



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Sistema para la Administración de Pagos Internos
y Externos de un Centro de Capacitación
Computacional (SAPIECC)**

T E S I S

Que para obtener el título de

Ingeniero en Computación

Presenta:

Muñoz Reverte Belem Mitzi

Director de tesis: Ing. Lucila Patricia Arellano Mendoza



MÉXICO, CIUDAD UNIVERSITARIA 2012

AGRADECIMIENTOS

A mis padres muestra de apoyo incondicional y perseverancia en todos los aspectos que conozco. Gracias por darme la oportunidad de estar aquí y siempre estar ahí.

A la UNAM por darme todas las herramientas necesarias para contribuir al avance tecnológico de mi país.

A todos mis profesores porque gracias a sus enseñanzas y conocimientos ahora puedo enfrentarme al mundo con la seguridad de que tengo la preparación de la mejor universidad de Iberoamérica.

ÍNDICE

I. PLANTEAMIENTO DEL PROBLEMA Y REQUERIMIENTOS DEL SISTEMA	1
1.1. <i>Situación Actual de la Institución</i>	1
1.2. <i>Justificación del Sistema</i>	2
1.3. <i>Requerimientos del Sistema</i>	2
1.4. <i>Restricciones del Sistema</i>	12
1.5. <i>Funciones del Sistema</i>	13
1.6. <i>Características de los Usuarios</i>	14
II. FUNDAMENTOS TEÓRICOS	15
2.1 <i>Sistemas Operativos</i>	15
2.1.1 <i>Funciones Básicas</i>	15
2.1.2 <i>Tipos de Sistemas Operativos</i>	16
2.1.3 <i>Ventajas y Desventajas de Algunos Sistemas Operativos</i>	20
2.2 <i>Ingeniería del Software</i>	21
2.2.1 <i>El Proceso del Software</i>	22
2.2.2 <i>Modelos de Proceso del Software</i>	22
2.3 <i>Planeación de un Proyecto de Programación</i>	26
2.3.1 <i>Definición del Problema</i>	26
2.3.2 <i>Desarrollo de una Estrategia de Solución</i>	28
2.3.3 <i>Planeación de un Proceso de Desarrollo</i>	29
2.4 <i>Sistema Manejador de Base de Datos</i>	32
2.4.1 <i>Arquitectura de un Sistema de Base de Datos</i>	32
2.4.2 <i>Funciones del Administrador de la Base de Datos y el Diccionario de Datos</i>	33
2.5 <i>Diseño de la Base de Datos</i>	34
2.5.1 <i>Modelo de Datos</i>	34
2.6 <i>Modelo Relacional</i>	34
2.6.1 <i>Arquitectura del Modelo Relacional</i>	34
2.6.2 <i>Modelo Conceptual</i>	35
2.6.3 <i>Modelo Lógico</i>	35
2.6.4 <i>Modelo Físico</i>	36

2.7 Lenguajes de Programación	36
2.7.1 Clasificación de los Lenguajes de Programación	37
2.7.2. Lenguajes de Programación para la Web	39
III. PROPUESTA DE SOLUCIÓN	41
3.1. Estrategias de Solución	41
3.1.1 Programación Estructurada vs Programación Orientada a Objetos.....	41
3.1.2 Modo Consola vs. Modo Gráfico	43
3.1.3 Software Libre vs. Software con Licencia Comercial	44
3.2. Estudio de Factibilidad de las Características del Sistema	45
3.3. Definición Formal de la Estrategia a Utilizar	48
3.4. Prioridades para las Características del Sistema.....	49
3.5. Criterios de Aceptación del Sistema.....	49
IV HERRAMIENTAS A UTILIZAR.....	50
4.1. Sistema Manejador de Bases de Datos.....	50
4.1.1 Características Principales	50
4.1.2 Ventajas y Desventajas	52
4.2. Lenguajes de Programación	53
4.2.1 Características Principales	53
4. 2.2 Ventajas y Desventajas	55
4.3. Herramientas Seleccionadas	56
V. ANÁLISIS DEL SISTEMA	58
5.1. Planeación y Definición de Requisitos.....	58
5.1.1 Plan de Proyecto.....	58
5.2. Definición de Requisitos	61
5.2.1 Modelado de Datos	61
5.2.2 Modelado Funcional	62
5.2.3 Modelado de Comportamiento	64
5.2.4 Especificación de Requisitos del Sistema.....	65
VI. DISEÑO	69
6.1 Diseño de Datos	69
6.2 Diseño Arquitectónico	71
6.3 Diseño Detallado	72

6.4 Instrumentación	81
6.5 Plan de Pruebas de Aceptación	93
VII. PRUEBAS Y MANTENIMIENTO	94
7.1 Tipos de Pruebas.....	94
7.1.1 Pruebas de Unidad	95
7.1.2 Pruebas de Integración	124
7.1.3 Pruebas de Validación	127
7.1.4 Pruebas de Sistema	131
7.2 Administración de la Configuración	132
7.3 Métricas Técnicas del Software	134
7.3.1 Métricas del Modelo de Análisis	134
7.3.2 Métricas del Modelo de Diseño	137
7.3.3 Métricas del Código Fuente	139
7.3.4 Métricas para las Pruebas.....	140
7.3.5 Métricas del Mantenimiento	140
CONCLUSIONES.....	142
DICCIONARIO DE DATOS DE SAPIECC	143
MANUAL DE USUARIO	158
BIBLIOGRAFÍA	184

I. PLANTEAMIENTO DEL PROBLEMA Y REQUERIMIENTOS DEL SISTEMA

1.1. Situación Actual de la Institución

Un Centro de Capacitación Computacional desea desarrollar un sistema informático para la administración de los cursos que imparte en cuanto a logística y pago. En este momento la organización del centro posee las siguientes características:

Los clientes se dividen en dos grandes grupos: clientes físicos y clientes morales. Los clientes físicos son personas que generalmente llegan a inscribirse individualmente; para lo cual se les solicitan datos como: nombre, dirección, teléfono, e-mail, y RFC. Una vez que se registran estos el sistema asigna una matrícula que los identificará de manera única durante toda su estancia en el centro.

En el caso de los clientes morales o empresas, generalmente se tiene un contacto que proporciona la razón social de la Empresa, su dirección fiscal, teléfono y RFC. De igual forma que los clientes físicos cada cliente moral tiene una clave única asignada. Además el contacto envía una lista que incluye a todos los colaboradores que tomarán cursos en el centro. Cada uno de esos empleados al ser registrado en el sistema debe contar con una matrícula única, su nombre y la clave de la empresa a la que pertenece.

Una vez que se tienen almacenados los datos personales de los clientes se procede a la asignación del curso. Existen seis tipos de cursos: semi-intensivo 90 min, semi-intensivo 60 min, intensivo, sabatino, mensual y SNE (Servicio Nacional de Empleo). Además del tipo de curso se asigna el nombre de curso o concepto, el nivel, el periodo y el horario.

Para cada uno de los cursos tomados por los distintos tipos de clientes se lleva un control de pago el cual cambia de acuerdo al cliente. En el caso de clientes físicos se extiende una nota de pago que se identifica por un número de nota y que contiene los datos del cliente, el curso pagado, el monto y la fecha de pago. Mientras que para las empresas o clientes morales el registro de pago se lleva a través de facturas, las cuales tienen un número de factura, los datos de la empresa, la fecha de pago, los datos de los cursos pagados y el monto total pagado con su IVA desglosado. Es importante mencionar que a diferencia de las notas de pago las facturas se utilizan como un comprobante fiscal.

Los grupos se van formando de acuerdo al número de alumnos inscritos en un mismo tipo de curso del mismo nivel en un determinado horario y periodo.

Cada grupo tiene un sólo profesor asignado y un profesor sólo puede tener un grupo asignado en un determinado horario y periodo. De acuerdo al número de grupos asignados a un profesor en un determinado periodo es posible calcular el número de horas trabajadas por este facilitando considerablemente el cálculo de su nómina.

1.2. Justificación del Sistema

El Sistema para la Administración de Pagos Internos y Externos de un Centro de Capacitación Computacional (SAPIECC) tiene como principal objetivo mejorar la organización tanto administrativa como financiera del centro.

Pretende reducir tiempos, al sustituir el uso de varias aplicaciones por una sola herramienta integral que no sólo lleve un mejor control de la información al mantenerla almacenada en una sola base de datos sino que contará con módulos que permitirán realizar de manera automática el alta de clientes de distintos tipos, la asignación de cursos, la generación de pagos y consultas especiales. Además el sistema permitirá la impresión de distintos comprobantes de pago y será capaz de generar reportes que en la actualidad requieren del análisis e interpretación cuidadosa de los coordinadores del centro ya que el sistema tiene como uno de sus objetivos que tales reportes no requieran de la intervención de los usuarios más que para definir intervalos de los datos que pretenda reportar. El sistema incluirá también un módulo especial para asignar profesores de manera automática a los grupos facilitando el cálculo de las horas trabajadas por cada uno de los docentes. Además este mismo módulo permitirá ajustes de parte de los usuarios.

El sistema evitará que se generen varios libros de trabajo permitiendo que en la misma base de datos se almacenen los periodos de más de un año como se hace actualmente con los archivos que se utilizan. Esto hará posible mediante consultas especiales realizar comparativos distintos no sólo entre periodos, como se realiza en la actualidad, sino comparando información de varios años. Permitiendo observar así el crecimiento real del centro a través del tiempo.

Se propone que el sistema sea amigable y de interfaz gráfica permitiendo llevar un orden en las acciones que se realizan para ejecutar diferentes procedimientos, esto quiere decir que cuando un cliente llega el sistema se va enlazando a través de botones a la información que el sistema va requiriendo almacenar y que van desde capturar los datos generales del cliente, elegir un curso, generar el pago y asignar un grupo. Sin embargo también el sistema tiene la opción de entrar directamente en algún módulo en particular por ejemplo al realizar un pago.

El sistema contará también con un sistema de ayuda o tutorial, el cual tendrá como objetivo orientar a los usuarios en el uso del sistema y aclarar cualquier duda que pudiera llegar a surgir durante el uso del mismo. Esta opción estará disponible mediante el módulo de ayuda.

1.3. Requerimientos del Sistema

Antes de que el software se pueda construir es importante comprender el sistema en el que residirá. Para lograrlo se deben definir los objetivos generales; esto significa que es necesario identificar el papel del hardware, software, personas, bases de datos, procedimientos y otros elementos del sistema; así como los requerimientos operacionales deben ser identificados; analizados, especificados, modelados, validados y gestionados. De acuerdo a eso a continuación se describen

las herramientas y procedimientos que utiliza el centro de capacitación computacional actualmente para organizar la logística y administración de los cursos que imparte.

Para llevar el control de los cursos impartidos en el centro actualmente se cuenta con dos herramientas: el ICS (Inform Counted Students, Informe de Conteo de Estudiantes) y el Programing (Programación). El modo en que opera cada uno de ellos se describe a continuación.

El ICS consiste en archivo de Excel formado por 24 hojas de cálculo, dos por mes, correspondientes a los periodos en que se inician cursos en la institución. En la figura 1.1 se muestra como luce dicho archivo.

Como se puede observar cada hoja de cálculo se encuentra dividida en seis secciones que corresponden a los distintos tipos de cursos que imparte el centro y que se identifican como: semi-intensivos 90 min, semi-intensivos 60 min, intensivos, sabatinos, mensuales y Servicio Nacional de Empleo (SNE). Las principales características de cada uno de estos tipos de curso son:

- Semi-intensivos 90 min
 - ❖ Avance: 1 nivel cada 2 semanas.
 - ❖ Duración: 1 hora 20 minutos de clase.
 - ❖ De lunes a viernes.
 - ❖ Horarios: 7:00 a 8:20 a.m., 8:30 a 9:50 a.m., 10:10 a 11:30 a.m., 11:30 a 1:00 p.m., 3:00 a 4:20 p.m. 4:30 a 5:50 p.m., 6:10 a 7:30 p.m., 7:40 a 9:00 p.m.
- Semi-intensivos 60 min:
 - ❖ Avance: 1 nivel cada 3 semanas.
 - ❖ Duración: 55 minutos de clase.
 - ❖ De lunes a viernes.
 - ❖ Horarios: 6:00 a 6:55 a.m., 1:00 a 1:55 p.m., 2:00 a 2:55 p.m. y 3:00 a 3:55p.m.
- Intensivos
 - ❖ Avance: 1 nivel cada semana.
 - ❖ Duración: 2 horas 40 minutos de clase.
 - ❖ De lunes a viernes.
 - ❖ Horarios: A elegir entre 7:00 a.m. y 9:00 p.m.

- Sabatinos
 - ❖ AM
 - Avance: 1 nivel cada 2 sábados.
 - Duración: 6 hrs. de clase cada sábado.
 - Horario: 8:30 A.M. a 14:30 P.M
 - ❖ PM
 - Avance: 1 nivel cada 4 sábados.
 - Duración: 3 hrs. de clase cada sábado.
 - Horario: 3:00 P.M. a 6:00 P.M
- Mensuales
 - ❖ Avance: 1 nivel cada 4 semanas.
 - ❖ Duración: 1 hora 20 minutos de clase.
 - ❖ Martes y jueves.
 - ❖ Horarios: 7:00 a 8:20 a.m., 8:30 a 9:50 a.m., 10:10 a 11:30 a.m., 11:30 a 1:00 p.m., 3:00 a 4:20 p.m. 4:30 a 5:50 p.m., 6:10 a 7:30 p.m., 7:40 a 9:00 p.m.
- Servicio Nacional de Empleo(SNE)
 - ❖ Avance: 1 nivel cada semana.
 - ❖ Duración: 2 horas 40 minutos de clase.
 - ❖ De lunes a viernes.
 - ❖ Horarios: A elegir entre 7:00 a.m. y 9:00 p.m.

Cabe resaltar que los cinco primeros cursos se encuentran abiertos tanto a clientes físicos como morales. Sin embargo en lo que se refiere a Servicio Nacional de Empleo (SNE) consiste en un acuerdo que se tiene con las diferentes delegaciones del Distrito Federal. Así que sólo se manejan en los periodos de marzo a septiembre, en diferentes horarios de acuerdo a las necesidades de cada delegación. Aunque es importante considerar que cada contrato que se consigue de este tipo consta de 10 semanas ininterrumpidas de cursos en ese horario.

Además de las secciones correspondientes a los tipos de curso el ICS consta de una sección de Totales que se encuentra en la parte inferior izquierda y que únicamente tiene el objetivo de brindar estadísticas generales. Todas las hojas de ICS tienen exactamente el mismo formato y lo único en lo que difieren es el nombre que corresponde al periodo quincenal que está organizando.

Para cada periodo las secciones correspondientes a los tipos de cursos se encuentran divididas en los horarios en que se puede impartir y a su vez cada horario está dividido en tres secciones rotuladas en la parte inferior de cada horario. Dichas secciones son: Gr (Group, Grupo), N/X (New X, Nuevos Ingresos-Reingresos) y SS (Student Status, Estatus de Estudiantes), las cuales se muestran en la figura 1.2.

Gr	N/X	SS	Gr	N/X	SS
0	0	0	0	0	0

Figura 1. 2 Secciones de ICS

Gr o Grupo hace referencia al nombre del curso que se impartirá por ejemplo Word, Excel, etc. N/X o Nuevos Ingresos-Reingresos recibe su nombre debido a que tiene la función de colocar el número de ingresos nuevos a ese curso. Sin embargo un alumno se considera nuevo en tres casos: cuando se inscribe por primera vez, cuando se dio de baja durante más de dos periodos consecutivos y cuando se realiza un cambio de plantel. Por esta razón sus siglas son N / X y no N/ I. Finalmente SS ó Student Status indica el número total de alumnos en ese grupo en ese horario contabilizando los nuevos ingresos además de los alumnos que únicamente cambian de nivel o de horario.

Cada vez que un prospecto de alumno regular o nuevo cliente físico se presenta al centro a pedir informes sobre los próximos cursos disponibles el coordinador en turno revisa la hoja del ICS que corresponde al periodo de interés del cliente y observa todos los cursos que se impartirán así como los horarios y su tipo para brindar la información al prospecto. Cabe subrayar que la información que se observa en el ICS es colocada de manera manual, de acuerdo a la demanda de los cursos así como a la continuidad que tienen algunos paquetes considerados como diplomados. Si al cliente le interesa algún curso se le pide que llene una papeleta con información general tal como nombre, dirección, teléfono y la forma en la cual se enteró de la institución para que posteriormente la gente encargada de ventas pueda dar de alta al alumno utilizando módulos programados en el sistema AS400 el cual además de almacenar sus datos le proporciona una matrícula única. Una vez que se tiene la matrícula se proporciona una ficha de pago para realizar este en caja. Enseguida el coordinador debe actualizar manualmente el ICS contabilizando el nuevo ingreso en la celda N / X correspondiente y colocando un comentario en la celda del estatus de estudiantes SS en donde indica la matrícula del cliente y el monto del pago que se realizó, ya que este puede ser total o parcial. El comentario de la celda N / X siempre debe contener las matrículas de los alumnos de cada curso ya que servirá de ayuda para el coordinador al permitirle ubicar a los nuevos ingresos de cada curso que se abre.

Por otro lado cada fin de periodo el coordinador debe proporcionar fichas de pago a los alumnos que ya están inscritos en algún curso para continuar con el nivel siguiente y cada que uno de ellos le confirme su pago debe actualizar la columna SS del horario y nuevo curso correspondiente.

Para el caso de los alumnos morales el área de ventas debe comunicarse con el contacto para obtener los datos fiscales exactos de la empresa así como la relación de empleados que tomarán la capacitación ya que el vendedor encargado deberá programar las evaluaciones a los miembros de empresa. Una vez que se tienen los resultados de tales evaluaciones el vendedor hace una propuesta al cliente de los grupos, los horarios y la fecha de inicio de los cursos. Esta propuesta debe ser consultada con el coordinador ya que él debe preparar la logística para cumplir con todos sus compromisos físicos y morales. Si este la aprueba y el cliente está de acuerdo se firma el contrato y se inicia. Si no se modifica hasta que se llegue a un acuerdo para ambos. En el caso de una empresa o cliente moral el coordinador coloca en el ICS en la sección de Gr el nombre del curso seguido del nombre de la Empresa. Esto para cada uno de los cursos que se impartirán tanto en las instalaciones del centro como fuera de ellas. Es importante que el área de ventas consiga un documento con los nombres de todo el personal que tomará la capacitación ya que él será el encargado de darlos de alta en el sistema AS400.

Cada que se agrega un curso nuevo en cualquier horario las celdas que se encuentran debajo de los rótulos de Gr, N/X, SS. Se van modificando ya que tienen fórmulas que van contabilizando o

sumando las cantidades de cada horario con el fin de obtener estadísticas por ejemplo debajo de Gr se muestra el total de cursos abiertos en ese horario, bajo N/X se muestra el total de alumnos nuevos del horario y finalmente en SS el total de alumnos (inscritos + reinscritos) del horario. Estas estadísticas también son útiles para llenar las celdas del área de totales.

Podemos observar que esta sección consta de cinco columnas:

- Curso: Clasificación que tiene el centro de acuerdo a la demanda de los cursos que imparte.

