de un centro de información o biblioteca, con una arquitectura cliente - servidor, el funcionamiento de SIABUC está basado en módulos, cada módulo corresponde a una tarea específica dentro de la biblioteca, además es ampliamente utilizado por instituciones nacionales e internacionales.

A continuación se mencionan las características SIABUC9, entre los que se encuentran los requerimientos Recomendados:

Para la PC Cliente

- PC Pentium IV ó superior con 1 Gb en RAM y para cualquier sistema operativo Windows.

Para el Servidor

- Procesador Core 2 Duo con 2 Gb en RAM y un sistema operativo servidor, Windows 2003 Server ó superior, 2Gb de espacio libre en disco duro, monitor de video con resolución mínima de 1024x768.

Como ya se mencionó la tarea principal consistía en el desarrollo del análisis de sistemas, integrada por cinco de las seis fases del método de ciclo de vida para el desarrollo de sistemas.

La primera etapa de la investigación consistió en una entrevista con la Maestra Blanca Miranda Valencia encargada de la administración de la biblioteca del Centro de Investigación en Computación.

## 7.2. Fase de determinación de los requerimientos del sistema

Se realizó el levantamiento de entrevistas orientada a requerimientos con el personal que administra la biblioteca del CIC, se documentó mediante el formato estándar UML de requerimientos.

Para poder realizar la fase de diseño de sistemas y la fase de prueba de sistemas al mismo tiempo, se tuvieron que hacer las siguientes instalaciones:

- Sistema SIABUC9 en el servidor de pruebas con sistema operativo Windows Server 2003.
- Sistema SIABUC9 en PC clientes con sistema operativo Windows XP.
- Plataforma PostgreSQL 8.2 en todas las computadoras.

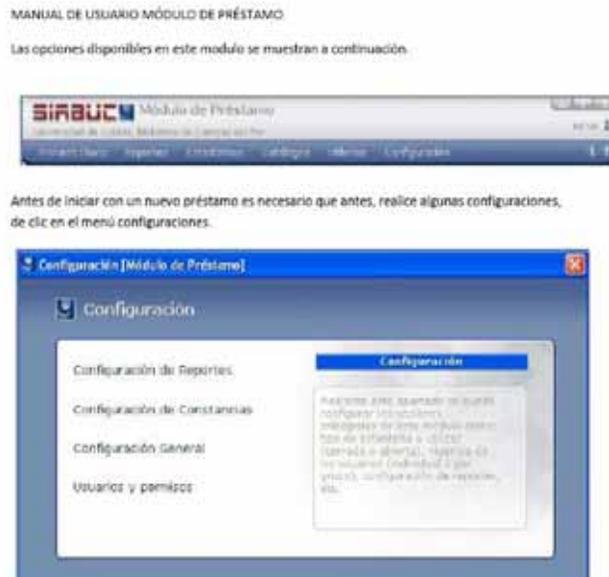
### 7.3. En la fase de diseño del sistema se desarrollo los siguientes diagramas:

- De proceso, del personal que intervienen en los procesos de préstamos y administración de la biblioteca.
- Actividad, de roles de usuario que participan en los módulos de préstamos y administración.
- Casos de uso, con las interacciones que los roles de usuario tienen con cada subsistema de SIABUC9.
- Navegación para cada módulo del sistema, donde fueron especificados: documentos y enlaces que intervienen en cada módulo.

### 7.4. En la fase de prueba de sistemas:

Se realizaron pruebas piloto donde se simula el préstamo de libros para el módulo de préstamos. En el caso del módulo de administración se simularon: solicitudes, adquisiciones, reportes, estadísticas y presupuestos.

Se desarrollaron dos manuales correspondientes a los roles que participan en el proceso de préstamos y administración de la biblioteca **Figuras 4 y5**. Además de la creación de un manual general que guía paso a paso el proceso de: préstamos, devoluciones, altas y bajas de libros.



