



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES
"ARAGÓN"

"SISTEMA DE ADMINISTRACIÓN DE
EVENTOS Y REGISTRO HISTÓRICO DE LA
UNIDAD DE EXTENSIÓN UNIVERSITARIA
DE LA ENEP ARAGÓN"

T E S I S

QUE PARA OBTENER EL TÍTULO DE :
INGENIERO EN COMPUTACIÓN
P R E S E N T A N:
JOSÉ ENRIQUE ATANACIO MIRANDA
ISRAEL TOLEDO SÁNCHEZ

ASESOR DE TESIS: ING. RODOLFO VÁZQUEZ MORALES

SAN JUAN DE ARAGÓN, EDO. MÉX.

2005

0350979



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos

A **Dios**, por haberme concedido el don de la vida, darme la paciencia y el empuje necesarios para desarrollarme plenamente.

A mi madre **Victoria Silvia Miranda Vázquez**, que con su amor, cuidados y ejemplo me hicieron un hombre de bien, así como por acompañarme en cada uno de mis pasos para mi formación profesional.

A mi hermano **Antonio Maya Miranda**, quien con su alegría me inspiró a seguir adelante y no darme por vencido.

A mi hermana **Erika Miranda Vázquez**, por el apoyo que siempre me brindó y la confianza que tuvo en mí.

A mis amigos **Abraham García Hernández** y **Emmanuel Mejía Valderrama**, por su gran ayuda para la elaboración de ésta tesis.

A mis amigos **Othón Navarrete Romero**, **Samuel Hernández Alfaro** y **Mónica Pérez Daniel**, por los momentos de alegría y consejos que me brindaron.

A mis **compañeros de generación** por su valiosa amistad.

A mi asesor y amigo **Ing. Rodolfo Vázquez Morales** e **Ing. Gladis Fuentes Chávez**, por apoyarme en todos los aspectos desde mi ingreso a este plantel, así como en la elaboración de la tesis.

A mis revisores: Ing. **Jeanett Figueroa Martínez**, Ing. **Norma Raquel Soto Arredondo**, Ing. **Esteban Ayala Peña** e Ing. **Mirna Miriam Ramírez Guerra** por dedicar su valioso tiempo a mis consultas y resolución de mis dudas.

A todo el **personal**, que labora en la escuela, que influyeron en mis estudios, durante mi paso en éste plantel.

A la **Universidad Nacional Autónoma de México** por haberme dado la oportunidad de desarrollarme profesionalmente en su seno.

Y muy especialmente a mi entrañable amigo **Isael Toiedo Sánchez** por apoyarme en todo el trayecto que recorrimos juntos, desde que ingresamos a la carrera hasta la culminación de nuestra tesis.

A todos ellos mil gracias.

José Enrique Atanacio Miranda

Agradecimientos

A **Nuestro Padre Celestial**, por haber nacido en el seno de una familia honorable y trabajadora además del haberme permitido gozar de salud para poder lograr mis objetivos.

A mis padres **Reynaldo Toledo Canseco** y **Eulalia Sánchez Valderrama**, que con sus atenciones, ternura, cuidados y cariño supieron conducirme por el camino del bien, y sus buenos ejemplos, lograron que llegara a este feliz momento.

A mi esposa **Lic. Carmen Sánchez Cadena**, por su amor, confianza y apoyo incondicional que mantuvo en mí desde el noviazgo, además de ser mi motivo de inspiración.

A mi hermano **Mtro. Reynaldo Toledo Sánchez** y su esposa **Lic. Ruth Morales Aponte**, por brindarme su apoyo desinteresado y oportuna asesoría.

A mi hermana **Rosalía Toledo Sánchez**, la más pequeña de la familia, por siempre ser una motivación para mí.

A mis **compañeros y amigos** de la FES Aragón, por su apoyo y la fraternidad que me mostraron.

A mi asesor y amigo **Ing. Rodolfo Vázquez Morales** e **Ing. Gladis Fuentes Chávez** por su apoyo desde el inicio de mi carrera hasta la terminación de mi tesis inclusive.

A mis revisores: **Ing. Jeanett Figueroa Martínez**, **Ing. Norma Raquel Soto Arredondo**, **Ing. Esteban Ayala Peña** e **Ing. Mirna Miriam Ramírez Guerra** por su apoyo, paciencia y buena voluntad hacia mi persona.

A todo el **personal docente, administrativo y manual** que labora en la escuela, que de manera directa e indirecta tuvieron ingerencia en el proceso de mi educación en este centro de estudios.

A la **Universidad Nacional Autónoma de México**, por ser mi alma mater.

A mi compañero de estudios y amigo **José Enrique Atanacio Miranda**, que desde el inicio de la carrera logramos integrarnos, primero para terminarla y segundo para cristalizar la tesis.

A todos ellos, le reitero mi consideración atenta y distinguida.

Gracias

Isael Toledo Sánchez

Índice

| | |
|--|----|
| Introducción | |
| Introducción..... | 2 |
| Capítulo I | |
| 1. Antecedentes..... | 5 |
| 1.1 Unidad de Extensión Universitaria..... | 5 |
| 1.2 Procesos de las áreas..... | 10 |
| 1.3 Problemáticas..... | 16 |
| 1.4 Propuesta de solución..... | 19 |
| Capítulo II | |
| 2. Definición de herramientas y uso de metodologías..... | 24 |
| 2.1 Sistema cliente-servidor..... | 26 |
| 2.2 Sistema web..... | 27 |
| 2.3 Metodologías para el desarrollo de sistemas..... | 30 |
| 2.3.1 Análisis..... | 36 |
| 2.3.2 Diseño..... | 37 |
| 2.3.3 Desarrollo o programación..... | 37 |
| 2,3,4 Implementación..... | 38 |
| 2.4 Herramientas..... | 38 |
| 2.4.1 Análisis..... | 38 |
| 2.4.1.1 Herramientas de análisis..... | 39 |
| 2.4.1.2 Diagrama de casos de uso..... | 45 |
| 2.4.1.3 Diagrama de clases..... | 46 |
| 2.4.1.4 Diagrama de objetos..... | 50 |
| 2.4.1.5 Diagrama de componentes..... | 51 |
| 2.4.1.6 Diagrama de flujo de datos (DFD)..... | 53 |
| 2.4.1.7 Diagrama entidad-relación..... | 54 |
| 2.4.2 Diseño..... | 55 |
| 2.4.2.1 Capa de usuario..... | 55 |
| 2.4.2.2 Diseño de interfaz gráfica..... | 55 |
| 2.4.2.3 Programación de cliente..... | 57 |
| 2.4.2.4 Capa de base de datos..... | 58 |
| 2.4.2.5 Capa de negocios..... | 59 |
| 2.4.2.6 Herramientas de diseño..... | 60 |
| 2.4.2.7 Presentación..... | 62 |
| 2.4.2.8 Reglas de negocio..... | 64 |
| 2.4.2.9 Acceso a datos..... | 67 |
| 2.4.2.10 Diferencia con la programación tradicional..... | 69 |
| 2.4.3 Desarrollo o programación..... | 76 |
| 2.4.3.1 Herramientas de desarrollo o programación..... | 77 |

| | |
|---------------------------|----|
| 2.4.4 Implementación..... | 81 |
|---------------------------|----|

Capítulo III

| | |
|--|-----|
| 3. Análisis y diseño del sistema..... | 84 |
| 3.1 Objetivo..... | 84 |
| 3.2 Alcance..... | 84 |
| 3.3 Requerimientos de infraestructura..... | 86 |
| 3.4 Diagrama de contexto..... | 87 |
| 3.5 Análisis de procedimientos..... | 88 |
| 3.6 Lista de procesos..... | 89 |
| 3.7 Diagrama de casos de uso..... | 89 |
| 3.7.1 Actores..... | 90 |
| 3.7.2 Casos de uso..... | 91 |
| 3.7.3 Descripción de casos de uso..... | 92 |
| 3.8 Diagramas de secuencia..... | 104 |
| 3.9 Diseño de interfaz (prototipo)..... | 111 |
| 3.10 Mapa de navegación..... | 115 |
| 3.11 Diagrama entidad-relación..... | 118 |
| 3.12 Reglas de negocio..... | 119 |

Capítulo IV

| | |
|---|-----|
| 4. Desarrollo e implementación..... | 122 |
| 4.1 Formulario de password..... | 122 |
| 4.2 Formulario principal de la aplicación..... | 123 |
| 4.3 Formulario del personal..... | 125 |
| 4.4 Formulario de acuerdos y/o convenios..... | 127 |
| 4.5 Menú elección de departamentos..... | 130 |
| 4.6 Módulo de botones..... | 131 |
| 4.7 Diccionario de datos..... | 133 |
| 4.8 Sistema de información..... | 137 |
| 4.8.1 Ciclo de vida de un sistema de información..... | 137 |
| 4.9 Implementación del sistema..... | 139 |
| 4.9.1 Infraestructura..... | 140 |
| 4.9.2 Instalación de la aplicación..... | 141 |
| 4.9.3 Pruebas de funcionamiento..... | 144 |
| 4.9.4 Capacitación..... | 145 |
| 4.9.5 Seguimiento..... | 145 |
| 4.9.6 Entrega de manuales..... | 147 |

Capítulo V

| | |
|----------------------------------|-----|
| 5. Mantenimiento y manuales..... | 149 |
| 5.1 Mantenimiento..... | 149 |
| 5.2 Manual de usuario EUSI..... | 152 |
| 5.2.1 Personal..... | 154 |

| | |
|-------------------------------------|-----|
| 5.2.2 Acuerdos..... | 157 |
| 5.2.3 Departamentos..... | 161 |
| 5.2.3.1 Actividades Culturales..... | 162 |
| 5.2.4 Reportes..... | 166 |
| Conclusiones | |
| Conclusiones..... | 170 |
| Bibliografía | |
| Bibliografía..... | 173 |
| Glosario | |
| Glosario..... | 177 |

Capítulo I

Antecedentes

1. Antecedentes

Uno de los programas estratégicos de la Facultad de Estudios Superiores Aragón, cuyo objetivo es contribuir a la formación integral del estudiante a través del fortalecimiento de la extensión y difusión de la cultura interna, además de ampliar el servicio que se brinda a la comunidad aledaña, es el programa de extensión y difusión de la cultura, el cual es desarrollado por la Unidad de Extensión Universitaria.

1.1 Unidad de Extensión Universitaria

Esta unidad es la encargada de administrar y coordinar las actividades artísticas, socioculturales, deportivas, recreativas y de vinculación, para cumplir con uno de los principales objetivos de la Universidad Nacional Autónoma de México. Se encuentra ubicada dentro de la estructura organizacional de la FES Aragón como se muestra en el siguiente diagrama. Nota: a la altura de la Unidad de Extensión Universitaria se encuentran diferentes áreas que para fines prácticos fueron omitidas.

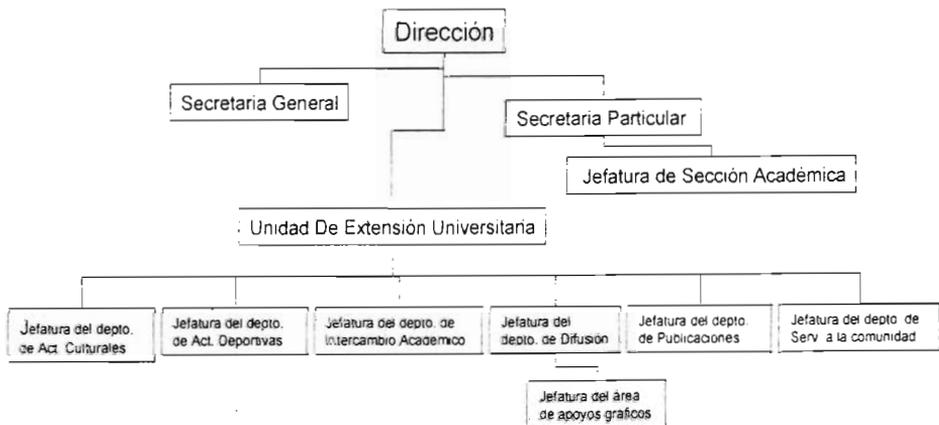


Figura 1: Organigrama de la FES Aragón

La Unidad de Extensión Universitaria, para realizar todas sus actividades, se divide en diferentes departamentos o áreas, como a continuación se muestran:

- Departamento de actividades culturales.
- Departamento de difusión.
- Departamento de intercambio académico y vinculación.
- Departamento de publicaciones.
- Coordinación de actividades deportivas y recreativas.
- Coordinación de servicios a la comunidad.
- Área de impresiones.
- Área de bolsa de trabajo.

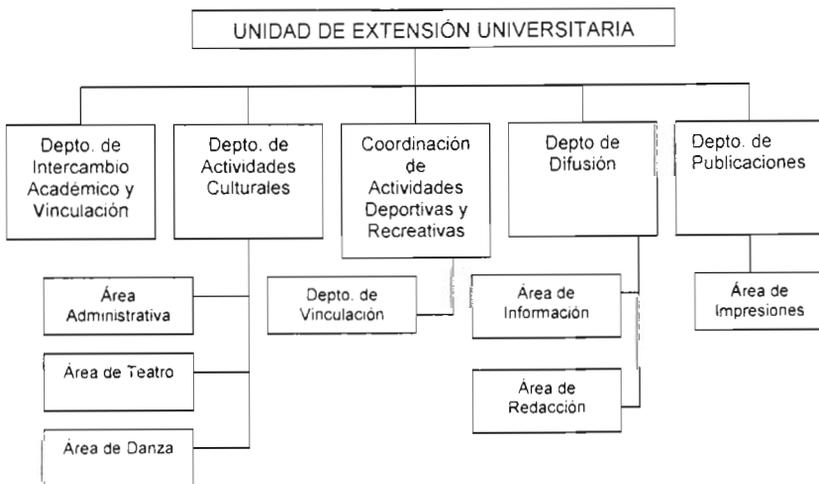


Figura 2: Organigrama de la Unidad de Extensión Universitaria

Estas ocho áreas son las principales cabeceras de la Unidad de Extensión Universitaria, todas se relacionan entre sí y además se generan otras subáreas, no menos importantes, y que no realizan tantas actividades como las principales.

Las actividades que realizan las diferentes áreas son:

1. **Departamento de actividades culturales.** Es el encargado de coordinar los siguientes eventos:

- Teatro.
- Danza.
- Cine.
- Conciertos.
- Proyección de películas.
- Conferencias.
- Talleres.
- Recitales.

La mayoría de estos eventos o actividades son realizadas en el Auditorio "José Vasconcelos", cada una de éstas se realizan por temporadas, las cuales se dan al menos dos por semestre y muchas otras son eventuales o en ocasiones especiales que elige la escuela, con todo esto, no existe mes en que no se realice una actividad en el auditorio, además de los diferentes foros que pueden ocupar las demás actividades dentro de la institución.

2. **Departamento de difusión.** Realiza lo siguiente:

- Coordinar la difusión de las actividades de todas las áreas.
- Enviar reportes a impresiones y publicación de los eventos.
- Difundir los eventos internos o externos.

Este departamento tiene la cualidad de estar directamente relacionado con el departamento de publicaciones, ya que la principal difusión dentro del Campus es escrita o gráfica, sin embargo, atiende diferentes tareas de la Institución en general y no sólo de la Unidad de Extensión Universitaria.

3. **Departamento de intercambio académico y vinculación.** Sus actividades son:

- Actualizar los convenios y acuerdos de la UNAM con otras instituciones.
- Establecer y realizar los intercambios académicos.
- Realizar exposiciones en instituciones aledañas a la FES para difundir el trabajo de la institución.
- Coordinar las actividades para la participación de la FES en las ferias universitarias.

Este departamento se encarga de renovar constantemente los diferentes convenios académicos de la Institución, así como de difundir los acuerdos escolares con diferentes planteles educativos para el beneficio de toda la población estudiantil.

4. **Departamento de publicaciones.** Se encarga de la publicación de:

- Propaganda.
- Trípticos.
- Volantes.
- Libros.
- Folletos.

Publica documentos e información de toda la institución, no sólo de la Unidad de Extensión Universitaria, además de publicar mensualmente la **Gaceta** del Campus, que es el diario informativo de más prestigio dentro de la FES, esta área siempre trabaja en conjunto con el departamento de difusión.

5. **Departamento de coordinación de actividades deportivas y recreativas.**

Coordina los torneos y eventos de:

- Fútbol soccer y rápido.
- Básquetbol.
- Voleibol.

- Ajedrez.
- Atletismo.
- Físico culturismo.
- Gimnasia olímpica.
- Lucha olímpica.
- Montañismo.
- Softbol.
- Tae kwon do.
- Béisbol.

En total, coordina más de cincuenta disciplinas deportivas. Se realizan por medio de competiciones internas o externas, debido a que la Institución maneja una serie de torneos con instituciones afines, ya sea dentro del Campus donde se cuentan con diferentes áreas deportivas o en otras instalaciones adecuadas, esto mantiene al departamento siempre ocupado en uno u otro deporte.

6. **Departamento de coordinación de servicios a la comunidad.** Sus actividades se apoyan con el Departamento de intercambio académico para coordinar los eventos a realizar con la comunidad aledaña, principalmente de carácter social.
7. **Área de impresiones.** Realiza las impresiones de todo el material que la institución necesita, auxiliado por el área de publicaciones, como los siguientes documentos:
 - Papelería de inscripción.
 - Diplomas y constancias.
 - Folletos.
 - Gaceta.
 - Libros.
 - Propaganda.
 - Carteles.

Se coordina con Publicaciones y aunque no sólo atiende a la Unidad de Extensión Universitaria, la mayor parte de trabajo proviene de ella.

8. **Área de bolsa de trabajo.** Se encarga de establecer relaciones con diferentes organizaciones y empresas para conocer las necesidades y perfiles requeridos de los recursos humanos a contratar.

A continuación se muestra un cuadro que presenta las principales actividades de todas las áreas.

Principales Actividades de las Áreas

| Áreas | Culturales | Difusión | Intercambio | Publicaciones | Deportivas | Comunidad | Impresiones | Bolsa |
|-------------|---|--|--|---|---|--|--|---|
| Actividades | Teatro Danza Cine Conciertos Proyección de películas Conferencias Talleres Recitales | Coordinar difusión de actividades Difusión oportuna de eventos internos o externos. | Actualizados los convenios y acuerdos Intercambios académicos Exposiciones en escuelas aledañas a la ENEP Forman representantes en las ferias universitarias. | Propaganda Tripticos Volantes Libros Folletos Gaceta | Torneos de fútbol soccer Básquetbol Voleibol Ajedrez Atletismo Físico culturismo Gimnasia olímpica Lucha olímpica Montañismo Softbol | Servicios Comunitarios a los lugares aledaños a la institución | Papelería de inscripción Folletos Gaceta Libros Propaganda Carteles | Bolsa de trabajo ofrecida por instituciones a los alumnos de la institución |

Tabla 1: Actividades de las 8 áreas de la Unidad de Extensión Universitaria

1.2 Procesos de las áreas

Para que las ocho áreas sean capaces de realizar todas sus funciones y actividades controladamente, es necesario que sigan por lo menos un proceso.

Cada una realiza procesos y planteamientos diferentes pero básicamente son similares en las siguientes características:

- Planificar un evento.
- Ejecutar dicho evento.
- Redactar un reporte de lo sucedido.
- Enviarlo a la dirección de Extensión Universitaria.
- Se adjunta con un oficio para mayor control.
- Se recopila y guarda físicamente en un archivero toda la información.
- Se almacena en la oficina de la dirección para su uso posterior.

Cada una de las áreas maneja diferente información, ya sean torneos, publicaciones o eventos y por lo tanto manejan diferentes procesos para el mejor uso de su información, además de ocupar una serie de formatos prediseñados donde se vacía la información necesaria. Sin embargo, dichos formatos no cumplen con todos los requisitos y son pobres en cuanto a la información requerida debido a que se limitan a una hoja tamaño carta, como a continuación se muestran:

| Tipo de Evento | Nº de Eventos Org. | Nº de Eventos Part. | N de Participantes |
|-------------------|--------------------|---------------------|--------------------|
| Atletismo | | | |
| Básquetbol | | | |
| Béisbol | | | |
| Montañismo | | | |
| Lucha Olímpica | | | |
| Fútbol | | | |
| Voleibol | | | |
| Taekwondo | | | |
| Físico culturismo | | | |

Figura 3: Formatos de reporte

| Carrera | Revistas | Libros | Apuntes | Planes | Doc. Trabajo | Total |
|------------------------|----------|--------|---------|--------|--------------|-------|
| Div. Ciencias Sociales | | | | | | |
| Comunicación | | | | | | |
| Div. Físico Mat. | | | | | | |
| Div. Humanidades | | | | | | |
| Dirección | | | | | | |
| Secretaría Académica | | | | | | |
| Unidad de Planeación | | | | | | |
| Otros | | | | | | |

| Actividades | Total de Eventos | Nº Participantes | Nº Espectadores |
|-------------------------------|------------------|------------------|-----------------|
| Exposiciones | | | |
| Danza | | | |
| Teatro | | | |
| Talleres | | | |
| Conciertos | | | |
| Visitas guiadas | | | |
| Actividades musicales | | | |
| Apoyo interno | | | |
| Apoyo externo | | | |
| Tertulia | | | |
| Muestra internacional de cine | | | |

Figura 3: Formatos de reporte (continuación)

Al analizar los procesos, nos encontramos con varios problemas ocasionados por estos reportes como:

- Duplicidad de procesos: cada una de las áreas, como ya se mencionó, maneja diferentes procesos y la información obtenida se tiene que reescribir y acomodar para su mejor archivamiento y recopilación.

- Diferentes tiempos de entrega: cada área entrega en diferentes tiempos los reportes, ya que se realizan cuando existen eventos que registrar, no se dan mensualmente o en determinado tiempo; sin embargo, deben juntarse todos al final, esto ocasiona búsquedas físicas en un gabinete para encontrar la información requerida.
- Movimiento lento por ser manual: ya que tienen que elaborar y vaciar la información en los diferentes formatos. si no se cuenta con ellos, solicitar una copia a la dirección, imprimirlos y enviarlos con un mensajero físico a la dirección, esperando tener disponibilidad inmediata para realizar dicho acto, lo cual no es seguro ni rápido.
- Procesos distintos: debido a que cada área tiene un proceso para el manejo de su información además de una forma de redactar sus registros, al final debe unificarse toda la información de la Unidad de Extensión Universitaria para poder presentarlos en un reporte que se realiza anualmente y debe ser entregado a la dirección general: por lo tanto, no tiene caso el reescribir toda la información de cada uno de los procesos.
- Archivamiento manual de los documentos: todos los reportes se almacenan en un gabinete ubicado en la dirección, sin embargo, es posible que no todos se guarden adecuadamente o se olvide archivar alguno, debido a fallas humanas o simplemente entre tanto movimiento en el cual intervienen más de dos personas. lo cual propicia la pérdida de información muy valiosa por errores de traspapeleo.
- Reescritura de información para su recopilación: debido a que es necesario recopilar y unificar toda la información enviada por cada una de las áreas, ocasiona la intervención de una tercera persona que debe recibir todos los reportes de las áreas y reescribirlos en formatos únicos en la computadora para poder entregarlo a la dirección, lo cual además de pérdida de tiempo puede ocasionar pérdida de información al reescribir.
- Búsqueda física de eventos anteriores: la dirección eventualmente debe reportar lo sucedido en sus áreas a la Dirección general del plantel, para esto

se debe buscar físicamente los eventos que tienen hasta la fecha y unificarlos para entregar reportes parciales necesarios a través del año lectivo.

Otro conflicto encontrado en los procesos, es que existen áreas que reportan cada mes, otras cada bimestre, semestre o año, por lo que no se lleva un buen control de los mismos. Esto ocasiona que cuando la Unidad de Extensión Universitaria necesita informar a la Dirección de la escuela las diferentes actividades que ha realizado a lo largo de un año, se tengan que buscar físicamente en un gabinete los reportes más los archivos de computadora y juntar lo de todas las áreas para entregar dicho trabajo.

Básicamente cada área maneja un proceso similar, dicho proceso consiste en planear una actividad y luego reportar lo sucedido, a la Unidad de Extensión Universitaria por medio de un formato que lleva algunas características del evento, junto con un oficio.

Para entender mejor lo antes expuesto, a continuación se muestra un diagrama de flujo mostrando la forma en que actualmente se ejecuta la generación de un evento, con las veces que se tiene que reescribir el mismo formato de reporte:

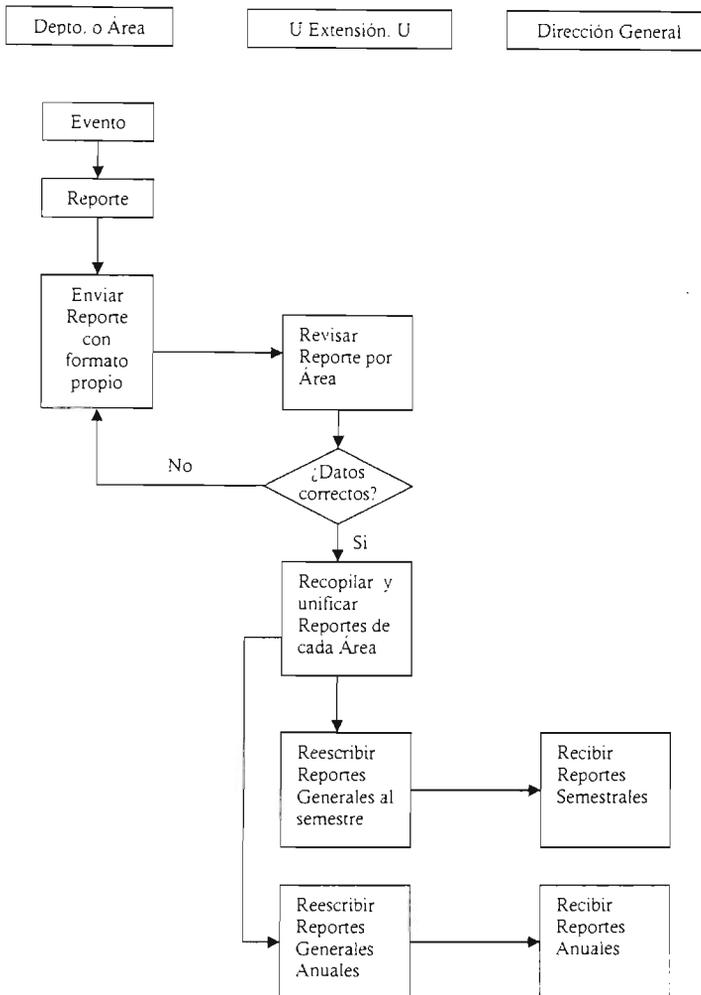


Figura 4: Diagrama de flujo de la generación de reporte de un evento.

Esto también produce que la información no viaje tan rápido, ya que se envía por medio de asistentes, llegando con días de retraso, además, después de que envía el reporte, una persona es encargada de revisarlo y compilarlo para tener una mejor

estructura de toda la información, esto ocasiona duplicidad de trabajo, así como pérdida de tiempo.

Esta última recopilación se hace con el objetivo de enviar un registro general a la Dirección de la escuela mencionando las actividades realizadas durante el semestre de la Unidad de Extensión Universitaria, así como después entregar un informe para el reporte anual de toda la Institución.

1.3 Problemáticas

Esta serie de problemas y riesgos nos genera pérdida de tiempo y gasto de energía del diferente personal para la entrega de sus reportes generales. Además, a todo esto, hay que agregar los diferentes acuerdos de las áreas con otras instituciones filiales a la UNAM, estos últimos datos no se reportan, simplemente se actualizan.

Actualmente, cada área mantiene todas sus operaciones con un proceso distinto, por lo tanto, no generan reportes al mismo tiempo ni con los mismos formatos, además de tener que hacer repetitiva la información para que llegue a las áreas correspondientes.

Sin embargo, cada área cuenta con un equipo de cómputo, la mayoría esta conectada en red, esperando que en un futuro no muy lejano todas estén en red.

Cada equipo de cómputo cubre los requisitos mínimos para implementar aplicaciones extras, y debido a su conexión, permiten una comunicación fácil entre las áreas.

A continuación se muestra un cuadro de resumen con las problemáticas generadas por los procesos:

| Problemas |
|---|
| Duplicidad de procesos |
| Diferentes tiempos de entrega |
| Movimiento lento por ser manual |
| Procesos distintos |
| Archivamiento físico de los documentos |
| Reescritura de información para su recopilación |
| Búsqueda física de eventos anteriores |
| Distintos formatos de reporte |
| Pérdida de tiempo |
| Dificultad de recopilación |

Tabla 2: Diversos problemas que ocasiona tener diferentes procesos

De todas las problemáticas mostradas, en su mayoría se realizan desaprovechando por completo los recursos con que cuenta el área, ya que aun teniendo equipos de cómputo conectadas en red, no las utilizan más que para redactar oficios o reportes, y lamentablemente no aprovechan al máximo dichos recursos. Además, las diferentes áreas con las que cuenta la Unidad de Extensión Universitaria, envían reportes físicamente siendo que los elaboran en el equipo de cómputo y desaprovechan la conexión de red que tienen entre sus computadoras.

Para ejemplificar todas estas problemáticas, a continuación mostramos un reporte de un evento cultural:

1. El día 6 de diciembre se planea realizar un recital de villancicos en el auditorio "José Vasconcelos", por lo tanto se envían diferentes oficios a las oficinas de la escuela para solicitar: permisos de puerta, seguridad, intendencia, cada una de éstas con un mensajero que lleva portando en la mano un oficio con el pedido.
2. Además de enviar solicitud con oficio al área de publicaciones e impresiones para que diseñe la propaganda necesaria para el evento e intentar tener oportunamente todos los folletos, al menos 15 días antes del evento para su difusión.
3. Al tener todos los requisitos previos se dispone a realizar el evento en la fecha deseada, y después de realizado se vacía la información del evento en los formatos prediseñados, los datos que incluye son: hora del evento, nombre del evento, pequeño resumen y número de participantes o espectadores, sin embargo no meten datos importantes como datos de economía o seguridad o algunas observaciones extras.
4. Después se elabora un oficio para adjuntar la hoja (única) del reporte, y así el área de actividades culturales, mediante un mensajero, se lo envía a la asistente de la Dirección de Extensión Universitaria, en la semana o mes posterior al evento; luego, la asistente reescribe el evento en su computadora y se lo muestra a su superior para que lo apruebe, entonces ya aprobado se guarda el diskette en la computadora y los papeles en el archivero físico.

En todo este proceso la información no es completa, se pierden demasiados datos importantes como: gastos, público general y universitario, comentarios, entre otros, además de tiempo y esfuerzo físico.

1.4 Propuesta de solución

Ante tales desventajas que presenta el actual sistema de ejecución, presentamos una solución práctica a dicho problema, la creación de un sistema de información que permita la automatización de todas estas actividades, ofreciendo las siguientes ventajas:

- Recibir toda la información mucho más minuciosa de los eventos: dado que el sistema establecerá los detalles y requerimientos de información.
- Unificación de los procesos: todos manejarán el mismo proceso, facilitando su recopilación, por medio de eventos (llamando evento a cualquier actividad cultural deportiva u otra acción de las áreas)
- Manejarán los mismos formatos: para todos los eventos por medio de pantallas fáciles para cada área.
- Todo se manejará por medio de Eventos: ya que lo que realiza cada área son simplemente eventos.
- Se mantendrá la comunicación con los oficios: ya que se adjuntarán números de oficios para su mejor manejo
- Tendiendo mayor control de dicha información: debido a que no hay pérdida y todo queda guardado en la base de datos
- Contará con un sistema de bitácora: para verificar quién entra al sistema y en qué momento.
- Control de los movimientos realizados al sistema: para que no exista un mal uso del mismo.
- Ahorro de tiempo: al no tener que reescribir toda la información se agilizan los procesos.
- Evitar duplicidad de trabajo: no se repite la descripción de un mismo reporte.
- Optimización de los reportes: más rápidos y sencillos

- Todos los datos se almacenan en un mismo lugar: una base de datos hospedada en la computadora principal de la Dirección.
- Fácil búsqueda de datos: ya que el sistema permite obtener datos de cualquier evento almacenado.
- Creación de reportes finales: el sistema permite imprimir diferentes partes de los reportes para su mejor uso.
- Aprovechamiento de recursos existentes (cómputo): se utilizarán de manera eficiente los recursos de cómputo con que cuenta cada área.

Esto implica que el nuevo sistema logrará un funcionamiento óptimo de los recursos existentes además de un manejo más fácil de su recurso más importante: su información interna. Tomando en cuenta que la información debe ser segura, confiable, oportuna y veraz, dicho sistema cumplirá como principal objetivo el manejo más cómodo y seguro de la información, para la toma de decisiones, así como unificar los diferentes procesos.