Totales				
Curso	Nv Ing	Tt Alu	Grups	Pro Al
Pico	0	0	0	0,00
N Pico	0	0	0	0,00
1ª Hora	0	0	0	0,00
Sb Am	3	3	1	3,00
Sb Pm	0	0	0	0,00
INT 1a	22	22	1	22,00
INT 2a	22	22	2	11,00
Mensual	0	0	0	0,00
SNE	0	0	0	0,00
Total	47	47	4	11,75
Poblacion Anterior				
SNE		0		
Total de alumnos		47		

Figura 1.3 Totales ICS

- ❖ Pico. Considera los cursos de cualquier tipo que se encuentran dentro de los siguientes horarios: 8:30 a 9:50 a.m., 10:10 a 11:30 a.m., 6:10 a 7:30 p.m., 7:40 a 9:00 p.m.
 - ❖ N Pico. Considera los cursos de cualquier tipo que se encuentran dentro de los siguientes horarios: 7:00 a 8:20 a.m., 11:30 a 1:00 p.m., 3:00 a 4:20 p.m. 4:30 a 5:50 p.m.
 - ❖ 1ª Hora. Considera todos los cursos de una hora que se abrieron en el periodo.
 - ❖ Sb Am. Abarca todos los cursos sabatinos AM del periodo.
 - ❖ Sb Pm. Considera todos los cursos sabatinos PM del periodo.
 - ❖ INT 1a. Corresponde a los cursos intensivos de la primera semana del periodo.
 - ❖ INT 2a. Considera todos los cursos intensivos de la segunda semana del periodo.
 - ❖ Mensual. Corresponde a los cursos mensuales del periodo.
 - ❖ SNE. Abarca los cursos del Servicio Nacional de Empleo del periodo.
- Nv Ing. Muestra el total de nuevos ingresos (Suma de N/X).
 - Tt Alu. Total de alumnos, despliega el total de alumnos (Suma de SS) .
 - Grups. Muestra el total de grupos abiertos (Suma de Gr).
 - Pro Al. Promedio de Alumnos, Grups / Tt de Alu

Todas estas cantidades se actualizan automáticamente mediante fórmulas que ya se encuentran incluidas en todas las hojas de cálculo.

La información contenida en la hoja del ICS de cada periodo debe estar lista a más tardar el día que termina el periodo anterior. Esto debido a que es la base para que el coordinador realice la programación de cursos y profesores.

Una vez completo el ICS el coordinador hace uso del archivo denominado programing para llevar un control de la logística de los cursos. El programing se maneja de forma semanal y tiene un aspecto como el que se muestra en la figura 1.4.

Como podemos observar en la figura 1.4 cada periodo que integra el programing o programación consiste en una hoja de cálculo. Todos los programing de un año se encuentran almacenados en el mismo libro de trabajo siguiendo un orden cronológico. Debido a esto cuando el coordinador se dispone a realizar la programación de un nuevo periodo lo primero que debe hacer es colocar en la parte superior derecha las fechas correspondientes al periodo que está organizando.

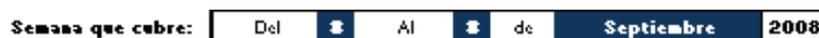


Figura 1. 5 Periodos Programing

Posterior a esto se inicia con el llenado del formato. El programing está formado por distintas secciones, la más amplia que podemos observar es la que abarca el rotulo Lunes a Viernes, la cual contiene subsecciones que corresponden a los horarios de mayor demanda en el centro. Cada horario se encuentra dividido en tres secciones: Gr (Grupo), NI (Nuevos Ingresos) y Al (Alumnos Totales). Estas celdas tienen el objetivo de almacenar los mismos datos que provienen del ICS tales como Grupo, Nuevos Ingresos y Alumnos Totales. Esta división también se puede observar en la sección inferior izquierda que corresponde a los horarios sabatinos en su modalidad AM y PM.

Sábado	1, 15	de	Noviembre					
8:30		11:40			15:00			
Gr	NI	Al	Gr	NI	Al	Gr	NI	Al

Figura 1. 6 Opciones Programing

Debido a que la función primordial del programing es distribuir los cursos disponibles entre los diferentes salones y profesores. En la primera columna del lado izquierdo podemos observar los nombres de los profesores del centro y su disponibilidad. Por facilidad se encuentran divididos en tres secciones AM ó matutino, Mid ó intermedio y PM ó Vespertino. También se puede observar que aparecen los nombres de los coordinadores del plantel y los de los profesores disponibles en sábado, los cuales por política deben ser todos, aunque esto varía dependiendo de la demanda que se tenga en ese periodo.

Para empezar a llenar el programing el coordinador inicia por los cursos que en el ICS tienen un número total de alumnos mayores a tres, el resto se traslada para el siguiente periodo salvo casos especiales en los que se trate de un curso individual o cuando el curso ya se haya retrasado más de una vez. Es obligación del coordinador trasladar los datos de tales cursos al siguiente periodo del ICS y de la gente de ventas ponerse en contacto con el cliente para informarle el retraso del curso. Mientras que para los cursos que si se pueden abrir el coordinador debe asignarlo a alguno de los profesores disponibles en el horario que debe impartirse colocando el nombre del curso seguido del salón en donde se impartirá en la celda Gr que corresponde a la intersección del horario y profesor que se desea asignar. Así también se coloca el número de ingresos nuevos y el total de alumnos, con el objetivo de brindar al profesor apoyo en cuanto a la magnitud del grupo que va a manejar.

De esta forma debe ir cubriendo todos los cursos que se abrieron con los profesores y salones disponibles. Es importante subrayar que cada que se asigna un grupo a un profesor en la celda a la derecha del nombre del profesor se coloca de forma automática la cantidad de horas trabajadas por él, esto debido a que ya se tiene una fórmula para realizar la operación correspondiente. Es deseable que todos los profesores tengan en promedio el mismo número de horas trabajadas. Sin embargo esto no es siempre posible ya que no todos los profesores manejan todos los cursos. También se debe considerar que sólo se tienen 7 salones habilitados por lo que no se deben abrir más de siete cursos en un mismo horario, además de que en promedio sólo se tienen cinco profesores por turno por lo que el coordinador debe considerar que se deben cubrir todos los cursos con los recursos humanos y materiales disponibles. Toda esta asignación se lleva a cabo de forma manual comparando el ICS con el Programing hasta que estos coincidan en los cursos abiertos y asignados.

Finalmente de igual forma que el ICS el Programing consta de una sección denominada conteo.

C O N T E O				
Curso	Grupos	Ingresos	Alms	Pro Al
PICO	4	12	20	5,00
NO PICO	0	0	0	0,00
1 HORA	0	0	0	0,00
ST A.M.	3	6	12	4,00
ST P.M.	0	0	0	0,00
Intensivo	0	0	0	0,00
Especial	0	0	0	0,00
Personal	0	0	0	0,00
Personal INT	0	0	0	0,00
Total	7	18	32	9,00

Figura 1. 7 Conteo Programing

Esta tabla se llena de manera automática ya que contiene fórmulas que nos generan de forma inmediata los siguientes resultados:

- Curso: Clasificación que tiene el centro de acuerdo a la demanda de los cursos que imparte.

- ❖ Pico. Considera los cursos de cualquier tipo que se encuentran dentro de los siguientes horarios: 8:30 a 9:50 a.m., 10:10 a 11:30 a.m., 6:10 a 7:30 p.m., 7:40 a 9:00 p.m.
- ❖ N Pico. Considera los cursos de cualquier tipo que se encuentran dentro de los siguientes horarios: 7:00 a 8:20 a.m., 11:30 a 1:00 p.m., 3:00 a 4:20 p.m. 4:30 a 5:50 p.m.
- ❖ 1ª Hora. Considera todos los cursos de una hora que se abrieron en el periodo.
- ❖ Sb Am. Abarca todos los cursos sabatinos AM del periodo.
- ❖ Sb Pm. Considera todos los cursos sabatinos PM del periodo.
- ❖ INTENSIVO. Corresponde a los cursos intensivos del periodo.
- ❖ Especial: Corresponde a los cursos de horarios especiales del periodo.
- ❖ Personal. Abarca los cursos orientados a una sola persona.
- ❖ Personal Int. Abarca los curso de intensivos para una sola persona.
- Grupos. Muestra el total de grupos abiertos (Suma de Gr).
- Ingresos. Muestra el total de nuevos ingresos (Suma de NI).
- Alms. Total de alumnos, despliega el total de alumnos (Suma de Al).
- Pro Al. Promedio de Alumnos por grupo, Alms / Grupos

Aunque el llenado de esta sección es automático el coordinador debe asegurarse que corresponda numéricamente con la información del ICS.

Cuando este archivo es terminado se coloca en la sala de profesores ya que les servirá de horario, debido a que pueden observar junto a su nombre que cursos van a impartir, donde y a cuantos alumnos. Por esta razón es importante que también se encuentre listo antes de que inicie el periodo.

El programing semanal de cada periodo también se enviara a la administración ya que en base a las horas trabajadas que se reflejan en él se realizara el cálculo la nómina de los profesores y coordinadores. Aunque el coordinador siempre tiene que notificar si hay ajustes a este documento debido a faltas, retardos, etc. Esto se hace platicando directamente con el personal del área de administración quienes utilizan NOI para estos ajustes y todo lo relacionado con la nómina.

Finalmente la información concentrada en los cuadros de totales del ICS y conteo del Programing son enviados a la dirección para que esta pueda realizar un comparativo respecto a los ingresos y alumnos totales en cada periodo. Generando así nuevas estrategias ó bien informes sobre los incrementos o perdidas del centro. Toda esta información se discute y estudia en las juntas corporativas de la institución.

Esta es la forma en que actualmente se resuelve la problemática expuesta en el tema anterior. Sin embargo con el desarrollo del Sistema para la Administración de Pagos Internos y Externos de un Centro de Capacitación Computacional se pretende cambiar los métodos y herramientas para realizar estos movimientos. Aunque es importante que los datos de entrada para el sistema y las restricciones así como las salidas de datos sean las mismas que se obtienen con los métodos actuales. Todos estos detalles se establecerán de manera formal en las secciones posteriores.

1.4. Restricciones del Sistema

El Sistema para la Administración de Pagos Internos y Externos de un Centro de Capacitación Computacional (SAPIECC) tiene como principal objetivo facilitar los movimientos de logística y administración en el centro de capacitación. Sin embargo para que el sistema se considere aceptable debe cumplir con las siguientes restricciones:

- El sistema deberá ser capaz de realizar la captura y el procesamiento de la información sin hacer uso de elementos de software externo.
- El sistema deberá tener la capacidad de realizar cálculos matemáticos y estadísticos de manera automática mediante la programación de rutinas.
- El sistema deberá ser amigable y fácil de utilizar por el usuario.
- El sistema deberá contar con un manual de usuario además, una vez terminado el desarrollo del sistema se brindará capacitación para su uso.
- El sistema deberá ser confiable y eficiente, no se permitirán errores en pagos y no debe existir duplicidad en clientes o grupos asignados a profesores.
- Además deberá soportar más de 1000 registros (Promedio de alumnos inscritos anualmente considerando todos los planteles), sin problemas de consistencia y en caso de caer debe recuperarse rápidamente.
- El sistema deberá ser capaz de operar durante quince horas diarias sin interrupción. Ya que el horario laboral del centro es de 7:00 a.m. a 10:00 p.m.
- El sistema deberá producir respuestas en tiempo real.
- En caso de ser necesario el sistema deberá permitir agregar o eliminar fácilmente módulos sin afectar su unidad y funcionamiento.
- El sistema deberá funcionar correctamente en cualquier plataforma.
- El sistema deberá residir en menos de 1 GB de memoria principal y deberá emplear con eficiencia esa memoria. Esto debido a que son las características mínimas de los equipos del departamento en donde se utilizará.
- El sistema deberá ser consistente y seguro, aunque puede ser usado por todo el personal administrativo el único capaz de tener acceso al código fuente es el programador del sistema.

- El sistema deberá disminuir los tiempos para efectuar altas, pagos y obtener reportes del centro de capacitación en más de un 100%.
- El sistema deberá ser capaz de generar reportes imprimibles de manera automática sin necesidad de análisis por parte de los usuarios y sin presentar errores.

1.5. Funciones del Sistema

De acuerdo con la descripción de la situación actual de la institución así como la forma en que actualmente el centro soluciona esta problemática y considerando algunas de las ventajas que se pretenden obtener del uso de un sistema computacional se considera que las principales funciones con las que deberá cumplir el Sistema para la Administración de Pagos Internos y Externos de un Centro de Capacitación Computacional (SAPIECC) son las siguientes:

- El sistema (SAPIECC) será capaz de generar el alta de nuevos clientes respetando los tipos de cliente y la información requerida para cada uno.
- El sistema permitirá almacenar los diferentes tipos de comprobantes de pagos realizados, conservando la información de cada uno de ellos y utilizándola para realizar cálculos estadísticos y financieros del centro.
- El sistema estará habilitado para asignar cursos a cada cliente considerando las restricciones en cuanto a tipo, horario, periodo y niveles disponibles que ya se encuentran preestablecidos para el centro.
- El sistema facilitará la creación de grupos considerando las características que ya se mencionaron entre las que destaca que las personas que solicitan el curso sean más de tres.
- Otra función del sistema será permitir la asignación de profesores a cada uno de los grupos abiertos en cada periodo y registrar esto para llevar la contabilidad de las horas trabajadas por cada docente.
- Además dentro de las ventajas que se pretende agregar al sistema se encuentra la visualización en pantalla e impresión de consultas y reportes relacionados con:
 - ❖ Monto total de pagos realizados
 - ❖ No. De clientes inscritos por periodo
 - ❖ No. De grupos abiertos por periodo.
 - ❖ No. De personas por grupo
 - ❖ Saldo e historial de cada cliente sin importar su tipo.
 - ❖ Saldo e historial de cada empresa
 - ❖ Grupos asignados a cada profesor
 - ❖ Horas trabajadas por cada profesor.

- ❖ Además de otras consultas más específicas
- El sistema también contará con un sistema de ayuda o tutorial y un manual de usuario. Esto con el objetivo de auxiliar a los usuarios en caso de tener algún problema al realizar algún procedimiento en el sistema.

1.6. Características de los Usuarios

El sistema (SAPIECC) será utilizado por distintos tipos de usuarios entre los que destacan en primer lugar el administrador del sistema o programador, quien será el encargado de diseñarlo, actualizarlo, proporcionar los permisos y brindarle mantenimiento. Esta persona deberá tener amplios conocimientos de la problemática del centro así como de las herramientas que se utilizarán para su resolución.

Enseguida otro tipo de usuarios serán los coordinadores comerciales. Ellos serán los encargados de dar de alta a los clientes así como de asignar cursos, formar grupos, asignar profesores y realizar ajustes en las horas trabajadas por cada uno de los docentes durante cada periodo. Así como de obtener los diferentes tipos de reportes que el sistema será capaz de generar. Debido a la importancia de la función que tienen los coordinadores se recomienda que estos tengan conocimientos de paquetería básicos. Considerando con esto el manejo de interfaces gráficas, módulos de impresión e interpretación de la información que el sistema generará y mostrará no sólo en pantalla sino también de forma impresa.

Finalmente el departamento de administración recibirá los comprobantes de pagos, por lo que es importante que sus colaboradores tengan acceso y comprendan como funciona esta sección del sistema, ya que si es necesario realizar un ajuste en esta sección ellos serán los encargados de realizarlo y deben tener los conocimientos así como los privilegios para realizar esto.

II. FUNDAMENTOS TEÓRICOS

2.1 Sistemas Operativos

2.1.1 Funciones Básicas

La clasificación más común que existe para el software considera dos grandes tipos de programas: el software de aplicación y el software de base o de sistema. El software de aplicación consiste en todos aquellos programas que tiene una función perfectamente definida y que facilitan alguna tarea mediante el uso de la computadora. Por otro lado el software de base o de sistema considera todo el conjunto de programas que permiten controlar el comportamiento de la máquina. Dentro de este grupo se encuentran los sistemas operativos.

Un sistema operativo es un conjunto de programas que amplían la funcionalidad propia de los dispositivos de hardware para facilitar el desarrollo y operación de aplicaciones. (Sistema operativo como maquina extendida). Aunque también se considera que el sistema operativo es un conjunto de programas que se dedican a administrar recursos del equipo: procesador, memoria, dispositivos de entrada y salida, el sistema de archivos y el software disponible. (Sistema Operativo como administrador de recursos).

Un sistema operativo desempeña 5 funciones básicas en la operación de un sistema informático. Estas son:

- **Suministro de interfaz al usuario:** Parte que permite comunicarse con él sistema, de tal manera que se puedan cargar programas, acceder archivos y realizar otras tareas. Existen tres tipos básicos de interfaces: las que se basan en comandos, las que utilizan menús y las interfaces gráficas de usuario.
- **Administración de recursos:** Sirven para administrar los recursos de hardware y de redes de un sistema informático como la CPU, memoria, dispositivos de almacenamiento secundario y periféricos de entrada y de salida.
- **Administración de archivos:** Programas que controlan la creación, borrado y acceso de archivos de datos y de programas. También mantienen el registro de la ubicación física de los archivos en los discos magnéticos y en otros dispositivos de almacenamiento secundarios.
- **Administración de tareas:** Programas que administran la realización de las tareas informáticas de los usuarios finales. Los programas controlan qué áreas tienen acceso al CPU y por cuánto tiempo. Las funciones de administración de tareas pueden distribuir una parte específica del tiempo del CPU para una tarea en particular, e interrumpir al CPU en cualquier momento para sustituirla con una tarea de mayor prioridad.

- **Servicio de soporte y utilidades:** Los servicios de soporte de cada sistema operativo dependerán de la implementación particular de éste con la que estemos trabajando. Estos servicios de soporte suelen consistir en:
 - Actualización de versiones.
 - Mejoras de seguridad.
 - Inclusión de alguna nueva utilidad.
 - Controladores para manejar nuevos periféricos.
 - Corrección de errores de software.
 - Otros.

2.1.2 Tipos de Sistemas Operativos

Sistemas Operativos por su Estructura

De acuerdo a su estructura los sistemas operativos se dividen en:

- sistemas operativos de estructura monolítica
- sistemas operativos de estructura jerárquica,
- sistemas operativos tipo máquina virtual
- sistemas operativos Cliente-Servidor (Microkernel)

Los sistemas operativos de estructura monolítica se caracterizan por la construcción del programa final a base de módulos compilados separadamente que se unen a través del ligador. Carecen de protecciones y privilegios al entrar a rutinas que manejan diferentes aspectos de los recursos de la computadora, como memoria, disco, etc. Generalmente están hechos a medida, por lo que son eficientes y rápidos en su ejecución y gestión, pero por lo mismo carecen de flexibilidad para soportar diferentes ambientes de trabajo o tipos de aplicaciones. (Ver Fig. 2.1)

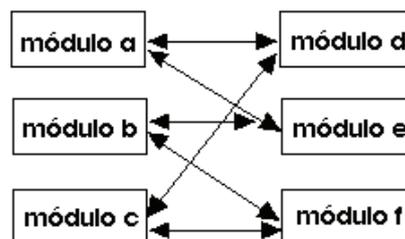


Figura 2. 1 Sistemas Operativos de Estructura Monolítica

Los sistemas operativos de estructura jerárquica esquemáticamente se pueden describir de acuerdo a la figura 2.2.



Figura 2. 2 Sistemas Operativos de Estructura Jerárquica

Otra forma de ver este tipo de sistema es la denominada de anillos concéntricos o "rings" (Ver Fig. 2.3). En este tipo de sistema cada anillo tiene una apertura, conocida como puerta o trampa (trap), por donde pueden entrar las llamadas de las capas inferiores. De esta forma, las zonas más internas del sistema operativo o núcleo del sistema estarán más protegidas de accesos indeseados desde las capas más externas. Las capas más internas serán, por tanto, más privilegiadas que las externas.

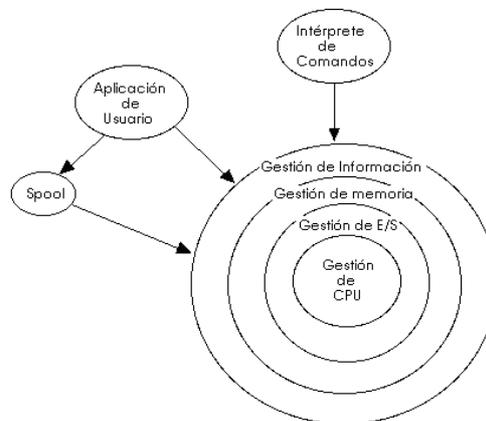


Figura 2. 3 Zonas Sistemas Operativos de Estructura Jerárquica

Los sistemas operativos tipo máquina virtual presentan una interface a cada proceso, mostrando una máquina que parece idéntica a la máquina real subyacente. Estos sistemas operativos separan dos conceptos que suelen estar unidos en el resto de sistemas: la multiprogramación y la máquina extendida. El objetivo de los sistemas operativos de máquina virtual es el de integrar distintos sistemas operativos dando la sensación de ser varias máquinas diferentes (Ver Fig. 2.4).