**Figura 4.** Manual de usuario para el módulo de usuario.



**Figura 4.** Manual de usuario para el módulo de usuario.

### 7.5. En la fase de implementación y evaluación:

Se realizaron las siguientes instalaciones:

- Sistema SIABUC9 en el servidor final con sistema operativo Windows Server 2003.
- Sistema SIABUC9 en PC clientes con sistema operativo Windows XP.
- Plataforma PostgreSQL 8.2 en todas las computadoras.
- Capacitación al personal de la biblioteca en el manejo del: sistema, administración de usuarios y en los procesos del módulo de préstamos y administración en el sistema SIABUC9.

Personal capacitado: Lic. Leticia Quiroz, Lic. Guadalupe Sánchez, M. en B. Blanca Miranda Valencia.

## **8. Proyecto 3: Desarrollo de un sistema WEB para el proyecto FONCICYT 93829**

Como ya se mencionó el objetivo era desarrollar un sistema WEB capaz de enlazar al CIC-IPN y a otras universidades nacionales e internacionales, para de esta manera compartir la información del proyecto y que cada miembro de esta red de investigación pudiera tener acceso en todo momento a los datos obtenidos en el proyecto FONCICYT 93829. En este proyecto los investigadores tenían la tarea de construir robots de servicio sobre ruedas, cuyo objetivo principal era de detectar obstáculos (autos o peatones, entre otros), a partir de cámaras.

## 8.1. Descripción Técnica

Siguiendo el método de ciclo de vida para el desarrollo de sistemas se desarrolló este sistema WEB, la investigación preliminar, consistió en este caso de una entrevista con el Doctor Juan Humberto Sossa Azuela profesor investigador del Centro de Investigación en Computación y líder de proyecto FONCICYT 93829, en esta entrevista me comento que en este proyecto participaban cuatro universidades y cincuenta investigadores de diferentes países, la Universidad Libre de Berlín, la Universidad Complutense de Madrid, la Universidad de Guadalajara y el Centro de Investigación en Computación del IPN, por lo cual se requería un medio de enlace de información entre los participantes, y de esta forma no depender tan solo del correo electrónico; debido a la incomodidad además de que la información relacionada a dicho proyecto, no se encontraba de forma accesible en ningún momento por los investigadores participantes.

Por otro lado era necesario garantizar que el acceso a esta información fuera restringida.

Con base en las premisas anteriores se desarrollo el sistema WEB con Java Struts Action Framework, Spring Framework y Hibernate.

Struts Framework es un Framework de código abierto, que sirve para agilizar la creación de sistemas WEB con el modelo vista controlador (MVC) sobre la plataforma Java.

Spring Framework es una infraestructura ligera para integración de servicios en aplicaciones Java (IoC), y que da soporte a JDBC (Java Database Connectivity), JPA, Hibernate y a otras tecnologías.

Hibernate es una infraestructura de código abierto de persistencia para Java, que permite el mapeo flexible y de amplio uso en la industria.

El modelo vista controlador (MVC) puede definirse como un modelo que solo le permite al usuario interactuar con la vista del sistema (pagina web), de tal manera que no es posible que el usuario pueda acceder al controlador del sistema, en otras palabras no puede tener acceso al mecanismo que controla la atención de peticiones, la lógica de navegación, ni a la lógica de negocio.

El sistema web desarrollado está basado en una arquitectura cliente servidor, el usuario se registra en el sistema, accede mediante un nombre de usuario y contraseña, el usuario es capaz de compartir cualquier tipo de información electrónica mediante una herramienta de subida de archivos programada en el lenguaje Java. Para poder hacerlo es necesario elegir qué tipo de información se va a compartir, artículos, tesis, libros imágenes etc.

Requerimientos del Servidor:

- Apache tomcat 6.0.
- PostgreSQL 8.4.
- JAVA EE JDK 6.0
- Sistema Operativo: Linux, Windows.