Además de todo esto el sistema mejorará notablemente el sistema actual, reduciendo costos y tiempos de trabajo que se manejan en este momento, así como la conducción de dos rubros importantes como:

- Operación diaria. Actualmente los reportes se realizan cada vez que sucede un evento, es decir eventualmente así que no se puede tener toda la información oportuna, mientras que con nuestro nuevo sistema tendrá una operación diaria por lo que la información viajará más rápido y se podrán realizar reportes en cualquier instante que sea requerido.
- Soporte de información para la toma de decisiones. Ya que será posible tomar decisiones, referente a los eventos de una forma más directa y segura que de la otra forma.

Tomando en cuenta que cada una de las áreas cuenta con una PC, van a poder conectarse al sistema (exceptuando quien no tiene red) y lo harán por medio de una serie de contraseñas, similares a las que cuentan actualmente para entrar al entorno de Windows.

Dicho sistema contará con una pantalla de recepción de datos de cada una de las áreas, la cual se creará de acuerdo a su perfil y necesidades, se podrá mandar a la máquina principal como si mandarían un correo electrónico. Todo esto tendrá una interfaz muy amigable similar a la que manejan en un programa de texto como Word, para que su uso sea muy sencillo y permita la recopilación de mucha mayor información, debido a que contará con más campos de información de los que tienen ahora.

Las ventajas de un Sistema de Información en comparación con el esquema actual de operación se muestran en el siguiente cuadro comparativo:

| <i>Sistema Actual</i> | <i>Sistema Nuevo</i> |
|---|---|
| Duplicidad de trabajo | Evitar duplicidad de trabajo |
| Información pobre | Recibir la información más detallada |
| Procesos distintos | Se unificarán los procesos |
| Pérdida de tiempo | Ahorro de tiempo |
| Distintos formatos de reporte | Manejarán los mismos formatos |
| Archivamiento físico de documentos | Todos los datos se almacenarán en una base de datos |
| Reescritura de información para su recopilación | Controlando movimientos realizados al sistema |
| Búsqueda física de eventos anteriores | Se extraerán datos fácilmente |
| Dificultad de Recopilación | Se manejará por medio de eventos |
| Movimiento lento por ser manual | Tener mayor control de información |
| Diferentes tiempos de entrega | Imprimirán reportes finales |
| Desaprovechamiento de recursos existentes | Aprovechamiento de recursos existentes (cómputo) |
| Oficio adjunto | Se conservarán los oficios |
| No tienen sistema de bitácora | Contará con sistema de bitácora |
| Reportes sencillos | Optimización de los reportes |

Tabla 3: Cuadro comparativo de las diferentes ventajas del sistema actual con el nuevo

Tomando todo esto en cuenta, en el momento de analizar la información, para el nuevo sistema se pueden mejorar los procesos y homogenizar los reportes. Para entender esto mejor, a continuación se ejemplifica el mismo evento anteriormente mencionado.

1. El día 6 de diciembre se planea realizar un recital de villancicos en el auditorio "José Vasconcelos", por lo tanto se envían peticiones por computadora para solicitar permisos de puerta, seguridad, intendencia, incluido en el correo el número de oficio, lo cual se realiza en un instante y se recibe la confirmación inmediatamente.
2. Además de enviar electrónicamente la petición a publicaciones e impresiones para que diseñe la propaganda necesaria para el evento y teniendo oportunamente todos los folletos, 15 días o más antes del evento para su difusión ya que la petición se realiza en cualquier momento del día no sólo en horarios de oficina
3. Al tener todos los requisitos previos se dispone a realizar el evento en la fecha deseada y después de realizado se vacía la información del evento en los formatos prediseñados en la computadora, los datos que incluye abarca todos los necesarios del evento así como tener espacio para observaciones importantes.
4. Se adjunta un número de oficio y se envía electrónicamente a la asistente de la Dirección de Extensión Universitaria, en el momento deseado, al mismo tiempo lo ve su superior y al aprobarlo automáticamente queda guardado en la base de datos del sistema, listo para su recuperación cuando sea necesario,

En todo este proceso la información es completa, no se pierde nada y no se realiza ninguna otra actividad física. Para poder implementar una solución de este tipo, es fundamental realizar un análisis formal que nos asegure tomar en cuenta todos los aspectos y requerimientos que los usuarios necesiten en su nuevo sistema.

Capítulo II
Definición de
herramientas y
uso de
metodologías

2. Definición de herramientas y uso de metodologías

En el capítulo anterior hemos establecido la necesidad de implementar un sistema para dar solución a la serie de problemas que se nos presentan con el fin de hacer más rápida y confiable cualquiera de las tareas a las que se pretende llegar con dicho sistema.

En principio debemos cuantificar y cualificar los recursos con los que se cuenta en la Unidad de Extensión Universitaria, que son:

- Como mínimo una PC por departamento trabajando todas ellas bajo la plataforma Windows 98.
- En cada equipo se cuenta con la suite Office de Microsoft para todas las labores ofimáticas que en ella se realizan.
- Se cuenta con una memoria RAM por equipo que oscila entre los 32 y 128 MB.
- Cada PC no rebasa en su procesador la velocidad de 550 Mhz.
- Cuentan con una red interna que tiene acceso a Internet y como una vía de intercambio de información entre departamentos.
- La mayoría de los usuarios a quienes se destinará el sistema que se va a desarrollar no cuentan con un alto nivel de conocimiento en cuanto al uso de sistemas operativos distintos a Windows o suites Ofimáticas que no sean Microsoft Office.
- Algunos departamentos se encuentran en polos opuestos del Campus, y para trasladarse de uno a otro se toma un tiempo bastante considerable.
- No se cuenta con un servidor que pueda dar alojamiento a un sistema robusto.

Como consecuencia de los puntos mencionados debemos planear la plataforma de desarrollo con los siguientes requisitos mínimos:

- Deben trabajar bajo plataforma Windows.
- Compatibilidad con las distintas aplicaciones de Microsoft.
- Su aplicación en un entorno de red debe ser confiable para el intercambio de información en forma segura.
- El flujo de información a través de la red no debe provocar cargas excesivas.
- Debe ser fácil de utilizar.

Tomando en cuenta los recursos y necesidades anteriores, podemos hablar ahora de que los sistemas informáticos que existen, y que además es posible implementar en situaciones como ésta, son demasiados. Pero si pensamos un momento en que una solución informática debe ser práctica, real y objetiva, encontramos el hecho de que hay dos esquemas viables para su implementación:

- **Sistema Web**¹
- **Cliente-Servidor**²

Entendemos por **Sistema Web** como un sistema al que los usuarios puedan tener acceso mediante Internet para realizar cualquier transacción, además del hecho de que se cuenta con un servidor de bases de datos y un servidor de aplicaciones Web, lo que hará más fácil y viable la utilización de una intranet para su aprovechamiento total sin consumir en exceso los

¹ <http://www.mug.org.ar/FoxProGufa/ArticFox/default.aspx>

² <http://www.mug.org.ar/FoxProGufa/ArticFox/default.aspx>

recursos de nuestro servidor, como habría ocurrido si desde un principio se hubiese optado por realizar un sistema así, contando con recursos limitados.

En un esquema **Cliente-Servidor** se tiene la misma forma de operación, sólo que no se puede acceder a él mediante Internet, y no es necesario contar con un servidor como tal, pues se puede implementar en una PC que tome el papel de uno, mediante la instalación de un programa servidor y las demás PC's van a contar con un programa cliente.

De estos dos esquemas debemos decidir cuál es el más viable para la infraestructura de operación del sistema que cubrirá las necesidades de la Unidad de Extensión Universitaria, de tal modo que debemos establecer los puntos a favor de cada uno para poder llegar a una decisión final.

A continuación se enumeran algunas de las ventajas que ofrecen ambas alternativas, de tal forma que se pueda ver a los dos como sistemas que pueden ayudar en mayor o menor medida a la solución de nuestro problema.

2.1 Sistema Cliente-Servidor

- Permite la programación en varias capas, lo que deriva en una mejor administración del ancho de banda disponible.
- No requiere la disponibilidad de un servidor Web.
- No requiere la existencia de un servidor de bases de datos potente, por lo que no supone un gasto mayor.
- Permite la implementación de distintos niveles de acceso a la base de datos disponibles.
- Dispone de un programa Cliente y un Servidor, por lo que los datos no se encuentran dispersos entre terminales de trabajo.
- Puede trabajar a velocidades aceptables a pesar de que la red en la que se implemente sólo alcance los 10 Mbps.

- Se puede basar en software de Microsoft, por lo que abarca un 80% de los usuarios normales de PC's.
- Al implementarse en la red que se tiene disponible no se tendría que hacer una inversión extra en equipo de cómputo.
- Puede trabajar correctamente desde los 32 Mb en RAM.

2.2 Sistema Web

- Permite la consulta de una base de datos a través de Internet.
- Cuenta con un Servidor de Bases de Datos que permite una mayor integridad en la información.
- Permite la utilización de distintos niveles de seguridad en cuanto al acceso a la información.
- Es posible la administración remota del sistema.
- La velocidad con la que el servidor realiza una transacción es mayor que en un sistema convencional.
- El acceso a la información siempre esta disponible.
- Permite la administración real del ancho de banda, así como de los recursos que estarán disponibles y los que no lo estarán.
- Toda la información utilizada esta concentrada en un solo lugar, lo que disminuye la fragmentación de los distintos bloques del sistema.
- Es posible adaptarlo a grandes cantidades de información.

Ambos esquemas son una buena solución para el problema existente en la Unidad de Extensión Universitaria, así que debemos apegarnos a las necesidades que tiene, así como a los medios disponibles, de tal modo que el esquema elegido resulte ser el que más conviene a la institución, de aquí que también se deban listar algunos requerimientos que tienen ambos esquemas, para poder sopesar las desventajas que se tienen con sus consecuencias directas e indirectas sobre el desarrollo de la ya citada unidad.

| Características generales | | Cliente-Servidor | Sistema Web |
|---|--|-------------------------|--------------------|
| | <i>Servicio</i> | * | * |
| | <i>Recursos compartidos</i> | * | |
| | <i>Protocolos asimétricos</i> | * | * |
| | <i>Transparencia de localización</i> | * | * |
| | <i>Independencia de HW y SW</i> | | * |
| | <i>Intercambio basado en mensajes</i> | * | * |
| | <i>Encapsulamiento del servicio</i> | * | * |
| | <i>Escalabilidad</i> | * | * |
| | <i>Integridad</i> | * | * |
| | <i>Programación en múltiples capas</i> | * | * |
| | <i>Administración remota</i> | | * |
| | <i>Acceso a través de Internet</i> | | * |
| Tipos de servidores en que se pueden implementar | <i>Servidores de archivos</i> | | * |
| | <i>Servidores de bases de datos</i> | * | * |
| | <i>Servidores de transacciones</i> | * | * |
| | <i>Servidores de Groupware</i> | | * |
| | <i>Servidores de objetos / componentes</i> | | * |
| | <i>Servidores de aplicaciones Web</i> | | * |
| Requerimientos para un ambiente distribuido | <i>Procesamiento de transacciones</i> | * | * |
| | <i>Seguridad</i> | * | * |
| | <i>Agentes móviles</i> | | * |
| | <i>Mejor administración de datos, tradicionalmente se almacenan en DBMS. XML fundamento.</i> | * | * |
| | <i>Entidades inteligentes auto-administradas</i> | | * |
| | <i>Middleware inteligentes</i> | * | * |
| | <i>Ancho de Banda mayor de 10 Mbps</i> | | * |
| | <i>Servidores Independientes del resto de los equipos de computo</i> | | * |
| | <i>Administración remota</i> | | * |
| | <i>Inversión extra</i> | | * |
| <i>Perfil de Usuario Alto</i> | | * | |

Tabla 4: Cuadro comparativo de las características, servidores y requerimientos que utilizan un Sistema Cliente-Servidor y Sistema Web

Como se puede apreciar en la Tabla 4, un Sistema Web es mucho más potente que uno Cliente-Servidor, a pesar de que tienen como base una programación multicapa, y que se asemejan en muchos puntos más, pero para los recursos disponibles en este momento * lo más indicado es la implementación de un sistema es esquema Cliente-Servidor, esto debido a que para implementar un Sistema Web debemos utilizar más recursos de los que tenemos disponibles, por otro lado, no se cuenta con la disponibilidad de un servidor auténtico, por lo que se deberá utilizar una PC cualquiera para que desarrolle esta función, y no se sabe si lo estará en un mediano plazo.

Derivado de los recursos que tenemos disponibles, debemos economizar el uso de espacio en disco duro, cantidad de RAM utilizada y ancho de banda, por lo que deberemos utilizar software ligero tanto para el desarrollo como para la implementación.

Esta característica la ofrecen los dos sistemas, pero debemos utilizar el que menos recursos consuma, así como pensar en la posibilidad de que crezca la base de datos, o que se asignen más recursos a esta Unidad, para lo cual no importa cual de los dos se desarrolle debido a que partiendo de un Sistema Cliente-Servidor se puede dar paso a un Sistema Web.

Todo esto sin necesidad de realizar el análisis previo, pues son muy similares hasta el punto en el que se debe dar acceso de un lugar fuera de mi red privada, para lo cual se va a utilizar un Sistema Web creado a partir del estudio que se realizó para implementar un Sistema Cliente Servidor, lo que nos aclara que este sistema tiene amplias posibilidades de crecimiento sin grandes inversiones.

* pp 24

2.3 Metodologías para desarrollo de sistemas

"Al iniciar el desarrollo de sistemas de información, o de cualquier tipo de software, uno de los retos que enfrentamos es la elección de una metodología adecuada para su desarrollo. En este capítulo es necesario establecer lo que es y lo que implica una metodología." ³

Una Metodología es una colección de técnicas basadas sobre una filosofía común o una forma de trabajo, que se establece en conjunto sobre una plataforma llamada Ciclo de Vida del Desarrollo de Sistemas o simplemente Ciclo de Vida.

Corresponde a un conjunto de modelos, lenguajes y otras herramientas, que nos facilitan la representación de los datos de cada fase del proceso¹ junto con las reglas que permiten el paso de una fase a la siguiente.

Una metodología nos proporciona las herramientas necesarias para detectar en forma cualitativa y cuantitativa los distintos problemas que se tienen en los procesos que se llevan a cabo en la labor diaria de una empresa, así como también para calificar la eficacia de los productos finales, ya sean una aplicación sencilla o un complejo sistema de información.

³ <http://www.geocities.com/trescapas/introduccion.htm>

En principio una metodología nos auxilia en varios aspectos; como se observa en la siguiente figura.

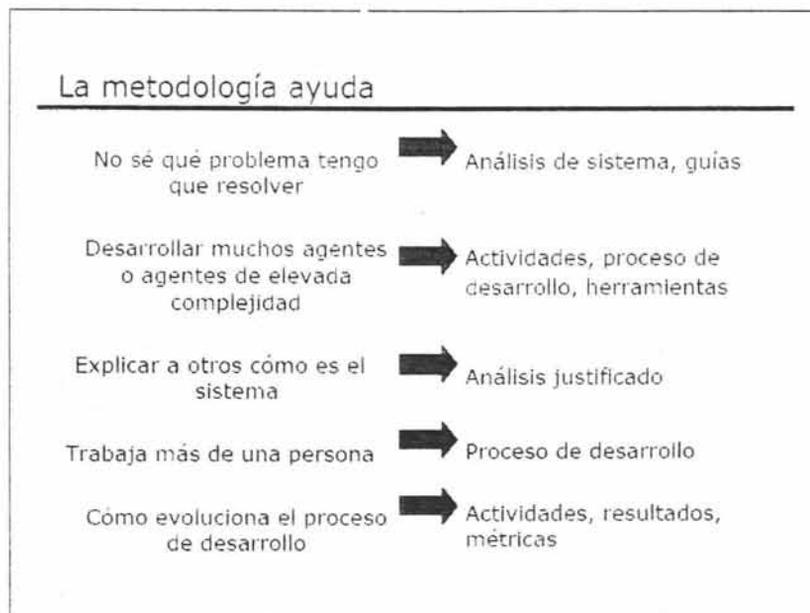


Figura 5: Utilidad de la Metodología

El enfoque que se da a un problema es el mismo que se lleva en una empresa ya establecida, tomando en cuenta los problemas que acarrea el hecho de no contar con una metodología formal, dichos problemas pueden ser de diversas índoles, como la falta de coordinación, coherencia y disponibilidad del recurso más valioso: **La Información**.

A medida que los volúmenes de información a manejar en las organizaciones aumentan, también crece la necesidad de disponer de herramientas de gestión, que soporten de forma ágil y eficaz el funcionamiento habitual de los procedimientos de trabajo de la Organización: **ágil**, pues debe ser capaz de cambiar y adaptarse a los diversos cambios que se presentan a

lo largo del funcionamiento de la empresa; **eficaz**, en cuanto a dar una solución útil y adecuada a las necesidades del personal.

Para asegurar que los productos y servicios estén diseñados de forma que satisfagan las necesidades de los usuarios es fundamental la colaboración entre grupos de diferentes áreas y, por tanto, indispensable el establecer mecanismos y reglas de comunicación entre las áreas usuarias y los diseñadores de Sistemas o el Departamento de Informática.

La solución para realizar con éxito el diseño y desarrollo de los productos deberá tomar en cuenta la utilización de una metodología que facilite la integración y utilización de ideas e información de los grupos, así como el avance rápido y eficiente del proyecto.

La inexistencia de estos mecanismos ocasiona, sin duda, el tomar como parte del funcionamiento habitual del Desarrollo e Implantación de Sistemas, problemáticas tales como:

- La escasa o nula documentación de las aplicaciones, lo que genera ineficacias durante el desarrollo, implantación y especialmente el mantenimiento y, en consecuencia, incumplimiento de los plazos.
- La falta de comunicación con los usuarios, lo que genera fuertes desviaciones en el grado de adecuación de los productos resultantes a las expectativas generadas en los usuarios.
- La inclusión de elementos innecesarios en la etapa de desarrollo, como consecuencia directa de la falta de comunicación con los usuarios, lo que genera una inadecuación en el producto final, que es la aplicación en sí.

- Asignación inadecuada de los distintos niveles de acceso a la información, generando así una dependencia innecesaria entre áreas que son totalmente distintas, y nada tienen que ver una con la otra.
- La duplicidad de procesos, lo que genera un aumento considerable en los tiempos de realización para tareas rutinarias.
- La falta de información adecuada para el desarrollo e implantación de productos finales, lo que genera un funcionamiento inadecuado de los mismos para los fines de la empresa en la que se lleva a cabo el desarrollo de un sistema o aplicación.
- Falta de conocimiento de los procesos que se llevan a cabo, derivando en una inadecuación de las aplicaciones o productos que se desarrollen a la vida productiva de una empresa.

La utilización de una metodología para la planificación, análisis, construcción e implantación de los Sistemas de Información, permite generar un entorno de trabajo en equipo con el objetivo de construir los sistemas para dar solución a las necesidades de los usuarios y que estén preparados para facilitar el mantenimiento y adecuación a los cambios futuros de la organización.

Parte del éxito de la implantación de la metodología se basa en la utilización de un vocabulario común, accesible para el usuario, que le permita involucrarse de forma confortable en las fases en las que es el protagonista principal, sin renunciar al empleo de los conceptos y las técnicas de la Ingeniería de los Sistemas de Información.

Las metodologías ofrecen un marco de trabajo en el que se define:

- Una estructura de proyecto que sirve de guía al equipo de trabajo, establece responsabilidades de los miembros e involucra a los usuarios en el desarrollo y toma de decisiones en los puntos decisivos.
- Un conjunto de productos finales a desarrollar.
- Un conjunto de técnicas para obtener los productos finales. Con este ambicioso objetivo, las metodologías se estructuran en fases que permiten que:

A partir de la estructura de fase, se establezca un conjunto formal de productos a entregar por el equipo de trabajo al final de cada fase. Esto permite dividir los proyectos en una serie de objetivos preestablecidos que facilita las labores de Planificación y Control de los Proyectos.

- El final de cada fase requiera la aceptación formal de las conclusiones a las que se ha llegado al término de la misma por parte de los responsables de validar los productos.
- El producto final de cada fase constituya el punto de partida para la ejecución de la siguiente.

Desde nuestro punto de vista, la implantación de una Metodología de Desarrollo de Sistemas en la que se defina el conjunto de métodos, técnicas y herramientas para la construcción de los Sistemas de Información, es el primer paso para garantizar a la empresa la consecución de los objetivos de:

- Satisfacer todas las necesidades de los departamentos usuarios implicados en cada proyecto.
- Generar la documentación asociada, para facilitar su mantenimiento posterior.

- Mejorar la productividad y la calidad del servicio prestado por los Departamentos de Informática y Tecnologías de la Información.

Una vez establecida y analizada la necesidad de utilizar una metodología, debemos hablar acerca de la inexistencia de un camino obligatorio para resolver cualquier problema, y mucho menos hay reglas estrictas para desarrollar sistemas de información, es por ello que hemos dividido el desarrollo del sistema al cual debemos dar forma en este tratado en cuatro partes importantes, dichas partes se observan a continuación:

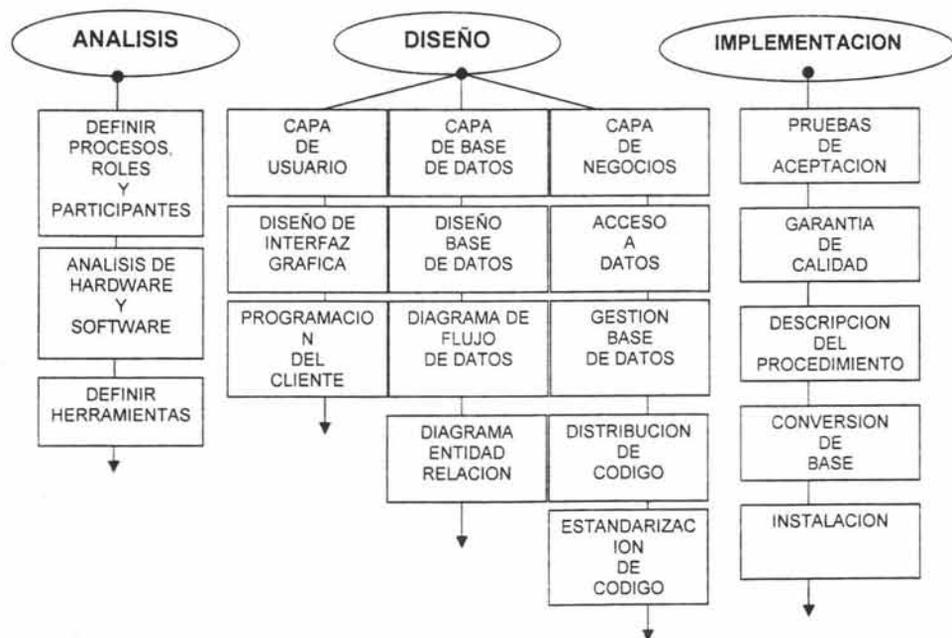


Figura 6: Etapas de la Metodología.

2.3.1 Análisis

Durante esta etapa del desarrollo de sistemas, se tiene como objetivo precisamente el estudio a profundidad de las condiciones en las que vamos a trabajar, de tal forma que como producto final debemos obtener un panorama general, y también particular, de los procesos que se desarrollan en el lugar donde se va implementar dicho sistema.

De tal forma que, como productos finales de la etapa de ANALISIS, se obtiene un conocimiento total de los casos que se pueden presentar, los procesos que se van a llevar a cabo para cada uno de ellos, así como los actores que van a intervenir en cada punto de dichos casos, y por obviedad debemos detectar los puntos críticos en dónde existen problemas en cuanto al flujo continuo y velos de la información. Además de las características de software y hardware con el que se cuenta.

Todo esto se lleva a cabo en forma independiente de la programación, es decir, que no importa en que lenguaje nos acomode más la programación, pues este análisis nos servirá de base si en un futuro deseamos cambiar o modernizar nuestros sistema ya terminado, pues nos ofrece una perspectiva bastante completa y desde distintos puntos de vista de la problemática a la que nos enfrentamos.

2.3.2 Diseño

Es propiamente la elección de la arquitectura bajo la cual se desarrollará nuestro sistema, esto basado en las características de los procesos que queremos automatizar o cambiar mediante la implantación del mismo.

El llamado diseño abarca desde la arquitectura a utilizar, hasta la interfaz de usuario a implementar, las tareas que va a realizar nuestro sistema, así como su distribución, en base a las necesidades del cliente y a las posibilidades que se tengan, tanto de infraestructura, como de recursos humanos. Cabe mencionar que esta etapa tiene que tomar en cuenta a los usuarios finales, así como sus características, responsabilidades y aptitudes.

2.3.3 Desarrollo o Programación

Durante esta etapa se lleva a cabo la programación o codificación, del sistema a desarrollar en base a los resultados arrojados en la etapa anterior, de tal forma que es aquí donde la etapa de diseño se materializa, pero no en una forma burda y sin control alguno, como en la mayoría de los casos, si no en una forma organizada y controlada en la cual se toman en cuenta las características de la infraestructura con la que se cuenta, los recursos humanos, así como las necesidades que se deben cubrir al implementar el sistema que vamos a desarrollar.

Antes de pasar a la etapa de programación como tal deberemos definir las herramientas que se van a utilizar, dichas herramientas engloban términos como el lenguaje de programación, el gestor de la base de datos, generador de reportes, etc.

2.3.4 Implementación

En esta etapa final se tiene como objetivo llegar a la implementación total del sistema en un ambiente de trabajo totalmente real, así como llevarlo a su cien por ciento de funcionalidad, libre de cualquier problema de operación o de programación, es decir, debe cubrir en un cien por ciento las necesidades establecidas en un principio, y debe realizar los procesos de la forma que se acordó en conjunto con los usuarios finales.

Debe buscarse la aprobación final del cliente, así como la documentación completa del sistema desarrollado, es decir, realizar manuales de usuario, con toda la documentación necesaria para su operación.

2.4 Herramientas

Ya que hemos mencionado el objetivo de cada etapa, debemos desarrollarlas y llevar a buen término cada una de ellas. Para este fin, debemos definir cuáles serán las herramientas que vamos a utilizar para cada una de ellas. Es por esta razón que a continuación se presentan las herramientas que vamos a utilizar en cada etapa de desarrollo de nuestro sistema.

2.4.1 Análisis

Para comenzar enunciaremos las actividades y esquemas a realizar para la etapa de análisis:

En esta actividad se desarrolla algo conocido como modelo de implantación del usuario. Se concentra básicamente en las especificaciones de la frontera humano-máquina (las partes que realizará alguien manualmente y

las partes que se implantarán en la máquina) y de la interfaz hombre máquina (el diseño de pantallas y los diálogos en línea entre el usuario y la máquina).

1.- Definir procesos, roles y usuarios: En este apartado se deben definir los procesos que se llevan a cabo en la institución para la cual se diseña un sistema, de tal modo que se puedan identificar aquellos a los que se dará asistencia mediante el mismo.

2.- Flujo de información: Aquí se debe determinar de dónde viene, hacia dónde va, y quién maneja la información en cada punto de los procesos que se realizan.

3.- Análisis de Hardware y Software: En esta parte del análisis, se deben considerar dos cosas: con cuáles recursos se va a desarrollar el sistema, y con cuáles recursos se va a implementar el sistema. Aunque parece lo mismo no lo es; en el primero se analiza el hardware (hw) y el software (sw) con el que se cuenta para desarrollar el sistema, y en el segundo se analiza el hw y el sw en el que se implementará el sistema.

4.- Definir Herramientas: Aquí se hace necesaria la comparación entre las distintas herramientas de las que disponemos para desarrollar y/o implementar cada capa de nuestro sistema, de tal forma que se tome una decisión acertada en cuanto a cuál es la mejor opción para cada caso particular.

2.4.1.1 Herramientas de Análisis

Con el fin de llevar a buen término este proyecto debemos establecer la existencia de una metodología denominada **Lenguaje Unificado de Modelado**⁴ (UML por sus siglas en inglés), que nos permite ver un sistema

⁴ Conallen, J. "Modeling Web Applications with UML" Conallen, Inc. 9-Mar-1999

desde puntos de vista distintos, de acuerdo a cada uno de los esquemas que su estructura incluye, y lo más importante en este punto es el hecho de que nos hace posible visualizar un sistema en forma independiente de cualquier lenguaje de programación, lo que da al estudio previo un mayor potencial, pues debemos adaptar la programación a la planeación y el análisis previo, y no sujeta la planeación al lenguaje elegido en un principio. A partir de aquí cabe mencionar algunos puntos importantes a saber sobre **UML**.

UML es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos (OO). Un artefacto es una información que es utilizada o producida mediante un proceso de desarrollo de software.

UML se quiere convertir en un lenguaje estándar con el que sea posible modelar todos los componentes del proceso de desarrollo de aplicaciones. Sin embargo, hay que tener en cuenta un aspecto importante del modelo: no pretende definir un modelo estándar de desarrollo, sino únicamente un lenguaje de modelado. Otros métodos de modelaje como **OMT** (Técnica de modelado de objetos) o Booch sí definen procesos concretos. En **UML** los procesos de desarrollo son diferentes según los distintos dominios de trabajo. No puede ser el mismo el proceso para crear una aplicación en tiempo real, que el proceso de desarrollo de una aplicación orientada a gestión, por poner un ejemplo.

Las diferencias son muy marcadas y afectan a todas las fases del proceso. El método del UML recomienda utilizar los procesos que otras metodologías tienen definidos.

Lo que se intenta es lograr con esto que los lenguajes que se aplican siguiendo los métodos más utilizados sigan evolucionando en conjunto y no por separado. Y además, unificar las perspectivas entre diferentes tipos de

sistemas (no sólo software, sino también en el ámbito de los negocios), al aclarar las fases de desarrollo, los requerimientos de análisis, el diseño, la implementación y los conceptos internos de la OO.

El **Lenguaje Unificado de Modelado** prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación. **UML** ofrece nueve diagramas en los cuales modelar sistemas.

- Diagramas de Casos de Uso para modelar los procesos 'business'.
- Diagramas de Secuencia para modelar el paso de mensajes entre objetos.
- Diagramas de Colaboración para modelar interacciones entre objetos.
- Diagramas de Estado para modelar el comportamiento de los objetos en el sistema.
- Diagramas de Actividad para modelar el comportamiento de los Casos de Uso, objetos u operaciones.
- Diagramas de Clases para modelar la estructura estática de las clases en el sistema.
- Diagramas de Objetos para modelar la estructura estática de los objetos en el sistema.
- Diagramas de Componentes para modelar componentes.
- Diagramas de Implementación para modelar la distribución del sistema.

Por el hecho de que **UML** no es una metodología totalmente restrictiva, y de que no es necesario implementar todos sus esquemas al pie de la letra, se ha optado por el desarrollo de este proyecto utilizando un esquema de esta.

Para elegir la metodología **UML** se llevó a cabo la comparación entre ésta y otras metodologías utilizadas para el desarrollo de sistemas, y dicha comparativa arroja resultados que se tabularon y se incluyen en la tabla 5.