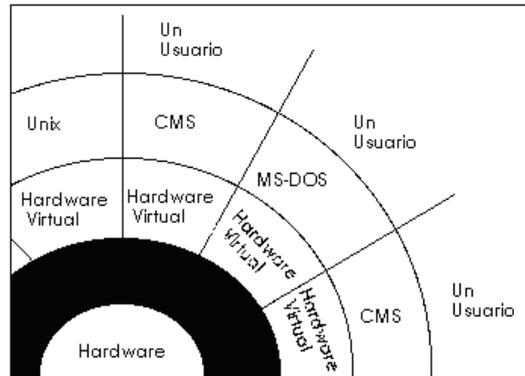


Figura 2. 4 Sistemas Operativos Tipo Maquina Virtual

El tipo más reciente de sistemas operativos es el denominado Cliente-Servidor. El núcleo tiene como misión establecer la comunicación entre los clientes y los servidores. Los procesos pueden ser tanto servidores como clientes. Por ejemplo, un programa de aplicación normal es un cliente que llama al servidor correspondiente para acceder a un archivo o realizar una operación de entrada/salida sobre un dispositivo concreto. A su vez, un proceso cliente puede actuar como servidor para otro. Este paradigma ofrece gran flexibilidad en cuanto a los servicios posibles en el sistema final, ya que el núcleo provee solamente funciones muy básicas de memoria, entrada/salida, archivos y procesos, dejando a los servidores proveer la mayoría que el usuario final o programador puede usar. Estos servidores deben tener mecanismos de seguridad y protección que, a su vez, serán filtrados por el núcleo que controla el hardware.

Sistemas Operativos por Servicios

Esta clasificación es la más comúnmente usada y conocida desde el punto de vista del usuario final. Esta clasificación se comprende fácilmente con el cuadro sinóptico que a continuación se muestra.

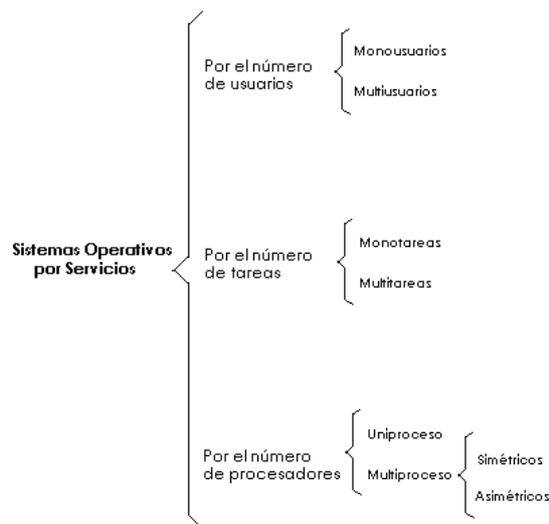


Figura 2. 5 Sistemas Operativos por Servicios

Monousuarios

Los sistemas operativos monousuarios son aquéllos que soportan a un usuario a la vez, sin importar el número de procesadores que tenga la computadora o el número de procesos o tareas que el usuario pueda ejecutar en un mismo instante de tiempo. Las computadoras personales típicamente se han clasificado en este renglón.

Multiusuarios

Los sistemas operativos multiusuarios son capaces de dar servicio a más de un usuario a la vez, ya sea por medio de varias terminales conectadas a la computadora o por medio de sesiones remotas en una red de comunicaciones. No importa el número de procesadores en la máquina ni el número de procesos que cada usuario puede ejecutar simultáneamente.

Monotareas

Los sistemas monotarea son aquellos que sólo permiten una tarea a la vez por usuario. Puede darse el caso de un sistema multiusuario y monotarea, en el cual se admiten varios usuarios al mismo tiempo pero cada uno de ellos puede estar haciendo solo una tarea a la vez.

Multitareas

Un sistema operativo multitarea es aquél que le permite al usuario estar realizando varias labores al mismo tiempo. Es común encontrar en ellos interfaces gráficas orientadas al uso de menú y el ratón, lo cual permite un rápido intercambio entre las tareas para el usuario, mejorando su productividad.

Uniproceto

Un sistema operativo uniproceto es aquél que es capaz de manejar solamente un procesador de la computadora, de manera que si la computadora tuviese más de uno le sería inútil. El ejemplo más típico de este tipo de sistemas es el DOS y MacOS.

Multiproceto

Un sistema operativo multiproceto se refiere al número de procesadores del sistema, que es más de uno y éste es capaz de usarlos todos para distribuir su carga de trabajo. Generalmente estos sistemas trabajan de dos formas: simétrica o asimétricamente. Cuando se trabaja de manera asimétrica, el sistema operativo selecciona a uno de los procesadores el cual jugará el papel de procesador maestro y servirá como pivote para distribuir la carga a los demás procesadores, que reciben el nombre de esclavos. Cuando se trabaja de manera simétrica, los procesos o partes de ellos (threads) son enviados indistintamente a cualquiera de los procesadores disponibles, teniendo, teóricamente, una mejor distribución y equilibrio en la carga de trabajo bajo este esquema.

Sistemas Operativos por la Forma de Ofrecer sus Servicios

Esta clasificación también se refiere a una visión externa, que en este caso se refiere a la del usuario, el cómo accesa los servicios. Bajo esta clasificación se pueden detectar dos tipos principales: sistemas operativos de red y sistemas operativos distribuidos.

Los sistemas operativos de red se definen como aquellos que tiene la capacidad de interactuar con sistemas operativos en otras computadoras por medio de un medio de transmisión con el objeto de intercambiar información, transferir archivos, ejecutar comandos remotos y un sin fin de otras actividades. El punto crucial de estos sistemas es que el usuario debe saber la sintaxis de un conjunto de comandos o llamadas al sistema para ejecutar estas operaciones, además de la ubicación de los recursos que desee acceder.

Los sistemas operativos distribuidos abarcan los servicios de los de red, logrando integrar recursos (impresoras, unidades de respaldo, memoria, procesos, unidades centrales de proceso) en una sola máquina virtual que el usuario accesa en forma transparente. Es decir, ahora el usuario ya no necesita saber la ubicación de los recursos, sino que los conoce por nombre y simplemente los usa como si todos ellos fuesen locales a su lugar de trabajo habitual.

2.1.3 Ventajas y Desventajas de Algunos Sistemas Operativos

Si comparamos Linux con Windows 95/98/XP/7 encontramos las siguientes ventajas y desventajas:

Precio

- Linux es libre, flexible, gratuito, configurable, eficiente, no requiere hardware caro, no obliga a cambiar continuamente la versión del software.
- Windows es un software no gratuito y poco flexible, además continuamente cambia la versión de software.

Compatibilidad

- Linux requiere más conocimientos de informática, no hay sustituto directo para todas las aplicaciones, algunos dispositivos de última generación no están soportados.
- Windows es un software mucho más compatible, tiene mucho mayor soporte en cuanto a dispositivos y requiere el mínimo de conocimientos de informática.

Usuarios

- Linux es un software menos comercial por lo cual es menos utilizado ya que como es gratuito la mayoría de usuarios no le tienen toda la confianza.
- Windows es un software muy publicitado lo cual lo convierte en un producto muy comercial además de que cuenta con un número mucho mayor de usuarios alrededor del mundo.

Sin embargo, la desventaja más negativa resulta que Microsoft va "jubilando" sus sistemas operativos con el tiempo. Por ejemplo, con la entrada de 2003 ha calificado de obsoletos a sus sistemas MS-DOS, Windows 3.x, Windows 95 y Windows NT 3.5. Windows ME, con tan sólo unos pocos años de vida, se jubilaron el 31 de diciembre de 2003.

A partir de entonces, Microsoft no ofrece más asistencia ni actualizaciones para ellos, lo que puede dejar desprotegidos a miles de usuarios de todo el mundo, que deberán actualizar su sistema operativo para disponer actualizaciones de seguridad.

2.2 Ingeniería del Software

La ingeniería del software se define como la disciplina tecnológica preocupada de la producción sistemática y mantenimiento de los productos de software que son desarrollados y modificados en tiempo y dentro de un presupuesto definido. Difiere de la programación tradicional en que se utilizan técnicas de ingeniería para especificar, diseñar, instrumentar, validar y mantener los productos dentro del tiempo y el presupuesto establecidos para el proyecto; además se preocupa por aspectos administrativos que quedan fuera del dominio normal de la programación.

Los cuatro principales objetivos de la ingeniería del software son: facilidad de modificación, eficiencia, confiabilidad y claridad. Mientras que los siete principios básicos de la ingeniería de software son: modularidad, localización, abstracción, ocultación de la información, comprobabilidad, uniformidad y completitud.

La ingeniería de software es una tecnología multicapa (Ver figura 2.6). El fundamento de la ingeniería de software es la capa de proceso, ya que es la unión de las capas de tecnología y permite un desarrollo racional y oportuno de la ingeniería de software.

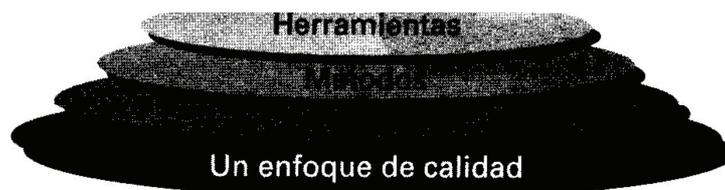


Figura 2. 6

El proceso define un marco de trabajo para un conjunto de áreas clave que se deben establecer para la entrega efectiva de la tecnología de la ingeniería de software. Las áreas claves del proceso forman la base del control de gestión de proyectos del software y establecen el contexto en el que se aplican los métodos técnicos, se obtienen los productos del trabajo, se establecen hitos, se asegura la calidad y el cambio se gestiona adecuadamente.

Los métodos de la ingeniería del software indican cómo construir técnicamente el software, abarcan una gran gama de tareas que incluyen análisis de requisitos, diseño, construcción de programas, pruebas y mantenimiento.

2.2.1 El Proceso del Software

Un proceso de software se puede caracterizar como se muestra en la figura 2.7. Es decir se establece un marco común del proceso definiendo un pequeño número de actividades del marco de trabajo que son aplicables a todos los proyectos del software sin importar su tamaño o complejidad, un conjunto de tareas, hitos y entregas. Finalmente las actividades de protección son independientes a cualquier parte del marco.

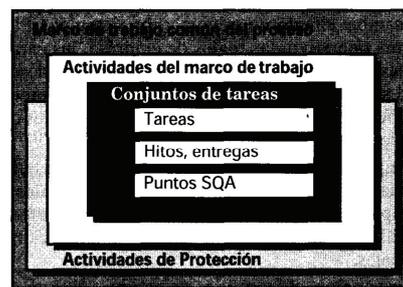


Figura 2. 7

2.2.2 Modelos de Proceso del Software

Para resolver los problemas reales de una industria, un ingeniero de software o un equipo de ingenieros deben incorporar una estrategia de desarrollo que acompañe al proceso, métodos y capas de herramientas. Esta estrategia a menudo se llama modelo de proceso o paradigma de ingeniería del software. Se selecciona un modelo de proceso para la ingeniería del software según la naturaleza del proyecto y de la aplicación, los métodos y las herramientas a utilizarse, y los controles y entregas que se requieren.

Todo el desarrollo del software se puede caracterizar como un bucle de resolución de problemas en el que se encuentran cuatro etapas distintas: estado actual, definición de problemas, desarrollo técnico e integración de soluciones. Conforme progresa el trabajo hacia un sistema completo, las etapas descritas anteriormente se aplican recursivamente a las necesidades del usuario y a la especificación técnica del desarrollador del software.

El Modelo Lineal Secuencial

Llamado algunas veces ciclo de vida básico o modelo en cascada, sugiere un enfoque sistemático secuencial para el desarrollo del software que comienza en un nivel de sistemas y progresa con el análisis, diseño, codificación, pruebas y mantenimiento. Este modelo comprende las siguientes actividades:

- a) Ingeniería y modelado de Sistemas/Información
- b) Análisis de los requisitos del software

- c) Diseño
- d) Generación de código
- e) Pruebas
- f) Mantenimiento

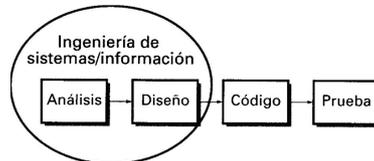


Figura 2. 8 Modelo Lineal Secuencial

El Modelo de Construcción de Prototipos

El paradigma de construcción de prototipos comienza con la recolección de requisitos. El desarrollador y el cliente encuentran y definen los objetivos globales para el software, identifican los requisitos conocidos y las áreas del esquema en donde es obligatoria más definición. Entonces aparece un diseño rápido. El diseño rápido se centra en una representación de esos aspectos del software que serán visibles para el usuario/cliente. El diseño rápido lleva a la construcción de un prototipo el cual es avaluado por el cliente/usuario y se utiliza para refinar los requisitos del software a desarrollar. La iteración ocurre cuando el prototipo se pone a punto para satisfacer las necesidades del cliente, permitiendo al mismo tiempo que el desarrollador comprenda mejor lo que se necesita hacer.

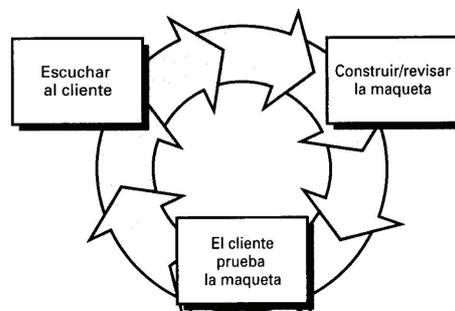


Figura 2. 9 Modelo de Construcción de Prototipos

El Modelo DRA

El Desarrollo Rápido de Aplicaciones(DRA) es un modelo de proceso del desarrollo del software lineal secuencial que enfatiza un ciclo de desarrollo extremadamente corto. Es una adaptación a alta velocidad del modelo lineal secuencial en el que se logra el desarrollo rápido utilizando una construcción basada en componentes. Si se comprenden bien los requisitos y se limita el ámbito del proyecto, el proceso DRA permite al equipo de desarrollo crear un sistema completamente funcional dentro de periodos cortos de tiempo. Cuando se utiliza principalmente para aplicaciones de sistemas de información, el enfoque DRA comprende las siguientes fases: modelado de gestión, modelado de datos, modelado del proceso, generación de aplicaciones, pruebas y entrega.

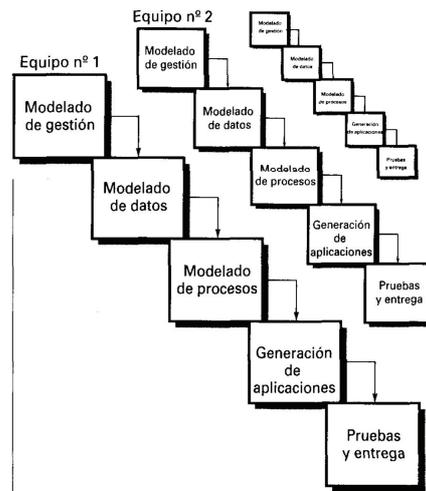


Figura 2. 10 Modelo DRA

Modelos Evolutivos del Proceso del Software

Los modelos evolutivos son iterativos. Se caracterizan por la forma en que permiten a los ingenieros del software desarrollar versiones cada vez más completas del software.

El Modelo Incremental

El modelo incremental combina elementos del modelo lineal secuencial con la filosofía interactiva de construcción de prototipos. Este modelo aplica secuencias lineales de forma escalonada. Cada secuencia lineal produce un incremento del software. Cuando se utiliza un modelo incremental, el primer incremento a menudo es un producto esencial. Es decir, se afrontan requisitos básicos, pero muchas funciones suplementarias quedan sin extraer. El cliente utiliza el producto central. Como resultado de la utilización y/o evaluación, se desarrolla un plan para el incremento siguiente. Este proceso se repite siguiendo la entrega de cada incremento, hasta que se elabore el producto completo.

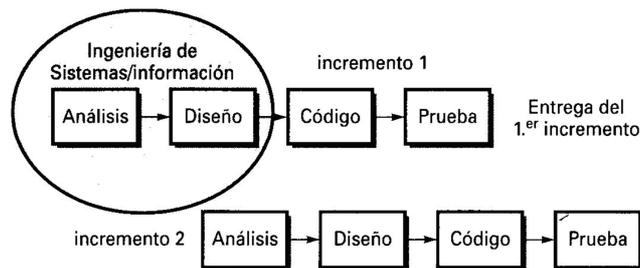


Figura 2. 11 Modelo Incremental del Proceso del Software

El Modelo Espiral

El modelo en espiral conjuga la naturaleza iterativa de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial. En este modelo, el software se desarrolla en una serie de versiones incrementales. El modelo en espiral se divide en un número de actividades

de marco de trabajo, también llamadas regiones de tareas. Generalmente existen entre tres y seis regiones de tareas, entre las cuales pueden estar: comunicación con el cliente, planificación, análisis de riesgos, ingeniería, construcción y acción, evaluación del cliente, etc. Cada una de las regiones está compuesta por un conjunto de tareas de trabajo, llamado conjunto de tareas que se adaptan a las características del proyecto que va a emprenderse. Cuando empieza el proceso evolutivo, el equipo de ingeniería de software gira alrededor de la espiral en la dirección de las agujas del reloj, comenzando por el centro. A diferencia del modelo de proceso clásico, el modelo en espiral puede adaptarse y aplicarse a lo largo de la vida del software de computadora. El modelo en espiral es un enfoque realista del desarrollo de sistemas y del software a gran escala.

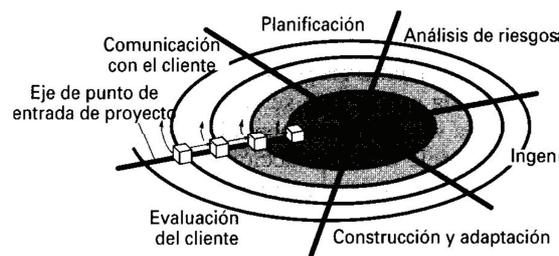


Figura 2. 12 Modelo Espiral del Proceso del Software

El Modelo de Desarrollo Basado en Componentes

El modelo de desarrollo basado en componentes incorpora muchas de las características del modelo en espiral. Sin embargo este modelo configura aplicaciones desde componentes preparados de software llamados clases. La actividad de ingeniería comienza con la identificación de clases candidatas. Esto se lleva a cabo examinando los datos que se van a manejar por parte de la aplicación y el algoritmo que se va a aplicar para conseguir el tratamiento. Los datos y los algoritmos correspondientes se empaquetan en una clase. Las clases creadas en los proyectos de ingeniería del software anteriores se almacenan en una biblioteca de clases o diccionario de datos. Una vez identificadas las clases candidatas, la biblioteca de clases se examina para determinar si estas clases ya existen. En caso de que así fuera, se extraen de la biblioteca y se vuelven a utilizar. Si no es así, se aplican los métodos orientados a objetos. Se compone así la primera iteración de la aplicación a construirse, mediante las clases extraídas de la biblioteca y las clases nuevas construidas para cumplir las necesidades únicas de la aplicación. El flujo del proceso vuelve a la espiral y volverá a introducir por último la iteración ensambladora de componentes a través de la actividad de ingeniería. El modelo de desarrollo basado en componentes conduce a la reutilización del software, y la reutilización proporciona beneficios a los ingenieros de software.

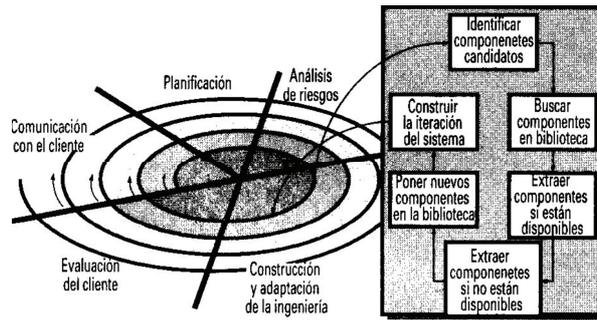


Figura 2. 13 Modelo de Desarrollo Basado en Componentes

El Modelo de Métodos Formales

El modelo de métodos formales comprende un conjunto de actividades que conducen a la especificación matemática del software de computadora. Los métodos formales permiten que un ingeniero de software especifique, desarrolle y verifique un sistema basado en computadora aplicando una notación muy rigurosa y matemática.