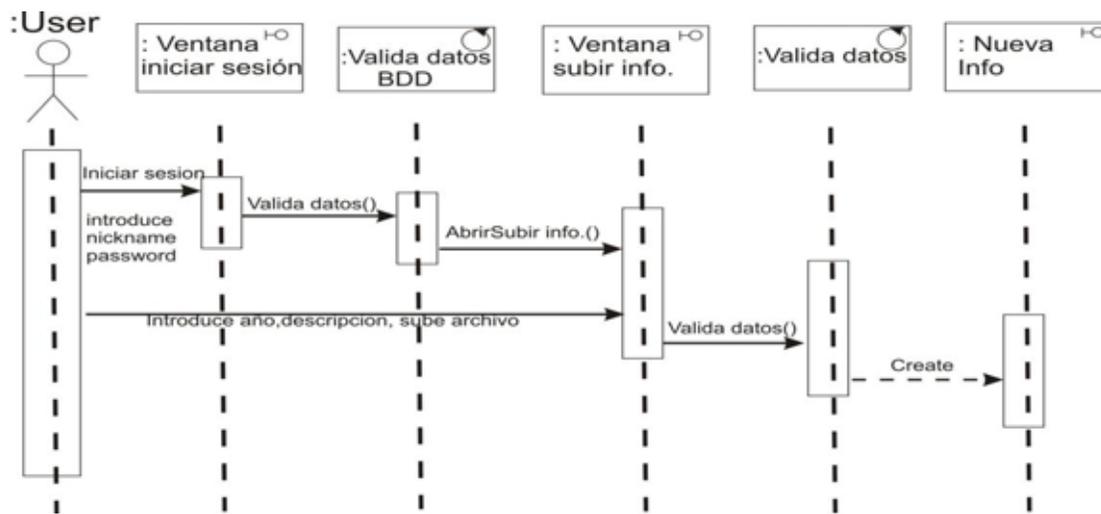
## 8.2. Determinación de los requerimientos del sistema

Se realizó un levantamiento de entrevistas orientadas a requerimientos. El Doctor Juan Humberto Sossa Azuela, líder de proyecto FONCICYT, fue la persona encargada y la información se procesó de acuerdo a la documentación de los requerimientos del formato IEEE-830.

## 8.3. Fase de diseño del sistema

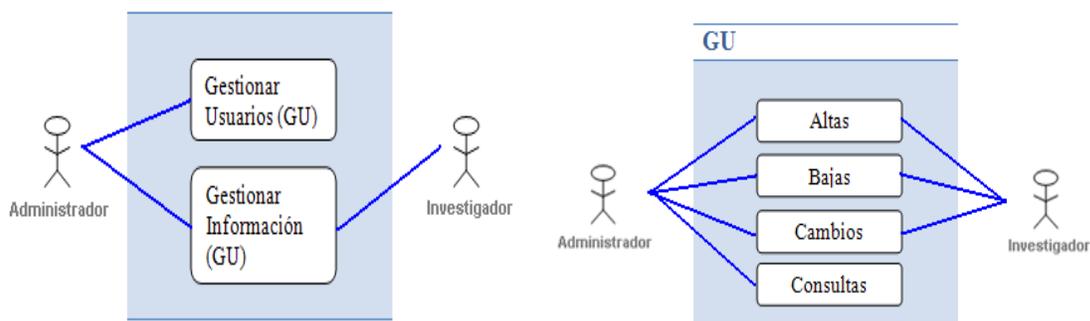
Se desarrollaron los diagramas de proceso, éstos detallan qué actividades intervienen en el proceso, en el que, un investigador desea compartir algún tipo de información útil para el proyecto.

A continuación se detallan los diagramas de actividad, estos diagramas especifican qué actividades intervienen cuando un usuario del sistema desea realizar un proceso: acceder al sistema, compartir una tesis, compartir una imagen, programa etc. o cuando desea acceder a cualquier información, para mayor detalle ver **Figura 5**.