Las metodologías que forman parte de la comparativa son:

- **HDM** (Método de Diseño Hipermedia)
- **RMM** (Metodología de Administración de Relaciones)
- **EORM** (Metodología de Relaciones de Objetos Mejorada)
- **OOHDM** (Método de Diseño Hipermedia Orientado a Objetos)
- **SOHDM** (Metodología de Diseño Hipermedia orientada a objetos y basada en escenarios)
- **WSDM** (Método de Diseño de Sitios Web)
- **WAE-Proceso Conallen** (Extensión de Aplicación Web para UML)
- **UML** (Lenguaje Unificado de Modelado)

| | Proceso | Técnica de modelado | Representación gráfica | Notación | Herramienta de soporte |
|------------------------------|--|---------------------------|--|--|------------------------|
| HDM | 1. Desarrollo a largo plazo 2. Desarrollo a corto plazo | E-R (Entidad-Relación) | 1.-2. Diagrama E-R | 1. E-R | |
| RMM | 1. Diseño E-R 2. Diseño <i>Slice</i> 13 3. Diseño de navegación 4. Diseño de protocolo de conversión 5. Diseño de UI14 6. Diseño de comportamiento en tiempo de ejecución 7. Prueba y construcción | E-R | 1. Diagrama E-R 2. Diagrama <i>Slice</i> 3. Diagrama RMDM | 1. E-R 2.3. Propio | <i>RMCase</i> |
| EORM | 1. Clases del entorno de desarrollo 2. Composición del entorno de desarrollo 3. Entorno de desarrollo de UI | OO | 1. Diagrama de clases 2. Diseño GUI | 1. OMT | <i>ONTOS Studio</i> |
| OOHDM | 1. Diseño conceptual 2. Diseño navegacional 3. Diseño abstracto de la UI 4. Implementación | OO | 1. Diagrama de clases 2. Diagrama navegacional, clase + contexto 3. Diagrama de configuración de ADV + Diagrama ADV | 1. OMT/ UML 2. Propio 3. ADVs | <i>OOHDM -Web</i> |
| SOHDM | 1. Análisis del dominio 2. Modelo en OO 3. Diseño de la vista 4. Diseño navegacional 5. Diseño implementación 6. Construcción | Escenarios Vistas-OO | 1. Diagramas de escenarios de actividad 2. Diagrama de estructura de clase 3. Vista OO 4. Esquema de enlace navegacional 5. Esquema de páginas | 1.-5. Propio | |
| WSDM | 1. Modelado del usuario 2. Diseño conceptual 2.1. Modelo objetos 2.2. Diseño navegacional 3. Diseño implementación 4. Implementación | E-R/ OO | 1. Diagrama de E-R o clase 2. Capas de navegación | 1. E-R/ OMT 2. Propio | |
| WAE-Proceso Conalle n | 1. Manejo de proyecto 2. Captura de requerimientos 3. Análisis 4. Diseño 5. Implementación 6. Prueba 7. Desarrollo 8. Configuración y manejo de cambios | OO | 2.-5. Diagramas UML | UML | <i>Rational Rose</i> |
| UML | 1. Análisis 2. Diseño 3. Programación 4. Implementación | Rol- Participante | 1. Diagrama Rol- Participante 2. Diagrama E_R | 1. Propio 2. E-R | |

Tabla 5: Cuadro comparativo entre UML y otras metodologías

Algunas de las metodologías citadas están orientadas al diseño de **Sistemas Web**, esto es debido a que la metodología **UML** puede llevarse a este mismo nivel, simplemente al diseño de la interfaz de usuario o capa de presentación se agrega la conectividad vía red o Internet, aunque básicamente los pasos son los mismos. Por ello el enfoque que se le da nos permite hacer una comparativa con estas metodologías.

La tabla anterior muestra que en tanto las metodologías tradicionales involucran procesos tediosos, **UML** es algo más sencillo, pues la mayor parte del tiempo se enfoca en el diseño, nos permite contemplar a los participantes en todos los procesos que tenemos que llevar a cabo, y además nos da cierta libertad antes de llegar a la traza del Diagrama Entidad-Relación.

Una ventaja más es que toma en cuenta las necesidades de los usuarios finales, permite el diseño flexible en cuanto a bases de datos y reglas de negocio, pues hace énfasis en el hecho de que debemos estandarizar y dar granularidad tanto al código como las tablas en una base de datos.

Si nos enfocamos en el problema presente, ésta es la mejor opción debido a que nos permite específicamente la utilización de software de Microsoft, no sólo por ser el más utilizado, sino porque en la Unidad de Extensión Universitaria sólo se tienen en producción PC's con este software, y por lo tanto el utilizar otro tipo de herramientas supone el hecho de que además de capacitar al personal en el uso y manejo de este nuevo sistema se debe capacitar en el uso del software desconocido para ellos.

Cabe mencionar que este sistema debe ser fácil de usar, pues el perfil de usuario que se tiene en la Unidad de Extensión Universitaria es el básico, y esta 100% dirigido a la utilización de Windows. Para tales fines debemos tomar en cuenta la realización de los siguientes esquemas de **UML**:

2.4.1.2 Diagrama de casos de uso⁵

Se emplean para visualizar el comportamiento del sistema, una parte de él o de una sola clase. De forma que se pueda conocer como responde esa parte del sistema. El diagrama de uso es muy útil para definir como debería ser el comportamiento de una parte del sistema, ya que sólo especifica como deben comportarse y no como están implementadas las partes que define.

Por ello, es un buen sistema el documentar partes del código que deban ser reutilizables por otros desarrolladores. El diagrama también puede ser utilizado para que los expertos de dominio se comuniquen con los informáticos sin llegar a niveles de complejidad. Un caso de uso especifica un requerimiento funcional, es decir indica: "esta parte debe hacer esto cuando pase esto".

En el diagrama nos encontramos con diferentes figuras que pueden mantener diversas relaciones entre ellas:

1.- Casos de uso: representado por una elipse, cada caso de uso contiene un nombre, que indique su funcionalidad. Los casos de uso pueden tener relaciones con otros casos de uso.

2.- Include: Representado por una flecha, en el diagrama de ejemplo podemos ver como un caso de uso, el de totalizar el coste incluye a dos casos de uso.

3.- Extends: Una relación de un caso de Uso A hacia un caso de uso B indica que el caso de uso B implementa la funcionalidad del caso de uso A.

4.- Generalization: Es la típica relación de herencia.

⁵ <http://www.rational.com/uml>

5.- Actores: se representan por un muñeco.

6- Communicates: Comunica un actor con un caso de uso, o con otro actor.

7.- Parte del sistema (System boundary): Representado por un cuadro, identifica las diferentes partes del sistema y contiene los casos de uso que la forman.

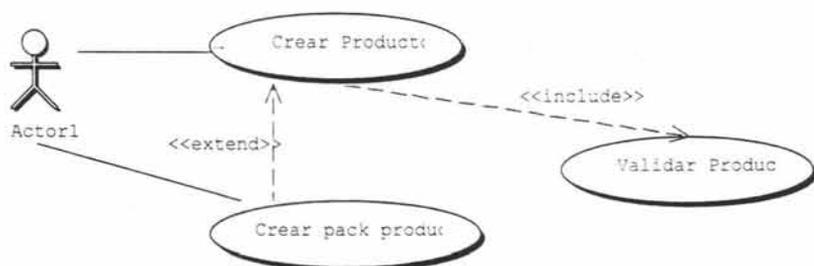


Figura 7: Casos de Uso

En este gráfico encontramos tres casos de uso: Crear producto utiliza Validar producto, y Crear pack productos es una especialización de Crear productos.

Podemos emplear el diagrama de dos formas diferentes, para modelar el contexto de un sistema, y para modelar los requisitos del sistema.

2.4.1.3 Diagrama de clases

Forma parte de la vista estática del sistema. En el diagrama de clases como ya hemos comentado será donde definiremos las características de cada una de las clases, interfaces, colaboraciones y relaciones de dependencia y generalización. Es decir, es donde daremos rienda suelta a nuestros

conocimientos de diseño orientado a objetos, definiendo las clases e implementando las ya típicas relaciones de herencia y agregación.

En el diagrama de clases debemos definir a éstas y a sus relaciones.

1.- La Clase: Una clase esta representada por un rectángulo que dispone de tres apartados, el primero para indicar el nombre, el segundo para los atributos y el tercero para los métodos. Cada clase debe tener un nombre único, que las diferencie de las otras.

2.- Atributo: Representa alguna propiedad de la clase que se encuentra en todas las instancias de la clase. Los atributos pueden representarse solo mostrando su nombre, mostrando su nombre y su tipo, e incluso su valor por defecto.

3.- Método u operación: Es la implementación de un servicio de la clase, que muestra un comportamiento común a todos los objetos. En resumen es una función que le indica a las instancias de la clase que hagan algo.

Para separar las grandes listas de atributos y de métodos se pueden utilizar estereotipos.

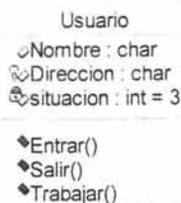


Figura 8: Ejemplo de una Clase

En la figura 8 vemos un ejemplo: La clase usuario contiene tres atributos. Nombre que es public, dirección que es protected y situación que es private. Situación empieza con el valor 3. También dispone de tres métodos Entrar, Salir y Trabajar.

4.-Relaciones entre clases: Existen tres relaciones diferentes entre clases, Dependencias, Generalización y Asociación. En las relaciones se habla de una clase destino y de una clase origen. El origen es desde la que se realiza la acción de relacionar. Es decir desde la que parte la flecha, la destino es la que recibe la flecha. Las relaciones se pueden modificar con estereotipos o con restricciones.

5.-Dependencias: Es una relación de uso, es decir una clase usa a otra, que la necesita para su cometido. Se representa con una flecha discontinua va desde la clase utilizadora a la clase utilizada. Con la dependencia mostramos que un cambio en la clase utilizada puede afectar al funcionamiento de la clase utilizadora, pero no al contrario. Aunque las dependencias se pueden crear tal cual, es decir sin ningún estereotipo (palabreja que aparece al lado de la línea que representa la dependencia) *UML* permite dar más significado a las dependencias, es decir concretar más, mediante el uso de estereotipos.

6.- Estereotipos de relación Clase-objeto⁶

- **Bind:** La clase utilizada es una plantilla, y necesita de parámetros para ser utilizada, con Bind se indica que la clase se instancia con los parámetros pasándole datos reales para sus parámetros.
- **Derive:** Se utiliza al indicar relaciones entre dos atributos, indica que el valor de un atributo depende directamente del valor de otro. Es decir el atributo edad depende directamente del atributo Fecha nacimiento.
- **Friend:** Especifica una visibilidad especial sobre la clase relacionada. Es decir podrá ver las interioridades de la clase destino.
- **InstanceOF:** Indica que el objeto origen es una instancia del destino.

⁶ <http://www.rational.com/uml>

- **Instantiate:** Indica que el origen crea instancias del destino.
- **Powertype:** Indica que el destino es un contenedor de objetos del origen, o de sus hijos.
- **Refine:** Se utiliza para indicar que una clase es la misma que otra, pero más refinada, es decir dos vistas de la misma clase, la destino con mayor detalle.
- **Generalización:** Es la herencia, donde tenemos una o varias clases padre o superclase o madre, y una clase hija o subclase. UML soporta tanto herencia simple como herencia múltiple. Aunque la representación común es suficiente en el 99.73% de los casos UML nos permite modificar la relación de Generalización con un estereotipo y dos restricciones.

7.- Estereotipo de generalización.

- **Implementation:** El hijo hereda la implementación del padre, sin publicar ni soportar sus interfaces. Restricciones de generalización.
- **Complete:** La generalización ya no permite más hijos.
- **Incomplete:** Podemos incorporar más hijos a la generalización.
- **Disjoint:** Sólo puede tener un tipo en tiempo de ejecución, una instancia del padre sólo podrá ser de un tipo de hijo.
- **Overlapping:** Puede cambiar de tipo durante su vida, una instancia del padre puede ir cambiando de tipo entre los de sus hijos.
- **Asociación:** Especifica que los objetos de una clase están relacionados con los elementos de otra clase. Se representa mediante una línea continua, que une las dos clases. Podemos indicar el nombre, multiplicidad en los extremos, su rol, y agregación.

En este diagrama se han creado cuatro clases. La clase principal es Usuario, que tiene dos clases hijas **UsuarioADM** y **UsuarioINF**. El usuario mantiene una relación de asociación con la clase Clave, se indica que es

propietario de una clave, o de un número indeterminado de ellas. Se le crea también una relación de dependencia con la clase Perfil, es decir las instancias de usuario contendrán como miembro una instancia de Perfil.

2.4.1.4 Diagrama de objetos

Forma parte de la vista estática del sistema. En este diagrama se modelan las instancias de las clases del diagrama de clases. Muestra a los objetos y sus relaciones, pero en un momento concreto del sistema. Estos diagramas contienen objetos y enlaces. En los diagramas de objetos también se pueden incorporar clases, para mostrar la clase de la que es un objeto representado.

En este diagrama se muestra un estado del diagrama de eventos. Para realizar el diagrama de objetos primero se debe decidir qué situación queremos representar del sistema. Es decir, si disponemos de un sistema de mensajería, deberemos decidir que representaremos el sistema con dos mensajes entrantes, los dos para diferentes departamentos, dejando un departamento inactivo. Para el siguiente diagrama de clases:



Figura 9: Representación Diagrama de Objetos

Tendríamos un diagrama de objetos con dos instancias de Mensaje, más concretamente con una instancia de **MensajeDIR** y otra de **MensajeADM**, con todos sus atributos valorados. También tendríamos una instancia de cada una de las otras clases que deban tener instancia. Como **CanalEnt**, **INS**, **Distr**, y el **Buzon** correspondiente a la instancia de mensaje que se esté trabajando en ese momento. En la instancia de la clase **INS** se deberá mostrar en su miembro Estado, que está ocupado realizando una inserción. En un diseño no podemos encontrar con multitud de diagramas de objetos, cada uno de ellos representando diferentes estados del sistema.

2.4.1.5 Diagrama de componentes

Se utilizan para modelar la vista estática de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema.

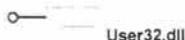
En él, situaremos librerías, tablas, archivos, ejecutables y documentos que formen parte del sistema. Uno de los usos principales es que puede servir para ver qué componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.



A UML Component diagram showing a single component named 'User32.dll'. The component is represented by a rectangle with a small semi-circular notch on its left side, indicating it is a component that provides an interface.

Figura 10: Representación de un Componente de Sistema.

En la figura 10 tenemos un componente del sistema de Windows. En el diagrama de componentes de Windows debe salir este componente, ya que sin el sistema no funcionaría.



A UML Component diagram showing a component named 'User32.dll'. The component is represented by a rectangle with a small semi-circular notch on its left side, indicating it is a component that provides an interface.

Figura 11: Representación de un Componente de Sistema con interfaz.

En esta otra (figura 11), tenemos el mismo componente, pero indicamos que dispone de una interfaz. Al ser una Dll la interfaz nos da acceso a su contenido. Esto nos hace pensar que la representación anterior es incorrecta, pero no es así, solo corresponde a un nivel diferente de detalle.

Como ya hemos indicado antes todo objeto UML puede ser modificado mediante estereotipos, los estándares que define UML son:

- Executable
- Library
- Table
- File
- Document.

Aunque por suerte no estamos limitados a estas especificaciones. Que pasa si queremos modelar un proyecto de Internet donde nuestros componentes son ASP, HTML, y Scripts, y queremos marcarlo en el modelo. Pues utilizamos un

estereotipo. Existen ya unos definidos WAE (Extensiones de Aplicaciones Web).

Podemos modelar diferentes partes de nuestro sistema, y modelar distintas entidades que no tienen nada que ver entre ellas.

- Ejecutables y bibliotecas.
- Tablas.
- API
- Código fuente.
- Hojas HTML.

2.4.1.6 Diagrama de Flujo de Datos (DFD)

Ilustra las funciones que el sistema debe realizar. Se utilizan para describir la transformación de entradas a salidas. Consiste en procesos, agregados de datos, flujos y terminadores.

Los **procesos** se representan por medio de círculos o burbujas, representan las diversas funciones individuales que el sistema lleva a cabo. Las funciones transforman las entradas en salidas.

Los **flujos** se muestran por medio de flechas curvas. Son las conexiones entre los procesos (funciones del sistema) y representan la información que dichos procesos requieren como entrada o la información que generan como salida.

Los **agregados de datos** se representan por medio de dos líneas paralelas o mediante una elipse. Muestran colecciones (o agregados) de datos que el sistema debe recordar por un periodo de tiempo. Cuando los diseñadores de

sistemas y los programadores terminan de construir el sistema, los agregados existirán como archivos o bases de datos.

2.4.1.7 Diagrama Entidad-Relación⁷

Hace énfasis en las relaciones entre los datos. En este apartado podremos implementar las herramientas necesarias para realizar un bosquejo y, a partir de éste, determinar en forma definitiva cómo estará distribuida la base de datos, así como las relaciones existentes entre agregados de datos. Constan de dos componentes principales:

Tipos de objetos: Se representan por medio de un rectángulo en el diagrama. Esto representa una colección o conjunto de objetos (cosas) del mundo real cuyos miembros juegan algún papel en el desarrollo del sistema; pueden además ser identificados de manera única y ser descritos por uno o más atributos.

Relaciones: Se representan por medio de líneas terminadas por rombos en el diagrama y son la serie de conexiones o asociaciones entre los tipos de objetos que están conectados con la relación por medio de flechas.

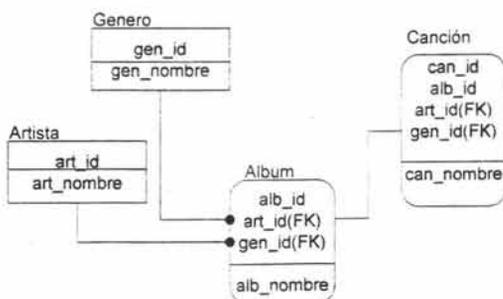


Figura 12: Diagrama Entidad-Relación.

⁷ <http://www.rational.com/uml>

2.4.2 Diseño

Es propiamente la elección de la arquitectura bajo la cual se desarrollará nuestro sistema, esto basado en las características de los procesos que queremos automatizar o cambiar mediante la implantación del mismo.

El llamado diseño abarca desde la arquitectura que vamos a utilizar, hasta la interfaz de usuario que vamos a implementar, las tareas que va a realizar nuestro sistema, así como su distribución, en base a las necesidades del cliente y a las posibilidades que se tengan, tanto de infraestructura, como de recursos humanos, y cabe mencionar que esta etapa tiene que tomar en cuenta a los usuarios finales, así como sus características, responsabilidades y aptitudes.

2.4.2.1 Capa de Usuario

Se refiere a aquella parte del sistema a la que el usuario final tendrá libre acceso, de tal forma que deberá diseñarse en base a quien la va a utilizar, para lo que se cuenta con una serie de requisitos a cubrir para que se adapte perfectamente a la situación de nuestros usuarios.

2.4.2.2 Diseño de Interfaz Gráfica

El diseño de la interfaz se basa en los procesos y tareas de los usuarios; de lo contrario, el sistema sería un fracaso. Se debe sensibilizar para únicamente ofrecer lo necesario para que el usuario trabaje.

Para la etapa de diseño de la interfaz gráfica se deben tomar en cuenta distintos aspectos, como son:

1.- Entender quien usará el sistema y para qué: Esta información se obtuvo en la definición de casos de uso y perfiles de usuario, es decir, es un "análisis de usuarios y tareas".

La necesidad de este análisis se basa en que se deben tener todos los elementos y secuencias necesarios para que el usuario interactúe con el sistema y realice sus operaciones en la base de datos, considerando sus características y necesidades específicas.

2.- Utilizar un estándar o normatividad de la empresa u organización: Si se cuenta con un estándar para el diseño de la interfaz es necesario ajustarse lo más posible a éste o, en caso contrario, justificar por qué no será así.

Es frecuente que se tengan sistemas de información o sistemas Web en producción, en donde ya se tienen definidos los colores, tamaños de letras, logos, frases, etc. por lo tanto se pueden analizar interfaces existentes con las cuales los usuarios han trabajado y después tomar y construir ideas para posteriormente utilizarlas en la nueva interfaz.

3.- Bosquejar un diseño: La descripción del bosquejo del diseño debe ser puesto en papel, el cual fuerza a que se piense acerca de los elementos de las pantallas y el orden que éstos tendrán, aunque la información se toma del análisis y los diagramas en ocasiones, dada la gran cantidad de elementos se pueden tener omisiones si se decide programar directamente.

4.- Crear un prototipo: Después de hacer el bosquejo y comprobar que se están considerando todos los requerimientos, si es necesario, se debe implementar un prototipo que se presente a los usuarios; obviamente, sin conexiones a la base de datos, para comprobar la congruencia de la interfaz con los requerimientos del usuario, aunque es un trabajo más detallado, en ocasiones puede ayudar a encontrar omisiones y un orden inadecuado.

5.- Evaluación con los usuarios: Existen muchos problemas que solamente aparecerán cuando el diseño se está probando con los usuarios. La evaluación debe de ser hecha con la gente cuyo conocimiento y expectativas se acerquen más a los usuarios reales.

Los usuarios deben ser cuestionados sobre el rendimiento de una o más tareas representativas que han sido diseñadas en el sistema para su mantenimiento. Ellos deben de ser cuestionados en la técnica de "pensamiento en voz alta" la cual consiste en que el usuario vaya narrando todo lo que está pensando al momento de realizar una tarea.

2.4.2.3 Programación de Cliente

Es aquí donde se engloban las actividades necesarias para llevar a cabo la codificación del cliente. Una vez que se ha comprobado la congruencia del diseño con los usuarios se puede empezar a construir la interfaz. Trate de anticipar cambios menores o secundarios con sencillos cambios de variables.

Por ejemplo: Si tiene que escribir su propia rutina de despliegue para un menú, no fije en el código parámetros como color, tamaño, o número de elementos y trate de anticipar cambios mayores con código muy claro y estructurado.

Si en una revisión posterior en el diseño se requiere que un menú sea cambiado por una función más genérica, estos cambios de código deberán de ser insignificantes. Para ello las herramientas para la construcción de interfaz son fundamentales.

2.4.2.4 Capa de Base de Datos

Considerando a la base de datos como la "base" del sistema, su buen diseño es fundamental para las etapas de desarrollo, implementación y utilización del sistema.

Diseño de la Base de Datos: Se debe comenzar por identificar a las entidades involucradas en el sistema; estas entidades tienen características o atributos que las describen.

Lo anterior no es más que la identificación de las tablas y sus campos, siempre se deberá cuidar la integridad referencial y evitar totalmente la redundancia. Esto se logra mediante la correcta y eficaz utilización de "llaves".

- **Llaves primarias:** Sirven para identificar y distinguir de manera única e irrepetible a los registros almacenados en la base de datos.
- **Llaves foráneas:** Se utilizan para relacionar a dos o más tablas, esto a través de campos en común.
- **Condiciones:** Son condiciones que debe respetar el contenido de los campos en particular.

Las relaciones entre las tablas pueden ser:

- **Uno a uno.** (Un alumno estudia una carrera).
- **Uno a muchos.** (Un alumno estudia muchas materias).
- **Muchos a muchos.** (Muchas materias se imparten en muchos grupos).

Relacionado con el diseño de la base de datos está la elaboración de un diccionario de datos. El diccionario de datos describe claramente el diseño físico de la base de datos, es decir, indica el nombre de cada tabla y de cada campo que la constituye, así como tipo de dato que se almacena y las condiciones que se deben cumplir. Como lo indica la siguiente tabla:

| Genero | | Guarda los géneros de música | |
|--------|-----|------------------------------|---------------------|
| | gen | int | identificador único |
| | gen | va | nombre del género |

| Artista | | Guarda los artistas | |
|---------|-----|---------------------|---------------------|
| | art | int | identificador único |
| | art | va | nombre del artista |

| Album | | Guarda los álbumes | |
|-------|-----|--------------------|---------------------|
| | alb | int | identificador único |
| | art | int | identificador único |
| | gen | int | identificador único |
| | alb | va | nombre del álbum |

| Canción | | Guarda las canciones | |
|---------|-----|----------------------|----------------------|
| | can | int | identificador único |
| | alb | int | identificador único |
| | art | int | identificador único |
| | gen | int | identificador único |
| | can | va | nombre de la canción |

Tabla 6: Ejemplo de un diccionario de datos

2.4.2.5 Capa de Negocios

En esta capa se encuentran la lógica y reglas de procedimientos necesarios para realizar las operaciones del sistema, esto es, las actividades operacionales, controles de consistencia, validaciones, cálculos, etc.

Esta capa interactúa entre la base de datos y la presentación ante algún requerimiento de búsqueda o acceso de datos por parte de los usuarios. Aquí se encuentran también los distintos "servicios" relacionados con el sistema.

Todos estos apartados ya han sido tratados al inicio del capítulo, pues ésta es exactamente la parte en la que se implementan todos los procesos ya mencionados en la lógica del negocio y las reglas del negocio. Esto también incluye los dos pasos siguientes, pues ya se han mencionado los mecanismos de acceso a la información, así como los distintos gestores.

Acceso a Datos:

Gestión de la Base de Datos:

Distribución del Código:

Estandarización del Código:

2.4.2.6 Herramientas de Diseño⁸

Una vez hecho el análisis previo, estableceremos la arquitectura del sistema, que es precisamente la herramienta a utilizar, esto incluye determinar si será un sistema simple diseñado para correr en una sola máquina, un sistema **Three-Tier** consistente en un cliente y un servidor, o un sistema **Multi-Tier** con objetos interfaz de usuario separados de los objetos business, separado de la base de datos, cada uno corriendo en plataformas distintas.

Una aproximación a dirigir el diagrama de clase para un sistema complejo es separar el diagrama en secciones que muestren la lógica de la aplicación, el diseño de la interfaz de usuario, y las clases implicadas con el almacenamiento de los datos. Esto se puede hacer físicamente segmentando el diagrama de

⁸ <http://www.microsoft.com/spanish/msdn/arquitectura/das/distapp.asp>

clase, usando diagramas separados para cada sección, o simplemente añadiendo una propiedad a cada clase que 'tracks' cada 'tier' al cual pertenece.

Para la elaboración de este tipo de sistemas la opción más acertada es la programación en **Multicapas**, ya que existe una gran variedad de opciones englobadas en este término, pero en su mayoría se enfocan a la simplificación en el manejo de la aplicación y la gestión rápida de la información, que en este caso nos resultará muy útil debido a la poca capacidad de los equipos disponibles.

La programación en múltiples capas es la técnica más efectiva en aplicaciones empresariales, debido a la fácil administración que ocasiona el dividir los componentes de la aplicación en capas y la rapidez que esto implica en programas orientados a Cliente-Servidor.

Esta Metodología consiste en dividir los componentes primarios de la aplicación, programarlos por separado y después unirlos en tiempo de ejecución.

El modelo COM de objetos es uno de los pilares de la programación Multitier (la llamaremos así de ahora en adelante, debido a que en la mayoría de escritos figura de esta forma) y asimismo, el modelo Windows DNA el cual define también Multitier como la mejor forma de programación.

¿Qué es Programación Multitier?

La programación multitier consiste en dividir los componentes primarios de la aplicación, programarlos por separado y luego unirlos, ya sea en tiempo de ejecución o en el mismo código. De esto naturalmente surge una duda ¿Cuáles son los componentes primarios de la aplicación? En este momento se habla de programación de cuatro capas, aunque se hablará sobre la programación en tres capas o **Three-Tier** que es el más documentado y que en este momento

permite mayor abstracción para aplicaciones empresariales. Estas tres capas las podemos denominar como: **Presentación, Reglas del Negocio y Acceso a Datos**. A continuación se explica cada una:

2.4.2.7 Presentación:

Esta capa permite al usuario visualizar y acceder al sistema a través de una interfaz gráfica de usuario (GUI Interfaz Gráfica de Usuario). También puede llamarse servicio de usuario, Front-End, cliente, aplicación o aplicación cliente.

En esta capa se diseña todo lo que forma la interfaz de usuario y la interacción del usuario con el software. Sus funciones se definen como sigue:

- Obtener información del usuario.
- Enviar la información del usuario a los servicios de negocios para su procesamiento.
- Recibir los resultados del procesamiento de los servicios de negocios.
- Presentar estos resultados al usuario.

En la actualidad podemos distinguir 3 formas de interfaz de usuario:

1.- Interfaz clásica de ventanas: Está presente en los programas a los que estamos acostumbrados en un 90%, una ventana principal y un menú van ofreciendo posibilidades al usuario que selecciona, e introduce la información para ser procesada, es construida con las herramientas de desarrollo tradicionales, tales como Visual Basic, Visual C++, Java, etc.

2.- Interfaz integrada en programas de gestión: Son las aplicaciones que el usuario ya tiene instalada en su equipo, por ejemplo, cualquier aplicación de la suite de **Office**, permite que un programador almacene código fuente y todos

los elementos adicionales necesarios para construir una aplicación en un documento de **Office**.

Cuando el usuario quiere trabajar con la aplicación, simplemente abre el documento, y toda la maquinaria se pone en marcha igual que si estuviéramos en un entorno tradicional. La diferencia es que estarán presentes los elementos de la aplicación servidora si no se han deshabilitado los menús de Word, las cajas de diálogo, etc. Tiene la ventaja de que no requiere de instalación, ya que la aplicación que lo maneja se encuentra ya instalada en el equipo del usuario.

3.- Navegadores: Éste es el sistema más fácil de instalar, ya que literalmente no requiere instalación, por lo menos no una independiente, pues se encuentra dentro de la instalación del sistema operativo, sea **Windows** o **Linux**, en forma predeterminada, pues hoy en día la navegación web es un estándar entre los usuarios comunes y avanzados, las páginas son descargadas de un sitio web y en ellas se encuentra el sistema de navegación, así como las reglas de negocio y, de haberla, la base de datos.

Aunque tiene varias desventajas, como el hecho de que su velocidad se determina por el ancho de banda de la conexión a Internet y por la concurrencia de usuarios simultáneos, de tal forma que una simple consulta puede llevarse a cabo en hasta 10 minutos cuando se está trabajando en una hora pico, o simplemente nos puede ser imposible en determinadas horas del día. Todo esto sin contar el hecho de que podríamos añadir componentes escritos en Java, lo cual hace mucho más lento el proceso.

2.4.2.8 Reglas del Negocio

En esta capa se encuentran las reglas y lógica de procedimientos necesarios para realizar las operaciones del sistema, es decir, las actividades operacionales, controles de consistencia, validaciones, cálculos, etc.

Esta capa interactúa entre la base de datos y la presentación ante algún requerimiento de búsqueda o acceso de datos por parte de los usuarios. Aquí se encuentran también los distintos servicios relacionados con el sistema.

Son todas las subrutinas creadas con el propósito de regular alguna acción del usuario. Por ejemplo, en una aplicación bancaria una regla del negocio podría ser que el cliente no debe retirar por taquilla más de U.S. \$10.000 y en caso de una petición de este tipo se genere un error.

Son el "puente" entre un usuario y los servicios de datos. Responden a peticiones del usuario (u otros servicios de negocios) para ejecutar una tarea de este tipo. Cumplen con esto aplicando procedimientos formales y reglas de negocio a los datos relevantes.

Cuando los datos necesarios residen en un servidor de bases de datos, garantizan los servicios de datos indispensables para cumplir con la tarea de negocios o aplicar su regla. Esto aísla al usuario de la interacción directa con la base de datos.

Una tarea de negocios es una operación definida por los requerimientos de la aplicación, como introducir una orden de compra o imprimir una lista de clientes. Las reglas de negocio (business rules) son políticas que controlan el flujo de las tareas.

Como las reglas de negocio tienden a cambiar más frecuentemente que las tareas específicas de negocios a las que dan soporte, son candidatos ideales para encapsularlas en componentes que están lógicamente separados de la lógica de la aplicación en sí. El nivel de servicios de negocios es responsable de:

1. Recibir la entrada del nivel de presentación.
2. Interactuar con los servicios de datos para ejecutar las operaciones de negocios para los que la aplicación fue diseñada a automatizar (por ejemplo, la preparación de impuestos por ingresos, el procesamiento de órdenes y así sucesivamente).
3. Enviar el resultado procesado al nivel de presentación.

Esta parte del sistema es la que nos permite la reutilización del código, dividiéndolo en **Componentes**, que no son otra cosa que fragmentos de código compilado, que reside en un archivo separado y que se registra en la máquina que va a utilizarlo. Una vez registrado, el componente puede ser utilizado por cualquier aplicación, y no sólo por aquella para la que ha sido pensada originalmente.

Por ejemplo, si creamos un componente para devolver el conjunto de clientes que pertenecen a una ciudad concreta, también podría ser usado por personal administrativo desde una herramienta como Office para crear listados de clientes para los representantes. Aunque, el que sea esto posible, depende de cómo haya sido construido el componente. Es aquí donde el concepto de estandarización se hace importante, como los componentes Active X, que han sido construidos utilizando la especificación **DCOM**⁹, que es una tecnología abierta, lo que permite que sea utilizada por cualquier desarrollador, pero que fue inventada por **Microsoft**.

⁹ <http://www.microsoft.com/spanish/msdn/arquitectura/das/distapp.asp>

Podemos crear componentes polivalentes, que pueden ser utilizados en diferentes contextos, haciendo que sea la lógica del negocio lo que importe, y no la lógica de las aplicaciones individuales. Una vez registrado, un componente se convierte en un servicio del sistema operativo.

Además, se pueden tener componentes **DCOM** en diferentes plataformas, de tal manera que una aplicación Windows puede llamar a componentes **DCOM** que residan en una máquina Sun-Solaris, sin problemas.

No sólo es posible en esa arquitectura, muchas aplicaciones corporativas de gran volumen, poseen componentes **COM** conectores para permitir que otras aplicaciones, utilizando programas de gestión clásicos o suites como **Office**, puedan manipular los datos producidos por ellas.

Asimismo si un componente es incluido en una página Web y un lector solicita la página, el componente se descarga junto con el resto de la información, un vez descargado se solicita permiso al usuario para la instalación en su máquina, y en caso de obtenerlo se registra quedando a disposición de los programas que lo soliciten.