2.3 Planeación de un Proyecto de Programación

Todo elemento desarrollado por el hombre primero es una idea en su mente. Los sistemas computacionales, como otros productos de la tecnología, se desarrollan en respuesta a requerimientos detectados. Las fuentes que producen las ideas de productos de programación incluyen las necesidades del cliente generadas externamente, las necesidades internas de la organización, planes de mercadotecnia, y los planes o misiones de las organizaciones. La mayor parte de las organizaciones que desarrollan productos de programación son muy selectivas al decidir qué productos desarrollarán. La decisión de llevar a cabo el proyecto se basa, generalmente, en el resultado de un estudio de factibilidad.

2.3.1 Definición del Problema

El primer paso en la planeación de un proyecto de programación es preparar, en la terminología del cliente, un enunciado breve del problema que se solucionará y de las restricciones que existen en su resolución. Debe de incluir una descripción de la situación actual y de las metas que debe lograr el nuevo sistema.

La definición del problema requiere de un entendimiento cabal del dominio del problema y del entorno de éste. Las técnicas para obtener este conocimiento, por parte del planeador, son entrevistas con el cliente, observación de las tareas problemáticas, y desarrollo de las reales. El planeador debe ser muy hábil en las técnicas de definición del problema, ya que distintos representantes del cliente tendrán diferentes puntos de vista, sesgos, y prejuicios que influirán en su percepción del área del problema.

El segundo paso en la planeación de un proyecto de programación es determinar lo apropiado de una solución computacional. Además de ser eficaz en términos de costo, un sistema debe aceptarse social y políticamente. Para ser eficiente en costo, un nuevo producto debe proporcionar los mismos servicios e información que el sistema antiguo, usando menos tiempo y personal, o proporcionar servicios e información que antes eran inaccesibles.

Habiendo determinado, por lo menos en una forma preliminar, que es apropiada una solución computarizada para el problema, la atención se centra en las funciones de los principales subsistemas del sistema computacional. Un sistema computacional está formado por los subsistemas de personal, equipo y de productos de programación, más las interconexiones entre ellos. El primer subsistema incluye operadores, personal de mantenimiento y usuarios finales. El segundo comprende el equipo de cómputo y de los dispositivos periféricos, y puede tener otros dispositivos como sensores y accionadores para control de proceso, o antenas rastreadoras y radares. El tercer subsistema contiene programas que deben desarrollarse, más programas que ya existen y que pueden emplearse como están o modificándolos.

Las funciones que debe realizar cada subsistema principal se deben identificar, se deben establecer las interacciones entre subsistemas y determinar las restricciones en el desarrollo y operación para cada subsistema principal. Las limitaciones especifican número y tipo de equipos, cantidad y habilidades de personal, y características del producto de programación como funcionamiento, precisión y nivel de confiabilidad. La asignación precisa de funciones entre equipo, programación y personal puede dificultarse durante la planeación preliminar, tal vez sea necesario desarrollar primero un análisis detallado. No obstante, debe intentarse la definición preliminar de las funciones de los subsistemas principales.

Dado el enunciado preciso del problema y la indicación de las restricciones que existen para su solución, se pueden formular metas y requisitos preliminares. Las metas son logros por alcanzar; sirven para establecer el marco de referencia para el proyecto de desarrollo del producto de programación. Estas se aplican tanto para el proceso de desarrollo como para los productos finales, y pueden ser cualitativas o cuantitativas.

Los requisitos especifican las capacidades que debe tener un sistema para la solución de un problema. Estos se establecen para la funcionalidad, el rendimiento, el equipo, la programación y las interfaces con el usuario. Los requisitos pueden establecer también estándares de desarrollo y de control de calidad tanto para el desarrollo como para el producto; deben ser cuantificados siempre que sea posible.

Las metas y los requisitos de alto nivel se pueden expresar en términos de atributos de calidad que el sistema deberá poseer. Es importante que los criterios de alto nivel para la aceptación del sistema se definan durante la fase de planeación. La falta de criterios cuantitativos de aceptación establecidos con claridad puede producir graves malentendidos entre el cliente y el encargado de desarrollo. Los criterios de aceptación se deben fijar de tal forma que se puedan verificar por métodos conocidos, como inspección, análisis, o pruebas realizadas en los productos finales. Cada requisito debe incluir el método que se empleará durante su verificación.

2.3.2 Desarrollo de una Estrategia de Solución

La tendencia de utilizar la primera solución que aparece es un problema importante en la ingeniería de software. Una manera de evitarlo es al desarrollar primero una estrategia de solución. Esta no es un plan detallado de solución, sino un enunciado general sobre la naturaleza de las posibles soluciones. Una estrategia de solución debe considerar todos los factores externos que son visibles para los usuarios del producto, y debe redactarse de tal manera que permita caminos alternos para el diseño de producto.

Se deben considerar varias estrategias de solución antes de elegir alguna, aunque los planificadores deben escoger una o más para poder realizar estudios de factibilidad y estimados de costo preliminares. La estrategia seleccionada proporciona un marco de referencia para el diseño y la instrumentación del producto de programación.

Las estrategias de solución se deben crear sin tomar en cuenta aspectos de factibilidad, puesto que no es posible ser creativo y crítico al mismo tiempo. La factibilidad de cada estrategia de solución propuesta se debe establecer por el análisis de las restricciones de la solución. Una estrategia de solución es factible si las metas y requisitos del proyecto se pueden satisfacer dentro de las restricciones de tiempo disponible, recursos y tecnología por medio de esa estrategia.

Las técnicas para determinar la factibilidad de una estrategia de solución comprenden el estudio de casos, análisis del peor caso, simulación y construcción de prototipos. Un prototipo difiere de un modelo de simulación en que aquél incorpora algunos componentes del sistema real. Las instrumentaciones de prototipos suelen tener funcionalidad limitada, poca confiabilidad y características de operación pobres. Los prototipos se construyen durante la fase de planeación para examinar aspectos técnicos y para simular despliegues al usuario, formatos de informes y diálogos. El último mecanismo es particularmente útil para obtener una mejor comprensión de las necesidades del usuario.

Cuando se recomienda una estrategia de solución, es muy importante documentar las razones por las que se rechazan las otras estrategias. Esto da justificación a la estrategia recomendada, y puede prevenir revisiones equivocadas en fechas posteriores.

Una estrategia de solución debe incluir una lista de prioridades de las características del producto; existen varias razones para establecer dichas prioridades. En algún momento posterior en el ciclo de desarrollo puede necesitarse posponer o eliminar algunas de las capacidades del sistema debido a inconsistencias en los requisitos, cuellos de botella técnicos, o excesos en tiempo o costo. En tales momentos, es esencial disponer de una guía de alto nivel que indique las prioridades de las características fundamentales, de las menos importantes. Sin esta guía, un diseñador o programador puede realizar serios errores de juicio, lo que provocaría la insatisfacción del cliente con el producto final. Las prioridades de las características del producto, también son útiles para indicar la manera en que las capacidades se pueden ir desarrollando en fases dentro de la evolución del sistema. Muchos ingenieros de programación proponen el desarrollo de sistemas como una serie de mejoras sucesivas hacia un sistema núcleo. Las prioridades del producto son útiles en la planeación de las versiones sucesivas que se construirán.

2.3.3 Planeación de un Proceso de Desarrollo

La planeación del proceso de desarrollo de un producto de programación comprende varias consideraciones importantes. La primera es definir un modelo para el ciclo de vida del producto. Este ciclo incluye todas las actividades requeridas para definirlo, desarrollarlo, probarlo, entregarlo, operarlo y mantenerlo. Es esencial definir un modelo de ciclo de vida para cada proyecto de programación, puesto que permite clasificar y controlar las diferentes actividades necesarias para el desarrollo y mantenimiento del producto. Un modelo de ciclo de vida entendido y aceptado por las partes interesadas en el proyecto mejora comunicación, permitiendo así una mejor administración, asignación de recursos, control de costos y calidad del producto. Los modelos de ciclo de vida que se analizan son los de: fases, costos, prototipos y versiones sucesivas.

Modelo de las fases del ciclo de vida

El modelo de fases divide el ciclo de vida del producto de programación en una serie de actividades sucesivas; cada fase requiere información de entrada, procesos y resultados, todos ellos bien definidos. Se necesitan recursos para terminar los procesos de cada fase, y cada una de ellas se efectúa mediante la aplicación de métodos explícitos, herramientas, y técnicas.

El modelo de fase está compuesto por las siguientes etapas: análisis, diseño, instrumentación, pruebas y mantenimiento. En ocasiones se denomina de cascada porque los productos pasan de un nivel a otro con suavidad.

El análisis consta de dos subfases: planeación y definición de requisitos. Las actividades principales durante la planeación incluyen la comprensión del problema del cliente, estudio de factibilidad, desarrollo de la estrategia de solución recomendada, determinación de los criterios de aceptación, y planeación del proceso de desarrollo. Los productos de la planeación son la Definición del sistema y el Plan de proyecto. La Definición, por lo regular, se expresa en español o en algún otro lenguaje natural, y puede contener cuadros, figuras, gráficas, y ecuaciones de distintos estilos. La notación exacta empleada en la definición depende mucho del área del problema.

El Plan del proyecto contiene el modelo de ciclo de vida que se utilizara, la estructura organizacional del proyecto, la programación preliminar del desarrollo, estimados preliminares de costos y recursos, así como de personal, herramientas, técnicas que se emplearán, y estándares que se seguirán.

La definición de requisitos se refiere a la identificación de las funciones básicas del componente de programación en un sistema de equipo / personal / programación. Se pone especial atención en las funciones y restricciones bajo las cuales se deben de desarrollar. El producto de la definición de requisitos, es una especificación que describe el ambiente de procesamiento, las funciones requeridas de los programas, restricciones de configuración sobre los programas (tamaño, velocidad, configuración de equipo), manejo de excepciones, subconjuntos y prioridades de instrumentación, cambios probables y modificaciones factibles, así como los criterios de aceptación del producto de programación.

En el modelo de fases, el diseño de la programación viene después del análisis. El diseño se refiere a la identificación de los componentes de la programación, especificando las relaciones entre ellos, la estructura de la programación, y manteniendo un registro de las decisiones, proporcionando un documento base para la instrumentación. El diseño se divide en estructural y detallado.

El diseño estructural comprende la identificación de los componentes de la programación, su desacoplamiento y descomposición en módulos de procesamiento y estructuras de datos conceptuales, y la especificación de las interconexiones entre componentes. El diseño detallado se refiere a detalles de cómo: cómo empacar módulos de procesamiento, y cómo instrumentar los algoritmos, las estructuras de datos y sus interconexiones.

La fase de instrumentación en el desarrollo del producto incluye la traducción de las especificaciones del diseño en código fuente, así como su depuración, documentación y pruebas. Los errores descubiertos durante la fase de instrumentación pueden ser errores en las interfaces de datos entre rutinas, errores lógicos en los algoritmos, errores en las estructuras de datos, y de falta de consideración de casos de procesamiento. Además, el código fuente puede contener: errores de requisitos, que indican alguna omisión de las necesidades del usuario en el documento de requisitos; errores de diseño, que reflejarán una mala traducción de requisitos en especificaciones y, por último, errores de instrumentación debidos a una mala traducción de especificaciones en código fuente.

Las pruebas del sistema comprenden dos tipos de actividades: pruebas de integración y aceptación. El desarrollo de una estrategia para integrar los componentes de un sistema de programación, en una entidad funcional, requieren una planeación cuidadosa de modo que se disponga de los módulos cuando éstos se necesiten. Las pruebas de aceptación se relacionan con la planeación y ejecución de varios tipos de pruebas para demostrar que el sistema de programación instrumentado satisface las necesidades establecidas en el documento de requisitos.

Una vez aceptado por el cliente, el sistema de programación se entrega para operación y se inicia la fase de mantenimiento del modelo de ciclo de vida por fases. Las actividades de mantenimiento incluyen mejora de las capacidades, adaptación a nuevos ambientes de procesamiento y corrección de fallas del sistema.

Modelo del costo de un proyecto

Otro punto de vista para el ciclo de vida de desarrollo de un producto de programación es la consideración del costo de la realización de las distintas actividades del proyecto. El costo de un proyecto es la suma de los costos incurridos en cada fase, y éstos, a su vez, incluyen los costos de la realización de los procesos y preparación de los documentos de esa fase, más los costos de verificación de la consistencia de estos productos con los de las fases previas.

El costo de producción de la Definición del sistema y del Plan del proyecto es el mismo de realizar la planeación y preparación de los documentos, más el costo de verificación de que la Definición del sistema establece precisamente las necesidades del cliente y el costo de verificación de que el Plan del proyecto es factible.

El costo de preparación de la Especificación de requisitos para la producción de software incluye el costo de definir requisitos y preparar documentos, más el costo de modificar y corregir la Definición del sistema y el Plan del proyecto, más el costo de verificación de que la Especificación este completa y sea consistente con respecto a la Definición del sistema y las necesidades del cliente.

De la misma manera, el costo del diseño es el costo de las actividades propias y la generación de los documentos, más el costo de modificar y corregir la Definición del sistema, el Plan del Proyecto y la Especificación de requisitos para la producción de software, más el costo de verificación del diseño contra los requisitos, la Definición del sistema y el Plan del proyecto.

El costo de la instrumentación del producto es el costo de la conversión, documentación, depuración y pruebas de código fuente, más el costo de la terminación del manual de usuario, el plan de verificación, los procedimientos de mantenimiento, y las instrucciones de instalación y entrenamiento, más el costo de modificar y corregir la Definición del sistema, el Plan de proyecto, la Especificación de requisitos para la producción del software, la Especificación del diseño, y el plan de verificación; más el costo de la verificación de que la instrumentación este completa y sea consistente con respecto a la Definición del sistema, la Especificación de requisitos para la producción de software y los documentos del diseño.

El costo de las pruebas del sistema incluye el costo de planear y llevar a cabo las pruebas, más el costo de las modificaciones al código fuente y a los documentos externos durante ellas, más el costo de verificación de que las pruebas validan adecuadamente el producto.

Por último, el costo del mantenimiento es la suma de los costos de llevar a cabo mejoras al sistema, hacer adaptaciones para nuevas condiciones de operación y encontrar fallas. Cada una de estas actividades puede comprender la modificación de alguno o de todos los documentos y la ejecución de un gran número de casos de prueba para verificar la corrección de la modificación.

Modelo de prototipo para el ciclo de vida

Este modelo subraya las fuentes de requisitos para el producto, puntos decisivos de continuar / detenerse, y el uso de prototipos. Un prototipo es una representación o modelo del producto de programación que, a diferencia de un modelo de simulación, incorpora componentes del producto real. Por lo regular, un prototipo tiene un funcionamiento limitado en cuanto a capacidades, confiabilidad o eficiencia.

La naturaleza y extensión del prototipo por desarrollar en un proyecto de programación depende de la naturaleza del producto. Se pueden desarrollar nuevas versiones de un producto ya existente en el modelo de las fases y sin ningún prototipo. El avance de un producto totalmente nuevo, tal vez requiera de prototipo durante las fases de planeación y análisis, o el producto se puede desarrollar como una serie de diseños e instrumentaciones.

Versiones sucesivas

El desarrollo de productos mediante el método de versiones sucesivas es una extensión del método de prototipos en el que se refina un esqueleto inicial del producto obteniendo así, cada vez más capacidades. En dicho método, cada versión es funcional y capaz de realizar trabajo útil. En este caso, las características de cada diseño sucesivo serán planteadas durante la fase de análisis. En realidad, el ciclo de desarrollo de un producto de programación es una combinación de los distintos modelos presentados.

2.4 Sistema Manejador de Base de Datos

Un DBMS es una colección de rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica. Las funciones principales de un DBMS son:

- Crear y organizar la base de datos
- Establecer y mantener las trayectorias de acceso a la base de datos,
- Manejar los datos de acuerdo con las peticiones de los usuarios.
- Mantener la integridad y la seguridad de los datos.
- Registrar el uso de las bases de datos

El DBMS interpreta y procesa las peticiones del usuario para recobrar información de la base. Las preguntas a la base pueden tener diferentes formas, en la mayoría de los casos, una petición de consulta tendrá que atravesar varias capas de software en el DBMS y el sistema operativo antes de que pueda acceder a la base de datos física. El DBMS responde a una pregunta llamando a los subprogramas apropiados, cada uno de los cuales realizará su función especial para interpretar la petición o localizar los datos deseados en la base y presentarlos en el orden solicitado. Así el DBMS protege a los usuarios de la tediosa programación que tendrían que hacer para organizar el almacenamiento de los datos, o accederlos una vez almacenados.

2.4.1 Arquitectura de un Sistema de Base de Datos

La arquitectura de una base de datos tiene tres niveles: interno, conceptual y externo. El nivel interno es el más cercano al almacenamiento físico, es decir, el que concierne a la manera como los datos se almacenan en realidad; el nivel externo es el más cercano a los usuarios, es decir, el que atañe a la manera cómo cada usuario ve los datos; y el nivel conceptual es un nivel de mediación entre los otros dos. Si el nivel externo se relaciona con las vistas de los usuarios individuales, el nivel conceptual puede considerarse como el que define una vista de la comunidad de usuarios. En otras palabras, habrá muchas vistas externas, cada una compuesta por una representación más o menos abstracta de alguna parte de la base de datos, y habrá una sola vista conceptual compuesta por una representación también abstracta de la base de datos en su totalidad. Asimismo, habrá una sola vista interna que representa la base de datos total tal como está almacenada.

2.4.2 Funciones del Administrador de la Base de Datos y el Diccionario de Datos

El administrador de bases de datos (DBA) es la persona (o grupo de personas) encargada del control general del sistema de bases de datos. Entre las responsabilidades del DBA se incluyen las siguientes:

- Decidir el contenido de la información de la base de datos
- Decidir la estructura de almacenamiento y la estrategia de acceso
- Vincularse con los usuarios
- Definir los controles de autorización y los procedimientos de validación
- Definir una estrategia de respaldo y recuperación
- Controlar el desempeño y responder a los cambios de requerimientos

Uno de los recursos más importantes de DBA es el diccionario de datos, el cual se estudia a continuación.

El diccionario de datos es un archivo que almacena información sobre los datos relativos al origen de estos, descripción, relación con otros datos, uso, responsabilidad y formato.

Los dos objetivos básicos de un diccionario de datos son la administración y el control de los datos como un recurso, en un lugar central, a través de las fases de diseño, realización y operación, así como el establecimiento de una comunicación efectiva entre todos los que estén interesados en la base de datos.

La siguiente es una lista de los requisitos deseables para formar un diccionario de datos.

Modelo conceptual. La información sobre los datos que se necesitan en el proceso de diseño del modelo conceptual incluye entidades, campos de datos o atributos que representan a las entidades y a las relaciones entre los campos. Los campos de datos, entidades y relaciones deben tener etiquetas, versiones, condiciones, descripciones textuales, sinónimos, miembros y campos de datos agrupados con los campos clave apropiados.

Modelo lógico. La siguiente información sobre el modelo lógico debe ser almacenada en el diccionario de datos. Las agrupaciones de campos de datos con los elementos clave, el modelo de datos fundamental, las relaciones de las agrupaciones basadas en el modelo de datos, los modelos externos contenidos en el modelo lógico, las transacciones lógicas, los programas y los módulos además de las referencias cruzadas entre las transacciones, los programas y los módulos. Otra información es el lenguaje de programación y el tipo de programa para los programas y las transacciones.

Modelo interno. La representación física de la información sobre los campos de datos, por ejemplo, longitud, forma, precisión, justificación o alineación, imagen, reglas de edición, algoritmo de derivación, secuencia, seguridad, medio, dispositivos donde se almacenan las bases de datos y la información de control de acceso deben ser almacenadas en el diccionario de datos.

2.5 Diseño de la Base de Datos

2.5.1 Modelo de Datos

Las entidades de una organización y las relaciones entre ellas pueden representarse por un modelo de datos. Los sistemas de manejo de base de datos comerciales disponibles en la actualidad, están basados en un modelo de red, jerárquico, relacional, o en una combinación o subconjunto de los tres.