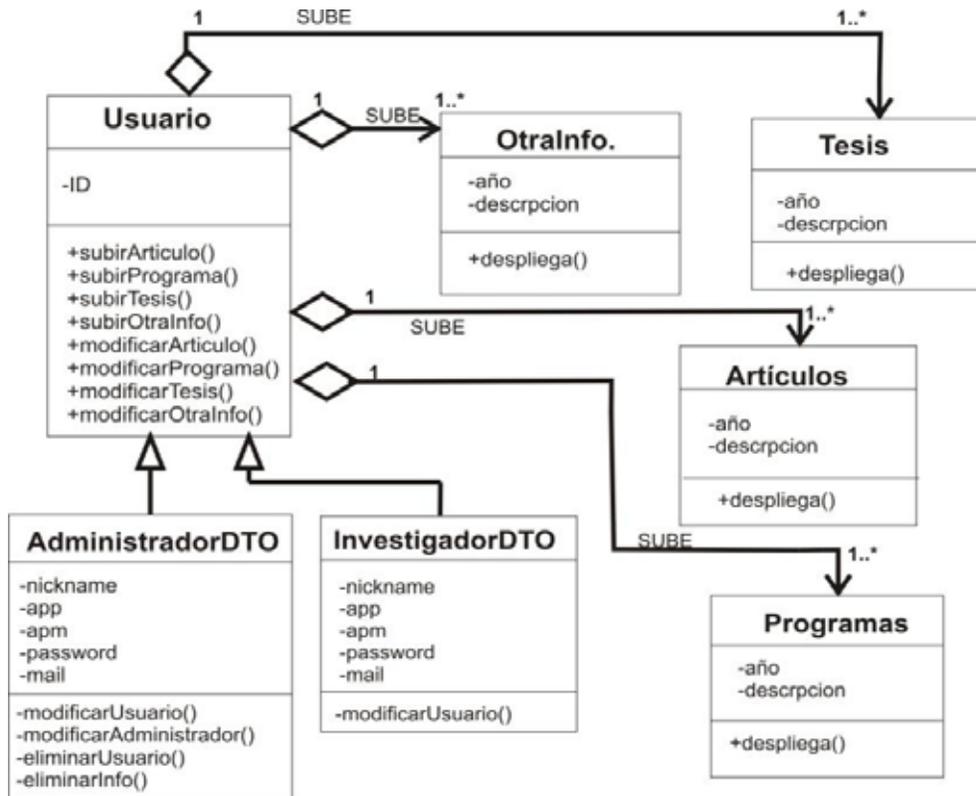
**Figura 5.** Diagrama general de actividad de acceso al sistema y de subida de información.

Diagramas de casos de uso, estos diagramas permitieron modelar el comportamiento de las clases, la interacción de los actores y sus relaciones **Figura 6**.



**Figura 6.** Diagrama de casos de uso del administrador de usuario y diagrama de casos de uso de gestión de información.

Desarrollo de diagramas de clases, en el diagrama **Figura 7**, describe la estructura general del sistema mostrando sus clases.



**Figura 7.** Diagrama general de clases del sistema.

Desarrollo de diseño del modelo entidad relación de la base de datos, es la representación del modelo de la base de este sistema.

Creación del diccionario de datos de la base de datos, a partir del modelo entidad relación, en este documento se detallaron las entidades identificadas describiendo el tipo de datos de sus atributos, las restricciones, formato, el nombre etc.

Desarrollo de diagrama de navegación para cada módulo web del sistema, en este diagrama se describió de manera grafica el esquema de modelación de la navegación del sistema web, donde se especifica los caminos de navegación para el usuario, funciones a ejecutar y la información requerida.

## 8.4 Desarrollo del software

En esta fase a partir del análisis y diseño se programó el modelo orientado objetos del sistema web para el proyecto descrito en el diagrama de clases, se utilizó el Struts Framework para controlar el MVC, la vista final del sistema se implementó en páginas jsp **Figura 7**. Para dar soporte al mapeo de las clases y sus atributos se utilizó Spring Framework y Hibernate. En el Anexo único se detalla el código de una interfaz que gestiona los usuarios que acceden al sistema.



**Figura 7.** Apariencia general del sistema web.

## 8.5 Prueba de sistemas

Durante la prueba de sistemas, se empleó el sistema de manera experimental para asegurarse de que el software no tuviera fallas, y de haberlas se corregían de manera inmediata. A la par de las pruebas piloto se desarrolló un manual de usuarios, el cual proporcionó una herramienta eficaz que permitió a los investigadores usar correctamente este nuevo sistema.