La idea es que cualquier aplicación que invoque el uso de uno de estos componentes puede hacerlo igual sobre una máquina remota que sobre una local.

Una buena razón para dividir la lógica de negocio en componentes, es la probabilidad de que esta lógica cambie, teniendo que reescribir el código de la aplicación. De este modo, se reescribe el componente correspondiente a la lógica de negocio afectada y no la totalidad del código.

2.4.2.9 Acceso a datos

En esta capa programamos todo lo que tiene que ver con el acceso a la base de datos. Esta capa queda encargada de tomar la información de la base de datos dada una petición de la capa de Reglas del Negocio, que a su vez es generada por la capa de presentación. Esta capa es la encargada de:

4. Almacenar los datos.
5. Recuperar los datos.
6. Mantener los datos.
7. La integridad de los datos.

Para tales efectos es necesario establecer un mecanismo de acceso a la información, que es la manera en que la aplicación cliente puede consultar y/o modificar los registros en una base de datos, actualmente el nivel de estandarización para dichos mecanismos es notable, a continuación se mencionan los principales:

1.- ODBC (Conectividad Abierta a Base de Datos)¹⁰

Se trata de un estándar que crea un controlador específico para la base de datos a manejar en el servidor adecuado. A partir de ahí, el programador utiliza una librería de objetos, como RDO o DAO para el acceso, de forma que todos los objetos de la base de datos aparecen ante él como objetos programables dentro de esa jerarquía.

Si se cambian los datos de un servidor a otro, la aplicación permanece intacta, solo hay que cambiar el controlador y todo puede seguir funcionando como estaba.

¹⁰ Booch, G. 1988. Object Oriented Development. Trans. of Soft. Eng. Vol. SE-12. Num. 2. Feb. 1986. pp. 211-221.

2.- Estrategia UDA(Acceso Universal a Datos)¹¹

El término **UDA** lo podemos definir como la idea de que los datos pueden ser accedidos desde cualquier punto en cualquier momento y sin problemas.

Dicho mecanismo, según la propuesta de Microsoft es la estrategia **UDA**, que se basa en una extensión de **OLE** denominada **OLE-OS**, que nos permite acceder a datos de formatos heterogéneos sin que eso suponga un problema de modificación de las formas de acceso.

Se puede abrir un correo electrónico, o un documento de Word, una hoja de Excel o incluso crear una tabla con el contenido de los archivos existentes en un directorio.

Lógicamente, habrá un traductor igual que sucedía con **ODBC**, ese traductor que permite el acceso es el proveedor **OLE-DB**, de los que existen ya un buen número, entre los que se cuenta un proveedor **OLE-DB** para **ODBC**, en el caso que el servidor de datos al que pretendemos acceder no tenga todavía disponible el proveedor **OLE-DB** nativo.

Con esta filosofía podemos poner en marcha la programación mediante la librería **ADO** (Objeto de Datos ActiveX), que para facilitar las cosas, se incluye de forma predeterminada en el sistema operativo **Windows 2000**.

Para el caso de Java, también existe un estándar de acceso a datos, denominado **JDBC** (Conectividad a Bases de Datos Java), que se comporta de forma similar a como lo hacen los controladores **ODBC** para cualquier otro lenguaje.

¹¹ Booch, G. 1988. Object Oriented Development. Trans. of Soft. Eng. Vol. SE-12, Num. 2, Feb. 1986, pp. 211-221.

Un aspecto importante de la capa de acceso a datos es la existencia de un Servidor de Datos, también llamado **DBMS** (Sistema Gestor de Bases de Datos) Sistema Manejador de Bases de Datos, que es un software utilizado para gestionar datos de una o más aplicaciones, los más populares son:

- SQL Server
- Oracle
- MySQL
- PostgreSQL
- DB2
- Informix
- Sybase

2.4.2.10 Diferencia con la programación tradicional

Probablemente se estará preguntando, ¿Por qué esto es mejor que programar todo de una sola vez y qué diferencia hay entre una Metodología y otra? Los esquemas siguientes nos ayudarán a visualizar mejor estas diferencias.

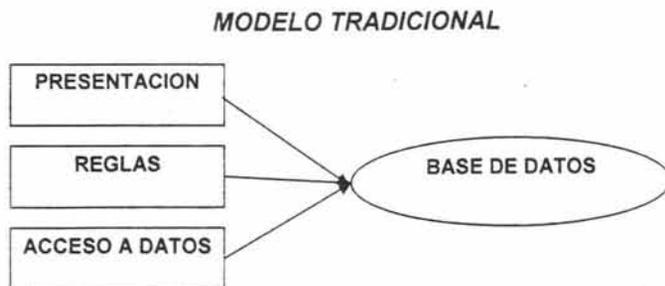


Figura 13: Representación del modelo Tradicional de Programación

En la figura 13, todas las capas están ubicadas en un único elemento y al actualizar un elemento se tendrán que afectar los otros dos.

También tenemos que los sistemas **Cliente-Servidor** cuentan con una estructura tradicional, la cual se representa como se observa en la figura 14:

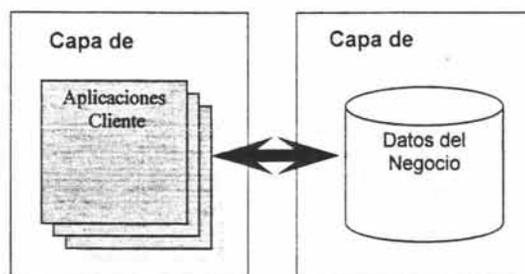


Figura 14: Estructura de Sistema Cliente-Servidor

En este esquema (Figura 14) existen dos capas, el cliente y el servidor, éste se encuentra ubicado normalmente en otra máquina, y suele ser un Gestor de Bases de Datos, como **SQL Server**, **Oracle** o **MySQL**, al que accedemos directamente desde nuestra aplicación cliente.

MODELO THREE-TIER



Figura 15: Representación del modelo Three-Tier

En la figura 15, el cambio en la capa de presentación no afectará reglas del negocio y se pueden actualizar las funciones de reglas del negocio sin tener que cambiar el acceso a datos.

Como podemos apreciar en las gráficas, resulta mucho más práctico programar las capas por separado, sobre todo en situaciones que tenemos un equipo de trabajo de diferentes programadores o el proyecto es para una empresa con varias sucursales con políticas independientes.

No podemos dejar de lado el hecho de que aquí se requiere un sistema **Cliente-Servidor**, por lo que a continuación se presenta la estructura de este sistema en tres capas o **Three-Tier**:



Figura 16: Estructura de Sistema Three-Tier

En este esquema se evita que la capa de aplicación tenga contacto directo con el gestor de la base de datos, pues se introduce una nueva capa en la que se encuentran las reglas de negocio y a la que el cliente acude en cada operación.

¿Cuándo es recomendable utilizar esta Arquitectura?

Como todo en nuestra querida profesión, se debe analizar muy detenidamente la Arquitectura a implementar, porque ésta puede implicar un costo muy alto o posibles fallas en el programa. Lo recomendable para utilizar la Arquitectura multitier es:

- Cuando el proyecto es muy grande.
- Cuando son varios los programadores y existen dificultades para coordinarlos, por ejemplo cuando el software se está desarrollando con programadores freelance que están muy lejos unos de los otros.
- Cuando el programa es para una empresa con varias sucursales y cada sucursal puede tener algunos métodos y políticas diferentes. Por ejemplo, una empresa tiene una sucursal en Los Ángeles y otra en Boston; la sucursal de Los Ángeles sólo permite compras a crédito por un valor de U.S\$10.000 y la de Boston de U.S \$12.000. Al utilizar un modelo multitier en este caso podemos cambiar el límite del crédito sin tocar la capa de presentación ni la de acceso a datos.
- Cuando el gestor de la base de datos no nos permite implementar todas las reglas de negocio con la flexibilidad que se desea.
- Cuando se utiliza más de un gestor de bases de datos, pues no es posible asegurar el cumplimiento de todas las reglas, y mucho menos conservar la integridad de la base de datos, pues se encuentra descentralizada.
- Cuando es absolutamente necesario el evitar que se introduzcan datos duplicados, o que estén ligados a registros inexistentes, en una empresa se puede dar el caso de la facturación y manejo de devoluciones.

¿Qué beneficios puedo esperar?

- La posibilidad de integrar aplicaciones que accedan a las mismas bases de datos de una forma sencilla.
- Separar las reglas de negocio de las interfaces especialmente en entornos multiplataforma permite que las reglas se cambien con un mínimo impacto sobre los usuarios de las aplicaciones.

- El uso de modelos Three-Tier aumenta increíblemente la flexibilidad a la hora de aplicar las posibilidades de la informática para aspectos específicos de la problemática del cliente.
- Es fácil construir nuevas aplicaciones desde los componentes instalados si las reglas del negocio están en unos servidores de aplicaciones más que en cada aplicación.
- Herramientas como el ODBC, que fuerza a las Bases de Datos a ser abstractas y transformarse en genéricos almacenes de datos, permiten permanecer completamente flexibles a la hora de determinar donde guardar los datos independientemente de la aplicación que se use.
- La posibilidad de utilizar las mismas reglas de negocio con distintas aplicaciones, sin la necesidad inherente de duplicar líneas de código para cada una de ellas, lo que ahorra tiempo, esfuerzo y sobre todo, dinero.
- Al poder existir varias aplicaciones que utilicen las mismas reglas de negocio, además de la misma base de datos, al mismo tiempo, podemos obtener una mejor distribución en la carga de trabajo.
- La descentralización y delegación de funciones entre cada uno de los usuarios involucrados en el manejo del sistema.

En base a estos puntos hemos llegado a la conclusión de que la mejor metodología para este caso es la **Arquitectura de Tres capas** o **Three-Tier**. Existen distintas modalidades de ella, pero todas coinciden en el número de componentes a utilizar o programar; es decir, las mencionadas tres capas. A continuación se da una reseña acerca de la programación en múltiples capas y las ventajas que ofrece programar en tres capas.

Existen muchas variantes de esta arquitectura, pero sin duda, al tener a nuestro alcance equipos que corren bajo Plataformas y Aplicaciones propias de Microsoft, debemos buscar una que se adecue a la situación existente en la

Unidad de Extensión Universitaria, por lo que se ha decidido la utilización de una propuesta hecha por el mismo Microsoft para desarrollar Sistemas de Información mediante la utilización de herramientas desarrolladas por esta misma compañía.

Por esto se propone la utilización de lenguajes de programación tales como Visual C++, Visual Basic, Visual Basic.Net, etc. para desarrollar la **Interfaz de Usuario**. Y la utilización de motores como Access, Microsoft SQL Server, entre otros para desarrollar la **Base de Datos**.

En tal situación lo propio es listar las etapas de desarrollo que se deben seguir para llevar a buen término esta metodología:

- Pensar en utilizar esta metodología obliga a establecer mecánicas de diseño algo particulares.
- En principio, se debe pensar en "roles" y "participantes", más que en estructuras de almacenamiento de información (Entidad – Relación).
- Es necesario aislarse de la "forma" concentrándose en la "función", en las características.
- Definir cada participante del "negocio" como un "objeto", determinando sus particularidades (Propiedades), las cosas que es capaz de hacer (Métodos), y aquellas cosas que podrían sucederle (Eventos), así como las dependencias que pueda tener con otros objetos.
- A partir de allí, trasladar a Objetos definidos como Clases para las Reglas de Negocio, determinando además los "Objetos de presentación" (Ventanas, páginas de un sitio Web, etc.).
- Cada Propiedad relevante de cada objeto debe persistir en el tiempo, lo cual implica definir campos en una tabla para cada una de ellas, lo cual va dibujando el diseño de Entidad-relación de la base de datos, que

determina además los elementos que van a constituir los servicios de datos.

- Establecer esta mecánica adecuadamente, facilita la conceptualización de la aplicación y su posterior desarrollo.
- Con el diseño bien determinado, es factible aumentar la granularidad del sistema de modo tal de poder "distribuir" el desarrollo de la misma en pequeños fragmentos de código, totalmente independientes, que pueden generarse en paralelo, disminuyendo marcadamente los tiempos de codificación.
- En este esquema, la carga de tiempo está en el diseño. Es allí donde es imperativo dedicar el mayor esfuerzo para evitar "contramarchas" posteriores.
- Tecnológicamente, deberán tomarse algunas consideraciones antes de comenzar la codificación, pero esto resulta fácil, luego de realizar la primera aplicación.

Como puntos importantes:

- El acceso a datos debe encapsularse en un componente. Esto permite, si fuese necesario, cambiar el origen de los datos o sencillamente reemplazar dicho componente por otro que optimice el rendimiento para el tipo de base de datos específica de que se trate.
- Las reglas de negocio NO deben presindir información. Esto es importante en términos de escalabilidad. Para obtener mejores resultados, es marcadamente conveniente que las mismas corren dentro de **Microsoft Transaction Server**.
- Esto permite que los objetos creados puedan "compartirse" entre diversos clientes, pero es imperativo que ninguno "guarde" características entre una llamada y otra al objeto, dado que, en ese caso, **MTS** se ve obligado a "reservar" cada objeto para cada cliente en

función de evitar "confusiones" de información. Tomemos como ejemplo un objeto "Tarjeta de Crédito".

- Si el mismo fuese llamado para realizar una compra por una aplicación cliente, y ésta le pasa el número de tarjeta para, en una segunda operación, realizar la compra, cabría la posibilidad que otra aplicación cliente, llamando al mismo objeto, le pase otro número, quedando entonces la primera operación, registrada a la segunda tarjeta. Por evitar esto, **MTS NO** permite reusar objetos que tengan propiedades "persistibles". Se dice entonces que los objetos, para ser adecuados para MTS, no deben conservar estados (ser "stateless").
- Dejar las transacciones a cargo de **MTS**. Esto es altamente recomendable, dado que dicho servicio se encarga de consistir la información de modo tal que las operaciones se realicen completas (todo o nada). Lamentablemente, esto depende de ciertas características que las bases de datos deben soportar. En principio, ser compatibles con Coordinador y Distribuidor de Transacciones de Microsoft (MS DTC).

Esto determina que sólo algunas bases de datos puedan operar adecuadamente. En estos momentos, tanto **SQL Server** en sus versiones 6.5 como 7.0, **Oracle** desde la versión 7.3 en adelante y ciertas acciones realizadas en **AS 400** a través de **COMTI** (Integrador de Transacciones COM), sean compatible con esta facilidad.

2.4.3 Desarrollo o Programación

Para poder llevar a cabo el desarrollo de un sistema debemos contar con las herramientas de programación adecuadas para cada caso, pues no todos los lenguajes son aptos para todos los sistemas, y mucho menos si en la etapa de análisis se obtuvo como resultado el hecho de que nuestra infraestructura es limitada, por lo cual realizamos una comparativa ligera para determinar que

herramientas se utilizarán, para lo cual se tomaron en cuenta los siguientes puntos a desarrollar:

- **Modelado de la Base de Datos**
- **Acceso a datos, gestión de la base de datos**
- **Diseño de Interfaz Gráfica, Programación de Cliente**
- **Generación de reportes**

2.4.3.1 Herramientas de Desarrollo o Programación

Aunque cabe mencionar que la decisión final se tomará principalmente en función de la capacidad de la infraestructura con la que se cuenta, así como la potencia, funcionalidad y consumo de recursos.

1.- Modelado de la Base de Datos

Para poder modelar una base de datos coherente existen varias herramientas en el mercado, tanto de **Microsoft** como de otros fabricantes, es por ello que basamos nuestro análisis en los dos principales y más comunes:

| | PLATAFORMA | VENTAJAS | DESVENTAJAS |
|-----------------------------|------------|--|--|
| DB BUILDER ERWIN 4.0 | -WINDOWS | -Capacidad para modelar grandes bases de datos - Soporte a nivel Industrial - Amplia gama de funciones para generar bases de datos | -Precio elevado -No disponible en español -Interfaz de usuario poco amigable |

| | PLATAFORMA | VENTAJAS | DESVENTAJAS |
|-------------------|------------|--|---|
| VISIO 2003 | -WINDOWS | - Disponible en español - Interfaz amigable e intuitiva - Versatilidad - Permite generar una base de datos compatible con Access -Precio relativamente económico - Compatibilidad con toda la gama de productos Microsoft | - Al manejar grandes cantidades de tablas disminuye en gran medida su performance - No es una aplicación muy robusta, es decir, es inestable |

Tabla 7: Herramientas para el modelado de base de datos

2.- Acceso a datos, gestión de la base de datos

En esta etapa en particular existen infinidad de productos, de igualmente variados fabricantes, sólo por mencionar algunos tenemos Microsoft SQL Server, Oracle, Informix, Paradox, Access, Dbase, My SQL, entre otros, pero en esta ocasión nos limitaremos a comparar las de más común utilización:

| | PLATAFORMA | VENTAJAS | DESVENTAJAS |
|-----------------------------|-----------------|--|--|
| ORACLE | -LINUX -UNIX | <ul style="list-style-type: none"> - Estabilidad - Soporte Técnico - Manejo de Gran Cantidad de registros - Consultas simultaneas - Seguridad | <ul style="list-style-type: none"> - Precio Elevado - Altos costos de mantenimiento - Difícil mantenimiento - Requiere amplio espacio de almacenamiento - Poco Portable |
| MICROSOFT SQL SERVER | -WINDOWS | <ul style="list-style-type: none"> - Estabilidad - Multiusuario - Útil en manejo de grandes cantidades de información - Fácil administración - Soporte técnico - Seguridad | <ul style="list-style-type: none"> - precio elevado - Altos costos de mantenimiento - Nivel de fragmentación medio |
| ACCESS 2003 | -WINDOWS | <ul style="list-style-type: none"> - Fácil instalación - Portable - Administración simple - Compatible con software de Microsoft para desarrollo de aplicaciones - Ligero - Precio accesible - Soporte de instrucciones SQL | <ul style="list-style-type: none"> - Poca seguridad - Medianamente inestable - Maneja un numero limitado de registros - No apto para grandes empresas |

Tabla 8: Cuadro comparativo entre gestores de base de datos

3.- Diseño de Interfaz Gráfica, Programación de Cliente

Para estas tres etapas se requiere sólo de una aplicación, pues existen en el mercado algunas opciones similares entre sí que nos proporcionan la opción de programar en base a eventos ligados a una interfaz gráfica de diseño propio, por lo que nos limitaremos a mencionar sólo unas cuantas.

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

| | PLATAFORMA | VENTAJAS | DESVENTAJAS |
|-------------------------|------------|---|--|
| VISUAL BASIC 6.0 | -WINDOWS | <ul style="list-style-type: none"> - Rápido aprendizaje - Compatible con software de Microsoft para desarrollo de aplicaciones - Ofrece una interfaz gráfica Bastante clara - Permite el desarrollo de aplicaciones prácticamente sin limitación - Es uno de los lenguajes de programación más comerciales - Amplia gama de material gráfico, escrito y multimedia que sirven de apoyo para su utilización. | <ul style="list-style-type: none"> - Poca seguridad - Medianamente Inestable - Presenta problemas al interactuar con bases de datos de gran tamaño - Mala gestión de los recursos disponibles en cuanto a hardware |
| VISUAL C++ | - WINDOWS | <ul style="list-style-type: none"> - Rápido aprendizaje - Compatible con software de Microsoft para desarrollo de aplicaciones - Ofrece una interfaz gráfica Bastante clara - Permite el desarrollo de aplicaciones prácticamente sin limitación - Es uno de los lenguajes de programación más comerciales - Amplia gama de material gráfico, escrito y multimedia que sirven de apoyo para su utilización. | <ul style="list-style-type: none"> - Es de un nivel más alto de programación. - Poco portable - Uso ligeramente más complicado que visual Basic |

Tabla 9: Cuadro comparativo entre programas de interfaz gráfica

4.- Generación de reportes

Esta no es en sí una etapa de la metodología, pero supone una importante porción de la programación de la aplicación cliente, pues está vinculado a ésta, ya que en caso de ser requerido por el usuario, nuestro sistema debe generar reportes impresos de la información que se maneja con él, de tal forma que se hace necesaria la utilización de software especializado para la generación de reportes, para lo cual se encontró que dentro de los productos disponibles, además de la opción de generarlo mediante código, el

más viable es **Crystal Reports** de la compañía **Seagate**, pues es sencillo, ligero y flexible, aunque no por eso deja de ser poderoso.

Este programa es compatible con **Visual Basic**, y como plus, maneja un esquema propio de fórmulas, así como el utilizado por **Visual Basic**, aunque sólo a partir de la versión 8.0, lo que lo hace aún más práctico.

| | PLATAFORMA | VENTAJAS | DESVENTAJAS |
|----------------------------|------------|--|--|
| CRYSTAL REPORTS 7.0 | - WINDOWS | <ul style="list-style-type: none"> - Rápido aprendizaje - Compatible con software de Microsoft para desarrollo de aplicaciones. - Permite el desarrollo de reportes en una forma rápida y versátil. - Amplia gama de material gráfico, escrito y multimedia que sirven de apoyo para su utilización. - Elaboración de informes en el web. - Rendimiento. | <ul style="list-style-type: none"> - Costo considerablemente alto - Interfaz gráfica poco amigable. - Uso de fórmulas complejas |

Tabla 9: Características de Crystal Reports 7.0

Por último, pero no menos importante, tenemos la conclusión de todo el proceso, que es la etapa de Implementación.

2.4.4 Implementación

1.- Pruebas de aceptación: En esta etapa se llevan a cabo distintas pruebas orientadas a determinar en qué grado hemos llegado a satisfacer las necesidades de los usuarios finales, de tal forma que se tome en cuenta el punto de vista del mismo.

2.- Garantía de Calidad: Se conoce también como la prueba final o la prueba de aceptación. Requiere como entradas los datos de la prueba de aceptación generada por la actividad anterior.

3.- Descripción del Procedimiento: Este apartado da como resultado un manual de usuario, donde se describen los procesos que se llevan a cabo mediante el sistema resultante de este proceso, así como la forma correcta de manejo tanto de la información como del sistema.

4.- Conversión de la Base de datos: Aunque pudiera no existir una base de datos en el sistema actual, en el caso general, requiere como entrada la base de datos actual del usuario y la especificación del diseño producida por la actividad del mismo nombre.

5.- Instalación: En algunos casos no es más que comenzar a utilizar el sistema, pero en otro puede implicar el tiempo de recepción de los manuales de usuario, capacitación de los usuarios y la utilización del sistema.

Para poder desarrollar aplicaciones funcionales, es necesario entender todo el entorno de desarrollo y operación que éste necesitará, el cual, apoyado en una metodología como la de ciclo de vida, nos indica las actividades, funciones, productos terminados y herramientas en cada una de las etapas, disminuyendo el riesgo de fracasos y acercándonos a nuestro objetivo.

Capítulo III
**Análisis y diseño
del sistema**

3. Análisis y diseño del sistema

Apoyados en una metodología adecuada, que nos indique cuáles son las etapas y actividades a realizar, empezaremos el desarrollo del sistema con el análisis, siendo aparte de la etapa inicial, la más crítica, pues será la base de toda la recopilación de la información que dará forma a nuestro sistema.

3.1 Objetivo

Debido a todas las necesidades mencionadas en el capítulo 1, nuestro nuevo sistema tiene como objetivo el manejo más cómodo y seguro de la información, para la toma de decisiones, así como unificar los diferentes procesos que se manejan dentro de la Unidad de Extensión Universitaria, y presentarlo con una interfaz gráfica amigable, donde los usuarios manejarán su información de una manera fácil y segura.

3.2 Alcance

Este sistema y la forma en que será manejada nos brindará un beneficio en tiempo, así como en esfuerzo y comodidad a la hora del manejo de información, debido a la conexión en red de sus áreas y con esto se logrará una comunicación óptima.

El sistema brindará los siguientes beneficios:

- Recibir toda la información mucho más minuciosa de los eventos: dado que en la computadora se puede detallar al punto que uno o desee
- Unificarán los procesos: todos manejarán el mismo proceso, por medio de la computadora, facilitando su recopilación, por medio de eventos
- Manejarán los mismos formatos: todos por medio de eventos y pantallas fáciles para cada área

- Se mantendrá la comunicación con los Oficios: ya que se adjuntarán números de oficios para su mejor manejo
- Teniendo mayor control de dicha información: debido a que no hay pérdida y todo queda guardado en la base de datos
- Contará con un sistema de bitácora: para verificar quién entra al sistema y en qué momento
- Controlando los movimientos realizados al sistema: para que no exista un mal uso del mismo
- Ahorro de tiempo: al no tener que reescribir toda la información
- Evitar duplicidad de trabajo: no se repite la descripción de un mismo reporte
- Optimización de los reportes: más rápidos y sencillos
- Todos los datos se almacenen en un mismo lugar: una base de datos hospedada en la computadora principal de la dirección
- Se extraerán datos fácilmente: ya que el sistema permite obtener datos de cualquier evento almacenado
- Imprimirán reportes finales: el sistema permite imprimir diferentes partes de los reportes para su mejor uso
- Aprovechamiento de recursos existentes (cómputo): se utilizan sus computadoras, conectadas en red, y se optimiza todo el proceso

Dicho sistema se encargará de actualizar los procesos que hasta este momento la Unidad de Extensión Universitaria lleva manejando de tal modo que todas las actividades que realicen las diferentes áreas se transformen en eventos para así unificarlos y realizar un registro completo de todos los eventos actuales así como los pasados y guardados dentro de una base de datos para su futura utilización.

La interfaz contará con un registro de usuarios con su contraseña respectiva donde cada área podrá ingresar su información para que se acumule con la demás y no

afectara con las demás áreas, toda la información se guardará en la computadora de la dirección de la Unidad de Extensión Universitaria.

Aunque en nuestro capítulo 2 preparamos todo para utilizar esta interfaz en página Web en la actualidad los recursos nos permiten simplemente realizarlo en una plataforma de Visual Basic en combinación con Access.

3.3 Requerimientos de infraestructura

Para la implementación de este sistema necesitaremos que cada área cuente con un equipo de cómputo conectado en red y que como características mínimas de cada equipo deben ser:

- Procesador Pentium II a 500 MHz
- Velocidad en Ram de 128 MB
- Disco Duro con capacidad de 20 GB
- Tarjeta de Red
- Plataforma Windows 98
- Office 2000 completo

Sin embargo la computadora principal debe contar con al menos un 50% de capacidad que las características mínimas debido a que esta computadora va a hospedar la base de datos que concentrará toda la información de la Unidad de Extensión Universitaria.

Nos auxiliaremos de la conexión de red que distribuye la institución de la FES Aragón, para no interferir con nuevas redes, debido a que ellos manejan una red general y actualmente se encuentra en funcionamiento.

La Institución maneja una red inalámbrica para la dirección de Unidad de Extensión Universitaria, y algunos de sus departamentos cuenta con redes alámbricas pero su conexión es buena y funcional.

3.4 Diagrama de contexto

Es la representación de las diversas entidades que van a interactuar con el sistema, así como su relación con el mismo. Nos ayuda a identificar fácilmente las entidades.

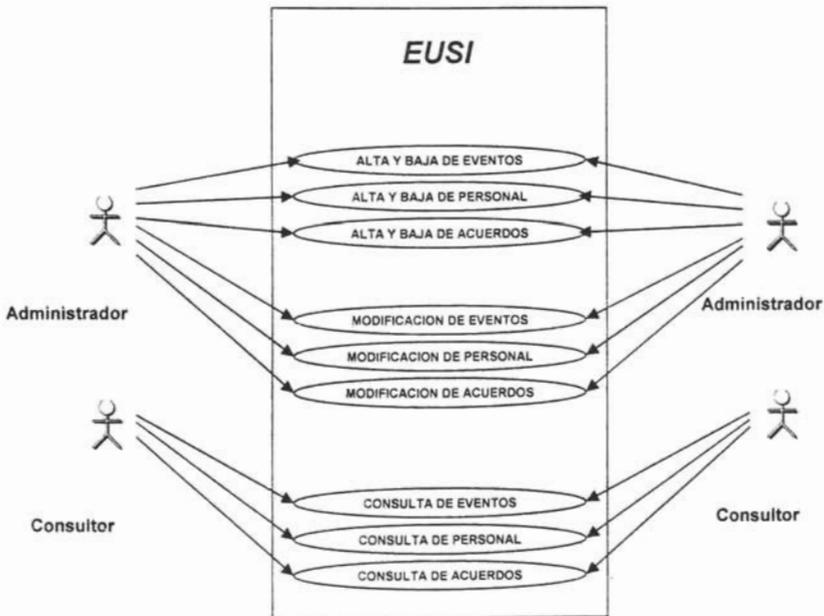


Figura 17: Diagrama de contexto

3.5 Análisis de procedimientos

Nos encontramos que cada una de las áreas maneja procesos diferentes por ejemplo.

El departamento de actividades deportivas:

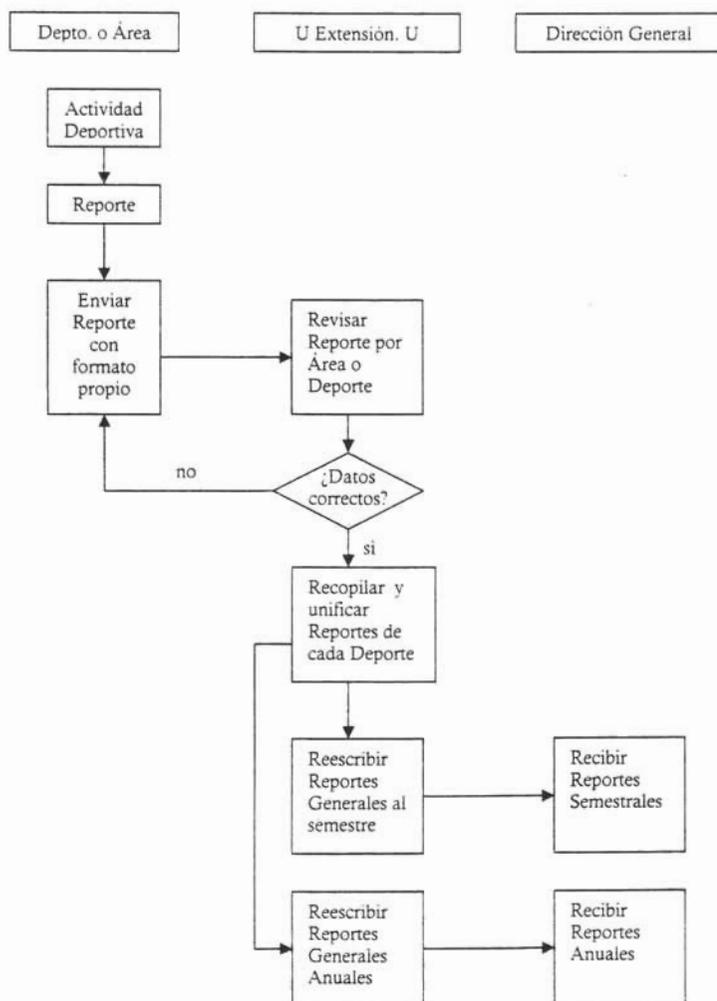


Figura 18: Diagrama de flujo del departamento de actividades deportivas

Así como este proceso cada una de las áreas cuenta con un proceso distinto pero que en su forma elemental son muy similares entre sí, esto ayudara a que nosotros manejeamos un proceso único optimizado.

3.6 Lista de procesos

1. Alta de Evento
2. Baja de Evento
3. Modificación de Evento
4. Consulta de Evento
5. Alta de Acuerdo y/o Convenio
6. Baja de Acuerdo y/o Convenio
7. Modificación Acuerdo y/o Convenio
8. Consulta Acuerdo y/o Convenio
9. Alta de Usuario
10. Baja de Usuario
11. Modificación de Usuario
12. Consulta de Usuario

3.7 Diagrama de casos de uso

Casos de uso es una técnica para capturar información de cómo un sistema o negocio trabaja, o de cómo se desea que trabaje. No pertenece estrictamente al enfoque orientado a objeto, es una técnica para captura de requisitos.

Casos de uso

Los casos de uso (Ivan Jacobson) describen bajo la forma de acciones y reacciones el comportamiento de un sistema desde el punto de vista del usuario. Permiten definir los límites del sistema y las relaciones entre el sistema y el entorno, son descripciones de la funcionalidad del sistema independientes de la implementación.

Cubren la carencia existente de métodos previos (OTM, Booch) en cuanto a la determinación de requisitos, particionan el conjunto de necesidades atendiendo a la categoría de los usuarios que participan en el mismo, están basados en el lenguaje natural, es decir, es accesible por los usuarios.