La principal diferencia entre los tres tipos de datos reside en la representación de las relaciones entre las entidades. Una relación es una unión o enlace entre dos conjuntos de datos. Estas pueden ser uno a uno, uno a vario o varios a varios.

2.6 Modelo Relacional

2.6.1 Arquitectura del Modelo Relacional

La arquitectura relacional se puede expresar en términos de tres niveles de abstracción, consta de los siguientes componentes:

Modelo relacional de datos. En el nivel conceptual, el modelo relacional de datos está representado por una colección de relaciones almacenadas (también llamadas tablas base). Cada registro de tipo conceptual en un modelo relacional de datos se implanta como un archivo almacenado distinto.

Submodelo de datos. Los esquemas externos de un sistema relacional se llaman submodelos relacionales de datos; cada uno consta de una o más vistas (views) para describir los datos requeridos por una aplicación dada. Una vista puede incluir datos de una o más tablas de datos. Cada programa de aplicación está provisto de un buffer (también llamado área de trabajo del usuario, user work area, UWA), donde el DBMS puede depositar los datos recuperados de la base para su procesamiento, o puede guardar temporalmente sus salidas antes de que el DBMS las escriba en la base de datos.

Esquema de almacenamiento. En el nivel interno, cada tabla base se implanta como un archivo almacenado. Para recuperaciones sobre las claves principal o secundaria se pueden establecer uno o más índices para acceder un archivo almacenado.

Sublenguaje de datos. El sublenguaje de datos es un lenguaje de manejo de datos para el sistema relacional. Originalmente, Codd propuso dos sublenguajes de datos: algebra relacional y cálculo relacional. Cualquier relación que pueda derivarse de una o más tablas de datos también se puede derivar con un solo comando del sublenguaje.

2.6.2 Modelo Conceptual

Una de las principales responsabilidades del administrador de la base de datos es desarrollar un modelo conceptual, también llamado modelo de la empresa o de la organización. El modelo expresa las entidades y sus relaciones y es la herramienta usada para representar la organización conceptual de los datos. El modelo conceptual es una herramienta de comunicación entre los diversos usuarios de los datos, y como tal se desarrolla sin tomar en cuenta la representación física de los mismos. El modelo conceptual se utiliza para organizar, visualizar, planear y comunicar ideas, y debe ser independiente del sistema de manejo de base de datos.

El modelo conceptual tiene que transcribirse a un modelo lógico, usado como una estructura fundamental de un sistema de manejo de base de datos y el modelo lógico debe transportarse a un modelo físico (también llamado modelo interno o estructura física). El modelo lógico puede ser un modelo relacional, jerárquico o de red.

El sistema de manejo de base de datos no es un factor para el diseño de un modelo conceptual, pero el diseño de un modelo lógico sí depende del DBMS que se vaya a usar.

2.6.3 Modelo Lógico

Un modelo relacional consta de varias relaciones (tablas). Al convertir el modelo conceptual en un modelo lógico, estaremos definiendo las relaciones y sus atributos. Uno o varios atributos de estas relaciones representarán a la clave primaria.

Derivar un modelo jerárquico lógico de uno conceptual no es un proceso directo pues se pueden hacer, aparentemente varias elecciones y no existe un resultado correcto. Sin embargo los pasos a seguir se pueden agrupar en categorías con reglas y directrices que se pueden hacer dentro de cada categoría. Estas categorías son las siguientes:

- a) Derivar un modelo jerárquico sin tomar en cuenta ningún sistema de manejo de base de datos (DBMS) en particular.
- b) Modificar el modelo de datos para eliminar conflictos con las reglas del DBMS que se usarán.
- c) Refinar el modelo de datos modificado de acuerdo con algunas consideraciones obvias de funcionamiento
- d) Simplificar los nombres de las claves
- e) Añadir las relaciones que existen entre los datos, pero que no aparecieron en el modelo lógico.

Para convertir el modelo conceptual en un modelo de red tampoco se tiene un modelo directo. Deben hacerse muchas elecciones arbitrarias y no existe ningún resultado correcto. Pueden seguirse los mismos pasos para la conversión a un modelo jerárquico.

2.6.4 Modelo Físico

El modelo físico es una estructura de la base de datos que se ha de almacenar en dispositivos físicos. Es absolutamente necesario evaluar las características de funcionamiento del modelo físico antes de implantar la base de datos. La predicción del funcionamiento es un análisis multivariable, en el cual la principal variable es el modelo físico. Se debería efectuar un análisis cuantitativo del modelo físico con frecuencias promedio de ocurrencias de grupos de datos, con las estimaciones esperadas de espacio auxiliar y con estimaciones esperadas de tiempo para la recuperación y el mantenimiento de los datos.

Un diseñador de bases de datos generalmente trata de optimizar el modelo físico, en cuanto a consideraciones de espacio y tiempo. Se realizan concesiones para obtener equilibrio entre espacio y tiempo en la mayoría de los DBMS. Sin embargo es importante no olvidar los requerimientos comerciales.

Se requiere que el diseñador del modelo físico de una base de datos sea diestro en tres áreas importantes:

- Conocimiento de las funciones del DBMS
- Comprensión de las características de los dispositivos de acceso directo
- Conocimiento de las aplicaciones

Al diseñar el modelo físico de una base de datos, necesitamos preocuparnos de los aspectos físicos de la base de datos, es decir, la disposición del registro en disco, tamaño de los bloques, tamaño de los buffers y características de entrada / salida entre otros aspectos. Si esto no se considera en el modelo físico la base de datos puede fracasar como consecuencia de un funcionamiento pobre.

2.7 Lenguajes de Programación

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina.

Un lenguaje de programación permite a uno o más programadores especificar de manera precisa sobre qué datos debe operar una computadora, cómo estos datos deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. Todo esto, a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural, tal como sucede con el lenguaje léxico. Una característica relevante de los lenguajes de programación es precisamente que más de un programador puedan tener un conjunto común de instrucciones que puedan ser comprendidas entre ellos para realizar la construcción del programa de forma colaborativa.

Los procesadores usados en las computadoras son capaces de entender y actuar según lo indican programas escritos en un lenguaje fijo llamado lenguaje de máquina. Todo programa escrito en otro lenguaje puede ser ejecutado de dos maneras:

Mediante un programa que va adaptando las instrucciones conforme son encontradas. A este proceso se lo llama interpretar y a los programas que lo hacen se los conoce como intérpretes.

Traduciendo este programa, al programa equivalente escrito en lenguaje de máquina. A ese proceso se le llama compilar y al programa traductor se le denomina compilador.

2.7.1 Clasificación de los Lenguajes de Programación

Clasificación según el estilo de programación

De acuerdo con el estilo de programación, podemos clasificar los lenguajes en las siguientes categorías:

- **Imperativos:** Son aquellos lenguajes, que basan su funcionamiento en un conjunto de instrucciones secuenciales, las cuales, al ejecutarse, van alterando las regiones de memoria donde residen todos los valores de las variables involucradas en el problema que se plantea resolver. Es decir, se cambia progresivamente el estado del sistema, hasta alcanzar la solución del problema.
- **Declarativos:** En este paradigma, más que el ¿cómo? desarrollar paso a paso un proceso, nos interesa el ¿qué? deseamos obtener a través del programa. Quizás el lenguaje declarativo que nos sea más familiar, es SQL, el cual es utilizado para interactuar con la información de bases de datos, concentrándose sólo en los resultados que van a ser obtenidos, dejándole al traductor la tarea de cómo llegar a ellos y presentárnoslos
- **Orientados a Objetos:** Los programas de este tipo, se concentran en los objetos que van a manipular, y no en la lógica requerida para manipularlos. Cabe mencionar con más detalle algunos de los elementos fundamentales que deben de poseer este tipo de lenguajes:
 - ❖ **Abstracción:** Determinación de las características de los objetos, que sirven para identificarlos y hacerlos diferentes a los demás.
 - ❖ **Encapsulamiento:** Es el proceso que agrupa y almacena los elementos que definen la estructura y el comportamiento de una abstracción, en un mismo lugar.
 - ❖ **Modularidad:** Es la propiedad de agrupar las abstracciones que guardan cierta relación lógica, y a la vez minimizar la interdependencia entre las diversas agrupaciones.
 - ❖ **Jerarquía:** Consiste en establecer un orden o una clasificación de las abstracciones.

Clasificación según el modo de traducción

Ahora bien, si tomamos como referencia las herramientas usadas en el proceso de traducción y ejecución de los programas: programa fuente, traductor, programa objeto; vamos a tener la siguiente clasificación de lenguajes:

- **Lenguajes Ensamblados:** Se refieren al lenguaje ensamblador, que viene a ser una representación simbólica de las instrucciones correspondientes al lenguaje ensamblador de alguna arquitectura específica, con lo que, casi siempre, la correspondencia entre las instrucciones de este lenguaje, y las del lenguaje máquina son de 1 a 1, si bien existen algunas excepciones, que dan lugar a lo que se conoce como lenguajes macro-ensambladores.
- **Lenguajes Compilados:** Son aquellos, que son traducidos de un lenguaje de alto nivel (como FORTRAN o PASCAL) a lenguaje máquina o bien a lenguaje ensamblador, produciendo un programa objeto permanente.
- **Lenguajes Interpretados:** Estos lenguajes, tienen la particularidad, de que no producen código objeto, sino que cada instrucción es analizada y ejecutada a la vez, lo que ofrece mucha interacción con los usuarios, pero a la vez resultan ineficientes, cuando se desea ejecutar repetitivamente un programa.
- **Lenguajes Preprocesados:** Son lenguajes que son traducidos primeramente a un lenguaje intermedio de más bajo nivel, para posteriormente volverlos a traducir y producir el programa objeto. Este tipo de lenguajes fueron creados, con la idea de proporcionar un lenguaje más potente que el lenguaje intermedio, mediante la implementación de algunas macroinstrucciones.

Clasificación según el entorno de programación

Finalmente, existen otros conceptos tomados en cuenta para agrupar los lenguajes, que dan origen a diversas clasificaciones, entre los que destacan las siguientes:

- **Lenguajes de cuarta generación 4GL:** Estos lenguajes se distinguen por formar parte de un entorno de desarrollo, que comprende el manejador de una base de datos, y todo lo que de esto se deriva, como la administración de un diccionario de datos, el control de accesos, el manejo de la consistencia de la información y otras características enfocadas a facilitar los programas de acceso y explotación de la información. Como ejemplos podemos citar a los 4 grandes: POSTGRE, SYSDBASE, INFORMIX, y ORACLE.
- **Lenguajes Visuales.** Se les llama de esta manera a los lenguajes que forman parte de una aplicación dotada de una interface gráfica, la cual por medio de iconos y otras herramientas visuales y simbólicas, pretenden facilitar las tareas rutinarias de los programadores, como son el diseño y desarrollo de formularios e informes. Los ejemplos más comerciales de estos lenguajes son: VISUAL BASIC, VISUAL CAFE, VISUAL FOX, etc.
- **Metalenguajes:** Son lenguajes como XML, SGML y HTML que sirven para definir otros lenguajes, cuyo objetivo es llevar a cabo la estructuración de textos mediante un conjunto de etiquetas, de manera tal, que puedan ser entendidos por los humanos y también procesado por los

ordenadores. Estos lenguajes están teniendo un gran auge sobre la plataforma de Internet, en la cual son usados para la creación de documentos, y el intercambio o transferencia de información.

- **Lenguajes de propósito específico:** Son aquellos lenguajes desarrollados con la finalidad de resolver problemas de una naturaleza muy determinada, tal como SPSS para problemas estadísticos, MATLAB para cálculos científicos y de ingeniería, CAD/CAM para el diseño de piezas y programación de máquinas de control numérico, como tornos y fresadoras, GPSS para simulación de sistemas, CORBA para el manejo de interfaces en ambientes cliente-servidor, etc.
- **Lenguajes Script:** Son lenguajes como JAVASCRIPT, VBSCRIPT o PHP, que se utilizan en ambientes clientes servidor, mediante la incrustación de código en las páginas HTML, y así permitir la programación del lado del cliente, buscando fundamentalmente, hacer más atractivos los interfaces gráficos de las páginas.

Esta gran cantidad de lenguajes, señala de manera clara que existe un esfuerzo continuo en la creación, y mejora de los lenguajes de programación, en aras, de hacer más fácil la tarea del programador y/o hacer un uso más eficiente de los recursos computacionales.

La búsqueda de los objetivos antes mencionados, así como la guerra mercantil de las compañías dedicadas a la producción de herramientas de software, han diversificado las opciones que los programadores pueden elegir. Sin embargo, hasta nuestros días, podemos decir que realmente no existe ningún lenguaje, o grupo de ellos, que destaque en la totalidad de las aplicaciones informáticas que se desarrollan actualmente, ya que cada uno, tiene cualidades que lo hacen más convenientes para algunos propósitos, pero al mismo tiempo, cuentan con inconvenientes para otros.

2.7.2. Lenguajes de Programación para la Web

La programación Web, parte de las siglas WWW, que significan World Wide Web. Para realizar una página con la programación Web, se deben tener claros, tres conceptos fundamentales los cuales son, el URL(Uniform Resource Locators), es un sistema con el cual se localiza un recurso dentro de la red, este recurso puede ser una página web, un servicio o cualquier otra cosa.

El siguiente concepto dentro de la programación Web, es el protocolo encargado de llevar la información que contiene una página Web por toda la red de internet, como es el HTTP(Hypertext Transfer Protocol).

Y por último el lenguaje necesario cuya funcionalidad es la de representar cualquier clase de información que se encuentre almacenada en una página Web, este lenguaje es el HTML(Hypertext Markup Language). En la programación Web, el HTML es el lenguaje que permite codificar o preparar documentos de hipertexto, que viene a ser el lenguaje común para la construcción de una página Web.

Actualmente existen diferentes lenguajes de programación para desarrollar en la web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas. Desde los inicios de Internet,

fueron surgiendo diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. A medida que paso el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación para la web dinámicos, que permitieran interactuar con los usuarios y utilizaran sistemas de Bases de Datos.

Entre los lenguajes más conocidos se encuentran:

- Lenguaje HTML
- Lenguaje Java script
- Lenguaje PHP
- Lenguaje ASP
- Lenguaje ASP.NET
- Lenguaje JSP
- Lenguaje Python
- Lenguaje Ruby

III. PROPUESTA DE SOLUCIÓN

3.1. Estrategias de Solución

Como ya se mencionó anteriormente antes de que el software se pueda construir es importante comprender el sistema en el que residirá. Para lograrlo se deben definir los objetivos generales del sistema; esto significa que es necesario identificar el papel del hardware, software, personas, bases de datos, procedimientos y otros elementos del sistema; así como los requerimientos operacionales deben ser identificados; analizados, especificados, modelados, validados y gestionados.

Los ingenieros en sistemas son los encargados de comprender los requisitos del sistema en colaboración con el cliente, los futuros usuarios y otras partes interesadas. Una vez entendido esto se realiza una propuesta para la solución del problema. Sin embargo la tendencia a utilizar la primera solución que aparece es un problema importante en la ingeniería de software. Una manera de evitarlo es al desarrollar primero una estrategia de solución. Esta no es un plan detallado de solución, sino un estudio general sobre la naturaleza de las posibles soluciones. Los factores estratégicos que se incluyen para el desarrollo del sistema (SAPIECC) son: programación estructurada o programación orientada a objetos; modo consola o modo gráfico y software libre o software con licencia comercial. Las características de cada uno de estos enfoques se mencionan a continuación.

3.1.1 Programación Estructurada vs Programación Orientada a Objetos

La programación estructurada es una forma de escribir programas de manera ordenada y clara utilizando únicamente tres bases principales: secuencia, selección e iteración y no permitiéndose el uso de instrucciones de transferencia incondicional.

Con este estilo de programación podemos obtener las siguientes ventajas:

- Los programas son más fáciles de entender, ya que pueden ser leídos de forma secuencial.
- La estructura del programa es clara, puesto que las instrucciones están más ligadas o relacionadas entre sí.
- Reducción de esfuerzo en las pruebas. El seguimiento de los fallos o errores del programa se facilita debido a la estructura más visible, por lo que los errores se pueden detectar y corregir más fácilmente.
- Reducción de costos de mantenimiento de los programas.
- Programas más sencillos y más rápidos.

- Los bloques de código son auto explicativos, lo que facilita la documentación.
- Un programa escrito de acuerdo a estos principios no sólo tendrá una mejor estructura sino también una excelente presentación.

El principal inconveniente de esta metodología de programación es que se obtiene un único bloque de programa, el cual cuando se hace demasiado grande puede resultar problemático en su manejo; esto se resuelve empleando la programación modular, la cual consiste en definir módulos interdependientes programados y precompilados por separado.

La programación estructurada propicia que el programador sólo piense en términos de procedimientos ó funciones y en segundo lugar en las estructuras de datos que esos procedimientos manejan. Sin embargo existe una metodología llamada programación orientada a objetos (OOP). La cual aporta un enfoque nuevo, convirtiendo la estructura de datos en el centro sobre el que pivotan las operaciones.

Algunas de las principales características que definen a un lenguaje de programación como orientado a objetos son: abstracción, encapsulamiento, ocultación, polimorfismo y herencia entre otras. Además la OOP proporciona las siguientes ventajas:

- **Uniformidad.** Ya que la representación de los objetos implica tanto el análisis como el diseño y la codificación de los mismos.
- **Comprensión.** Tanto los datos que componen los objetos como los procedimientos que los manipulan, están agrupados en clases, que corresponden con las estructuras de información que el programa trata.
- **Flexibilidad.** Al tener relacionados los procedimientos que manipulan los datos con los datos a tratar, cualquier cambio que se realice sobre ellos quedará reflejado automáticamente en cualquier lugar donde estos aparezcan.
- **Estabilidad.** Dado que permite un tratamiento diferenciado de aquellos objetos que permanecen constantes en el tiempo sobre aquellos que cambian con frecuencia permite aislar las partes del programa que permanecen inalterables en el tiempo.
- **Reusabilidad.** La noción de objeto permite que programas que traten las mismas estructuras de información reutilicen las definiciones de datos empleadas en otros programas e incluso los procedimientos que los manipulan. De esta forma, el desarrollo de un programa puede llegar a ser una simple combinación de objetos ya definidos donde están relacionados de una manera particular.

Una de sus principales desventajas es que dado que la estructura y los métodos están separados es difícil realizar un análisis rápido para entender el funcionamiento del programa, además de que el procesamiento es más complejo así como el diseño de pruebas y errores es más complejo con respecto a la programación estructurada.

3.1.2 Modo Consola vs. Modo Gráfico

La interfaz de usuario es el medio a través del cual el usuario puede comunicarse con una computadora. Básicamente está compuesta por menús, ventanas, teclado, ratón y en general por todos aquellos canales que hacen posible la comunicación hombre –computadora. Atendiendo a como el usuario puede interactuar con una interfaz tenemos principalmente dos tipos de interfaces de usuario: Interfaces alfanuméricas (Interprete de comandos o modo consola) e Interfaces gráficas de usuario (GUI, Graphics User Interface).

Un intérprete de órdenes, interprete de línea de órdenes, interprete de comandos, terminal, consola, Shell o su acrónimo en inglés CLI (Command Line Interface) es un programa informático que actúa como interfaz de usuario para comunicar al usuario con el sistema operativo mediante una ventana que espera órdenes escritas por el usuario en el teclado, los interpreta y los entrega al sistema operativo para su ejecución. En su forma más simple una CLI es una ventana que muestra un prompt, señalizando su disposición a recibir órdenes. El usuario teclea una orden y la ingresa habitualmente con la tecla Intro / Enter.

Entre sus ventajas están:

- Es interactiva, es decir, usuario y máquina se comunican en forma sucesiva.
- El uso de la línea de comandos permite realizar algunas tareas más rápidamente que en modo gráfico.
- La CLI consume menos recursos de la CPU.
- Es más rápida y más fácil de utilizar a través de una red de computadoras

Mientras que algunas de sus desventajas son:

- Hay que memorizar los nombres de comandos y sus parámetros.
- El manejo de CLI no es intuitivo, tal como sucede en un ambiente gráfico.
- Solo se puede ver un proceso a la vez, mientras que en los ambientes gráficos las aplicaciones muestran múltiples ventanas sobre la pantalla.