## 8.6 Implementación y evaluación

El sistema para el proyecto FONCICYT fue exitosamente liberado, el sistema web se montó en un servidor con sistema operativo Linux, se instaló Apache tomcat 6.024, PostgreSQL 8.4 y JAVA EE JDK 6.0.

El proyecto FONCICYT al ser un proyecto internacional, y debido a que se requería la capacitación de los usuarios, el manual desarrollado en la fase de prueba del sistema, fue enviado a los investigadores vía correo electrónico. De esta manera el sistema fue usado exitosamente por los investigadores durante un periodo de 6 meses, se compartieron tesis, artículos, y sobre todo código de programa referente a la autonomía de los robots.

## Conclusiones

La experiencia adquirida en estos proyectos no la hubiese podido obtener sin las habilidades necesarias, en cada proyecto la rapidez con la que concluyera cada proyecto era fundamental. Es por ello que pongo en alta estima el trabajo bajo presión, velocidad, eficacia y determinación sobre cualquier virtud. Estas habilidades puedo decir con orgullo, fueron forjadas en la Universidad Autónoma Metropolitana - Azcapotzalco pues es gracias al método de estudio, y a la exigencia que se nos exige en cada materia como podemos alcanzar estas virtudes.

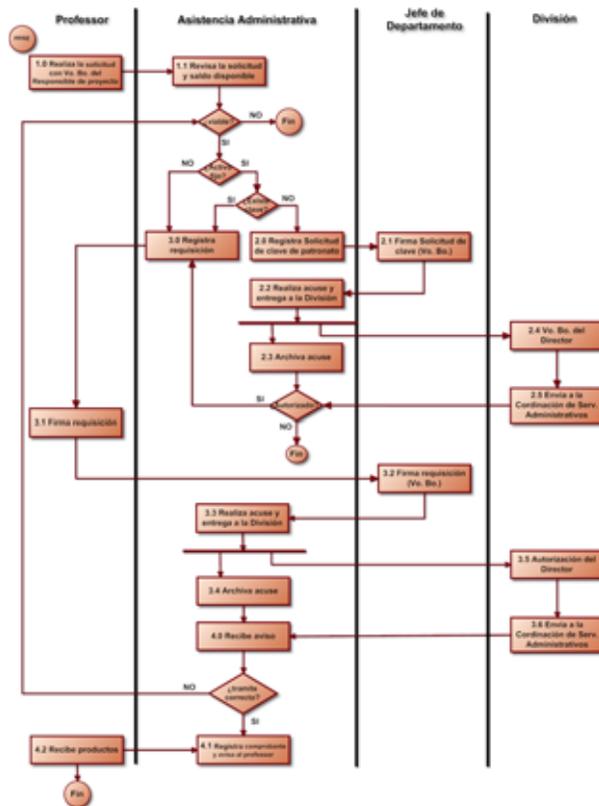
Puedo decir que si valoramos aquello que tenemos podremos afrontar cualquier tipo de reto sin importar que tan grande sea, es así como puedo decir que mi paso por el CIC del IPN fue exitoso.

## Bibliografía

- [1] L. Craig, "UML y Patrones", Edit. Pearson Alhambra, 2003.
- [2] G Booch, J Rumbaugh, I Jacobson," THE UNIFIED MODELING LANGUAGE USER GUIDE", Edit. Addison-Wesley, 1999.