Actores:

- Principales: personas que usan el sistema
- Secundarios : personas que mantienen o administran el sistema
- Material externo: dispositivos, materiales imprescindibles que forman parte del ámbito de la aplicación y deben ser utilizados
- Otros sistemas: sistemas con los que el sistema interactúa.

La misma persona física puede interpretar varios papeles como actores distintos, el nombre del actor describe el papel desempeñado.

3.7.1 Actores

| Actor | Descripción |
|-----------------------|--|
| Administrador General | Encargado de dar altas, bajas, modificaciones, consultas e impresiones de todos los eventos de la Unidad de Extensión Universitaria. Encargado de la administración de los usuarios (Altas, Bajas, Consultas y Modificaciones). |
| Administrador Local | Encargado de dar altas, bajas, modificaciones, consultas e impresiones de todos los eventos de cada departamento de extensión universitaria. |
| Consultor General | Encargado de Consultar los eventos e impresiones para registro en papel de todos los departamentos. |
| Consultor Local | Encargado de Consultar los eventos e impresiones para registro en papel de cada departamento. |

Tabla 10: Actores que intervienen en el sistema EUSI y su descripción.

3.7.2 Casos de Uso

| Caso de Uso | Descripción |
|-----------------------------------|---|
| Alta de Evento | Mediante este caso de uso, los actores pueden ingresar un nuevo evento a la base de datos. |
| Baja de Evento | Mediante este caso de uso, los actores pueden eliminar un evento de la base de datos. |
| Modificación de Evento | Mediante este caso de uso, los actores pueden modificar un evento de la base de datos. |
| Consulta de Evento | Mediante este caso de uso, los actores pueden consultar e imprimir un evento de la base de datos. |
| Alta de Acuerdo y/o Convenio | Mediante este caso de uso, los actores pueden ingresar un nuevo Acuerdo y/o Convenio a la base de datos. |
| Baja de Acuerdo y/o Convenio | Mediante este caso de uso, los actores pueden eliminar un Acuerdo y/o Convenio de la base de datos. |
| Modificación Acuerdo y/o Convenio | Mediante este caso de uso, los actores pueden modificar un Acuerdo y/o Convenio de la base de datos. |
| Consulta Acuerdo y/o Convenio | Mediante este caso de uso, los actores pueden consultar e imprimir un Acuerdo y/o Convenio de la base de datos. |
| Alta de Usuario | Mediante este caso de uso, los actores pueden ingresar un nuevo Usuario a la base de datos. |
| Baja de Usuario | Mediante este caso de uso, los actores pueden eliminar un Usuario de la base de datos. |
| Modificación de Usuario | Mediante este caso de uso, los actores pueden modificar un Usuario de la base de datos. |
| Consulta de Usuario | Mediante este caso de uso, los actores pueden consultar e imprimir un Usuario de la base de datos. |

Tabla 11: Casos de Uso que intervienen en el sistema EUSI y su descripción.

3.7.3 Descripción de Casos de Uso.

| | |
|--------------|--|
| Caso de uso: | Alta de evento |
| Actores: | Administrador General y Local |
| Propósito: | Ingresar un nuevo evento |
| Tipo: | Primario |
| Descripción: | Mediante este caso de uso, los actores pueden ingresar un nuevo evento a la base de datos. |

| Curso de Eventos: | | | |
|--------------------------|--|----|---|
| No | Acción del actor | No | Respuesta del sistema |
| 1 | Ingresar Login | 2 | El sistema valida el login ingresado para poder mostrar el formulario. |
| 3 | Selecciona el departamento al que quiere ingresar | 4 | El sistema muestra la lista de tareas que se pueden realizar. |
| 5 | El administrador elige la opción Alta de Evento | 6 | El sistema muestra una forma en donde solicita uno o más datos para ingresar un evento. |
| 7 | El administrador ingresa los datos del evento. | 8 | El sistema valida los datos ingresados y verifica que sean los necesarios para dar de alta el evento y pregunta si se desea guardar o no los datos. |
| 9 | El administrador acepta o rechaza la petición del sistema. | 10 | El sistema ingresa los datos en la base. |

Tabla 12: Caso de Uso de alta de un evento y su curso de eventos

| | |
|--------------|---|
| Caso de uso: | Baja de evento |
| Actores: | Administrador General y Local |
| Propósito: | Eliminar un evento existente |
| Tipo: | Primario |
| Descripción: | Mediante este caso de uso, los actores pueden eliminar un evento de la base de datos. |

| Curso de Eventos: | | | |
|--------------------------|--|-----------|---|
| No | Acción del actor | No | Respuesta del sistema |
| 1 | Ingresa Login | 2 | El sistema valida el login ingresado para poder mostrar el formulario. |
| 3 | Selecciona el departamento al que quiere ingresar | 4 | El sistema muestra la lista de tareas que se pueden realizar. |
| 5 | El administrador elige la opción Baja de Evento | 6 | El sistema muestra una forma en donde se indica que evento se quiere eliminar. |
| 7 | El administrador ingresa los datos del evento. | 8 | El sistema valida los datos ingresados y verifica cual evento coincide con lo ingresado, lo muestra al usuario y pregunta si se desea eliminar. |
| 9 | El administrador acepta o rechaza la petición del sistema. | 10 | El sistema elimina los datos de la base. |

Tabla 13: Caso de Uso de baja de un evento y su curso de eventos

| | |
|--------------|---|
| Caso de uso: | Modificación de evento |
| Actores: | Administrador General y Local |
| Propósito: | Modificar los datos de un evento existente |
| Tipo: | Secundario |
| Descripción: | Mediante este caso de uso, los actores pueden modificar los datos de un evento de la base de datos. |

| Curso de Eventos: | | | |
|--------------------------|--|----|---|
| No | Acción del actor | No | Respuesta del sistema |
| 1 | Ingresar Login | 2 | El sistema valida el login ingresado para poder mostrar el formulario. |
| 3 | Selecciona el departamento al que quiere ingresar | 4 | El sistema muestra la lista de tareas que se pueden realizar. |
| 5 | El administrador elige la opción Modificación de Evento | 6 | El sistema muestra una forma en donde se indica que evento se quiere eliminar. |
| 7 | El administrador ingresa los datos del evento que desea modificar. | 8 | El sistema valida los datos y verifica cual evento coincide con lo ingresado, lo muestra al usuario y pregunta si se desea modificar el registro. |
| 9 | El administrador acepta o rechaza la petición del sistema. | 10 | El sistema abre los datos de la base y permite su modificación. |
| 11 | El administrador modifica los datos que sean necesarios. | 12 | El sistema pregunta al usuario si desea guardar los cambios realizados. |
| 13 | El administrador acepta o rechaza la petición del sistema. | 14 | El sistema guarda los datos nuevos en el evento seleccionado de la base de datos. |

Tabla 14: Caso de Uso de modificación de un evento y su curso de eventos

| | |
|--------------|--|
| Caso de uso: | Consulta de evento |
| Actores: | Consultor General y Local |
| Propósito: | Consultar un evento existente, así como sus registros relacionados |
| Tipo: | Secundario |
| Descripción: | Mediante este caso de uso, los actores pueden consultar un evento de la base de datos sin modificarlo. |

| Curso de Eventos: | | | |
|--------------------------|--|-----------|---|
| No | Acción del actor | No | Respuesta del sistema |
| 1 | Ingresa Login | 2 | El sistema valida el login ingresado para poder mostrar el formulario. |
| 3 | Selecciona el departamento al que quiere ingresar | 4 | El sistema muestra la lista de tareas que se pueden realizar. |
| 5 | El consultor elige la opción consultar Evento | 6 | El sistema muestra una forma en donde se indica que evento se quiere consultar. |
| 7 | El consultor ingresa los datos del evento. | 8 | El sistema valida los datos ingresados y verifica cual evento coincide con lo ingresado, lo muestra al usuario y pregunta si se desea imprimir. |
| 9 | El consultor acepta o rechaza la petición del sistema. | 10 | El sistema envía los datos de la base a la impresora y pregunta al consultor si desea realizar otra consulta. |
| 11 | El consultor acepta o rechaza la petición del sistema. | 12 | En caso de ser afirmativa la respuesta de el consultor. el sistema regresa al paso seis, si es negativa regresa al formulario principal. |

Tabla 15: Caso de Uso de consulta de un evento y su curso de eventos

| | |
|--------------|--|
| Caso de uso: | Alta de Acuerdo y/o Convenio |
| Actores: | Administrador General y Local |
| Propósito: | Ingresar un nuevo Acuerdo y/o Convenio |
| Tipo: | Primario |
| Descripción: | Mediante este caso de uso, los actores pueden ingresar un nuevo Acuerdo y/o Convenio a la base de datos. |

| Curso de Eventos: | | | |
|--------------------------|---|-----------|---|
| No | Acción del actor | No | Respuesta del sistema |
| 1 | Ingresar Login | 2 | El sistema valida el login ingresado para poder mostrar el formulario. |
| 3 | Selecciona el departamento al que quiere ingresar | 4 | El sistema muestra la lista de tareas que se pueden realizar. |
| 5 | El administrador elige la opción Alta de Acuerdo y/o Convenio | 6 | El sistema muestra una forma en donde solicita uno o más datos para ingresar un Acuerdo y/o Convenio. |
| 7 | El administrador ingresa los datos del Acuerdo y/o Convenio. | 8 | El sistema valida los datos ingresados y verifica que sean los necesarios para dar de alta el Acuerdo y/o Convenio y pregunta si se desea guardar o no los datos. |
| 9 | El administrador acepta o rechaza la petición del sistema. | 10 | El sistema ingresa los datos en la base. |

Tabla 16: Caso de Uso de alta de acuerdo y/o convenio y su curso de eventos

| | |
|--------------|---|
| Caso de uso: | Baja de Acuerdo y/o Convenio |
| Actores: | Administrador General y Local |
| Propósito: | Eliminar un Acuerdo y/o Convenio existente |
| Tipo: | Primario |
| Descripción: | Mediante este caso de uso, los actores pueden eliminar un Acuerdo y/o Convenio de la base de datos. |

| Curso de Eventos: | | | |
|--------------------------|---|----|---|
| No | Acción del actor | No | Respuesta del sistema |
| 1 | Ingresar Login | 2 | El sistema valida el login ingresado para poder mostrar el formulario. |
| 3 | Selecciona el departamento al que quiere ingresar | 4 | El sistema muestra la lista de tareas que se pueden realizar. |
| 5 | El administrador elige la opción Baja de Acuerdo y/o Convenio | 6 | El sistema muestra una forma en donde se indica que Acuerdo y/o Convenio se quiere eliminar. |
| 7 | El administrador ingresa los datos del Acuerdo y/o Convenio. | 8 | El sistema valida los datos ingresados y verifica cual Acuerdo y/o Convenio coincide con lo ingresado, lo muestra al administrador y pregunta si se desea eliminar. |
| 9 | El administrador acepta o rechaza la petición del sistema. | 10 | El sistema elimina los datos de la base. |

Tabla 17: Caso de Uso de baja de acuerdo y/o convenio y su curso de eventos

| | |
|--------------|--|
| Caso de uso: | Modificación de Acuerdo y/o Convenio |
| Actores: | Administrador General y Local |
| Propósito: | Modificar los datos de un Acuerdo y/o Convenio existente |
| Tipo: | Secundario |
| Descripción: | Mediante este caso de uso, los actores pueden modificar un Acuerdo y/o Convenio de la base de datos. |

| Curso de Eventos: | | | |
|--------------------------|--|----|---|
| No | Acción del actor | No | Respuesta del sistema |
| 1 | Ingresar Login | 2 | El sistema valida el login ingresado para poder mostrar el formulario. |
| 3 | Selecciona el departamento al que quiere ingresar | 4 | El sistema muestra la lista de tareas que se pueden realizar. |
| 5 | El administrador elige la opción Modificación de Acuerdo y/o Convenio | 6 | El sistema muestra una forma en donde se indica que evento se quiere eliminar. |
| 7 | El administrador ingresa los datos del Acuerdo y/o Convenio que desea modificar. | 8 | El sistema valida los datos y verifica cual Acuerdo y/o Convenio coincide con lo ingresado, lo muestra al administrador y pregunta si se desea modificar el registro. |
| 9 | El administrador acepta o rechaza la petición del sistema. | 10 | El sistema abre los datos de la base y permite su modificación. |
| 11 | El administrador modifica los datos que sean necesarios. | 12 | El sistema pregunta al administrador se desea guardar los cambios realizados. |
| 13 | El administrador acepta o rechaza la petición del sistema. | 14 | El sistema guarda los datos nuevos en el Acuerdo y/o Convenio seleccionado de la base de datos. |

Tabla 18: Caso de Uso de modificación de acuerdo y/o convenio y su curso de eventos

| | |
|--------------|---|
| Caso de uso: | Consulta de Acuerdo y/o Convenio |
| Actores: | Consultor General y Local |
| Propósito: | Consultar un Acuerdo y/o Convenio existente, así como sus registros relacionados |
| Tipo: | Secundario |
| Descripción: | Mediante este caso de uso, los actores pueden consultar e imprimir un Acuerdo y/o Convenio de la base de datos. |

| Curso de Eventos: | | | |
|--------------------------|---|----|---|
| No | Acción del actor | No | Respuesta del sistema |
| 1 | Ingresar Login | 2 | El sistema valida el login ingresado para poder mostrar el formulario. |
| 3 | Selecciona el departamento al que quiere ingresar | 4 | El sistema muestra la lista de tareas que se pueden realizar. |
| 5 | El consultor elige la opción consultar Acuerdo y/o Convenio | 6 | El sistema muestra una forma en donde se indica que Acuerdo y/o Convenio se quiere consultar. |
| 7 | El consultor ingresa los datos del Acuerdo y/o Convenio. | 8 | El sistema valida los datos ingresados y verifica cual Acuerdo y/o Convenio coincide con lo ingresado, lo muestra al consultor y pregunta si se desea imprimir. |
| 9 | El consultor acepta o rechaza la petición del sistema. | 10 | El sistema envía los datos de la base a la impresora y pregunta al consultor si desea realizar otra consulta. |
| 11 | El consultor acepta o rechaza la petición del sistema. | 12 | En caso de ser afirmativa la respuesta de el consultor, el sistema regresa al paso seis. si es negativa regresa al formulario principal. |

Tabla 19: Caso de Uso de consulta de acuerdo y/o convenio y su curso de eventos

| | |
|--------------|---|
| Caso de uso: | Alta de Usuario |
| Actores: | Administrador General y Local |
| Propósito: | Ingresar un nuevo Usuario |
| Tipo: | Primario |
| Descripción: | Mediante este caso de uso, los actores pueden ingresar un nuevo Usuario a la base de datos. |

| Curso de Eventos: | | | |
|--------------------------|--|-----------|--|
| No | Acción del actor | No | Respuesta del sistema |
| 1 | Ingresar Login | 2 | El sistema valida el login ingresado para poder mostrar el formulario. |
| 3 | Selecciona el departamento al que quiere ingresar | 4 | El sistema muestra la lista de tareas que se pueden realizar. |
| 5 | El administrador elige la opción Alta de Usuarios | 6 | El sistema muestra una forma en donde solicita uno o más datos para ingresar un Usuario. |
| 7 | El administrador ingresa los datos del Acuerdo y/o Convenio. | 8 | El sistema valida los datos ingresados y verifica que sean los necesarios para dar de alta el Usuario y pregunta si se desea guardar o no los datos. |
| 9 | El administrador acepta o rechaza la petición del sistema. | 10 | El sistema ingresa los datos en la base. |

Tabla 20: Caso de Uso de alta de usuario y su curso de eventos

| | |
|--------------|--|
| Caso de uso: | Baja de Usuario |
| Actores: | Administrador General y Local |
| Propósito: | Eliminar un Usuario existente |
| Tipo: | Primario |
| Descripción: | Mediante este caso de uso, los actores pueden eliminar un Usuario de la base de datos. |

| Curso de Eventos: | | | |
|--------------------------|--|----|--|
| No | Acción del actor | No | Respuesta del sistema |
| 1 | Ingresar Login | 2 | El sistema valida el login ingresado para poder mostrar el formulario. |
| 3 | Selecciona el departamento al que quiere ingresar | 4 | El sistema muestra la lista de tareas que se pueden realizar. |
| 5 | El administrador elige la opción Baja de Usuarios. | 6 | El sistema muestra una forma en donde se indica que Usuario se quiere eliminar. |
| 7 | El administrador ingresa los datos del Usuario. | 8 | El sistema valida los datos ingresados y verifica cual Usuario coincide con lo ingresado, lo muestra al administrador y pregunta si se desea eliminar. |
| 9 | El administrador acepta o rechaza la petición del sistema. | 10 | El sistema elimina los datos de la base. |

Tabla 21: Caso de Uso de baja de usuario y su curso de eventos

| | |
|--------------|---|
| Caso de uso: | Modificación de Usuario |
| Actores: | Administrador General y Local |
| Propósito: | Modificar los datos de un Usuario existente |
| Tipo: | Secundario |
| Descripción: | Mediante este caso de uso, los actores pueden modificar un Usuario de la base de datos. |

| Curso de Eventos: | | | |
|--------------------------|---|-----------|--|
| No | Acción del actor | No | Respuesta del sistema |
| 1 | Ingresar Login | 2 | El sistema valida el login ingresado para poder mostrar el formulario. |
| 3 | Selecciona el departamento al que quiere ingresar | 4 | El sistema muestra la lista de tareas que se pueden realizar. |
| 5 | El administrador elige la opción Modificación de Usuarios | 6 | El sistema muestra una forma en donde se indica que Usuario se quiere eliminar. |
| 7 | El administrador ingresa los datos del Usuario que desea modificar. | 8 | El sistema valida los datos y verifica cual Usuario coincide con lo ingresado, lo muestra al administrador y pregunta si se desea modificar el registro. |
| 9 | El administrador acepta o rechaza la petición del sistema. | 10 | El sistema abre los datos de la base y permite su modificación. |
| 11 | El administrador modifica los datos que sean necesarios. | 12 | El sistema pregunta al administrador si desea guardar los cambios realizados. |
| 13 | El administrador acepta o rechaza la petición del sistema. | 14 | El sistema guarda los datos nuevos en el Usuario seleccionado de la base de datos. |

Tabla 22: Caso de Uso de modificación de usuario y su curso de eventos

| | |
|--------------|--|
| Caso de uso: | Consulta de Usuario |
| Actores: | Consultor General y Local |
| Propósito: | Consultar un Usuario existente, así como sus registros relacionados |
| Tipo: | Secundario |
| Descripción: | Mediante este caso de uso, los actores pueden consultar e imprimir un Usuario de la base de datos. |

| Curso de Eventos: | | | |
|--------------------------|--|-----------|--|
| No | Acción del actor | No | Respuesta del sistema |
| 1 | Ingresa Login | 2 | El sistema valida el login ingresado para poder mostrar el formulario. |
| 3 | Selecciona el departamento al que quiere ingresar | 4 | El sistema muestra la lista de tareas que se pueden realizar. |
| 5 | El consultor elige la opción consultar Usuario | 6 | El sistema muestra una forma en donde se indica que Usuario se quiere consultar. |
| 7 | El consultor ingresa los datos del Usuario. | 8 | El sistema valida los datos ingresados y verifica cual Usuario coincide con lo ingresado. lo muestra al consultor y pregunta si se desea imprimir. |
| 9 | El consultor acepta o rechaza la petición del sistema. | 10 | El sistema envía los datos de la base a la impresora y pregunta al consultor si desea realizar otra consulta. |
| 11 | El consultor acepta o rechaza la petición del sistema. | 12 | En caso de ser afirmativa la respuesta de el consultor, el sistema regresa al paso seis, si es negativa regresa al formulario principal. |

Tabla 23: Caso de Uso de consulta de usuario y su curso de eventos

3.8 Diagramas de Secuencia

Es una representación gráfica de los pasos que se tienen que seguir para realizar un proceso, considerando las actividades que tiene que realizar el usuario en los elementos/dispositivos así como la respuesta que estos emiten. A continuación se muestran los diagramas de secuencia de los siguientes procesos:

- Alta de Evento
- Baja de Evento
- Modificación de Evento
- Consulta de Evento
- Alta de Acuerdo y/o Convenio
- Baja de Acuerdo y/o Convenio
- Modificación Acuerdo y/o Convenio
- Consulta Acuerdo y/o Convenio
- Alta de Usuario
- Baja de Usuario
- Modificación de Usuario
- Consulta de Usuario