Por otro lado la interfaz gráfica de usuario (Graphical User Interface GUI) es un tipo de interfaz de usuario que utiliza un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Como ejemplo podemos citar el escritorio o desktop de Windows y también Aqua de Mac OS X.

Algunas de las ventajas de este tipo de interfaz son:

- Permite la comunicación de forma rápida e intuitiva representando gráficamente los elementos de control y medida.
- Ofrece una estética mejorada y un ambiente más amigable.
- Permite observar varios procesos a la vez mediante ventanas.

Algunas de sus desventajas son:

- Mayor consumo de recursos computacionales

- Mayor vulnerabilidad por su complejidad
- Reducción de la funcionalidad ofrecida.

3.1.3 Software Libre vs. Software con Licencia Comercial

El software libre (free software) es la denominación del software que brinda libertad a los usuarios sobre su producto adquirido y por tanto, una vez obtenido puede hacer uso de las cuatro libertades de los usuarios del software:

- La libertad de usar el programa, con cualquier propósito.
- De estudiar el funcionamiento del programa y adaptarlo a las necesidades;
- De distribuir copias.
- De mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie.

El software libre puede estar disponible de forma gratuita o bajo algún costo ya que su denominación de libre se debe a que se trata de programas de código abierto (Open Source) y no tiene relación con su costo. De acuerdo a su forma de distribución podemos encontrar términos como Freeware (Todo aquel programa que se distribuya sin ningún costo adicional), Shareware (Programas que se distribuyen con limitaciones, o bien como versión de demostración o de evaluación, con funciones o características limitadas o con un uso restringido a un límite de tiempo) y Adware (Programas gratuitos en su totalidad pero que incluyen publicidad en su programa).

Entre las principales ventajas del software libre se encuentran:

- Bajo costo de adquisición y libre uso.
- Innovación tecnológica.
- Requisitos de hardware menores y durabilidad de las soluciones.
- Escrutinio público.
- Independencia del proveedor.
- Adaptación del software.

Mientras que algunas de sus desventajas son:

- La curva de aprendizaje es mayor.
- El software no tiene garantía proveniente del autor.
- Las interfaces amigables con el usuario (GUI) y la multimedia apenas se están estabilizando.
- La mayoría de la configuración de hardware no es intuitiva.
- Únicamente los proyectos importantes y de trayectoria tienen buen soporte, tanto de los desarrolladores como de los usuarios.
- La diversidad de distribuciones, métodos de empaquetamiento, licencias de uso, herramientas con un mismo fin, etc., puede crear confusión.

Por otro lado el software no libre o comercial posee las siguientes características:

- Tiene licencias, las cuales están limitadas por los usuarios y tienen un costo. Estas licencias restringen las libertades de los usuarios para usar, modificar, copiar y distribuir el software.
- El desarrollo, programación y actualización de este software solo lo hace la empresa que tiene los derechos. Esto ocasiona retrasos tecnológicos y poca creatividad en los productos.
- El futuro del software que compro el usuario solo depende de una empresa comercial.

Algunas de sus principales ventajas son:

- Control de calidad.
- Recursos a la investigación.
- Personal altamente capacitado.
- Uso común por los usuarios.
- Software para aplicaciones muy específicas.
- Amplio campo de expansión de uso en universidades.
- Difusión de publicaciones acerca del uso y aplicación del software.

Pero dentro de sus desventajas se encuentran:

- Código fuente oculto
- Soporte técnico ineficiente.
- Ilegal o costosa la adaptación de un módulo del software a necesidades particulares.
- Derecho exclusivo de innovación.
- Ilegalidad de copias sin licencia para el efecto.
- Imposibilidad de compartir.
- Descontinuación de una línea de software.
- Dependencia a proveedores.

3.2. Estudio de Factibilidad de las Características del Sistema

Ahora que ya se conocen las características, ventajas y desventajas de distintos factores que se deben considerar para el diseño del sistema a continuación se muestran tres tablas en las que se evalúan algunos requerimientos importantes para el desarrollo del Sistema para la Administración de Pagos Internos y Externos de un Centro de Capacitación Computacional (SAPIECC).

Cuando se ha entendido claramente el problema a solucionar el primer paso antes de empezar a desarrollar el programa es establecer la perspectiva o enfoque bajo el cual se implementara la solución. Para conocer y evaluar los dos enfoques de programación más populares se presenta la siguiente tabla.

Programación Estructurada vs. Programación Orientada a Objetos

<i>Enfoque</i>	Claridad de los programas	Detección de errores	Costo de mantenimiento	Rapidez y sencillez de los programas	Documentación
Programación Estructurada	Estructura secuencial. Fácil de entender	Facilidad para detectar errores	Bajo costo de mantenimiento.	Programas rápidos y sencillos debido a la estructura secuencial.	Programas sencillos de documentar debido a su estructura.
Programación Orientada a Objetos	Propiedades y métodos separados. Por lo que es difícil de comprender.	Debido a su estructura es fácil detectar errores.	Bajo costo de mantenimiento debido a su código reutilizable	Programas sencillos una vez que se entiende la metodología y más lentos conforme aumentan las clases y métodos	Programas complejos de documentar debido a su estructura y a la inserción de código reutilizable.

Tabla 3.1

Haciendo un análisis de la información mostrada en la tabla anterior podemos observar que la programación estructurada tiene varias ventajas tales como la claridad en los programas desarrollados bajo este enfoque y la facilidad en cuanto a la documentación, el diseño de pruebas y el mantenimiento. Todo esto debido a su estructura secuencial. Sin embargo en contraparte el enfoque de programación orientada a objetos aunque requiere de mayor complejidad para su entendimiento es más eficiente al corregir errores ya que existe una clara división entre objetos de datos y funciones, además de facilitar la reusabilidad de su código para otras aplicaciones, aunque esto mismo genera dificultad para documentar los programas.

Como ya se mencionó un enfoque de programación nos permite definir como se atacará o dividirá un problema para darle solución. Sin embargo una vez definido esto es importante establecer la forma en que los usuarios finales de un sistema van a interactuar con él, es decir se comunica mediante texto en una consola o a través de una interfaz gráfica. Para facilitar la elección entre la forma de interactuar con el usuario se presenta la siguiente tabla

Modo consola vs. Modo Gráfico

Enfoque	Recursos que utiliza	Rapidez en la respuesta	Facilidad de Aprendizaje(usuarios)	Estética	Multitarea
Modo Consola	Pocos Recursos	Pocos ms para desplegar un resultado.	Lenta y difícil ya que el usuario debe memorizar todos los comandos y parámetros para interactuar con el sistema	Pantalla oscura sin elementos gráficos	Solo se puede realizar un proceso a la vez.
Modo Gráfico	Entre más avanzados, requiere más espacio en disco duro y RAM	Ligeramente mayor que en modo consola porque debe cargar todos los elementos gráficos en memoria.	Intuitivo y amigable, el programa te lleva de la mano para realizar cualquier procedimiento a través de menús y botones	Contiene imágenes, botones y menús.	El usuario puede realizar varias acciones mediante ventanas.

Tabla 3.2

En la tabla 3.2 podemos observar que entre las principales ventajas del modo consola se encuentran la poca demanda de recursos y su rapidez de respuesta aunque su apariencia no es muy agradable como lo son las interfaces gráficas las cuales además facilitan al usuario su uso y entendimiento.

Una vez establecida la forma en que se desea interactuar con el sistema (consola ó gráficamente), es importante elegir la naturaleza del software que se utilizará. De acuerdo a su origen el software se divide principalmente en software libre y software de propietario. Para facilitar una elección entre algunos de estos tipos de software se proporciona la siguiente información.

Software Libre vs. Software Propietario

Enfoque	Costo	Requisitos de Hardware	Independencia de Proveedor	Adaptabilidad	Control de Calidad	Difusión de Publicaciones
Software Libre	Bajo costo de adquisición y en ocasiones ninguno	Generalmente bajos. Aunque su configuración a veces no es intuitiva.	No depende de una persona en particular. Se puede descargar de varios sitios.	Dado que muchos trabajadores trabajan en el mismo proyecto se adapta rápidamente	Bajo debido a las múltiples y constantes modificaciones	No tienen mucha difusión. Debido a los diversos y constantes cambios es difícil estandarizar alguna versión
Software Propietario	Incrementa según la versión	Conforma aumenta la versión requiere mayor espacio en disco y RAM.	Dependen de un solo proveedor.	Lenta, ya que solo el dueño del software las genera conforme él considera necesario.	Alto, ya que debe pasar por pruebas y métricas estandarizadas y cuidadosamente revisadas	Debido a su popularidad tienen un gran número de publicaciones en el mercado y en internet.

Tabla 3.3

Como podemos observar en la tabla 3.3 el software libre presenta muchas ventajas frente a software propietario tales como el costo, la adaptabilidad a distintas situaciones y plataformas así como su baja demanda en cuanto a hardware. Sin embargo debido a que se desarrolla por varias personas en distintos lugares muchas veces no tiene mucha literatura de apoyo. Mientras que en contraparte el software de propietario pasa por un control de calidad y el proveedor es el encargado de difundirlo y dar soporte. Aunque es importante mencionar que dada la tendencia actual el software cada vez requiere mayor espacio en disco y RAM. Además de que muchas veces se instala sólo para un tipo de plataforma de trabajo.

3.3. Definición Formal de la Estrategia a Utilizar

Considerando todas las ventajas y desventajas de los distintos enfoques que se mencionaron en la sección anterior y considerando los requerimientos del sistema (SAPIECC) que se definieron en capítulos anteriores. Para el diseño de este sistema se utilizara un enfoque de programación estructurado ya que la claridad y sencillez del sistema, así como una correcta documentación del proyecto y un bajo costo de pruebas y mantenimiento es importante, en especial si se considera alguna ampliación futura del sistema. Además todo el sistema será diseñado por un solo programador quien será el encargado de diseñar, modificar y dar mantenimiento al sistema. Considerando estos factores el enfoque de programación estructurada en su modalidad de programación modular es apropiado para desarrollar el software que cumpla con todos los requerimientos del sistema (SAPIECC).

La forma en que interactuará el usuario con el sistema será a través de una interfaz gráfica debido a que como podemos recordar en la sección de requerimientos del sistema se mencionó que una de las principales virtudes que se busca en el sistema es que sea amigable y fácil de utilizar. Por tal motivo además de que la imagen que da el manejo de este tipo de sistemas en un centro de capacitación se opta por diseñar un sistema que trabaje en modo gráfico ya que aunque los recursos que exige son mayores y su velocidad de respuesta es levemente menor. Las ventajas en cuanto a facilidad de manejo, usando botones en vez de comandos y los permisos para realizar varias tareas a la vez a través de ventanas, así como lo agradable de la vista de este tipo de sistemas son de mayor importancia para el centro.

Finalmente debido a que es necesario que el sistema funcione correctamente en distintas plataformas de desarrollo, sobre equipos con diferentes características técnicas y en especial ya que el centro no desea realizar gastos por concepto software de aplicación el tipo de software que se utilizará será libre.

3.4. Prioridades para las Características del Sistema

Debido a las características del centro de capacitación y a los requerimientos del sistema (SAPIECC) el modelo de ingeniería de software que se utilizará para el desarrollo de este proyecto será el modelo en espiral propuesto por Boehm. Esto debido a que el sistema se planea ir desarrollando de forma paulatina mediante tareas realizadas periódicamente.

Es posible definir de acuerdo a los datos que se observan en la sección de requerimientos del sistema que el desarrollo de SAPIECC se puede iniciar con el módulo de Clientes, el cual tendría como objetivo únicamente almacenar la información de los distintos tipos de cliente de acuerdo a los requerimientos y restricciones que ya se mencionaron anteriormente. Enseguida y una vez que se siguieron todos los pasos del modelo de ingeniería de software que se escogió para la implementación de este módulo. Se puede continuar con el módulo de cursos el cual tendría el objetivo de almacenar todos los cursos y relacionar estos con los datos de los clientes ya registrados en el sistema. Posterior a esto se desarrollará el módulo de Pagos, el cual almacenará la información de los distintos comprobantes de pago de los clientes con respecto a los cursos que han tomado. El paso siguiente será desarrollar el módulo de Profesores y enseguida el de Grupos, este orden se debe a que este módulo relaciona información de los módulos anteriores. Enseguida se desarrollará el módulo de consultas y finalmente el de ayuda y ajustes.

Como se puede observar toda esta información está ligada y para el desempeño óptimo del sistema final es recomendable seguir esta secuencia de módulos pero siguiendo cada uno de los pasos del modelo de ingeniería en la implementación de cada uno de los módulos.

3.5. Criterios de Aceptación del Sistema

El sistema (SAPIECC) se considerará aceptable cuando su funcionamiento permita generar al menos los mismos resultados que se obtienen ahora utilizando herramientas distintas. Es decir el sistema debe cumplir con todos los aspectos que se describen en la sección de especificación de requisitos, así como con los requerimientos del sistema y la funcionalidad del mismo.

El sistema deberá también haber concluido satisfactoriamente los diferentes tipos de pruebas.

Finalmente antes de considerarse el sistema como concluido se brindaran dos sesiones de capacitación a los usuarios. La primera para que se familiaricen con el uso del sistema y la siguiente después de un periodo de tiempo que se acordará y que únicamente tendrá como objetivo aclarar dudas operativas por parte de los usuarios. Aún con esto es el sistema contará con su propio manual de usuario, el cual se entregará junto con el sistema terminado.

IV HERRAMIENTAS A UTILIZAR

4.1. Sistema Manejador de Bases de Datos

4.1.1 Características Principales

Para cualquier empresa, sin importar su giro o tamaño la información es uno de los elementos que necesita para su supervivencia y prosperidad. Con el paso del tiempo y como consecuencia del crecimiento económico y demográfico surge la necesidad creciente de administrar grandes cantidades de datos interrelacionados con fines comerciales y administrativos. Es así como surgen las bases de datos, las cuales podríamos definir de manera sencilla como un conjunto de datos relacionados entre sí almacenados bajo una estructura perfectamente definida. Esta estructura puede ser física como un archivero o lógica como un sistema de informático.

Como podemos recordar un Sistema Manejador de Base de Datos (Data Base Management System, DBMS) es una colección de rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica.

Para administrar y manipular eficientemente la información que maneja el centro de capacitación es importante que SAPIECC cuente con un sistema manejador de bases de datos que cumpla con los requerimientos mencionados anteriormente. Por tal motivo a continuación se mencionan algunas características de algunos de los DBMS libres más populares.

Apache Derby

- Apache Derby es un sistema gestor de base de datos relacional escrito en Java que puede ser embebido en aplicaciones Java y utilizado para procesos de transacciones online.
- Soporta cifrado completo, roles y permisos. Además posee SQL SCHEMAS para separar la información en una única base de datos y brindar control completo a usuarios.
- En su modo embebido solo soporta un único proceso en la base de datos. Sin embargo en su modo cliente servidor soporta el acceso de varios procesos simultáneos mediante el bloqueo de filas.
- Tiene un tamaño de 2 MB de espacio en disco.
- Inicialmente distribuido como IBM Cloudscape, Apache Derby es un proyecto open source licenciado bajo la Apache 2.0 License. Actualmente se distribuye como Sun Java DB.

Berkeley DB (DB)

- Berkeley DB es una base de datos embebida con API para C, C++, Java, Perl, Python, Ruby, TCL y muchos otros lenguajes.
- Soporta múltiples datos para una misma clave ya que los datos se almacenan en el formato nativo del lenguaje de programación.

- Berkeley DB cuenta con cache configurable para modificar el rendimiento. Además permite crear bloqueos de forma detallada. Y permite realizar copias de seguridad y replicación rápidamente.
- Berkeley DB permite miles de hilos de control manipulando bases de datos de hasta 256 terabytes en muchos sistemas, incluidos la mayoría de los tipo-UNIX y Windows, e incluso sistema operativos de tiempo real.
- Berkeley DB pertenece y es desarrollado por la compañía Sleepycat Software. Está disponible con código fuente y licencia de libre distribución (free software).

MySQL

- MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario subsidiaria de Sun Microsystems como software libre en un esquema de licenciamiento dual.
- MySQL funciona en diferentes plataformas.
- MySQL tiene la facilidad de añadir otro sistema de almacenamiento. Esto es útil si desea añadir una interfaz SQL para una base de datos propia. Además las funciones SQL están implementadas usando una librería altamente optimizada.
- El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor. También está disponible como biblioteca y puede ser incrustado (linkado) en aplicaciones autónomas. Dichas aplicaciones pueden usarse por sí mismas o en entornos donde no hay red disponible.
- MySQL se puede considerar eficiente en aspectos como seguridad, escalabilidad y límites.
- Los clientes pueden conectar con el servidor MySQL usando sockets TCP/IP en cualquier plataforma. La interfaz para el conector ODBC (MyODBC) proporciona a MySQL soporte para programas clientes que usen conexiones ODBC (Open Database Connectivity). La interfaz para el conector J MySQL proporciona soporte para clientes Java que usen conexiones JDBC. Estos clientes pueden ejecutarse en Windows o Unix.
- MySQL server tiene soporte para comandos SQL para chequear, optimizar, y reparar tablas. Estos comandos están disponibles a través de la línea de comandos y el cliente mysqlcheck.

PostgreSQL

- PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD. Dirigido por el PGDG (PostgreSQL Global Development Group).
- PostgreSQL permite alta concurrencia y amplia variedad de tipos nativos
- PostgreSQL posee claves ajenas también denominadas Llaves ajenas o Claves Foráneas (foreing keys), Disparadores (triggers) y Vistas.
- PostgreSQL brinda integridad transaccional.
- Permite herencia de tablas, diferentes tipos de datos y operaciones geométricas.

4.1.2 Ventajas y Desventajas

Una de las principales virtudes que busca tener SAPIECC es la exactitud en los resultados que arroja, así como su rapidez y la capacidad de que en un momento dado este sistema se pueda expandir hacia las distintas sucursales que maneja el centro manteniendo un control centralizado de la información.

Por tales razones se busca que el Sistema Manejador de Bases de Datos que se utilice cuente con la tecnología cliente/servidor, así como la propiedad de ser multiplataforma, ya que no siempre se podrá contar con sistemas operativos de licencia comercial en la computadora que de soporte al sistema. Otra característica que podemos buscar es la seguridad de los datos y su compatibilidad con lenguajes de programación apropiados para las rutinas que se van a implementar para cumplir con las características del sistema que se establecieron en capítulos anteriores.

Considerando los requerimientos del sistema y las características que se acaban de mencionar acerca de distintos DBMS podemos realizar un estudio de factibilidad tomando en consideración si cumplen o no con los requerimientos necesarios para ser utilizado como herramienta de software en el desarrollo del sistema.

En la tabla 4.1 se resumen algunas de las características que se acaban de mencionar respecto a los sistemas manejadores de bases de datos libres de la sección anterior.

Manejador de Bases de Datos	Compañía Distribuidora	Orientado a Redes e Internet	Multiplataforma	Seguridad en Datos	Lenguajes de Programación que soporta (API)
Apache Derby	Sun Java DB	Si	Si	Si	Java
Berkeley DB	Sleepycat Software	No	Si	Si	C, C++, Java, Perl, Python, Ruby, TCL.
MySQL	Sun Microsystems	Si	Si	Si	C, C++, Eifel, Java, Perl, PHP, Phyton, Ruby, TCL.
PosgresSQL	PosgresSQL Global Development Group	Si	Si	Si	C, C++, Gambas, Java, Perl, PHP, Phyton, Ruby, TCL, Scheme.

Tabla 4.1

Mediante esta tabla se puede observar que además de no tener un costo, tres de los cuatro DBMS cumplen con los requerimientos necesarios para soportar el diseño del sistema. Sin embargo MySQL y PostgreSQL proporcionan mayor apertura en cuanto a los lenguajes de programación que permiten manejar.