## Apéndice

**Diagrama A1.** Ejemplo de diagrama de proceso para solicitud de requisición de activo fijo de consumo.





Código de la clase `AdministradorUsuariosImpl` que implementa la interfaz `AdministradorUsuarios` encargada de gestionar los usuarios que acceden al sistema.

```
/**
 *
 */
package com.analisis.bussines.local;
import com.analisis.bussines.AdministradorUsuarios;
import com.analisis.datos.InvestigadorDTO;
import com.analisis.datos.UserDTO;
import org.hibernate.criterion.Order;
import org.hibernate.criterion.Property;

import java.util.*;

import org.hibernate.*;
/**
 * @author CARLOS
 */
public class AdministradorUsuariosImpl implements AdministradorUsuarios
{
    private int _estado;
    private String _salida;
    private List<Integer> _indices;
    private UserDTO _user;
    private AdministradorUsuarios _administradorusuarios;
    private SessionFactory _sessionFactory;

    public UserDTO leerUsuario(long id)
    {
        Session session = getSession();
        return ( UserDTO )session.get( UserDTO.class, id );
    }

    public InvestigadorDTO leerInvestigador( long id )
    {
        Session session = getSession();
        //System.out.println("leer investigador");
        return ( InvestigadorDTO )session.get( InvestigadorDTO.class, id );
    }

    public long insertarInvestigador( InvestigadorDTO inv )
    {
        Session session = getSession();
        //return 0;
        session.save( inv );
        return (inv.getId());
    }

    public long insertarUsuario( UserDTO usuario )
    {
        Session session = getSession();
        session.save( usuario );
        return (usuario.getId());
    }

    public void setSessionFactory( SessionFactory sessionFactory )
    {
        _sessionFactory = sessionFactory;
    }

    private Session getSession()
```

```

{
    return _sessionFactory.getCurrentSession();
}

public void setAdministradorusuarios( AdministradorUsuarios
administradorusuarios )
{
    _administradorusuarios = administradorusuarios;
}
public void setEstado( int estado )
{
    _estado = estado;
    System.out.println(_estado);
}
public void setSalida( String salida )
{
    _salida = salida;
    System.out.println(_salida);
}
public void setIndices( List<Integer> indices )
{
    _indices = indices;
    System.out.println(_indices.get( 0 ).intValue());
    System.out.println(_indices.get( 1 ).intValue());
}

public void setUser( UserDTO user )
{
    _user = user;
    System.out.println(_user.getId());
    System.out.println(_user.getNombre());
}

}

public void actualizarUsuario( UserDTO usuario )
{
    Session session = getSession();
    session.update( usuario);
}

}

public void borrarUsuario( long id )
{
    Session session = getSession();
    UserDTO inv =(UserDTO )session.load(UserDTO.class, id );
    session.delete( inv );
}

}

public List<UserDTO> leerUsuarioQuery()
{
    Session session = getSession();
    Query query = session.createQuery( "SELECT t FROM UserDTO t" );
    return query.list();
}

}

public List<UserDTO> leerUsuarios( FiltroUsuarios filter )
{
    Session session = getSession();

```

```

Query query = session.getNamedQuery( "leer_usuarios_filtro" );
query.setString( "nombre", "%" + filter.getNombre() + "%" );
return query.list();
}

public Map<Long, String> cargarUsuarios()
{
    List<UserDTO> t = leerUsuarios();
    Iterator<UserDTO> i = t.iterator();
    UserDTO tt = new UserDTO();
    Map<Long, String> mapa = new TreeMap<Long, String>();
    while(i.hasNext())
    {
        tt = i.next();
        mapa.put( new Long(tt.getId()), tt.getNombre() );
    }
    return mapa ;
}

public List<UserDTO> leerUsuarios()
{
    Session session = _sessionFactory.getCurrentSession();
    Query query = session.getNamedQuery( "leer_usuarios" );
    return query.list();
}

public List<UserDTO> buscarTemasCriterioNombre( FiltroUsuarios filter )
{
    Criteria criteria = getSession().createCriteria( UserDTO.class );
    String nombre = "%" + filter.getNombre() + "%";
    criteria.add( Property.forName( "nombre" ).like( nombre ) );
    criteria.addOrder( Order.asc( "id" ) );
    return criteria.list();
}

public UserDTO leerUsuarioLoginPassword( String login, String password )
{
    Session session = _sessionFactory.getCurrentSession();
    Query query = session.getNamedQuery( "leer_usuario_login_pass" );
    query.setString( 0, login );
    query.setString( 1, password );
    List<UserDTO> lista = query.list();
    return lista.isEmpty() ? null : lista.get( 0 );
}
}

```