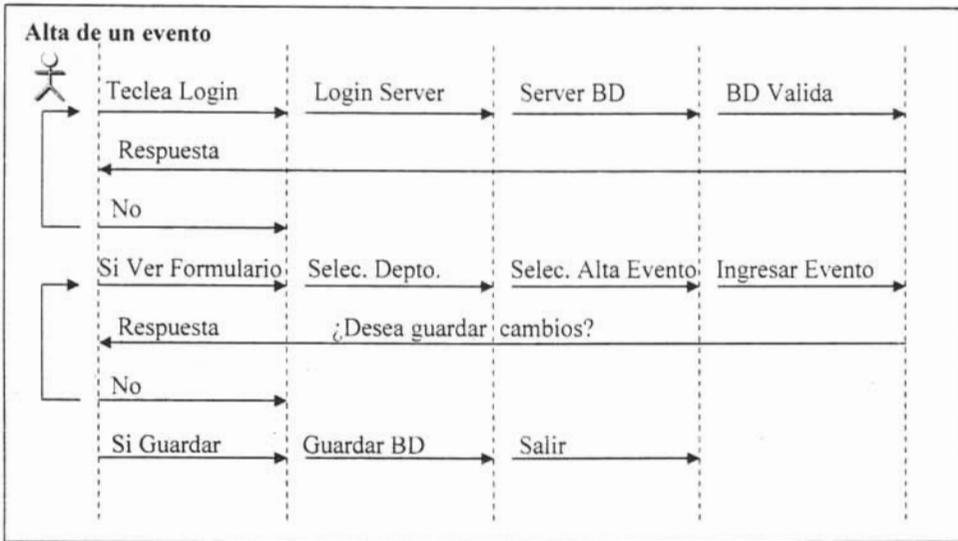


Tabla 24: Diagrama de secuencia de alta de un evento

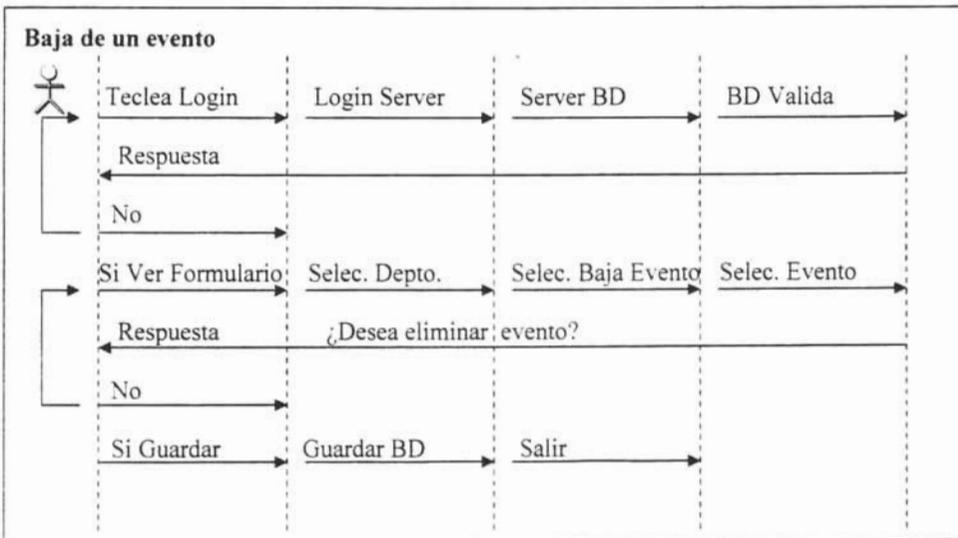


Tabla 25: Diagrama de secuencia de baja de un evento

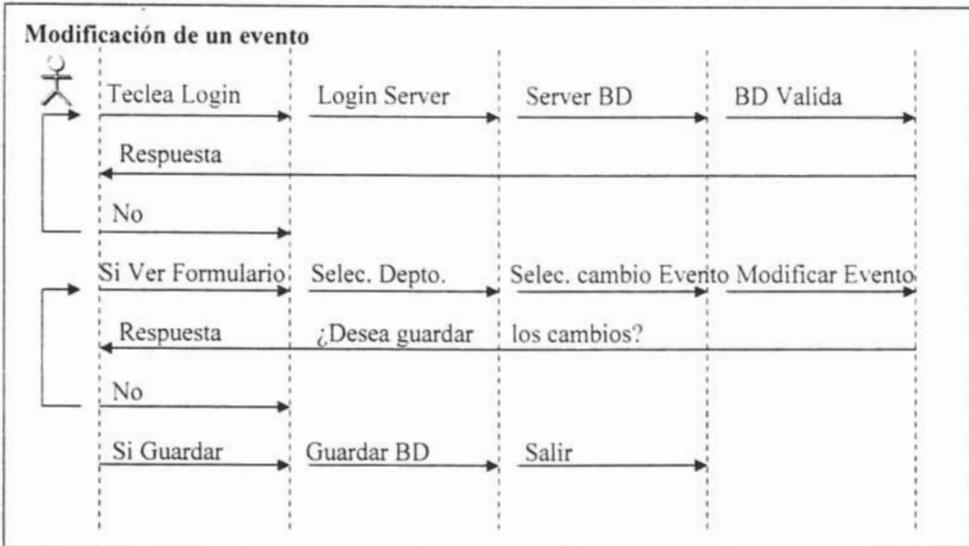


Tabla 26: Diagrama de secuencia de modificación de un evento

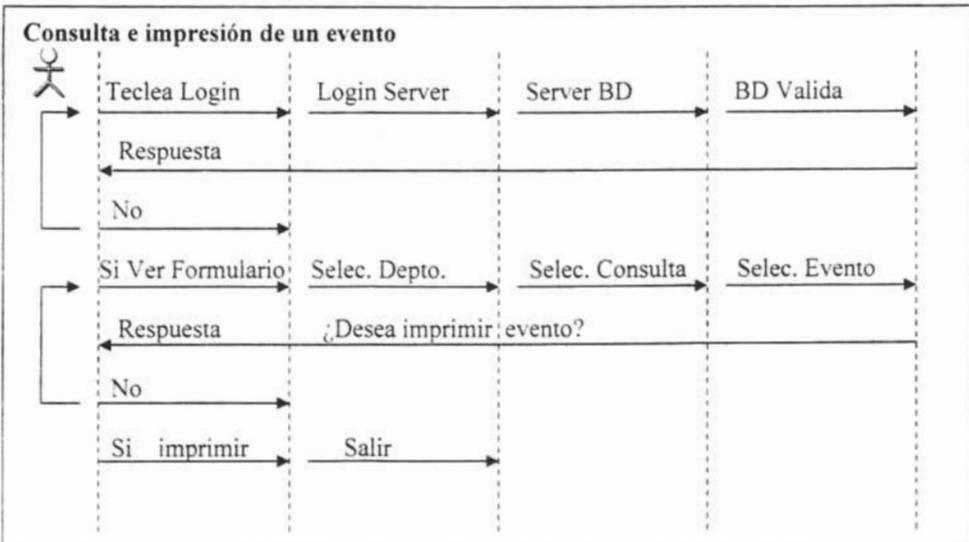


Tabla 27: Diagrama de secuencia de consulta de un evento

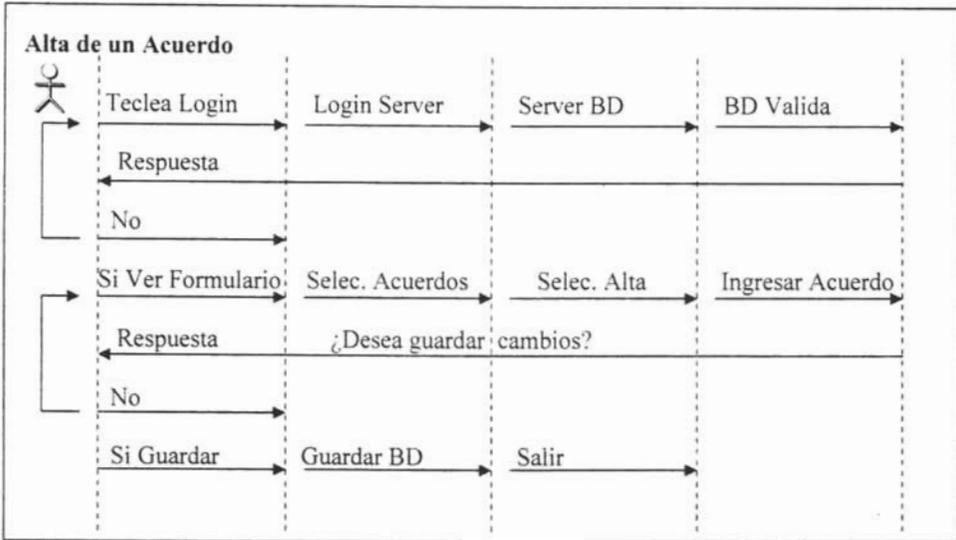


Tabla 28: Diagrama de secuencia de alta de un acuerdo

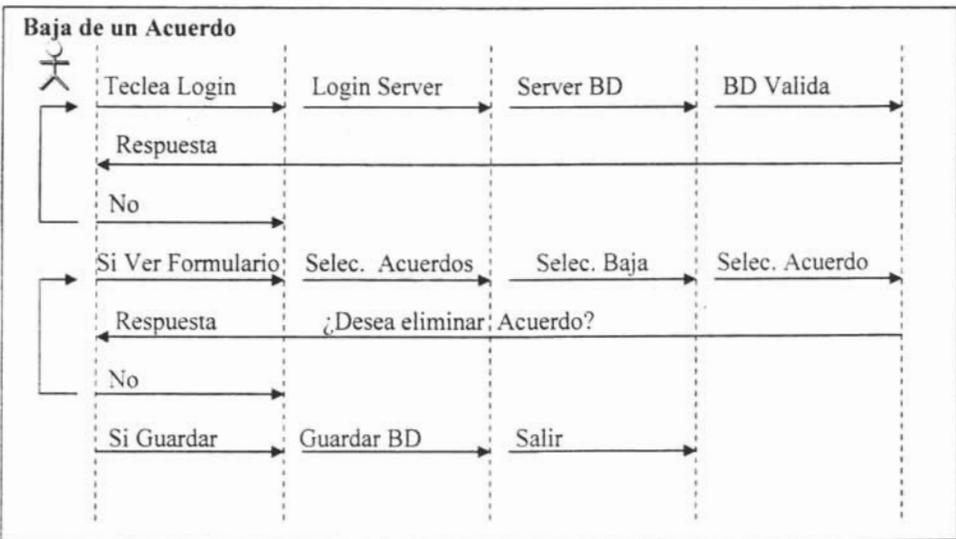


Tabla 29: Diagrama de secuencia de baja de un acuerdo

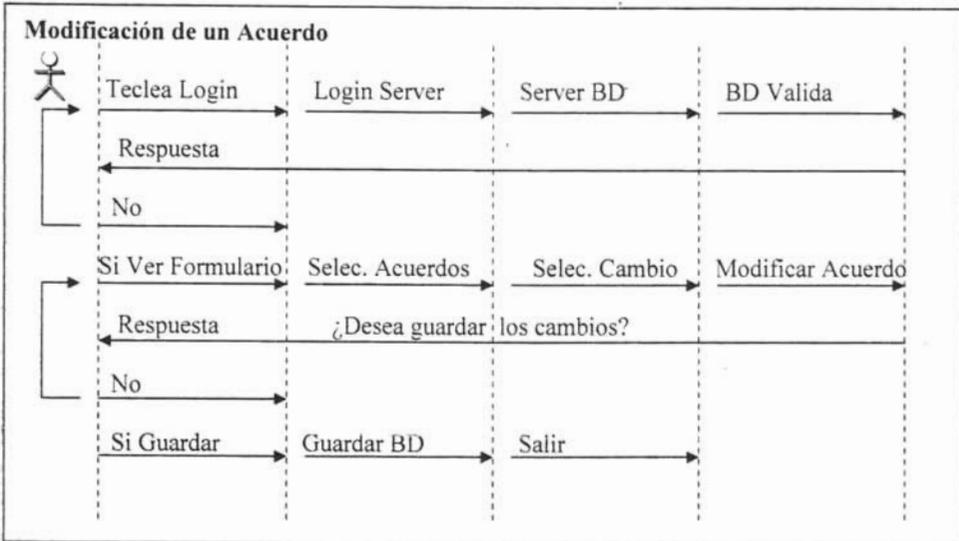


Tabla 30: Diagrama de secuencia de modificación de un acuerdo

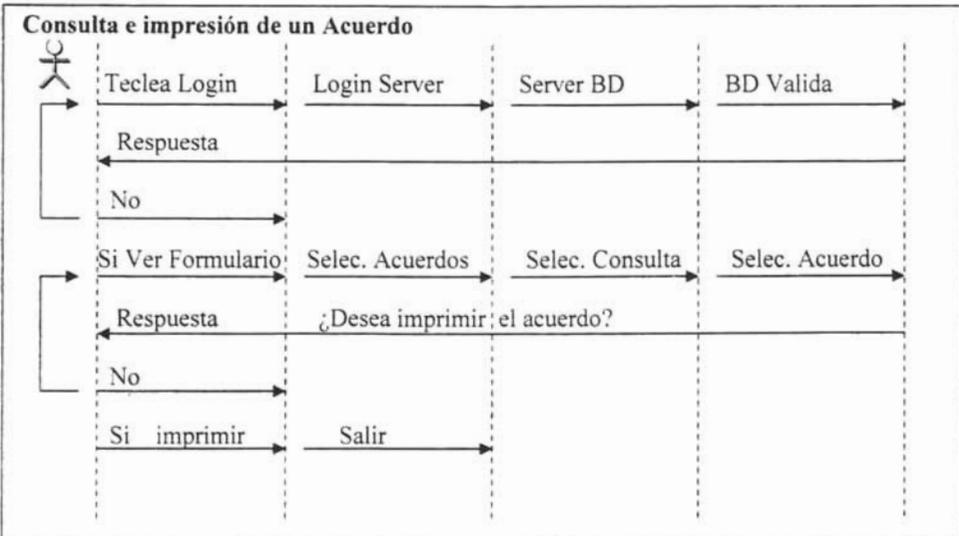


Tabla 31: Diagrama de secuencia de consulta e impresión de un acuerdo

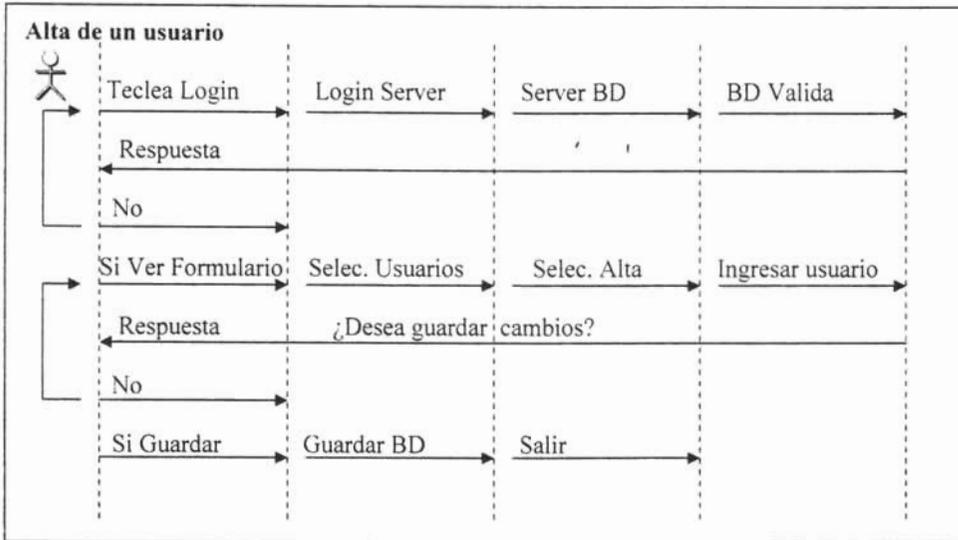


Tabla 32: Diagrama de secuencia de alta de un usuario

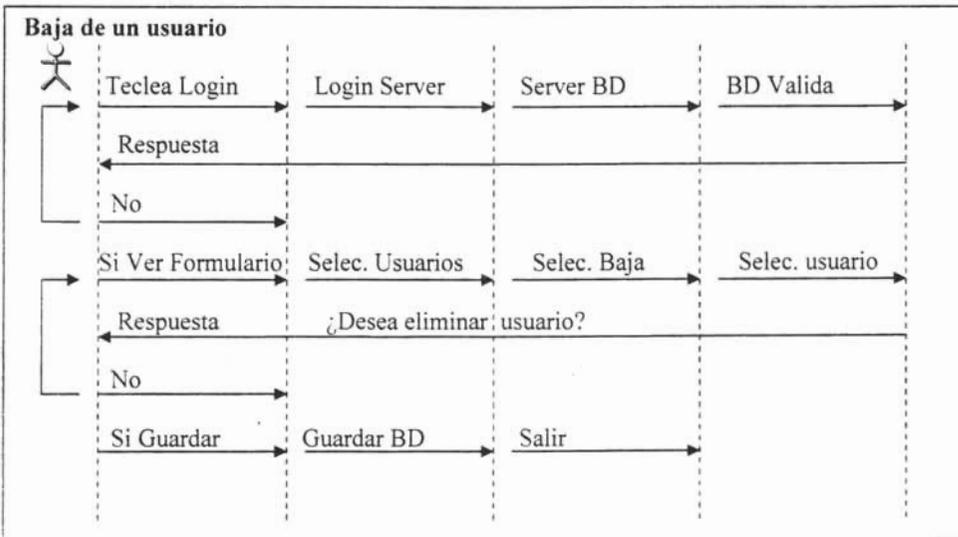


Tabla 33: Diagrama de secuencia de baja de un usuario

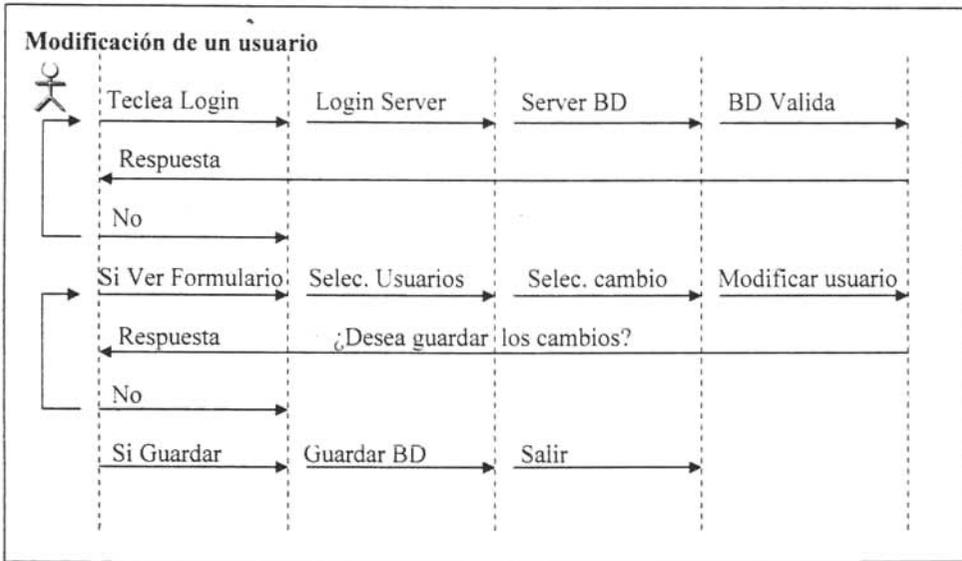


Tabla 34: Diagrama de secuencia de modificación de un usuario

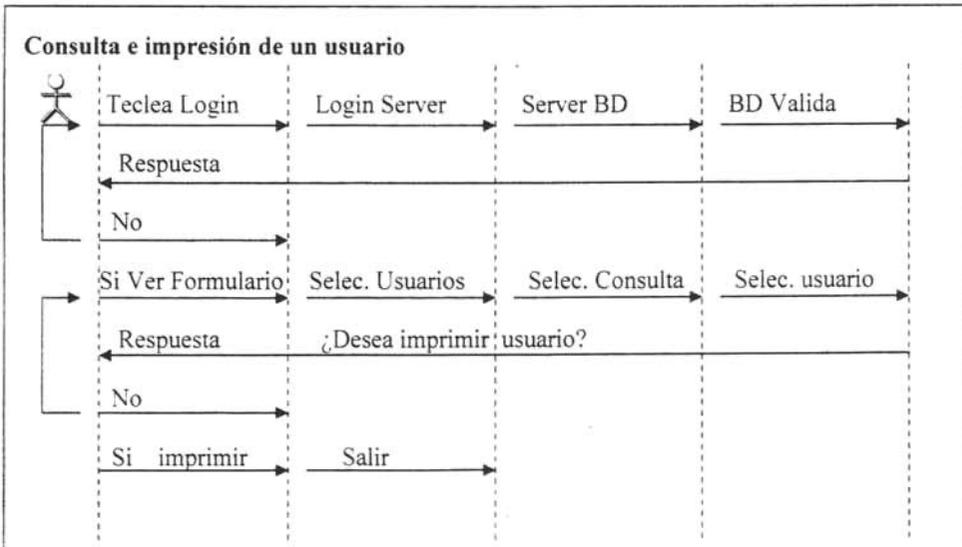


Tabla 35: Diagrama de secuencia de consulta e impresión de un usuario

3.9 Diseño de Interfaz (Prototipo)

Principios del diseño.

Algunos aspectos fundamentales que se deben procurar en el diseño son:

- **Diseño amigable:** toda interfaz debe de ser amigable, ya que el usuario la visualiza como un elemento de trabajo, aunque la funcionalidad del sistema lo puede hacer olvidar lo vistoso de las páginas normales, siendo fundamentales el diseño y la forma
- **Congruencia con los procesos:** el sistema debe estar estrictamente apegado a los procesos establecidos en el análisis, siendo fundamental su funcionalidad y organización, una interfaz amigable no sirve de nada si los procesos no son los necesarios o adecuados.
- **Distribución lógica:** los elementos de la interfaz como formularios, botones, listas, tablas, celdas, imágenes, mensajes, etc. deben tener una distribución lógica para lograr la funcionalidad y entendimiento.
- **Navegación lógica:** el orden y la secuencia en que se navegue entre los elementos de la interfaz y sus pantallas deben ser lógicos y adecuados según los procesos y perfiles del usuario.
- **Retroalimentación:** se refiere a la información o respuesta que da el sistema a través de los mensajes, a las acciones o procesos que realiza el usuario, se debe recordar que en ocasiones las transmisiones pueden ser lentas, por lo que el usuario puede pensar que algo está mal, por lo que es conveniente el uso de mensajes para indicarle que se está procesando o cargando información y que no es un problema o error de conexión.
- **Consistencia:** la uniformidad de la distribución, navegación, diseño, entradas y salidas, uso de colores, formato de mensajes, etc. de todas las pantallas debe

conservarse siempre, pues el usuario se adaptará rápidamente al uso del sistema con un comportamiento homogéneo de éste.

- **Implementación de ayuda:** en ocasiones es necesaria la implementación de módulos de ayuda, esto se decide a través del análisis de perfiles de usuario.

Sistema de Informacion



Figura 19: Pantalla de presentación

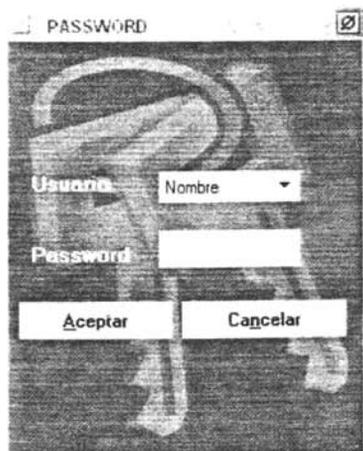


Figura 20: Pantalla de password



Figura 21: Menú principal

Personal [?] [0]

| | | |
|-----------|--------------------|----------|
| | id | 001 |
| Alta | Nombre | INVITADO |
| | Apellido_paterno | INVITADO |
| | Apellido_materno | INVITADO |
| Modificar | Cargo | INVITADO |
| Consultas | Departamento | INVITADO |
| Reportes | Password | ***** |
| | Confirmar Password | |

[H] [Personal] [H]

Figura 22: Formulario del personal



Figura 23: Menú de departamentos

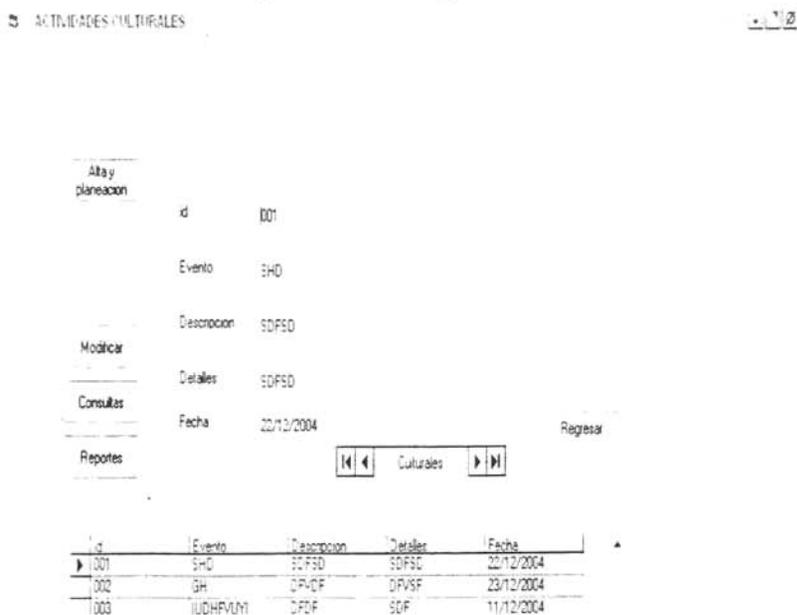


Figura 24: Menú general para departamentos

3.10 Mapa de navegación

Una parte importante del diseño de la interfaz corresponde al equilibrio entre la estructura y las relaciones entre los formularios y sus procesos. El objetivo es construir una jerarquía de menús y formularios que parezcan naturales y bien estructurados al usuario.

Los modelos tradicionales son: lineales, planos, jerárquicos, vertical y vertical con opciones, sin embargo en sistemas como este la estructura nos la dan los procesos y generalmente es del tipo vertical con opciones

La estructura de los formularios y menús debe ser clara funcional y con una continuidad gráfica entre los diferentes componentes y subsecciones del sistema, de manera que la relación entre componentes sea ordenada.

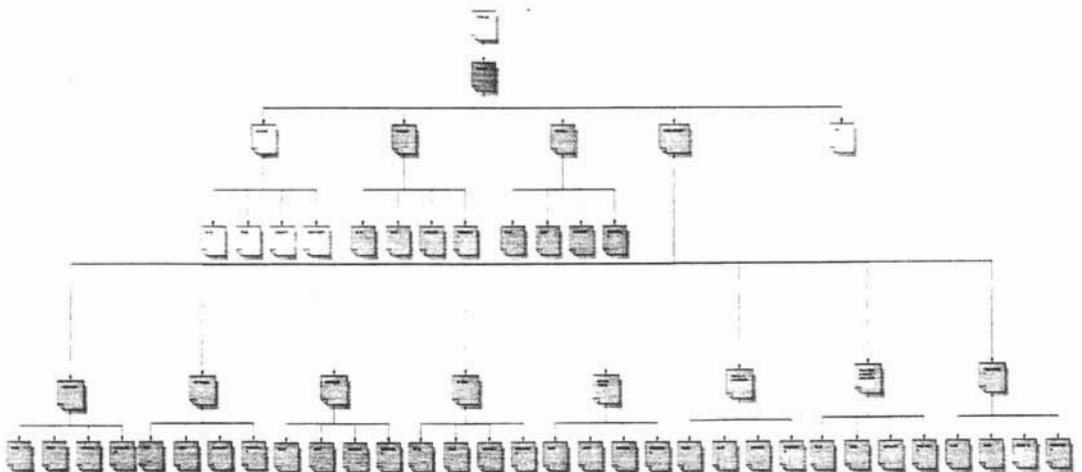


Figura 25: Mapa llave

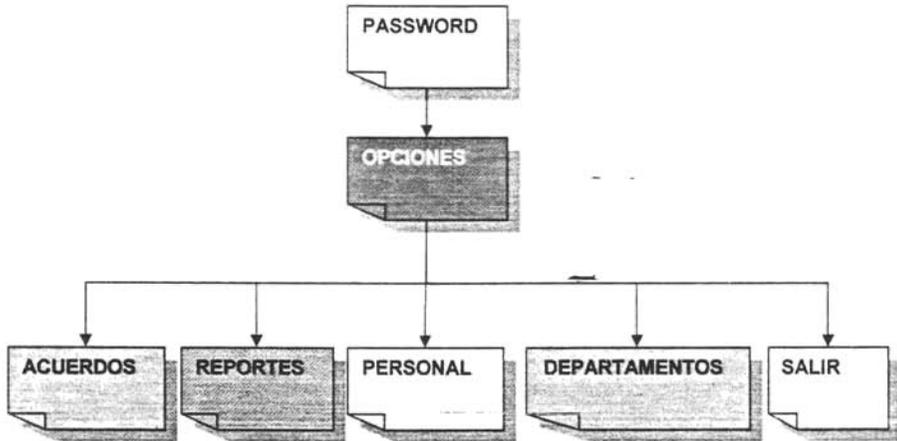


Figura 26: Mapa subsecuente

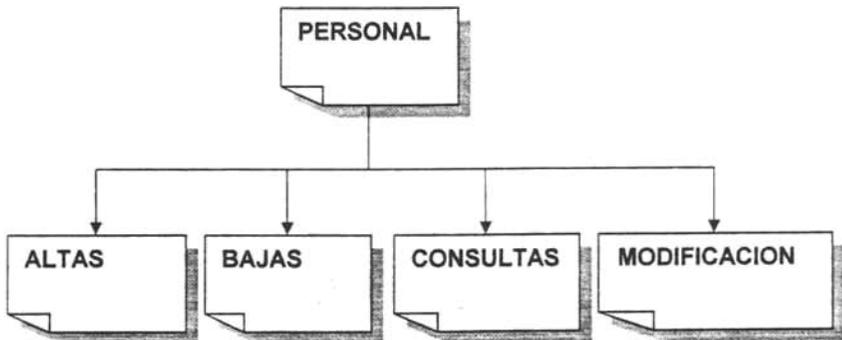


Figura 27: Mapa del personal



Figura 28: Mapa de acuerdos



Figura 29: Mapa de reportes

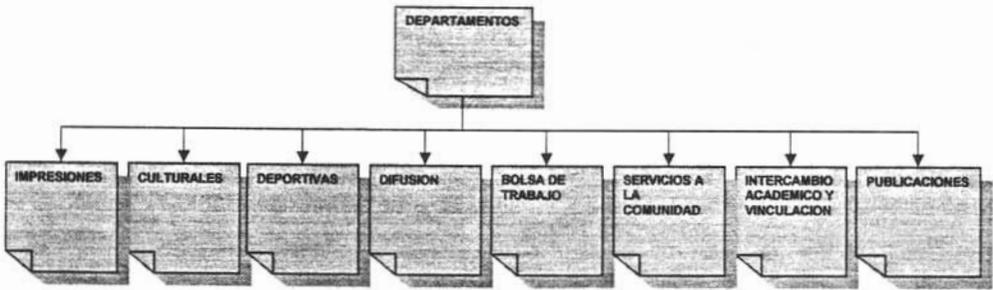


Figura 30: Mapa de departamentos

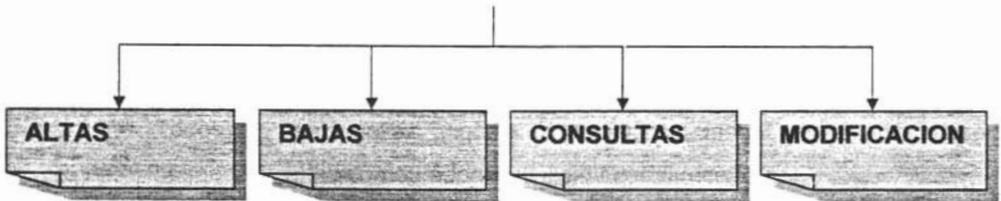


Figura 31: Opciones de los departamentos

3.11 Diagrama Entidad-Relación

Muestra la relación que existe entre agregados de datos. Constan de dos componentes principales:

Tipos de Objetos: Se representan por medio de un rectángulo en el diagrama. Esto representa una colección o conjunto de objetos (cosas) del mundo real cuyos miembros juegan algún papel en el desarrollo del sistema; pueden además ser identificados de manera única y ser descritos por uno o más atributos.

Relaciones. Se representan por medio de líneas terminadas por rombos en el diagrama y son la serie de conexiones o asociaciones entre tipo de objetos que están conectados con la relación por medio de flechas.

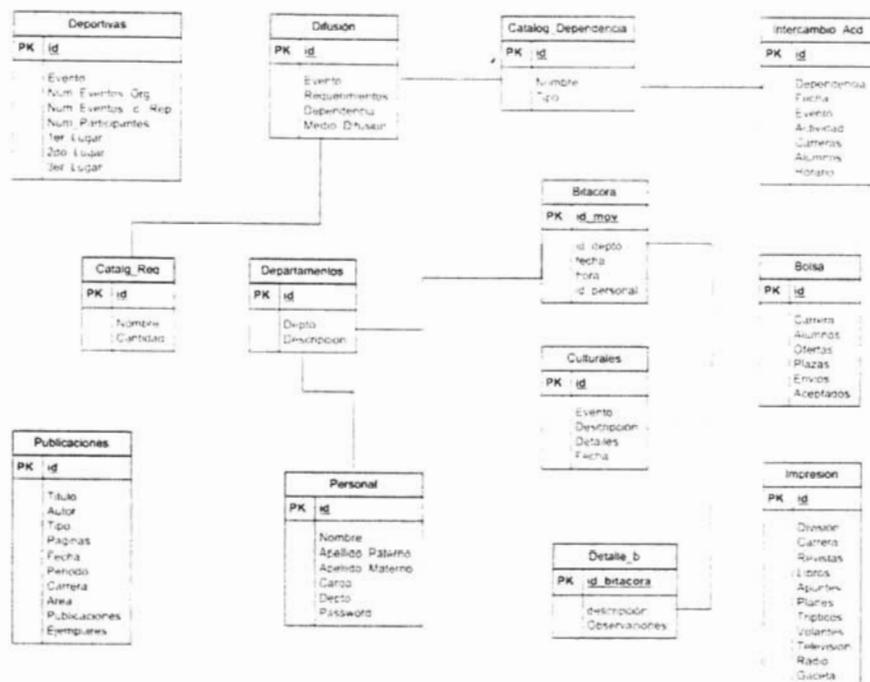


Figura 32: Diagrama entidad-relación

3.12 Reglas de Negocio

Toda aplicación trata de reflejar parte del funcionamiento del mundo real, para automatizar tareas que de otro modo serían llevadas a cabo de modo menos eficiente, o bien, no podrían realizarse. Para ello, es necesario que cada aplicación refleje las restricciones que existen en el negocio dado, de modo que nunca sea posible llevar a cabo acciones no válidas.

A las reglas que debe seguir la aplicación para garantizar esto se les llama reglas de negocio, o business rules.

Existen dos tipos de reglas importantes:

El primer grupo de reglas de negocio engloba todas aquellas reglas que se encargan de controlar que la información básica almacenada para cada atributo o propiedad de una entidad u objeto es válida:

- No hay precios de artículos negativos
- El sexo de una persona sólo puede ser masculino o femenino
- Una fecha siempre debe ser una fecha válida (no existe el 30 de febrero ¿cierto?)

A estas reglas las llamaremos reglas del modelo de datos.

Otro grupo importante de reglas incluye todas aquellas reglas que controlan las relaciones entre los datos. Estas reglas especifican, por ejemplo, que todo pedido debe ser realizado por un cliente, y que el mismo debe estar dado de alta en nuestro sistema: además, una vez que el cliente haya hecho algún pedido, se deberá garantizar que no es posible eliminarlo, a menos que previamente se eliminen todos los pedidos. Estas reglas constituyen las reglas de relación.

El login de los usuarios debe contener por lo menos cuatro dígitos

Los eventos deben tener un nombre y un departamento

Todos los eventos deberán tener fecha completa de su realización

Para almacenar datos todos los espacios deberán estar con datos

Para poder realizar bien el sistema es fundamental realizar con cuidado los diferentes pasos que nos indica realizar el análisis y diseño del sistema, para el mejor aprovechamiento de todos nuestros recursos, así como entender cada uno de los procesos de dicho sistema, apoyándonos en los casos de uso y diagramas de secuencia.

Capítulo IV

Desarrollo e Implementación

4. Desarrollo e Implementación

Tomando en cuenta el análisis y diseño del sistema, además de verificar los diferentes aspectos que esto nos brindó, estamos listos para poder realizar el desarrollo del sistema y la implementación del mismo.

4.1 Formulario de Password

En este formulario el sistema decide si el usuario que se logea es o no un usuario autorizado para ingresar, cada usuario que se registra está almacenado en una tabla de la base de datos del sistema, por lo cual si se desea denegar el acceso a un usuario, se debe eliminar de la base de datos.

A continuación se muestran la apariencia del formulario y el código que se ejecuta con cada botón:

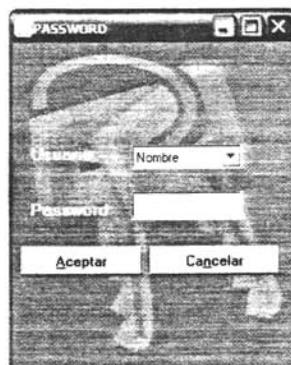


Figura 33: Pantalla de password

```

Private Sub Form_Load() Aquí se establece la conexión a la base de datos
User.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" + UNIDAD +
"EUSI.mdb;Persist Security Info=False"
User.RecordSource = "SELECT * FROM Personal"
User.Refresh
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer) En esta sección se oculta el password
If KeyAscii = 13 Then
    Call aceptar_Click
End If
End Sub

Private Sub aceptar_Click() A hacer click en el boton aceptar debe mostrar el formulario Principal.
pass.Hide
MDIForm1.Show
End Sub

Private Sub Cancelar_Click() Cuando se cancela el proceso, el sistema debe cerrarse
End
End Sub

Private Sub DataCombo1_Click(Area As Integer)
If Area = 2 Then
    Text1.Locked = False
    Text1.SetFocus
End If
End Sub

```

Tabla 36: Parte del código de l password

4.2 Formulario principal de la aplicación

Este formulario sólo funciona como llave para acceder a las distintas partes del sistema, es decir, desde aquí se pueden dar altas, bajas, cambios y consultas de cualquiera de las opciones disponibles, sólo funciona como separador entre las distintas partes del sistema.

Por obviedad el código que se utiliza en este formulario sólo muestra u oculta los formularios que de él se derivan.

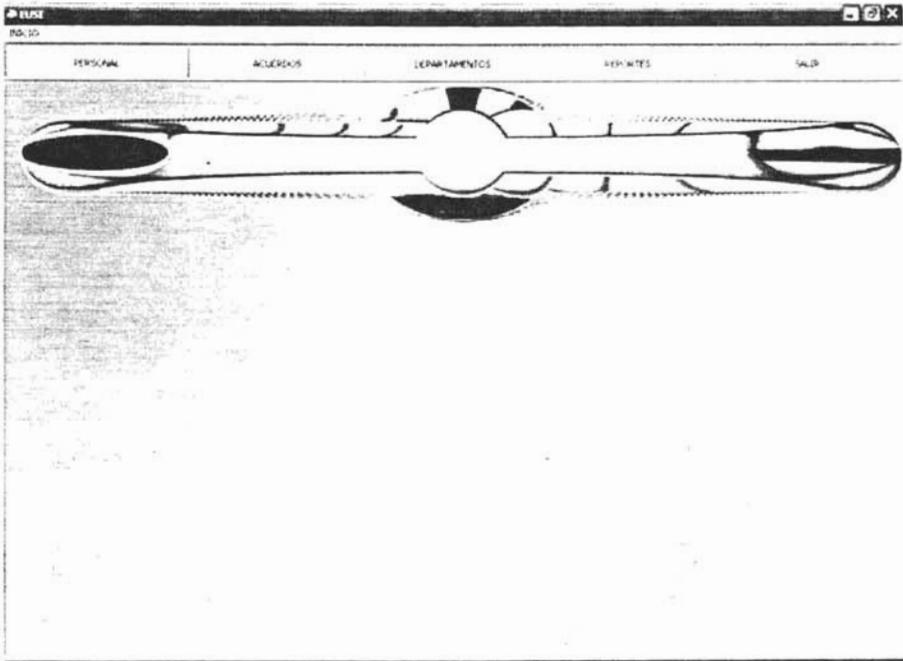


Figura 34: Pantalla del Menú principal

```

Private Sub Command5_Click()
    Perso.Show
End Sub

Private Sub Command1_Click()
    Acuerdos.Show
End Sub

Private Sub Command2_Click()
    menu_depto.Show
End Sub

Private Sub Command4_Click()
    Unload Me
End
End Sub
    
```

Tabla 37: Parte del código del menú principal

4.3 Formulario del personal

El código de este formulario incluye la llamada a un módulo que nos permite habilitar o deshabilitar los botones de la izquierda, que son las opciones principales de cada formulario, las etiquetas y cajas de texto están ligadas entre si de tal forma que su propiedad Caption es rellena automáticamente con los nombres de los campos de la tabla en la base de datos a la que se conecta.

Figura 35: Pantalla del Formulario de Personal

```
Private Sub Form_Load() ' Rellena la propiedad caption de cada etiqueta
```

```
For i = 0 To Personal.Recordset.Fields.Count - 1
```

```
Label1(i).Caption = Personal.Recordset.Fields(i).Name
```

```
Next i
```

```
Command2.Enabled = False
```

```
Command3.Enabled = FalseEnd Sub
```

```
Private Sub Command1_Click() ' Prepara la tabla para agregar un registro, llama a la  
funcion BOTONES y habilita o deshabilita los botones  
segun sea necesario
```

```
resp = False
```

Tabla 38: Parte del código del formulario del personal

```

Text1(0).Enabled = True
On Error GoTo Err_Addnew
Personal.Recordset.AddNew
Label4.Visible = True
Text3.Visible = True
For i = 0 To 6
    Text1(i).Locked = False
Next i
Call BOTONES(Perso, False, True, True, False, False, False, False)
Text1(0).Locked = True
Text1(0) = Format(Personal.Recordset.RecordCount, "000")
Text1(1).SetFocus
Exit Sub
Err_Addnew:
MsgBox Err.Description
End Sub

Private Sub Command2_Click() 'Este botón guarda los registros nuevos en la base de
                                datos
    verif = Text1(6).Text
    If verif <> Text3.Text Then
        Text1(6).Text = ""
        Text3.Text = ""
        MsgBox ("Verificar Password")
        Text1(6).SetFocus
    Exit Sub
    Else
        On Error GoTo err_update
        Personal.Recordset.UpdateBatch adAffectAllChapters
        Label4.Visible = False
        Text3.Visible = False
        For i = 0 To 6
            Text1(i).Locked = True
        Next i
        Call BOTONES(Perso, True, False, False, True, True, True, True)
        Personal.Refresh
    Exit Sub
    End If
err_update:
MsgBox Err.Description
End Sub

Private Sub Command3_Click() 'En caso de no desear continuar con la captura, este
                                botón interrumpe el proceso y vuelve al formulario
                                principal.
    On Error GoTo err_cancel
    Personal.Recordset.CancelBatch adAffectAllChapters
    Personal.Refresh
    For i = 0 To 6
        Text1(i).Locked = True
    Next i
    Call BOTONES(Perso, True, False, False, True, True, True, True)

```

Tabla 38: Parte del código del formulario del personal (continuación)

```

Label4.Visible = False
Text3.Visible = False
Exit Sub
err_cancel:
MsgBox Err.Description
End Sub

Private Sub Command4_Click() Prepara los registros de una tabla para ser modificados
Dim aux As Integer, A As Integer
For i = 0 To 6
Text1(i).Locked = False
Next i
Call BOTONES(Perso, False, True, True, False, False, False, False)
Text1(1).SetFocus
Label4.Visible = True
Text3.Visible = True
End Sub

Private Sub Command7_Click()
Perso.Hide
End Sub

```

Tabla 38: Parte del código del formulario del personal (continuación)

4.4 Formulario de Acuerdos y/o Convenios

A partir de aquí, todos los formularios presentan la misma mecánica, pues debe existir una homogeneidad en los formularios, de tal forma que un usuario normal se encuentre con una interfaz uniforme en cada proceso que lleve a cabo.

Los formularios son similares tanto en código como en función, esto es debido a que los procesos que se llevan a cabo en la Unidad de Extensión Universitaria son muy similares entre sí, por lo que se eligió conformar una tabla para cada departamento existente, basada en los formatos que se manejan ahí, estos formatos nos fueron proporcionados durante la etapa de análisis.