4.2. Lenguajes de Programación

4.2.1 Características Principales

Como se mencionó en el tema anterior la información nos permite conocer diferentes aspectos de cualquier empresa. Sin embargo para tener una ventaja real de este recurso no es suficiente almacenarla sino que requiere procesarla. Para esto se hace uso de diferentes aplicaciones, y en especial cuando se va a diseñar el sistema encargado de almacenar y procesar información se utiliza la categoría de software denominada software de desarrollo en donde se encuentran los diferentes lenguajes de programación.

Existen diferentes clasificaciones para los distintos lenguajes de programación que existen en la actualidad. Sin embargo para este proyecto nos enfocaremos en los lenguajes para desarrollar en la web (ya que el SAPIECC trabajara en línea), los cuales han ido surgiendo debido a las tendencias y necesidades de las plataformas.

Desde los inicios de Internet, fueron surgiendo diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. A medida que paso el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas que solucionar. Esto dio lugar a desarrollar lenguajes de programación para la web dinámicos, que permitieran interactuar con los usuarios y utilizar sistemas de Bases de Datos. A continuación se dará una breve descripción de los lenguajes de programación para la web más populares.

Lenguaje HTML

- Desde el surgimiento de internet se han publicado sitios web gracias al lenguaje HTML.
- Es un lenguaje estático para el desarrollo de sitios web (acrónimo en inglés de HyperText Markup Language, en español Lenguaje de Marcas Hipertextuales). Desarrollado por el World Wide Web Consortium (W3C).
- Los archivos pueden tener las extensiones (htm, html) . Además HTML tiene la ventaja de permitir diseñar páginas incluso con un simple editor de texto. Generando archivos pequeños de despliegue rápido.
- Lo admiten todos los exploradores.
- Las etiquetas son limitadas y en ocasiones guarda muchas de ellas que pueden convertirse en basura y dificultar la corrección.

Lenguaje Java Script

- Este es un lenguaje interpretado, no requiere compilación. Fue creado por Brendan Eich en la empresa Netscape Communications.

- Utilizado principalmente en páginas web. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias. La mayoría de los navegadores en sus últimas versiones interpretan código Java script. El código Java script puede ser integrado dentro de nuestras páginas web.
- Para evitar incompatibilidades el World Wide Web Consortium (W3C) diseñó un estándar denominado DOM (en inglés Document Object Model, en su traducción al español Modelo de Objetos del Documento).
- El código Java script se ejecuta en el cliente.
- Es un lenguaje seguro y fiable.

Lenguaje JSP

- Es un lenguaje para la creación de sitios web dinámicos, acrónimo de Java Server Pages. Está orientado a desarrollar páginas web en Java. JSP fue desarrollado por Sun Microsystems
- JSP es un lenguaje multiplataforma. Creado para ejecutarse del lado del servidor.
- Posee un motor de páginas basado en los servlets de Java pero para su funcionamiento se necesita tener instalado un servidor Tomcat.
- Permite separar la parte dinámica de la estática en las páginas web.
- Los archivos se encuentran con la extensión (jsp). Y el código JSP puede ser incrustado en código HTML.
- Los elementos que pueden ser insertados en las páginas JSP son los siguientes: código (se puede incrustar código “Java”), Directivas (permite controlar parámetros del servlet) y Acciones (permite alterar el flujo normal de ejecución de una página).

Lenguaje Python

- Es un lenguaje de programación creado en el año 1990 por Guido van Rossum, es el sucesor del lenguaje de programación ABC. Algunos usuarios lo consideran como un lenguaje más limpio para programar.
- Permite la creación de todo tipo de programas incluyendo los sitios web.
- Su código no necesita ser compilado, por lo que se llama que el código es interpretado.
- Es un lenguaje de programación multiparadigma, lo cual fuerza a que los programadores adopten por un estilo de programación particular: Programación orientada a objetos, Programación estructurada, Programación funcional, Programación orientada a aspectos.
- Es un lenguaje de propósito general, portable, orientado a objetos, que posee gran cantidad de funciones y librerías.
- El multiplataforma y maneja licencia de código abierto.

Lenguaje Ruby

- Es un lenguaje interpretado de muy alto nivel y orientado a objetos.
- Desarrollado en el 1993 por el programador japonés Yukihiro “Matz” Matsumoto. Su sintaxis está inspirada en Python, Perl.
- Es distribuido bajo licencia de software libre (OpenSource).
- Ruby es un lenguaje dinámico para una programación orientada a objetos rápida y sencilla.
- Es multiplataforma y maneja licencia de código abierto.

- Para los que deseen iniciarse en este lenguaje pueden encontrar un tutorial interactivo de Ruby. Se encuentra también a disposición de estos usuarios un sitio con informaciones y cursos en español.

Lenguaje PHP

- PHP es un acrónimo recursivo que significa “PHP Hypertext Pre-processor”, (inicialmente se llamó Personal Home Page). Surgió en 1995, desarrollado por PHP Group.
- PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse.
- Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP.
- La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas.
- Los archivos cuentan con la extensión (`php`).
- El multiplataforma y maneja licencia de código abierto.
- PHP cuenta con la capacidad de conexión con la mayoría de los manejadores de base de datos.
- Es un lenguaje de propósito general, orientado a objetos y que posee gran cantidad de funciones.

4. 2.2 Ventajas y Desventajas

Para el desarrollo de SAPIECC es importante considerar distintos aspectos tales como que el lenguaje de programación que se elija debe ser multiplataforma, ya que no debe existir problemas debidos a la naturaleza del sistema operativo en donde se instale el sistema. Además dado que los usuarios van a interactuar con el sistema y este responderá a algunas de sus peticiones es importante que el lenguaje que se elija permita realizar páginas dinámicas. Otra característica que debe tomarse en cuenta para elegir el lenguaje es su capacidad para conectarse con un sistema de base de datos y la facilidad de implementar funciones.

De acuerdo a esto y en base a las características que se acaban de presentar respecto a distintos lenguajes de programación para web tenemos los siguientes resultados.

Lenguaje de Programación	Estático / Dinámico	Multiplataforma	Libre	Comunicación con DBMS	Lenguaje de Propósito General
HTML	Estático	Si	Si	No	No
Java Script	Estático	Si	Si	No	No
JSP	Dinámico	Si	Si	Si	Sí
Phyton	Dinámico	Si	Si	Sí	Sí
Ruby	Estático	Si	Si	No	No
PHP	Dinámico	Si	Si	Si	Si

Tabla 4.2

En esta tabla se puede observar que los lenguajes de programación que cumplen con los requerimientos para ser usados en el desarrollo de SAPIECC son JSP, Phyton y PHP. Sin embargo para tomar una decisión definitiva se tomaran en cuenta aspectos como la compatibilidad con los manejadores de bases de datos disponibles para el desarrollo del sistema, entre otros aspectos más específicos.

4.3. Herramientas Seleccionadas

Aunque PHP se utiliza en su mayoría para aplicaciones para la web, para muchos es un lenguaje de propósito general y su uso dependerá en gran parte de las necesidades del programador. En el caso del desarrollo de SAPIECC, se acopla perfectamente a las necesidades ya que entre algunas de las características de PHP se encuentra su capacidad para el desarrollo de aplicaciones de escritorio y sin duda la mayor virtud de PHP sobre sus competidores es las integración con los sistemas de bases de datos y el soporte nativo a distintas bases de datos existentes libres y comerciales. Las razones principales que argumentan los desarrolladores en PHP para su uso son:

- Evitar redundancias
- Evitar programas complicados
- Búsquedas
- Seguridad
- Arquitectura n-tier

Una arquitectura cliente/servidor es una 2-tier, una n-tier disgrega aún más funciones, por ejemplo en la web tenemos una 3-tier

1. Visualización: Navegador Web
2. Lógica: Servidor Web + programas o scripts PHP
3. Almacenamiento de datos: Base de datos

La comunicación entre el tier-1 y el tier-2 es a través de http como soporte de comunicación y HTML como representación de datos. La comunicación entre el tier-2 y el tier-3 es a través del middleware, en nuestro caso PHP y las funciones de MySQL que se conectan al servidor aunque también pueden hacerse mediante ODBC.

Además PHP tiene numerosas funciones que se compaginan perfectamente con MySQL. Por ejemplo PHP tiene funciones que permiten al programador acceder a las diferentes estructuras que conforman la base de datos y la implementación de PHP y MySQL permite realizar conexiones simultáneas y todo tipo de consultas para obtener información.

Otra característica importante de la implementación PHP y MySQL es que a partir de la versión 5 de PHP se incluye soporte para SQLite que es una librería escrita en lenguaje C que implementa un motor de base de datos accesible por varios lenguajes (PHP, Python, C, etc.). SQLite no permite usuarios múltiples accediendo a la vez en modo escritura a la base de datos, debido a que el mecanismo de bloqueo que utiliza consiste en bloquear toda la base de datos. Además brinda soporte para un máximo de 2 terabytes de datos.

Sin embargo uno de los avances más importantes de PHP es la inclusión de MySQLi, una extensión que permite acceder a las funcionalidades provistas por MySQL a partir de la versión 4.1.2. Dos de sus principales características son: las consultas preparadas y la seguridad en las conexiones.

Las consultas preparadas son una especie de molde o plantilla que PHP manda a MySQL con una serie de instrucciones. MySQL lo valida y lo guarda para que posteriormente PHP no tenga que enviar la serie de instrucciones completa sino que pueda enviarle solo los datos que podrían cambiar de una instrucción a otra. Esto disminuye las repeticiones.

Estas son las principales características por las cuales se eligió utilizar la implementación PHP y MySQL para el desarrollo de SAPIECC, ya que la combinación de los otros lenguajes con MySQL o PostgreSQL no brindan ventajas como estas. Además de que el uso de esta combinación en muchos sistemas proporciona confianza acerca de lo consistente y seguro de esta implementación.

V. ANÁLISIS DEL SISTEMA

5.1. Planeación y Definición de Requisitos

Uno de los principales factores que generan retrasos, incremento de costos y baja calidad en el desarrollo de cualquier producto es la falta de planeación. El software no es la excepción ya que según los expertos aunque es difícil establecer una metodología para resolver un problema, el hecho de tener una serie de lineamientos para desarrollar la solución facilita el proceso de software además de garantizar un producto de calidad.

Por facilidad el análisis del sistema SAPIECC se dividió en dos partes estableciendo en la primera sección la planeación del proyecto, es decir las estrategias metodológicas, tecnológicas y de análisis que se utilizarán para desarrollar el sistema, mientras que en la segunda parte se mencionan las características propias que deberá cumplir el sistema a través de la especificación de requisitos.

5.1.1 Plan de Proyecto

Antes de iniciar con la construcción del cualquier sistema es de gran utilidad definir algunos lineamientos del proyecto tales como el modelo de ciclo de vida, la estructura organizacional, los requisitos preliminares de personal y recursos, los mecanismos de supervisión y control del proyecto, las herramientas y técnicas que se emplearan, las formas de demostración y entrega, y algunas consideraciones de mantenimiento del sistema. Estos factores como podemos observar no indican cómo dar solución a un problema de la vida real sino que establecen cuál será el enfoque que se seguirá en distintos aspectos para encontrar una solución apropiada y lograr que el proyecto llegue a su objetivo final.

En el caso del sistema (SAPIECC) la metodología para el desarrollo del software presenta las características que se mencionan a continuación.

Modelo del ciclo de Vida

De acuerdo al análisis que se realizó en las secciones anteriores el sistema (SAPIECC) cuenta con ciertas características que permiten su desarrollo de manera secuencial. Es decir para obtener el sistema con la funcionalidad completa es necesario iniciar con un módulo que permita almacenar los datos de los clientes para que posteriormente se les pueda asignar un curso el cual podrá ser pagado y de acuerdo al número de ocurrencias similares permitirá formar un grupo que será asignado a un profesor. Como se puede ver el sistema se va formando de pequeñas aplicaciones que se van enlazando y agregando de manera paulatina. Es por este conjunto de características que el modelo de ciclo de vida ó modelo de proceso de software que se eligió para el desarrollo del sistema es un modelo en espiral cuya principal virtud es que el software se desarrolla en versiones incrementales cada vez más completas. Sin embargo aunque el sistema se desarrolle de forma

evolutiva cada nueva versión seguirá las fases de análisis, diseño, instrumentación, pruebas y mantenimiento.

Estructura Organizacional

Una vez definida la estrategia que se utilizará para el desarrollo del sistema es importante mencionar las características que tendrá el equipo de personas que seguirán dicha estrategia, es decir cómo estarán organizados los encargados de desarrollar el software.

El sistema será desarrollado por un sólo equipo de trabajo que se hará cargo del proyecto a lo largo de todas sus etapas. Esta estructura se conoce como Formato del Proyecto, además dentro del equipo no hay un líder definido sino que se tiene una jerarquía igual para todos los desarrolladores. Esto quiere decir que la estructura de programación corresponde a un grupo democrático.

Requisitos Preliminares de Personal y Recursos

Considerando a cada miembro del equipo de forma individual es importante mencionar que cada uno de los desarrolladores que participen en el sistema (SAPIECC) debe cumplir con algunas características tales como poseer capacidad de análisis y solución de problemas a través de soluciones informáticas, así como manejar el lenguaje de programación PHP y el manejador de base de datos MySQL (debido a que son las herramientas que se eligieron para desarrollar el sistema), otra característica importante para cada miembro del equipo es conocer y comprender a fondo la problemática del centro de capacitación computacional.

Por otro lado cada integrante necesita ciertos materiales para desarrollar el sistema entre los cuales destacan que cada uno de ellos debe tener al menos una computadora con los requerimientos necesarios para soportar PHP y MySQL, los cuales deberá de tener instalados y funcionando correctamente. Así también deberá contar con programas auxiliares como Project, Visio, Power Point, Word y Excel. Además de diferentes dispositivos de almacenamiento externo.

Mecanismos de Supervisión y Control del Proyecto

Aunque la metodología permite ir dividiendo el proyecto final en pequeñas versiones hasta llegar al producto de software final, el objetivo primordial es alcanzar la funcionalidad completa para el software con respecto a las necesidades del centro, para lo cual es necesario llevar un control de las actividades que se realizan en cada fase, esto implica revisar ciertos aspectos fundamentales del sistema para garantizar la claridad tanto de parte del cliente como del desarrollador por tal motivo en cada fase del proceso se revisarán algunos documentos para garantizar el avance correcto a lo largo de todo el proyecto así como para llevar un mejor control de las actividades realizadas. A continuación se mencionan cuáles son los documentos que se verificarían al final de cada fase.

Fase	Documento
Análisis	Definición de Requisitos
Diseño	Detalles Estructurales
Instrumentación	Código, Depuración, Pruebas de módulo
Pruebas	Integración, Definición de requisitos
Mantenimiento	Mejoras, Adaptación, Ajuste.

Herramientas y Técnicas que se emplearan

Como ya se mencionó en secciones anteriores el enfoque que se tiene para el desarrollo del sistema es a través de una metodología de programación estructurada en su forma modular. Utilizando un lenguaje de programación de distribución libre que permite desarrollar interfaces gráficas.

Para el desarrollo del sistema principalmente se utilizaran dos herramientas informáticas, un sistema para la administración de bases de datos (MySQL) y un lenguaje de programación para páginas dinámicas (PHP).

Estas características se definieron como resultado de los estudios realizados en los capítulos anteriores.

Formas de Entrega

Una vez concluido diseño del sistema y después de haber cumplido satisfactoriamente con las pruebas, el sistema (SAPIECC) se entregará en un disco compacto para su instalación. Todas las características especiales relacionadas con este aspecto se especificarán en el manual de usuario en la sección correspondiente a instalación.

Consideraciones de Mantenimiento

Finalmente el último aspecto que se considera para organizar el desarrollo del sistema a través del proceso de software es el mantenimiento, este se realizará de manera preventiva dos veces al año y correctiva según lo requiera el sistema. Esto debido a la gran cantidad de información que se almacena durante seis meses de uso del sistema y a que coincide con los dos periodos vacacionales que tiene el centro. Mientras que la parte correctiva se refiere a los casos en que las respuestas producidas por el sistema no sean las que se solicitaron en la especificación de requisitos del sistema o bien cuando el centro solicite bajo previo aviso un cambio en la programación del sistema.

5.2. Definición de Requisitos

En el ámbito de la ingeniería de software la definición de requisitos se refiere a la identificación de las funciones básicas de los componentes de programación en un sistema de equipo/personal/programación. Poniendo atención en las funciones y restricciones bajo las cuales se debe desarrollar el sistema computacional. Para un mejor entendimiento de esto se hace uso de herramientas de análisis estructurado tales como diagramas entidad-relación, diagramas de flujo y diagramas de transición entre otras.

El producto de la definición de requisitos, es una especificación que describe el ambiente de procesamiento, las funciones requeridas de los programas, restricciones de configuración, manejo de excepciones, subconjuntos y prioridades de instrumentación, cambios probables y modificaciones factibles, así como los criterios de aceptación del producto de programación. Su objetivo es especificar total y consistentemente los requerimientos técnicos del producto de una manera concisa y sin ambigüedades.

Para definir los requisitos del sistema (SAPIECC) se muestra en las secciones siguientes el modelado de datos, el modelado funcional y el modelado del comportamiento. Esto con el objetivo de conocer a fondo la composición y comportamiento del sistema orientado a la creación del sistema computacional. Los requerimientos formales resultado de los diferentes análisis se establecen en la especificación de requisitos que se muestra al final del capítulo.

5.2.1 Modelado de Datos

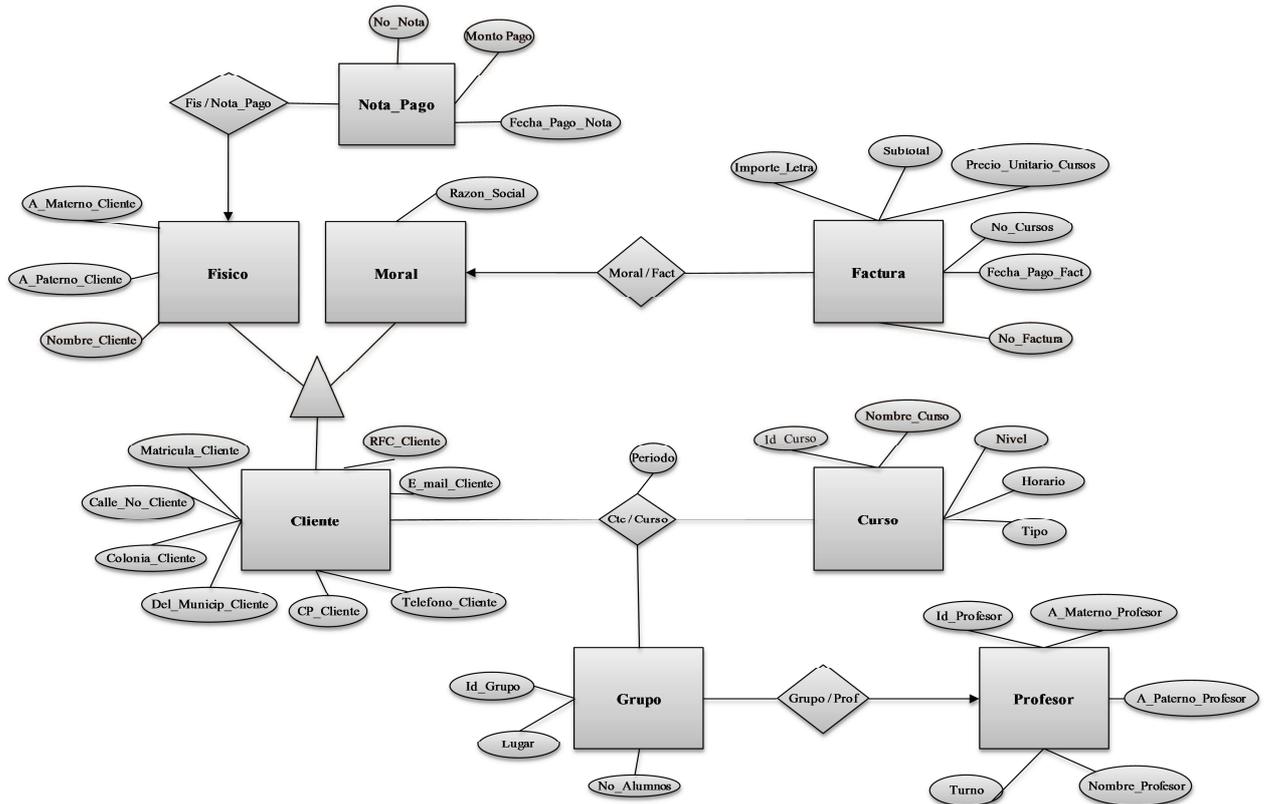
Para administrar los cursos que imparte el centro de capacitación computacional se requiere almacenar y manipular diferentes tipos de datos. Por ejemplo para dar de alta un cliente dependiendo de su tipo requiere que se proporcione información como su nombre, dirección, teléfono, e-mail, y RFC. Sin embargo para asignar un curso se solicita información como: nombre del curso, tipo, nivel y horario. Mientras que un pago requiere almacenar datos como cliente, monto y fecha.