The screenshot shows a Windows application window with a menu bar containing 'PERSONAL', 'ACUERDOS', 'DEPARTAMENTOS', 'REPORTES', and 'SALIR'. The 'ACUERDOS' menu is active. The main area contains a form with the following fields:

- ID_Acuerdo: 301
- ID_Division: 301
- Nombre Acuerdo: (empty)
- Fecha_Convenio: (empty)
- Nombre_Convenio: (empty)
- Responsable_Convenio: (empty)
- Propuesta: (empty)
- Fecha_inicio: (empty)
- Fecha_termina: 3
- Nombre_Servicio: (empty)
- Vigencia: (empty)

At the bottom, there is a data grid with the following columns: ID_Acuerdo, ID_Division, Nombre Acuerdo, Fecha_Convenio, Nombre_Convenio, Responsable_Convenio, Propuesta. The grid contains three rows of data:

| ID_Acuerdo | ID_Division | Nombre Acuerdo | Fecha_Convenio | Nombre_Convenio | Responsable_Convenio | Propuesta |
|------------|-------------|----------------|----------------|-----------------|----------------------|-----------|
| 301 | 301 | | | | | |
| 302 | | | | | | |
| 303 | | | | | | |

Figura 36: Pantalla del Formulario de Acuerdos y Convenios

```

Private Sub Form_Load()
For i = 0 To Convenios.Recordset.Fields.Count - 1
Label1(i).Caption = Convenios.Recordset.Fields(i).Name
Next i
Command2.Enabled = False
Command3.Enabled = False
Frame1.Visible = False
End Sub

Private Sub Command1_Click()
On Error GoTo Err_Addnew
Convenios.Recordset.AddNew
For i = 0 To 10
Text1(i).Locked = False
Next i
Call BOTONES(Acuerdos, False, True, True, False, False, False, False)
Convenios.Enabled = False
Text1(0) = Format(Convenios.Recordset.RecordCount, "000")
Text1(1).SetFocus
Text1(0).Locked = True
Departamento.Visible = True
Text1(2).SetFocus
Exit Sub
    
```

Tabla 39: Parte del código del formulario de acuerdos y convenios

```

Err_Addnew:
MsgBox Err.Description
End Sub

Private Sub Command10_Click()
If Command10.Caption = "Mostrar Acuerdo" Then
On Error GoTo error_doc
OLE1.Visible = True
OLE1.SourceDoc = (App.Path & "/" & Trim(Text1(0).Text) & ".doc")
OLE1.Action = 1 'indicamos que se visualise el documento
Command10.Caption = "Cerrar"

Else
OLE1.Visible = False
Command10.Caption = "Mostrar Acuerdo"
End If
Exit Sub
error_doc:
MsgBox "El Alumno no Tiene Expediente"
OLE1.Visible = False
End Sub

Private Sub mnuFileOpen_Click()
openfile (CommonDialog1.FileName)
End Sub

Private Sub Command11_Click()
CommonDialog1.ShowOpen
nombre = CommonDialog1.FileName

End Sub

Private Sub Command2_Click()
On Error GoTo err_update
Convenios.Recordset.UpdateBatch adAffectAllChapters
For i = 0 To 4
Text1(i).Locked = True
Next i
Call BOTONES(Acuerdos. True. False. False. True. True, True, True)
Convenios.Enabled = True
Exit Sub
err_update:
MsgBox Err.Description
End Sub

Private Sub Command3_Click()
On Error GoTo err_cancel
Convenios.Recordset.CancelBatch adAffectAllChapters
Convenios.Refresh
For i = 0 To 10
Text1(i).Locked = True
Next i

```

Tabla 39: Parte del código del formulario de acuerdos y convenios (continuación)

```

Call BOTONES(Acuerdos, True, False, False, True, True, True, True)
Convenios.Enabled = True
Exit Sub
err_cancel:
MsgBox Err.Description
End Sub

Private Sub Command4_Click()
For i = 0 To 10
Text1(i).Locked = False
Next i
Call BOTONES(Acuerdos, False, True, True, False, False, False, False)
Convenios.Enabled = False

End Sub

Private Sub Departamento_Click(Area As Integer)
If Area = 2 Then
Text1(1).Text = Departamento.BoundText
Departamento.Visible = False
Text1(2).SetFocus
End If
End Sub

```

Tabla 39: Parte del código del formulario de acuerdos y convenios (continuación)

4.5 Menú elección de departamentos

Este formulario es el acceso a cada departamento como tal, por lo cual su código sólo nos proporciona acceso a las distintas opciones de cada uno de ellos.

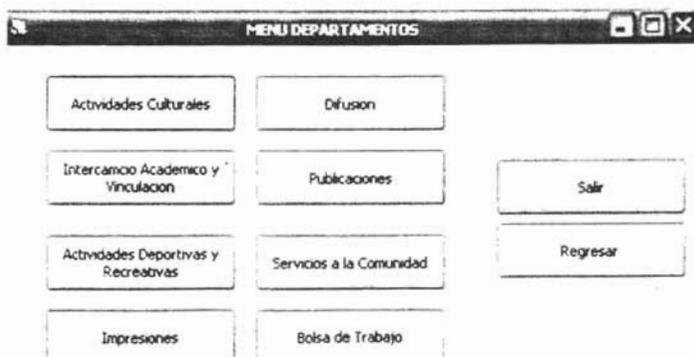


Figura 37: Pantalla del Menú de departamentos

```
Private Sub Command2_Click()  
Inter_av.Show  
End Sub  
  
Private Sub Command3_Click()  
Difu.Show  
End Sub  
  
Private Sub Command4_Click()  
Publica.Show  
End Sub  
  
Private Sub Command5_Click()  
Activ_DR.Show  
End Sub  
  
Private Sub Command6_Click()  
Impresion.Show  
End Sub  
  
Private Sub Command7_Click()  
Servicios.Show  
End Sub  
  
Private Sub Command8_Click()  
Bolsa.Show  
End Sub  
  
Private Sub Command9_Click()  
Unload Me  
End Sub
```

Tabla 40: Parte del código del formulario del menú de departamentos

4.6 Módulo de botones

En este módulo se tiene el código que se ejecuta cada vez que se presiona un botón dentro del sistema, no es otra cosa más que un arreglo que habilita o no los botones que deben estar disponibles en cada acción:

```

Public UNIDAD As String
Public ruta As String
Public USUARIO As String
Public Sub BOTONES(FORMULARIO As Form, AGREGA, GRABA, CANCELA, MODIFICA, BUSCA,
REPORTES, SALE As Boolean)
FORMULARIO.Command1.Enabled = AGREGA
FORMULARIO.Command2.Enabled = GRABA
FORMULARIO.Command3.Enabled = CANCELA
FORMULARIO.Command4.Enabled = MODIFICA
FORMULARIO.Command5.Enabled = BUSCA
FORMULARIO.Command6.Enabled = REPORTES
FORMULARIO.Command7.Enabled = SALE
End Sub

Public Sub RUTAS()
UNIDAD = "C:\Documents and Settings\Quique\Mis documentos\EUSA\
'ruta = "DSN=datos"
'Devoluciones.Picture = LoadPicture(UNIDAD + "FONDO.JPG")
End Sub

Public Sub BOTONES_C(FORMULARIO As Form, AGREGA, GRABA, MODIFICA, CANCELA, BUSCA,
IMPRIME, SALE, RECORRE As Boolean)
FORMULARIO.agregar.Enabled = AGREGA
FORMULARIO.grabar.Enabled = GRABA
FORMULARIO.modificar.Enabled = MODIFICA
FORMULARIO.cancelar.Enabled = CANCELA
FORMULARIO.buscar.Enabled = BUSCA
FORMULARIO.imprimir.Enabled = IMPRIME
FORMULARIO.salir.Enabled = SALE
FORMULARIO.Adodc2.Enabled = RECORRE
End Sub

```

Tabla 41: Parte del código del formulario del módulo de botones

4.7 Diccionario de datos

| Deportivas | | | | |
|-------------------|---------------|----------|-------|---------------------------|
| Nombre | Tipo Variable | Longitud | Nulos | Comentario |
| Id | Interger | 2 | N | Identificador deportivo |
| Evento | Varchar | 20 | N | Actividad realizada |
| Num_Eventos_Org | Numérico | 5 | N | Eventos organizados |
| Num_Eventos_c_Rep | Numérico | 5 | N | Eventos reportados |
| Num_Participantes | Numérico | 5 | N | Número de participantes |
| 1er_Lugar | Numérico | 2 | N | Ganador del 1er lugar |
| 2do_Lugar | Numérico | 2 | N | Ganador del 2er lugar |
| 3er_Lugar | Numérico | 2 | N | Ganador del 3er lugar |
| Difusión | | | | |
| Nombre | Tipo Variable | Longitud | Nulos | Comentario |
| Id | Interger | 2 | N | Identificador de difusión |
| Evento | Varchar | 20 | N | Actividad realizada |
| Requerimientos | Varchar | 20 | N | Requerimientos necesarios |
| Dependencia | Varchar | 20 | N | Dependencia |
| Medio_Difusión | Varchar | 20 | N | Medio utilizado |
| Bolsa | | | | |
| Nombre | Tipo Variable | Longitud | Nulos | Comentario |
| Id | Interger | 2 | N | Identificador de bolsa |
| Carrera | Varchar | 20 | N | Carrera |
| Alumnos | Numérico | 5 | N | Alumnos |
| Ofertas | Numérico | 5 | N | Ofertas |
| Plazas | Numérico | 5 | N | Plazas |
| Envios | Numérico | 2 | N | Envios |
| Aceptados | Numérico | 2 | N | Aceptados |

Tabla 42: Diccionario de datos del sistema EUSI

| Departamentos | | | | |
|---------------|---------------|----------|-------|--------------------------------|
| Nombre | Tipo Variable | Longitud | Nulos | Comentario |
| Id | Interger | 2 | N | Identificador de departamentos |
| Depto | Varchar | 20 | N | Departamento |
| Descripción | Varchar | 20 | N | Descripción del departamento |
| Catalog_Req | | | | |
| Nombre | Tipo Variable | Longitud | Nulos | Comentario |
| Id | Interger | 2 | N | Identificador de catalogo |
| Nombre | Varchar | 20 | N | Nombre del catalogo |
| Cantidad | Númeroico | 5 | N | Cantidad |
| Publicaciones | | | | |
| Nombre | Tipo Variable | Longitud | Nulos | Comentario |
| Id | Interger | 2 | N | Identificador de publicaciones |
| Título | Varchar | 20 | N | Título de la publicación |
| Autor | Varchar | 20 | N | Autor |
| Tipo | Varchar | 5 | N | Tipo |
| Paginas | Númeroico | 3 | N | Paginas |
| Fecha | Númeroico | 5 | N | Fecha |
| Periodo | Varchar | 10 | N | Periodo |
| Carrera | Varchar | 10 | N | Carrera |
| Área | Varchar | 10 | N | Área |
| Publicaciones | Varchar | 10 | N | Publicaciones |
| Ejemplares | Númeroico | 5 | N | Ejemplares publicados |

Tabla 42: Diccionario de datos del sistema EUSI (continuación)

Tabla 42: Diccionario de datos del sistema EUSI (continuación)

| Personal | | | | |
|-------------------|---------------|----------|-------|------------------------------|
| Nombre | Tipo Variable | Longitud | Nulos | Comentario |
| Id | Integer | 2 | N | Identificador del personal |
| Nombre | Varchar | 20 | N | Nombre del usuario |
| Apellido_paterno | Varchar | 20 | N | Apellido paterno del usuario |
| Apellido_materno | Varchar | 20 | N | Apellido materno del usuario |
| Cargo | Varchar | 10 | N | Cargo del usuario |
| Departamento | Varchar | 10 | N | Departamento del usuario |
| Password | Varchar | 5 | N | Contraseña |
| Culturales | | | | |
| Nombre | Tipo Variable | Longitud | Nulos | Comentario |
| Id | Integer | 2 | N | Identificador de culturales |
| Evento | Varchar | 20 | N | Actividad realizada |
| Descripción | Varchar | 20 | N | Descripción de la actividad |
| Detalles | Varchar | 20 | N | Detalles de la actividad |
| Fecha | Númeroico | 5 | N | Fecha de actividad |
| Bitácora | | | | |
| Nombre | Tipo Variable | Longitud | Nulos | Comentario |
| Id_mov | Integer | 2 | N | Identificador de difusión |
| Id_depto | Varchar | 20 | N | Actividad realizada |
| Fecha | Varchar | 20 | N | Requerimientos necesarios |
| Hora | Varchar | 20 | N | Dependencia |
| Id_personal | Varchar | 20 | N | Medio utilizado |

| Catalog_Dependencia | | | | |
|---------------------|---------------|----------|-------|------------------------------|
| Nombre | Tipo Variable | Longitud | Nulos | Comentario |
| Id | Interger | 2 | N | Identificador de dependencia |
| Nombre | Varchar | 20 | N | Nombre dependencia |
| Tipo | Varchar | 20 | N | Tipo dependencia |
| Detalle_b | | | | |
| Nombre | Tipo Variable | Longitud | Nulos | Comentario |
| Id_bitacora | Interger | 2 | N | Identificador de detalle |
| Descripción | Varchar | 20 | N | Descripción |
| Observaciones | Varchar | 20 | N | Observaciones |
| Impresión | | | | |
| Nombre | Tipo Variable | Longitud | Nulos | Comentario |
| Id | Interger | 2 | N | Identificador de impresión |
| División | Varchar | 10 | N | División |
| Carrera | Varchar | 10 | N | Carrera |
| Revistas | Varchar | 10 | N | Revistas |
| Libros | Varchar | 10 | N | Libros |
| Apuntes | Varchar | 10 | N | Apuntes |
| Planes | Varchar | 10 | N | Planes |
| Tripticos | Varchar | 10 | N | Tripticos |
| Volantes | Varchar | 10 | N | Volantes |
| Televisión | Varchar | 10 | N | Televisión |
| Radio | Varchar | 10 | N | Radio |
| Gaceta | Varchar | 10 | N | Gaceta |

Tabla 42: Diccionario de datos del sistema EUSI (continuación)

4.8 Sistema de información

En el presente trabajo de tesis, nos hemos encaminado a la resolución de cierto problema aplicando un sistema de información optimizado para el mejor manejo de toda la información que utiliza la Unidad de Extensión Universitaria, apoyada con una confiable base de datos, describiendo en general los pasos a seguir desde el análisis de factibilidad, hasta la validación y prueba del sistema.

Ahora iniciaremos con el ciclo de vida de un sistema de datos para entender mejor globalmente su funcionamiento.

4.8.1 Ciclo de vida de un Sistema de Información

Un sistema de información engloba todos los recursos que dentro de la organización se encargan de la recolección, administración y uso de la información, si el sistema de información emplea mecanismos computarizados, entonces el SGBD, el hardware empleado, el personal que usa y administra los datos, etc. todo esto queda incluido en el universo del sistema de información.

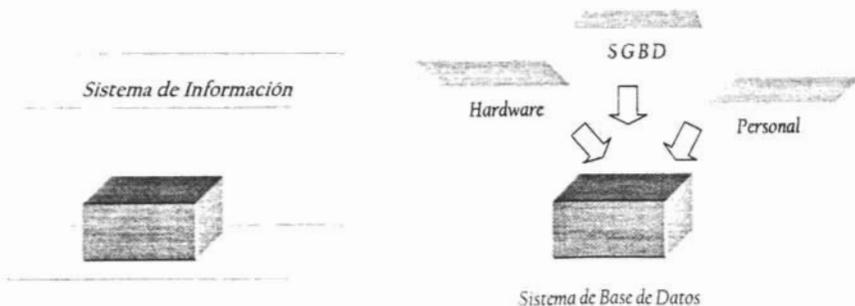


Figura 38: Sistema de aplicación de Base de Datos dentro de un sistema de Información

El ciclo de vida de un sistema de información incluye por lo general los siguientes pasos:

1. **Análisis de factibilidad:** Se ocupa de analizar las posibles áreas de aplicación, en esta fase se lleva a cabo estudios preliminares en donde se comparan los costos contra los beneficios, de igual manera se establecen prioridades entre las aplicaciones que se contemplan desarrollar dentro del sistema.
2. **Recolección y análisis de requerimientos:** En esta fase se obtienen los requerimientos detallados de propia mano de los posibles usuarios, con la finalidad de identificar sus problemas y necesidades específicas.
3. **Diseño:** Esta fase está formada por dos partes, por un lado se debe trabajar sobre el diseño de la base de datos y por el otro se trabaja sobre el diseño de los sistemas de aplicación, es decir los programas que usan y almacenan la información contenida en la base de datos.
4. **Implementación:** En esta fase el sistema de información se instala, la base de datos se carga y se hacen las pruebas de transacciones en la base de datos, esta fase termina con el programa puesto en marcha y con las pruebas preliminares llevadas a cabo satisfactoriamente.
5. **Validación y prueba de aceptación:** Se valida el sistema en cuanto a los requerimientos establecidos en la fase dos, el sistema es aprobado contra los requerimientos y especificaciones de comportamiento que se definieron junto con los futuros usuarios.
6. **Operación:** Esta fase operativa comienza después de que todas las funciones han sido probadas y validadas, en el proceso de operación, viene acompañado de un monitoreo periódico al sistema para supervisar el rendimiento y estar preparados para dar mantenimiento en caso de posibles fallas. Es posible que en esa fase se requieran modificaciones, en cuyo caso se debe de pasar de nuevo por todas las etapas anteriores.

En lo escrito hasta ahora, se han mencionado las fases que componen el ciclo de vida de un sistema de información y el ciclo de vida de un sistema de base de datos, para asegurar que hasta este momento se tenga una idea cronológica correcta de donde está situado cada proceso, a continuación se expone una gráfica en donde se precisa cada paso antecedente a cual otro.



Figura 39: Ciclo de vida del sistema de información

4.9 Implementación del sistema

A continuación se muestra una planeación general de la implementación del sistema, donde se muestran las diferentes etapas de la misma.

1. Verificación del entorno de la infraestructura para que ésta fuese la adecuada para nuestro sistema.
2. Instalación de la aplicación en cada uno de los equipos así como la base de datos en el equipo principal.
3. Pruebas de funcionamiento de los 2 puntos anteriores para verificar su buen funcionamiento
4. Impartición del curso de capacitación a los usuarios del sistema.

5. Seguimiento durante dos semanas para observar el desempeño del sistema y la adaptación de los usuarios.
6. Entrega de los manuales a los principales usuarios, para dudas futuras.

4.9.1 Infraestructura

Para poder implantar nuestro sistema necesitaremos diferentes características de los equipos de cómputo de cada área como a continuación se especifica.

- Tener una conexión de red alámbrica o inalámbrica, en caso de ser alámbrica que cuente con una tarjeta de red de 10/100 Mb conectada a un swicht general, o un acces point en caso inalámbrico tomando en cuenta que la red corra mínimo a 10 Mbps
- Espacio libre en disco duro de 100 Mb por cada equipo y 300 Mb en la computadora principal para hospedar la base de datos, para que tenga la suficiente libertad el sistema y tenga un mejor desempeño.
- Una resolución en pantalla mínima de 800 X 600 dpi u óptima de 1024 X 768 dpi, debido a que la programación de la interfaz y al tamaño de los objetos en la pantalla de la computadora se realizó en una resolución de 1024 X 768 dpi.
- Cada equipo de cómputo deberá contar con un lector de CD y unidad de floppy de 3 ½ ", para que por medio de estos lectores de datos podamos instalar el sistema.
- Además de contar con un sistema operativo de Windows 98 u otro más avanzado debido a que diferentes controles de VB únicamente funcionan a partir de Windows 98.

4.9.2 Instalación de la aplicación

Ya que se verificó el software y el hardware se procederá a instalar el sistema en cada una de las computadoras de la Unidad de Extensión Universitaria.

Primero se instalará la interfaz para cada área o departamento y después se implantará la base de datos en la computadora principal.

Entonces se realizarán diferentes pruebas para verificar la conexión del sistema para poder ingresar los nuevos datos

Para entender mejor la instalación del sistema a continuación se ejemplifica en un plan de trabajo que muestra las diferentes actividades que se desarrollaron.

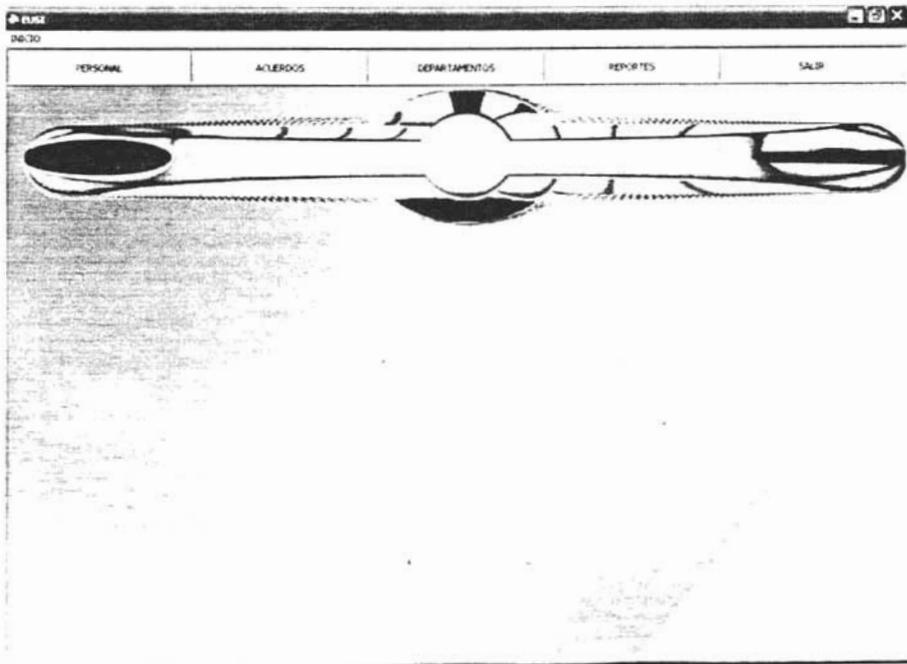


Figura 40: Pantalla del menú principal

Plan de Trabajo

Sistema: EUSI
Desarrolladores: José Enrique Atanacio Miranda
Isael Toledo Sánchez
Solicitante: Unidad de Extensión Universitaria
Responsable: Alicia Berthier

| Actividad | Responsable | Fecha de inicio | Fecha final | Fecha real |
|------------------------------------|----------------------------|-------------------|-------------------|------------|
| Recopilación de necesidades | Desarrolladores | 15 diciembre 2003 | 15 abril 2004 | |
| Análisis de información | Desarrolladores | 15 abril 2004 | 15 agosto 2004 | |
| Elección de Software | Desarrolladores | 15 agosto 2004 | 15 diciembre 2004 | |
| Diseño de sistema | Desarrolladores | 15 diciembre 2004 | 15 mayo 2005 | |
| Verificación del hardware | Desarrolladores | 15 mayo 2005 | 20 mayo 2005 | |
| Verificación del software | Desarrolladores | 15 mayo 2005 | 20 mayo 2005 | |
| Instalación de la interfaz | Desarrolladores | 20 mayo 2005 | 10 junio 2005 | |
| Implementación de la base de datos | Desarrolladores | 20 mayo 2005 | 10 junio 2005 | |
| Pruebas de funcionamiento | Desarrolladores | 10 junio 2005 | 11 junio 2005 | |
| Definición y creación de perfiles | Desarrolladores y Berthier | 11 junio 2005 | 15 junio 2005 | |
| Actualización de datos | Berthier | 11 junio 2005 | 15 junio 2005 | |
| Capacitación de usuarios | Desarrolladores | 15 junio 2005 | 1 julio 2005 | |
| Validación por parte usuarios | Berthier | 1 julio 2005 | 15 julio 2005 | |
| Seguimiento del sistema | Desarrolladores | 15 julio 2005 | 1 Agosto 2005 | |
| Entrega de manuales | Desarrolladores | 1 Agosto 2005 | 5 Agosto 2005 | |
| Entrega del proyecto | Desarrolladores | 5 Agosto 2005 | | |

Tabla 43: Plan de trabajo del sistema EUSI

Gráfica de Gantt

Tabla 44: Gráfica de Gantt del sistema EUSI

| Actividad | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------------------------------|------|-----|-----|-----|-----|-----|-----|-----|------|-----|-----|-----|-----|-----|-----|-----|------|-----|-----|-----|-----|--|--|--|--|
| Recoopilación de necesidades | | | | | | | | | | | | | | | | | | | | | | | | | |
| Análisis de información | | | | | | | | | | | | | | | | | | | | | | | | | |
| Elección de software | | | | | | | | | | | | | | | | | | | | | | | | | |
| Diseño del sistema | | | | | | | | | | | | | | | | | | | | | | | | | |
| Verificación del hardware | | | | | | | | | | | | | | | | | | | | | | | | | |
| Verificación del software | | | | | | | | | | | | | | | | | | | | | | | | | |
| Instalación de la Interfaz | | | | | | | | | | | | | | | | | | | | | | | | | |
| Implementación de la base de datos | | | | | | | | | | | | | | | | | | | | | | | | | |
| Pruebas de funcionamiento | | | | | | | | | | | | | | | | | | | | | | | | | |
| Definición y creación de perfiles | | | | | | | | | | | | | | | | | | | | | | | | | |
| Actualización de datos | | | | | | | | | | | | | | | | | | | | | | | | | |
| Capacitación de usuarios | | | | | | | | | | | | | | | | | | | | | | | | | |
| Validación por parte usuarios | | | | | | | | | | | | | | | | | | | | | | | | | |
| Seguimiento del sistema | | | | | | | | | | | | | | | | | | | | | | | | | |
| Entrega de manuales | | | | | | | | | | | | | | | | | | | | | | | | | |
| Entrega del proyecto | | | | | | | | | | | | | | | | | | | | | | | | | |
| | Dic | Ene | Feb | Mar | Abr | May | Jun | Jul | Ago | Sep | Oct | Nov | Dic | Ene | Feb | Mar | Abr | May | Jun | Jul | Ago | | | | |
| | 2003 | | | | | | | | 2004 | | | | | | | | 2005 | | | | | | | | |

4.9.3 Pruebas de funcionamiento

Ya implementado el sistema se procederán a validar las pruebas de funcionamiento como se muestra en los siguientes cuadros.

Pruebas de funcionamiento

| Actividad | Datos de Entrada | Resultado Esperado | Operaciones | Observaciones | Fecha |
|----------------------|-------------------------------|--|-------------|---------------|-------|
| Acceso al sistema | Clave de Usuario | Acceso al menú principal | | | |
| Selección del área | Elección de icono | Acceso al área seleccionada | | | |
| Alta de Usuario | Usuario, clave, detalles | Ingreso de nuevo usuario en el sistema | | | |
| Baja de Usuario | Usuario, clave, observaciones | Eliminación de usuario en el sistema | | | |
| Consulta de Usuario | Usuario, clave | Se muestra en pantalla el usuario | | | |
| Impresión de Usuario | Usuario, clave, impresión | Se imprimen datos del usuario | | | |
| Cambio de Usuario | Usuario, clave, cambios | Almacenan datos nuevos de usuario | | | |
| Alta de Evento | Evento, fecha, detalles | Ingreso de nuevo evento en el sistema | | | |
| Baja de Evento | Evento, fecha, detalles | Eliminación de evento en el sistema | | | |
| Consulta de Evento | Evento, fecha | Se muestra en pantalla el evento | | | |
| Impresión de Evento | Evento, fecha, impresión | Se imprimen datos del evento | | | |
| Cambio de Evento | Evento, fecha, cambios | Almacenan datos nuevos del evento | | | |
| Alta de Acuerdo | Acuerdo, fecha, detalles | Ingreso de nuevo acuerdo en el sistema | | | |
| Baja de Acuerdo | Acuerdo, fecha, detalles | Eliminación de acuerdo en el sistema | | | |
| Consulta de Acuerdo | Acuerdo, fecha | Se muestra en pantalla el acuerdo | | | |
| Impresión de Acuerdo | Acuerdo, fecha, impresión | Se imprimen datos del acuerdo | | | |
| Cambio de Acuerdo | Acuerdo, fecha, cambios | Almacenan datos nuevos del acuerdo | | | |

Tabla 45: Pruebas de funcionamiento del sistema EUSI

4.9.4 Capacitación

Nos pusimos de acuerdo con el personal que utilizará el sistema para impartir un curso de capacitación durante 15 días, donde se explico el uso general del sistema, dicho curso contó con los siguientes puntos:

1. **Acceso.** Se muestra el camino correcto para que los usuarios ingresaran al sistema sin ningún problema, así como el manejo de las contraseñas para lo diferentes usuarios.
2. **Elementos de Interfaz.** En este módulo se muestran las diferentes pantallas del sistema así como cada uno de los elementos o íconos que se muestran en dichas pantallas.
3. **Módulos del sistema.** Todo el sistema implantado está dividido para su mejor manejo en módulos, aquí se muestra cada uno de ellos y su función.
4. **Reportes.** Explicamos como efectuar los distintos reportes que puede generar el sistema.
5. **Salida.** Después del uso del sistema es conveniente que los usuarios salgan correctamente del sistema para un buen funcionamiento.
6. **Respaldos.** Por último se explica como darle mantenimiento periódicamente para la generación de respaldos de la información utilizada.

4.9.5 Seguimiento

Para dar seguimiento al sistema y verificar su funcionamiento, así como resolver las diferentes dudas que pudieran surgir dentro del uso continuo del sistema, sugerimos la siguiente encuesta:

Encuesta de servicios

| CON RELACIÓN AL PERSONAL | | Respuestas | | | |
|---|--|------------|---|---|---|
| E=Excelente, B=Bueno, R=Regular, M=Malo | | E | B | R | M |
| 1 | ¿Cómo califica el tiempo que se le asignó al desarrollo del proyecto? | | | | |
| 2 | El nivel de conocimientos de la persona que atendió su solicitud fue | | | | |
| 3 | ¿Cómo evalúa la actitud del personal que le atendió? | | | | |
| 4 | ¿Cómo califica la disponibilidad para atender sus dudas o comentarios? | | | | |
| 5 | ¿Cómo fue la comunicación con Usted para informarlo del avance del proyecto? | | | | |
| 6 | ¿Cómo califica de forma general la atención y solución de su solicitud? | | | | |
| CON RELACIÓN AL SERVICIO | | Respuestas | | | |
| Responda según corresponda a la pregunta S=Si, N=No ó | | S | N | | |
| E=Excelente, B=Bueno, R=Regular, M=Malo | | E | B | R | M |
| 1 | ¿Cuál es el nivel de cumplimiento del objetivo de su solicitud, de acuerdo con lo solicitado? | | | | |
| 2 | Califique el grado de cumplimiento de la fecha de entrega de la solución de su solicitud | | | | |
| 3 | ¿Se le entregó un documento de plan de trabajo sobre su requerimiento? | | | | |
| 4 | Califique el grado de cobertura del documento del plan de trabajo de su requerimiento | | | | |
| 5 | ¿Cuál fue el nivel de claridad del documento de plan de trabajo que se entregó para la solución de su requerimiento? | | | | |
| 6 | ¿Recibió la capacitación requerida? | | | | |
| 7 | ¿Cómo califica el manual y/o la capacitación recibida? | | | | |
| 8 | ¿Se entregó la documentación necesaria para entender y operar la nueva funcionalidad solicitada? | | | | |
| 9 | ¿El requerimiento presentó fallas al momento de empezar a utilizarlo en producción? | | | | |
| 10 | ¿Se atendieron con prontitud las fallas reportadas? | | | | |
| 11 | ¿Se recibió alguna documentación sobre la solución realizada para la falla reportada? | | | | |
| Comentarios Adicionales | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

Tabla 46: Encuesta de servicios del sistema EUSI

4.9.6 Entrega de manuales.

Finalmente, se realizó la entrega del manual para las diferentes dudas que surgieran en un futuro, además de las guías completas del sistema.

Se desarrolló e implementó satisfactoriamente el sistema, logrando una mejora notable a la Unidad de Extensión Universitaria, y por medio de diferentes planes de prueba nos encontramos con el voto aprobatorio de los usuarios, así que ahora procederemos a entregar diferentes manuales para el uso futuro de los manejadores del sistema.

Capítulo V
**Mantenimiento y
manuales**

5. Mantenimiento y manuales

Para poder considerar el desarrollo de un sistema como ya concluido, debemos haber llegado al punto en el cual el personal de la institución sea capaz de administrarlo por sí mismo, es decir, que se elimine la dependencia del desarrollador para laborar, pues en muchas ocasiones no se puede concluir una tarea sin que se haga necesaria una consulta con éste.

Tomando lo anterior en consideración, se hace necesaria la elaboración de un programa de mantenimiento mediante el cual se localicen y corrijan los errores típicos de uso, como faltantes de información, rutas cruzadas, etc. Así como también la corrección de errores de otros tipos.

Además del citado programa de mantenimiento, se hace necesaria una herramienta que permita al usuario en forma autodidacta o asistida la asimilación del funcionamiento del sistema. Para lo cual se elaboraron los siguientes textos:

5.1 Mantenimiento

En primer lugar, hay que establecer el hecho de que para acceder al sistema cada usuario debe tener una jerarquía distinta de acuerdo a las características de su puesto, de tal forma que encontramos a usuarios con niveles distintos en cuanto a lo que pueden y no hacer.

Para tal efecto, debemos clasificar a cada usuario dentro de alguna de las siguientes cuentas, a fin de limitar el acceso a ciertas partes del sistema a quienes tienen las facultades y los permisos necesarios:

| TIPO DE CUENTA | TABLAS A LAS QUE ACCESA: | NIVEL DE AFECTACION | EQUIVALENCIA | SERVICIOS QUE PROPORCIONA |
|----------------|-----------------------------|----------------------|---|--|
| ADMINISTRADOR | Todas | Lectura Escritura | Director | Alta, baja y modificación de usuarios y eventos, asignación de nivel de acceso |
| ADMINISTRATIVO | Solo las de su departamento | Lectura Escritura | Jefe de cada departamento | Alta, baja y modificación de eventos |
| CONSULTOR "A" | Todas | Lectura | Personas autorizadas en la dirección | Consulta e impresión de reportes de todos los departamentos |
| CONSULTOR "B" | Solo las de su departamento | Lectura | Personas autorizadas en cada departamento | Consulta e impresión de reportes en su departamento |

Tabla 47: Cuentas de usuario del sistema EUSI

Para realizar un mantenimiento eficaz tenemos un programa de mantenimiento en el cual se contemplan diversos puntos en cuanto a la revisión periódica de algunos aspectos que mencionaremos en dicho programa:

| ACTIVIDAD | | AFECTACION | PRIORIDAD | PERIODICIDAD |
|---|----------------|------------|-----------|--------------------|
| CAMBIO DE PASSWORD | ADMINISTRADOR | Software | Alta | Mensual |
| | ADMINISTRATIVO | Software | Media | Mensual |
| | CONSULTOR "A" | Software | Baja | Semestral |
| | CONSULTOR "B" | Software | Baja | Semestral |
| RESPALDO DE BASE DE DATOS | | Software | Alta | Semestral |
| REVISION DEL TAMAÑO DE LA BASE DE DATOS | | Software | Alta | Trimestral |
| DEFRAGMENTACION DE DISCO DURO AL EQUIPO PRINCIPAL | | Hardware | Media | Mensual |
| LIMPIEZA GENERAL A TODOS LOS EQUIPOS | | Hardware | Baja | Semestral |
| REVISIÓN DE LA CONECTIVIDAD EFECTIVA DE LA RED | | Software | Media | Trimestral |
| DEPURACIÓN Y/O BORRADO DE LA BASE DE DATOS | | Software | Media | Anual |
| FORMATEO DE DISCO DURO | | Hardware | Alta | Cuando se requiera |
| REINSTALACIÓN DE LA APLICACIÓN | | Software | Alta | Cuando se requiera |

Tabla 48: Programa de mantenimiento del sistema EUSI

Este programa se llevará a cabo a partir de la fecha en la que se implementa el sistema y se considere que esta funcionando correcta y eficazmente, la periodicidad de cada tarea se determina en función de los cambios que pueda presentar la parte del sistema a la que afecta, así como en base al tiempo que pueden considerarse estables.

5.2 Manual del Usuario EUSI

En este texto nos ocuparemos de la utilización del sistema EUSI desde cero, es decir, el primer paso será encender el computador. Para hacer esto, debemos presionar el botón de encendido de la PC, dicho botón es generalmente el de mayor tamaño al frente del CPU, que es la parte principal del computador.

Una vez encendido, deberemos esperar a que inicie el sistema operativo; es decir, debemos esperar a que se pueda iniciar cualquier actividad en nuestro sistema.

Para poder trabajar con un computador, hay ciertas cosas que debemos saber, por ejemplo el **dar click con el Mouse**, que significa presionar el botón izquierdo del Mouse una ocasión, con lo cual se inicia un proceso determinado, en tanto que se puede **dar doble click con el Mouse**, que significa presionar dos veces el botón izquierdo del Mouse en forma rápida y continua para lograr el mismo efecto.

Un **icono** es la representación gráfica de un elemento informático, que puede ser un programa, un texto, una imagen, etc., cada elemento es representado por una imagen alusiva a su utilización, por eso es que programas como Word tienen como representación una **W**.

Así mismo, el **icono** que representa a nuestro sistema es un computador, dicha imagen se encuentra en la pantalla del monitor, junto con un grupo de **iconos**, en la zona llamada escritorio, que es la parte que se visualiza una vez que se entra al sistema operativo.

En caso de que no se encuentre la imagen del sistema, se deberá dar click en la barra de inicio, que se localiza en la parte inferior izquierda de cualquier

sistema Windows. Exactamente es eso lo que se lee en este botón (Inicio), al dar click sobre este botón se desplegará un menú que contiene distintas opciones, se deberá buscar una opción marcada como programas, y dentro de este submenú se debe buscar otra opción denominada **EUSI**.

Una vez localizada esta opción, se dará click sobre este **icono** para iniciar el sistema, cuando se encuentre funcionando usted podrá visualizar la pantalla de carga del programa; enseguida se le presentará la ventana de login o password, en donde usted tecleará un nombre de usuario y una contraseña que le han sido suministrados por el administrador general del sistema.

Si su password es correcto, usted tendrá acceso a la pantalla principal del sistema, en la cual se pueden ver varios elementos.

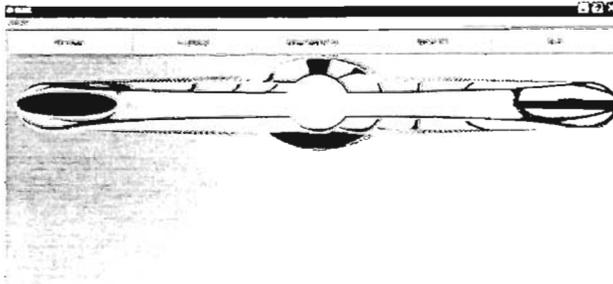


Figura 41: Pantalla Principal del Sistema

En esta pantalla usted podrá ver cinco opciones a elegir en la parte gris de la ventana, dichas opciones son: Personal, Acuerdos, Departamentos,

Reportes y Salir, cada una de las cuales son iguales a las opciones que aparecen al inspeccionar el menú que se presenta en la parte superior izquierda de la ventana.

Para poder iniciar cada opción, sólo es necesario hacer click sobre cualquiera de los iconos, una vez hecho esto se podrá acceder a las alternativas que cada apartado tiene disponibles.

5.2.1 Personal

Al dar click sobre esta opción se desplegará la ventana en la que se pueden ver las distintas actividades posibles de realizar en lo relacionado al personal que tenga acceso al sistema.

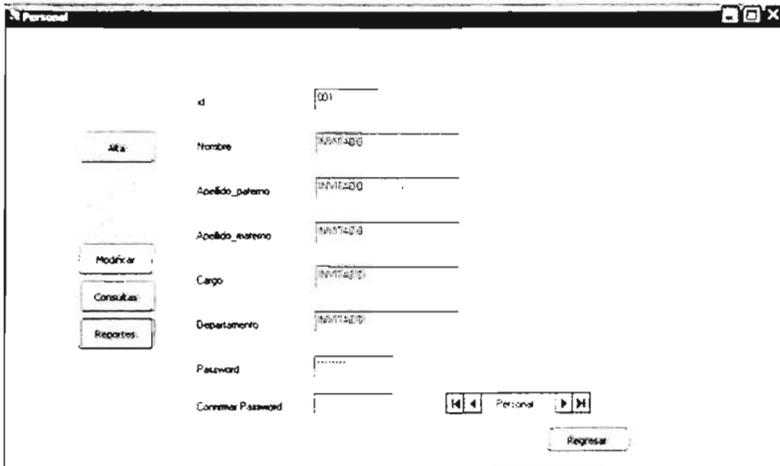


Figura 42: Ventana de Personal

Alta

Nos da la posibilidad de agregar un usuario autorizado para tener acceso al sistema y sus diversas funciones, al dar click sobre este botón se habilitan los botones **Guardar** y **Cancelar**, es decir que serán las únicas opciones disponibles, en tanto que los otros botones se bloquearán.

Una vez que elegimos esta función podremos ingresar los datos necesarios para dar de alta a un usuario nuevo, dichos datos se explican a continuación en la siguiente tabla:

| | |
|---------------------------|---|
| Id | Es el número de usuario que se le asignará, puede estar entre los valores 001 y 1000 |
| Nombre | El o los nombres del nuevo usuario |
| Apellido_paterno | El primer apellido del nuevo usuario |
| Apellido_materno | El segundo apellido del nuevo usuario |
| Cargo | Nivel de responsabilidad que tiene el nuevo usuario |
| Departamento | Dependencia o unidad en la que labora el nuevo usuario |
| Password | Clave de entrada que tendrá el usuario para acceder al sistema, debe ser mayor de 6 caracteres y no incluir signos de puntuación exclamación ni caracteres especiales |
| Confirmar Password | Clave de entrada que tendrá el usuario para acceder al sistema, debe ser idéntico a lo tecleado en la casilla anterior |

Tabla 49: Datos necesarios para dar de alta un usuario en el sistema EUSI

Si los datos ingresados son correctos, se almacenarán en la base de datos dando click en el botón **Guardar**, o si se prefiere abortar la operación y regresar al menú principal de esta opción se presionará el botón **Cancelar**.

Modificar

En caso de haber cometido algún error en la captura de los datos, o simplemente en caso de cambio, se pueden modificar los datos que identifican a cualquier usuario mediante esta opción, una vez presionado este botón se deberá digitar en la primer casilla el número de usuario que se quiere modificar para que el sistema nos presente la información correspondiente, los únicos botones disponibles en esta opción serán **Guardar** y **Cancelar**.

Una vez localizado el usuario se procederá a modificar la información que sea necesaria, si los datos ingresados son correctos, se almacenarán en la base de datos dando click en el botón **Guardar**, o si se prefiere abortar la operación y regresar al menú principal de esta opción se presionará el botón **Cancelar**.

Consultas

Si lo único que se requiere es una revisión rápida de los datos de algún usuario registrado se da click en este botón, con lo que aparecerán del lado derecho de la ventana el botón **Buscar**, y una casilla de texto, en la que se deberá teclear el número de usuario o su nombre y apellidos.

Para poder ver la información ingresada de un usuario se da click en el botón **Buscar** para que el sistema localice y visualice la búsqueda.

Reportes

Si lo que se requiere es una revisión rápida de los datos de algún usuario registrado, además de la **impresión** de ellos, se da click en este botón, con lo

que aparecerán del lado derecho de la ventana el botón **Buscar**, y una casilla de texto, en la que se deberá teclear el número de usuario o su nombre y apellidos.

Para poder ver la información ingresada de un usuario se da click en el botón **Buscar** para que el sistema localice y visualice la búsqueda.

Para poder imprimir la información consultada se deberá presionar el botón **Imprimir**.

5.2.2 Acuerdos

En esta opción se puede hacer todo lo relacionado con los Acuerdos y Tratados que firma la Unidad de Extensión Universitaria con distintas dependencias, instituciones u organizaciones ya sean internas o externas a la UNAM.

Es necesario remarcar que esta sección es de un manejo muy amplio en la Unidad, por lo que su variedad y obligatoria existencia en papel nos llevó a decidir que dentro del sistema sea sólo un registro general de dichos documentos, de tal forma que si se desea conocer información general este sea el medio utilizado para su consulta, pero en caso de requerir información más exacta y detallada, a cada documento se le ha asignado un **id**, que es un dígito identificador individual de cada Acuerdo, de tal forma que al ser localizado en el sistema también podrá encontrarse en el archivo físico mediante este mismo **id**. Su ventana es como sigue:

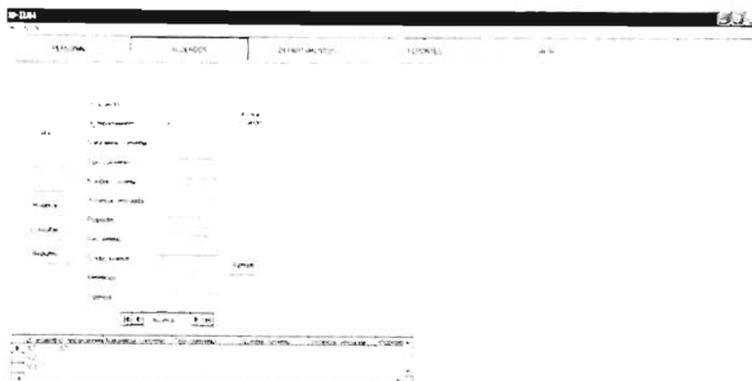


Figura 43: Ventana de Acuerdos

Alta

Nos da la posibilidad de agregar un acuerdo al sistema, al dar click sobre este botón se habilitan los botones **Guardar** y **Cancelar**, es decir que serán las únicas opciones disponibles, en tanto que los otros botones se bloquearán.

Una vez que elegimos esta función podremos ingresar los datos necesarios para dar de alta un acuerdo nuevo, dichos datos se explican a continuación:

| | |
|----------------------------|--|
| Id_acuerdo | Digito identificador de cada acuerdo |
| Id_departamento | Digito identificador de cada departamento, es proporcionado por el administrador general del sistema |
| Naturaleza_convenio | Interno o Externo |
| Tipo_convenio | Enfoque |
| Nombre_convenio | Nombre con el que se le deberá identificar |
| Instancia_vinculada | Institución, dependencia o asociación con la que se firma el acuerdo |
| Proposito | Meta a cumplir del acuerdo |
| Res_interno | Responsable directo en la unidad de dicho acuerdo |
| Grado_avance | Porcentaje de progreso en su ejecución al momento de su alta, puede ser modificado cada vez que presente cambios |
| Beneficios | Beneficios que representa el acuerdo para la Unidad o para la universidad |
| vigencia | Fecha a la cual deberá cumplirse o hasta la cual será valido |

Tabla 50: Datos necesarios para dar de alta un acuerdo nuevo en el sistema EUSI

Si los datos ingresados son correctos, se almacenarán en la base de datos dando click en el botón **Guardar**, o si se prefiere abortar la operación y regresar al menú principal de esta opción se presionará el botón **Cancelar**.

Modificar

En caso de haber cometido algún error en la captura de los datos, o simplemente en caso de cambio, se pueden modificar los datos que identifican a cualquier acuerdo mediante esta opción, una vez presionado este botón se deberá digitar en la primer casilla el número de identificación que se quiere modificar para que el sistema nos presente la información correspondiente, los únicos botones disponibles en esta opción serán **Guardar** y **Cancelar**.

Una vez localizado el acuerdo se procederá a modificar la información que sea necesaria; si los datos ingresados son correctos, se almacenarán en la base de datos dando click en el botón **Guardar**, o si se prefiere abortar la operación y regresar al menú principal de esta opción se presionará el botón **Cancelar**.

Consultas

Si lo único que se requiere es una revisión rápida de los datos de algún acuerdo registrado se da click en este botón, con lo que aparecerán del lado derecho de la ventana el botón **Buscar**, y una casilla de texto, en la que se deberá teclear el número de acuerdo o su vigencia.

Para poder ver la información ingresada de un acuerdo se da click en el botón **Buscar** para que el sistema localice y visualice la búsqueda.

Reportes

Si lo que se requiere es una revisión rápida de los datos de algún acuerdo registrado, además de la **impresión** de ellos, se da click en este botón, con lo que aparecerán del lado derecho de la ventana el botón **Buscar**, y una casilla de texto, en la que se deberá teclear el número de acuerdo o su vigencia.

Para poder ver la información ingresada de un acuerdo se da click en el botón **Buscar** para que el sistema localice y visualice la búsqueda.

Para poder imprimir la información consultada se deberá presionar el botón **Imprimir**.

5.2.3 Departamentos

Al dar click en esta opción se hará visible una ventana que contiene diez botones, cada botón corresponde a los ocho departamentos e instancias que conforman la Unidad de Extensión Universitaria, y a los botones **Salir** y **Regresar**, que nos permiten salir del sistema y regresar a la ventana principal de éste, respectivamente.

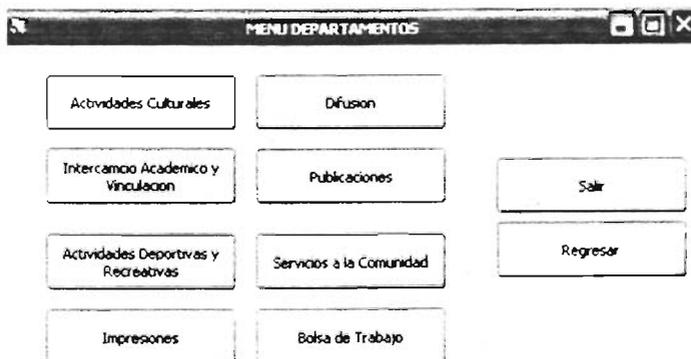


Figura 44: Ventana Opciones Departamentos

Al dar click en el botón correspondiente a cada departamento nos será permitido acceder a las opciones disponibles para estos, cabe señalar que el acceso a ellos esta restringido única y exclusivamente a los usuarios propios de un departamento, es decir, un usuario normal sólo tendrá acceso al departamento en el que labora.

Si algún usuario cambia de instancia debe reportarlo al administrador general para que lleve a cabo los cambios pertinentes a cada cuenta para que pueda acceder al departamento en que sea necesario.

5.2.3.1 Actividades Culturales

Al dar click sobre el botón con la leyenda Actividades Culturales se abrirá la ventana dedicada a las actividades de este departamento, se visualizará en pantalla algo como esto:

| id | Evento | Descripcion | Detalles | Fecha |
|-----|-----------|-------------|----------|------------|
| 001 | SHD | SDPFD | SDPFD | 22/12/2004 |
| 002 | GR | DPVDF | DPVDF | 22/12/2004 |
| 003 | LUCHPVAUT | DPDF | SCF | 11/12/2004 |
| 004 | | | | |

Figura 45: Vista Pantalla Actividades

A continuación se presentan las leyendas de los datos que se presentan en forma de casillas y el tipo de información con la que se debe rellenar cada una de ellas.

| | |
|--------------------|--|
| Id | Número de identificación del departamento, es proporcionado por el administrador general del sistema |
| Evento | Nombre del evento |
| Descripción | Tipo de evento, muestra, simposium, ciclo, congreso, obra de teatro, etc. |
| Detalles | Requerimientos especiales |
| Fecha | Fecha de realización |

Tabla 51: Datos necesarios para las actividades culturales en el sistema EUSI

Las opciones de este departamento son las mismas que en las anteriores, pues se busca uniformidad en los procesos, por ello las actividades que se pueden realizar están dadas en función de los botones de acción disponibles, dichos botones serán mencionados a continuación.

Alta

Nos da la posibilidad de agregar un evento al sistema, al dar click sobre este botón se habilitan los botones **Guardar** y **Cancelar**, es decir que serán las únicas opciones disponibles, en tanto que los otros botones se bloquearán.

Una vez que elegimos esta función podremos ingresar los datos necesarios para dar de alta un evento nuevo. La tabla anterior explica como se deben llenar los campos o casillas de texto.

Si los datos ingresados son correctos, se almacenarán en la base de datos dando click en el botón **Guardar**, o si se prefiere abortar la operación y regresar al menú principal de esta opción se presionará el botón **Cancelar**.

Modificar

En caso de haber cometido algún error en la captura de los datos, o simplemente en caso de cambio, se pueden modificar los datos que identifican a cualquier evento mediante esta opción, una vez presionado este botón se deberá digitar en la primer casilla el número de identificación que se quiere modificar para que el sistema nos presente la información correspondiente, los únicos botones disponibles en esta opción serán **Guardar** y **Cancelar**.

Una vez localizado el evento se procederá a modificar la información que sea necesaria; si los datos ingresados son correctos, se almacenarán en la base de datos dando click en el botón **Guardar**, o si se prefiere abortar la operación y regresar al menú principal de esta opción se presionará el botón **Cancelar**.

Consultas

Si lo único que se requiere es una revisión rápida de los datos de algún evento registrado se da click en este botón, con lo que aparecerán del lado derecho de la ventana el botón **Buscar**, y una casilla de texto, en la que se deberá teclear el número de evento o su fecha.

Para poder ver la información ingresada de un evento se da click en el botón **Buscar** para que el sistema localice y visualice la búsqueda.

Alta y planeación

Modificar

Consultas

Reportes

Buscar

Datos

id: 001

Evento: SHD

Descripción: SDFSD

Detalles: SDFSD

Fecha: 22/12/2004

Buscar Cerrar

Regresar

« « Culturales » »

| id | Evento | Descripción | Detalles | Fecha |
|-------|-----------|-------------|----------|------------|
| ▶ 001 | SHD | SDFSD | SDFSD | 22/12/2004 |
| 002 | GH | DFVDF | DFVSF | 23/12/2004 |
| 003 | IUDHFVUYI | DFDF | SDF | 11/12/2004 |
| 004 | | | | |

Figura 46: Pantalla Consultas

Reportes

Si lo que se requiere es una revisión rápida de los datos de algún evento registrado, además de la **impresión** de ellos, se da click en este botón, con lo que aparecerán del lado derecho de la ventana el botón **Buscar**, y una casilla de texto, en la que se deberá teclear el número de evento o su fecha.

Para poder ver la información ingresada de un evento se da click en el botón **Buscar** para que el sistema localice y visualice la búsqueda.

Para poder imprimir la información consultada se deberá presionar el botón **Imprimir**.

A partir de este punto, las tareas se vuelven totalmente repetitivas, pues al ser un formato homogéneo para todas las áreas involucradas, el proceso es

idéntico para cada tarea del sistema, para fines prácticos no se incluyen las secciones del manual pertenecientes al resto de los departamentos.

Por obviedad todos estos procesos serán incluidos en el manual de usuario y se verán a fondo en el proceso de capacitación de personal.

5.2.4 Reportes

Para acceder a esta opción hay que **dar click** en la opción con la leyenda **Reportes** de la ventana principal del sistema; una vez hecho esto, se visualizará una ventana en la que se deberá seleccionar el departamento del que se desea obtener un informe.

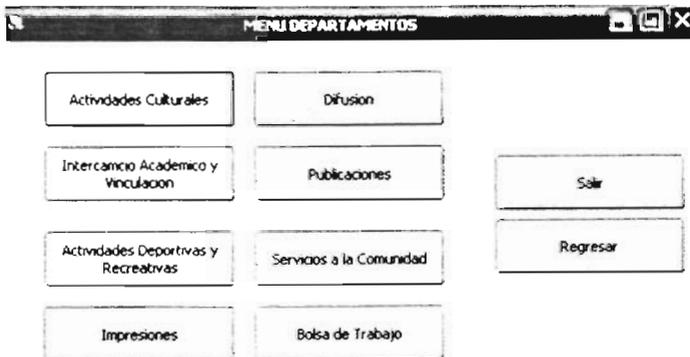


Figura 46: Pantalla Departamentos

Al **dar click** en cualquiera de los departamentos, se nos presentará la ventana del departamento elegido; esta parte del sistema es una copia al

carbón de la opción reportes de cada uno de los departamentos, la única diferencia sustancial es el hecho de que desde aquí no se puede hacer ninguna operación que no sea imprimir, cancelar o regresar.

The screenshot shows a web-based report interface. On the left is a vertical menu with options: 'Cancelar', 'Reportes', and 'Imprimir'. The main area contains a search form with five 'Label1' fields containing the values: '001', 'SHD', 'SDFSD', 'SDFSD', and '22/12/2004'. To the right is a 'Buscar' (Search) dialog box with a 'Datos' dropdown menu, 'Buscar' and 'Cerrar' buttons, and a 'Regresar' button below it. At the bottom center are navigation buttons: a left arrow, the text 'Culturales', and a right arrow. Below the search area is a table with the following data:

| id | Evento | Descripción | Detalles | Fecha |
|-----|-----------|-------------|----------|------------|
| 001 | SHD | SDFSD | SDFSD | 22/12/2004 |
| 002 | GH | DFVDF | DFVSF | 23/12/2004 |
| 003 | IUDHFVUYI | DFDF | SDF | 11/12/2004 |
| 004 | | | | |

Figura 47: Ventana de Reportes

El proceso de búsqueda de la información es idéntico al que se utiliza en cualquier departamento, de tal forma que la explicación de cómo utilizar las herramientas de esta sección ya se ha repasado en los puntos anteriores.

Esta modalidad se creó pensando en el hecho de que habrá personas que solo tengan acceso a las tablas como consultores, ya sea con o sin derecho a imprimir, por lo que es preferible que no tengan acceso al resto de las tablas ni a las opciones que de ellas se deriven.

Una vez más, al ser demasiado repetitivo este proceso, decidimos no incluirlo en este escrito, pero es claro que en la elaboración de el manual de usuario es menester su inclusión.

Mediante la lectura de este manual, se puede decir que el usuario será capaz de realizar cualquier tarea que se le asigne en el sistema, esto reforzado por el uso cotidiano del mismo, el apoyo de su instructor o de sus mismos compañeros de trabajo.

En cuanto al mantenimiento, se puede concluir que el programa sugerido abarca en su totalidad los aspectos que podrían generar problemas; por lo que si se sigue con la periodicidad sugerida no debe haber mayor complicación durante el ciclo de vida del sistema, salvo las generadas por una utilización errónea o una falla total del equipo en el que se encuentre alojado.

Conclusiones

Conclusiones

A la luz de la información que, de manera desinteresada, nos proporcionó la dirección de la Unidad de Extensión Universitaria con relación a sus procesos de trabajo, nos permitió tener una visión general del entorno, aunque sus procesos actuales son buenos y en base a la filosofía de que todo es perfeccionable, nos permitimos presentar algunas sugerencias.

Las áreas cuentan con diferentes procesos manuales inclusive, que se podrían optimizar y manejarse por un sistema de cómputo integrado facilitando los procesos.

Como resultado de las consultas obtenidas con el personal de las diferentes áreas y en base a sus necesidades resolvimos que el esquema cliente-servidor era la mejor opción, así como el uso de Visual Basic y Access, debido a su infraestructura.

Además, la etapa de análisis y diseño conceptual nos brindó herramientas que no sólo son aplicables a este sistema y se podrían utilizar en lo futuro para la implementación del sistema Web en cuanto su infraestructura se lo permita.

Automatizando los procesos manuales de la Unidad, nos encontramos con diversas ventajas como: información más detallada, unificación de procesos, evitar duplicidad de trabajo y la extracción de reportes finales, entre otros.

Actualmente la Unidad de Extensión Universitaria necesita estar actualizada en cuanto al archivo de los diferentes eventos que se realizan dentro de ella, y estamos seguros que nuestro sistema de información EUSI es la mejor opción en este momento, para mantenerla a la altura de las circunstancias.

Con base en todo esto, concluimos que nuestro sistema EUSI cumple con su objetivo y logrará grandes beneficios en el plantel facilitando las labores del área cultural y deportiva de la Facultad de Estudios Superiores "Aragón". Lo que nos permite demostrar nuestro tema de tesis.

Bibliografía

Bibliografía

- **Berntein, Douglas A.** "Real Time UML ".Pearson. Julio de 20044.
- **Conallen, J.** "Modeling Web Applications with UML". Ed. Conallen. 1994
- **Fowler, Martin.** "UML gota a gota". Pearson. 2003
- **Greiff W. R.** "Paradigma vs Metodología; El Caso de la POO (Parte II)". *Soluciones Avanzadas*. Ene-Feb 1994.
- **Jacobson.** "Applying UML in The Unified Process" Presentación. Rational Software. 1998.
- **Jacobson.** "Object-Oriented Software Engineering; A Use Case Driven Aproach. ACM Press". Adison-Wesley Publishing. 1992.
- **Larman, Graig.**"UML y patrones introducción al análisis". Pearson. 2003.
- **Lewis G.** "What is Software Engineering?" . DataPro (4015). Feb 1994.
- **Lopez, Nathalie.**" Integrar UML en los proyectos". Gestion 2000. 2003.
- **Microsoft 1997.** "Microsoft Solutions Framework 1.0.". Microsoft Corporation. USA. 1997.

- **Microsoft y Rational.** "A White Paper on the Benefits of Integrating Microsoft Solutions Framework and The Rational Process". Rational Software Corporation y Microsoft Corporation. 1998.
- **Muller.** "Modelado de objetos con UML". Gestion 2000. 2003.
- **Object Management Group.** "OMG Unified Modeling Language Specification". Versión 1.3. alfa R5. Addison-Wesley. Marzo de 1999.
- **Schmuller, Joseph.** "Aprendiendo UML en 24 horas". Pearson. 2004
- **Stevens, Perdita.** "Utilización De UML en Ingeniería De Software". Pearson. 2004

Fuentes

- http://www.4clerks.com/investigacion_y_desarrollo_para_la_generacion_de_soluciones_informaticas_.htm. **2004**
- <http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html#IngSoft>. **2005**
- http://www.artech.com.uy/gxdlsp/pub/genexus/csharp/docum/releasenotes/8.0/aplicacioneswintrescapas.htm#_toc63238102. **2003**
- <http://www.conalien.com/whitepapers/webapps/ModelingWebApplications.htm>. **2005**
- <http://www.elguille.info>. **2004**

- <http://www.geocities.com/trescapas/introduccion.htm>. **2005**
- http://www.inei - 1_2 alcance.htm. **2004**
- <http://www.madrimasd.org>. **2004**
- <http://www.microsoft.com/spanish/msdn/arquitectura/das/distapp.asp>. **2005**
- <http://www.mug.org.ar/FoxProGufa/ArticFox/default.aspx>. **2004**
- <http://www.nosolousabilidad.com/index.htm>. **2004**
- <http://www.rational.com/uml/software>. **2005**
- <http://www.rational.com/uml/tutorials>. **2005**
- <http://www.rational.com/uml/papers>. **2005**
- <http://www.recursos-as400.com/index.html>. **2005**
- <http://www.rincondelvago.com/>. **2003**

Glosario

Glosario

BOOCH: Herramienta de modelado similar a UML que, a diferencia de este, establece procesos a desarrollar en forma específica de igual forma que OMT.

EORM: (Metodología de Relaciones de Objetos Mejorada). Herramienta de modelado similar a UML

FREELANCE: Programador Independiente de cualquier compañía o empresa que realiza tareas por su cuenta.

GRANULARIDAD: "Distribuir" el desarrollo de la misma en pequeños fragmentos de código, totalmente independientes, que pueden generarse en paralelo.

HDM: (Método de Diseño Hipermedia). Herramienta de modelado similar a UML

MS DTC: (Coordinador y Distribuidor de Transacciones de Microsoft). Agente encargado de realizar operaciones en una base de datos implementada bajo Microsoft SQL.

MTS: (Servidor de Transacciones de Microsoft). Servidor implementado a nivel de software que permite realizar operaciones entre los distintos módulos de un Sistema de información.

MULTI-TIER: (Multi Capas). Modalidad de programación de aplicaciones y/o sistemas de información. Puede tener desde 2 hasta n Capas o niveles de programación.

OMT: (Técnica de de Modelado de Objetos). Herramienta de modelado similar a UML que, a diferencia de este, establece procesos a desarrollar en forma específica.

OOHDM: (Método de Diseño Hipermedia Orientado a Objetos). Herramienta de modelado similar a UML

RMM: (Metodología de Administración de Relaciones). Herramienta de modelado similar a UML

SGBD: (Sistema Gestor de Bases de Datos). Aplicación encargada de crear, modificar, eliminar y administrar numerosas bases de datos en equipos remotos o locales, manteniendo su integridad y permitiendo su estabilidad a pesar de la constante utilización.

SOHDM: (Metodología de Diseño Hipermedia orientada a objetos y basada en escenarios). Herramienta de modelado similar a UML

STATELESS: (Estático). Es el estado en que debe conservarse el MTS para evitar conservar las propiedades de cada objeto que registre.

THREE-TIER: (Tres Capas). Modalidad de programación de aplicaciones y/o sistemas de información. Consta de Capa del Cliente, Capa Intermedia, y Capa de Servidor.

UML: (Lenguaje Unificado de Modelado). Lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos (OO). Un artefacto es una información que es utilizada o producida mediante un proceso de desarrollo de software.

WAE: (Extensión de Aplicaciones Web). Herramienta de modelado similar a UML

WSDM: (Método de Diseño de Sitios Web). Herramienta de modelado similar a UML