Recordando que un sistema computacional se considera eficiente cuando es capaz de proporcionar los mismos servicios e información que el método actual usando menos tiempo y personal o cuando proporciona servicios e información que antes eran inaccesibles. Es importante definir algunas características del sistema relacionadas con los datos tales como ¿Cuáles son los objetos de datos que va a procesar el sistema? ¿Cuál es la composición de cada objeto de datos? ¿Cuál es la relación entre los objetos y los procesos que los transforman? Para responder a esto es necesario realizar el modelado de los datos.

Como ya se mencionó en la parte teórica los métodos de modelado de datos generalmente hacen uso de un diagrama entidad-relación (DER). Este diagrama considera los objetos de datos representados por un rectángulo, las relaciones existentes mediante un rombo y los atributos mediante círculos. Además la cardinalidad se indica mediante distintos tipos de flechas.

Para entender a detalle cuales son los datos que maneja el sistema (SAPIECC), como resultado del análisis de la problemática de la institución y de los requisitos que debe cumplir el sistema para generar una herramienta útil. A continuación se muestra una descripción de los objetos de datos y las relaciones que existen entre ellos utilizando un diagrama entidad –relación.

DIAGRAMA ENTIDAD-RELACIÓN SAPIECC



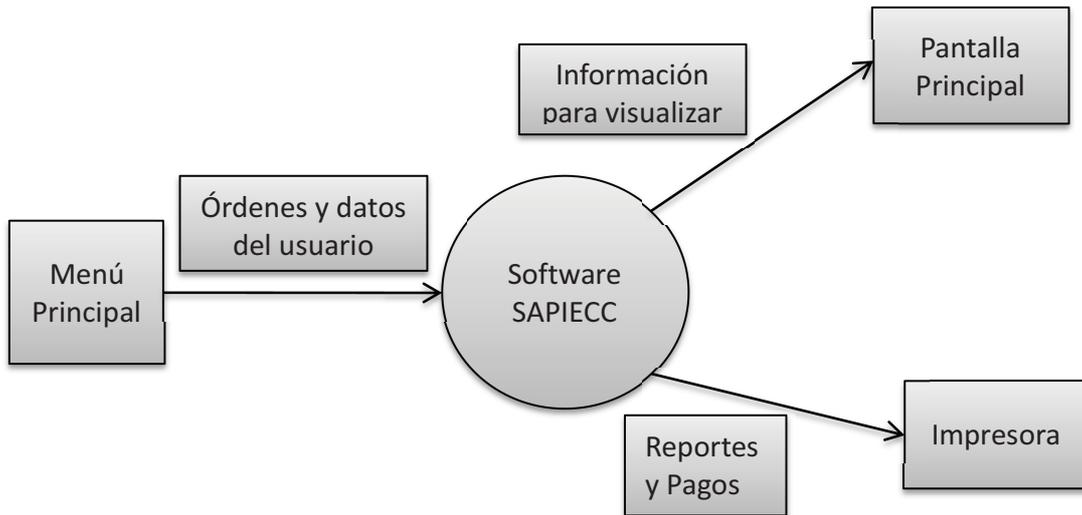
5.2.2 Modelado Funcional

Una vez definidos los datos que manipulará el sistema es importante recordar que de acuerdo a los requerimientos que este tiene los datos no sólo se almacenan de forma estática sino que se transforman a medida que se avanza en el sistema. Esto es posible gracias a las funciones (y subfunciones) que transforman el flujo de datos.

El análisis estructurado es una técnica de modelado del flujo y el contenido de la información, su principal herramienta es el diagrama de flujo de datos (DFD) el cual representa el flujo de información y las transformaciones que se aplican a los datos al moverse desde la entrada hasta la salida. Su notación utiliza un rectángulo para representar una entidad externa, esto es, un elemento del sistema u otro sistema que produce información para ser transformada por el software, o recibe información producida por el software. Un círculo representa un proceso o transformación que es aplicado a los datos y los modifica. Una flecha representa uno o más elementos de datos. Toda flecha en un diagrama de datos debe estar etiquetada. Las líneas en paralelo representan almacenamiento.

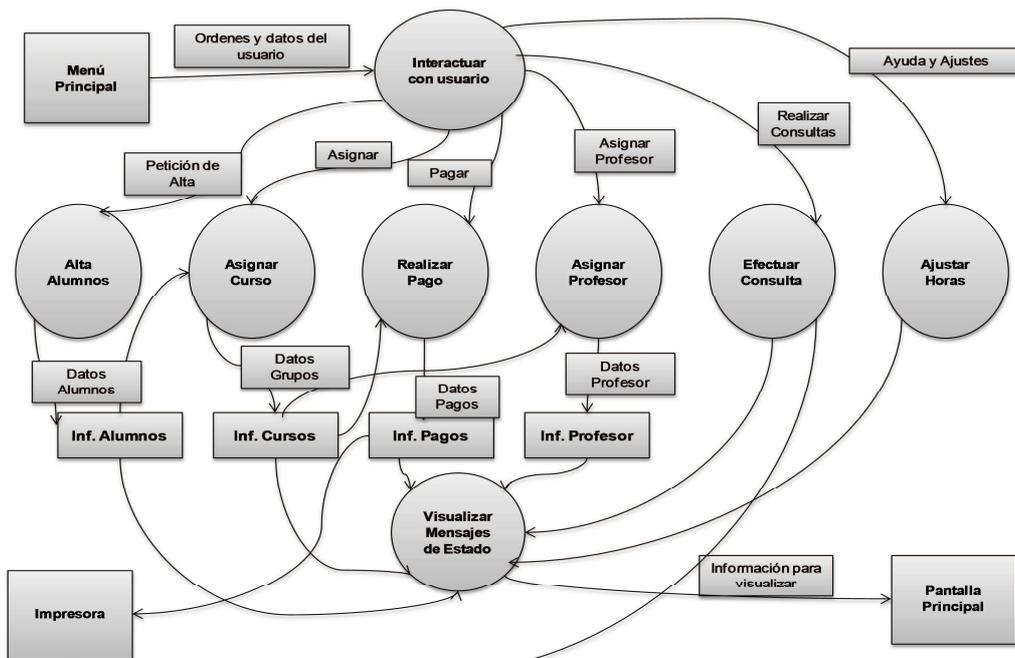
Recordando que un DFD de nivel 0, también denominado modelo fundamental del sistema o modelo de contexto, representa al elemento de software completo como una sola burbuja con datos de entrada y de salida representados por flechas de entrada y de salida, respectivamente. Aplicando esto a SAPIECC podemos observar en la figura cuál es su funcionamiento básico.

DFD DE NIVEL CONTEXTUAL DE SAPIECC



Para representar las subfunciones que realizara SAPIECC para cumplir con los requerimientos del centro se utilizara un DFD de nivel 1. Cada uno de los procesos representados en el nivel 1 es una subfunción del sistema general en el modelo del contexto.

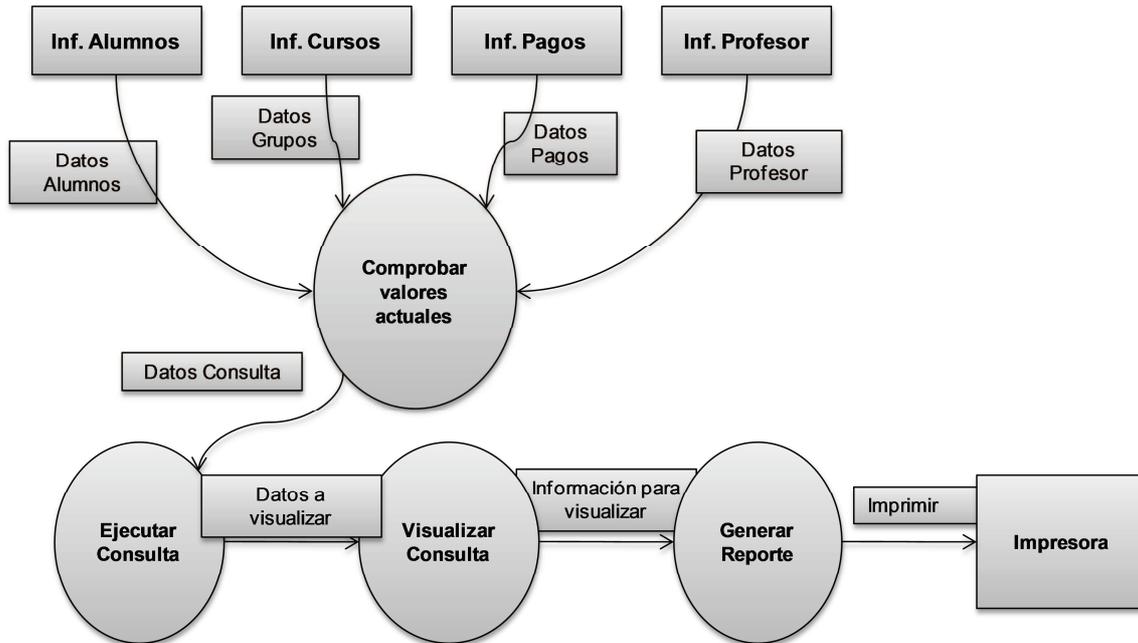
DFD DE NIVEL 1 DE SAPIECC



Se puede refinar cada una de las burbujas en distintos niveles para mostrar un mayor detalle. A continuación se muestra el DFD correspondiente a consultas del sistema SAPIECC.

DFD DE NIVEL 2 DE SAPIECC

EFECTUAR CONSULTAS



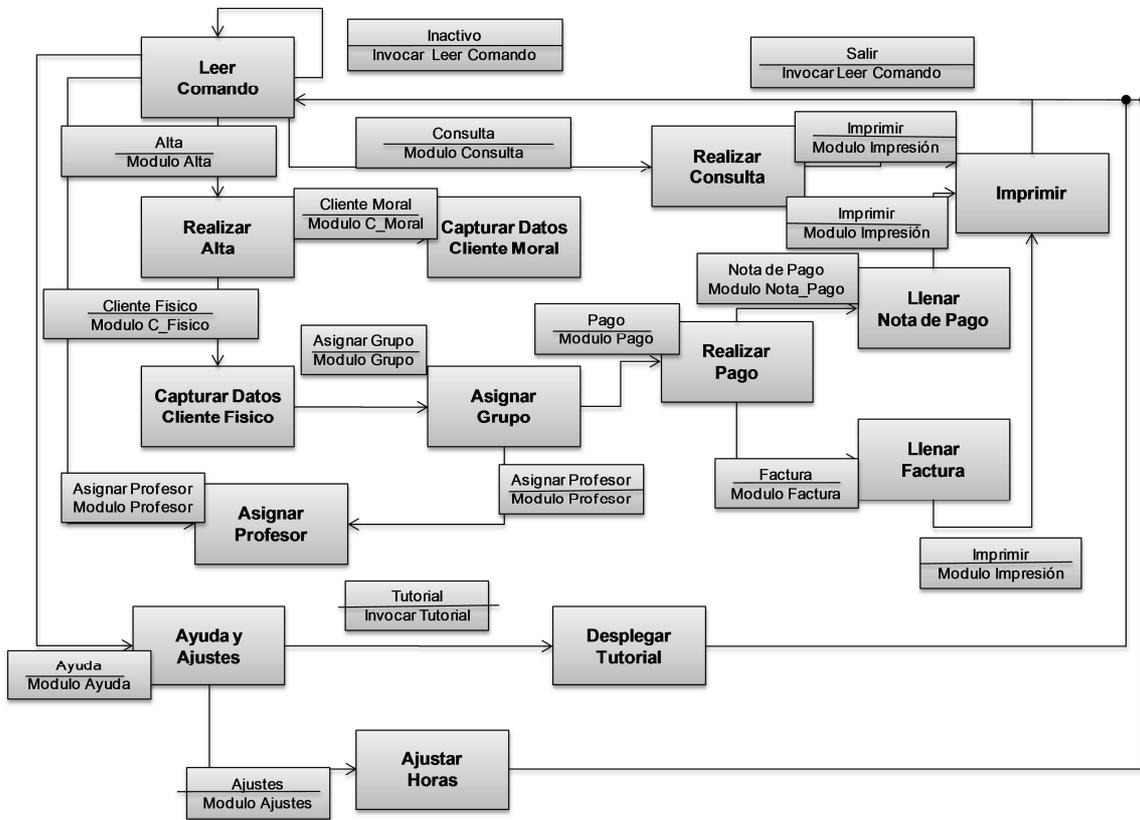
5.2.3 Modelado de Comportamiento

El modelado del comportamiento es parte fundamental de todos los métodos de análisis de requisitos. Para facilitar su desarrollo es común hacer uso de un diagrama de transición de estados, el cual, representa el comportamiento de un sistema que muestra los estados (podemos definir un estado como un modo observable de comportamiento) y los sucesos que hacen que el sistema cambie de estado. Además, el DTE indica qué acciones se llevan a cabo como consecuencia de un suceso determinado.

La notación que se utilizó para realizar es diagrama de transición de estados de SAPIECC que se muestra en la parte inferior es la siguiente: Los rectángulos representan los estados del sistema y las flechas representan las transiciones entre estados. Cada flecha esta etiquetada con una expresión en forma de fracción. La parte superior indica el suceso (o sucesos) que hace(n) que se produzca la transición. La parte inferior indica la acción que se produce como consecuencia del suceso.

Es importante mencionar que este comportamiento está basado en la descripción del sistema que se estableció en la sección de requerimientos del sistema.

DIAGRAMA DE TRANSICIONES DE SAPIECC



5.2.4 Especificación de Requisitos del Sistema

Como ya se mencionó en secciones anteriores la especificación de requisitos del sistema es una especificación técnica de los requisitos que debe cumplir el producto de software. Su objetivo es especificar total y consistentemente los requerimientos técnicos del producto de manera concisa y sin ambigüedades. Para lo cual se basa en la definición del sistema (requerimientos del sistema).

A continuación se presenta la especificación de requisitos del sistema (SAPIECC) dividida en cada una de las secciones que generalmente considera este documento.

Resumen y panorama del producto

Una Centro de Capacitación Computacional desea desarrollar un sistema para la administración de los cursos que imparte en cuanto a logística y pago. Actualmente para llevar a cabo esta administración se hace uso de distintas herramientas como hojas de cálculo y sistemas administrativos comerciales. Sin embargo se pretende que el sistema evite el uso de distintas aplicaciones por parte de los usuarios. Así como también se busca que los tiempos de respuesta se reduzcan y la generación de altas, asignaciones y diferentes tipos de reportes se ejecute de manera más rápida y sin errores.

Ambientes de desarrollo, operación y mantenimiento

El sistema (SAPIECC), será desarrollado por una sola persona quien seguirá una metodología en espiral para llevar a cabo el proceso de producción del software y quien hará uso principalmente de dos herramientas de software: un lenguaje de programación para páginas dinámicas (PHP) y un sistema de gestión para bases de datos (MySQL). Se pretende que el ambiente de desarrollo y operación sea el sistema operativo Windows pero dado que ambas herramientas son gratuitas y portables la posibilidad de que la plataforma de desarrollo o bien el ambiente de operación sea algún otro es abierta.

En un principio se pretende que el sistema se encuentre instalado en una computadora personal del departamento, por lo cual el sistema no tiene características y restricciones de aplicaciones compartidas o para redes, pero se pueden implementar posteriormente.

El mantenimiento preventivo del sistema se llevará a cabo dos veces al año, mientras que el correctivo se implementará cada vez que el sistema lo requiera.

Interfaces externas y flujo de datos

El sistema para la Administración de Pagos Internos y Externos de un Centro de Capacitación Computacional (SAPIECC) tendrá dos formas de salida de la información que maneja: resultados desplegados en la pantalla o mediante reportes impresos. En el caso de los reportes impresos estos siempre serán en tamaño carta y con un encabezado que contenga el nombre del centro, su dirección, y el título del reporte.

El flujo de datos que maneja el sistema se puede observar a detalle en el diagrama de flujo de datos. Mientras que el comportamiento que tendrá esta desglosado mediante el diagrama de transiciones del sistema.

La información detallada de los datos que se utilizaran en el sistema y que se muestran en el diagrama entidad-relación se encuentra almacenada en el diccionario de datos.

Requisitos funcionales

El sistema (SAPIECC) como ya se mencionó anteriormente tiene como objetivo facilitar la administración de sus cursos y mantener un mejor control de sus alumnos, profesores y pagos realizados. Para cumplir con esto el sistema deberá ser capaz de realizar las siguientes funciones.

El sistema contará con un módulo de alta para los alumnos, el cual será capaz de distinguir entre clientes físicos y morales y que será el mecanismo de entrada de todos los datos que corresponden a cada uno de estos.

El sistema contará también con un módulo de cursos, en el cual una vez dado de alta un cliente permitirá que se le asigne un curso de acuerdo a sus necesidades. Sin importar el tipo de cliente las características de los cursos son siempre las mismas.

Para llevar un control de los pagos del centro se diseñará un módulo de pagos el cual permitirá emitir facturas o notas de pago según sean las necesidades del cliente. Este módulo además de

Llevar un control de dichos documentos permitirá emitir los comprobantes necesarios (facturas o notas de pago).

Dado que es necesario llevar un control de los docentes que colaboran en la institución, se implementará un módulo de profesores, el cual tendrá como principal función asignar un maestro a cada grupo que se abra en un determinado periodo, horario y asignatura. Así como llevar un control de las horas trabajadas por cada profesor.

Además una de las principales novedades del sistema es que contará con un módulo de consultas, el cual permitirá observar en pantalla consultas tales como:

- Monto total de pagos realizados
- No. De alumnos inscritos por periodo
- No. De Grupos abiertos por periodo.
- No. De personas por grupo
- Saldo e historial de cada alumno sin importar su tipo.
- Saldo e historial de cada empresa
- Grupos asignados a cada profesor
- Horas trabajadas por cada profesor.
- Total de pagos de alumnos regulares
- Total facturado a empresas...

Finalmente el sistema (SAPIECC) contará con un módulo de Ayuda y Ajustes, el cual tendrá las capacidad de desplegar un tutorial para brindar ayuda o bien desplegar un módulo de ajustes para modificar las horas trabajadas por cada profesor.

Los datos que deberá almacenar cada módulo se definen a detalle en el diagrama entidad-relación del sistema. Mientras que la forma de interacción de cada una de estas funciones se puede visualizar en los distintos diagramas de flujo.

Subconjuntos iniciales y prioridades de instrumentación

Aunque todo el sistema está relacionado entre sí, considerando las necesidades del centro la jerarquía que deberá seguir para el desarrollo e implementación de los módulos es la siguiente: alta, cursos, pagos, profesores, consultas y finalmente ayuda y ajustes. Esta clasificación está basada en la importancia de almacenamiento de los datos y la forma en que se relacionan entre sí.

Modificaciones y mejoras previsibles

Se espera que en un futuro el sistema (SAPIECC) se adapte a equipos en red para que se pueda manipular desde otros equipos o bien a distancia. También se considera en un futuro y si es que el funcionamiento del sistema es aprobado por los usuarios agregar más módulos relacionados por ejemplo a datos completos de profesores o bien incluso asistencia y puntualidad para facilitar la detección de aquellos que merecen algún bono.

Criterios de aceptación

Para que el sistema se acepte es necesario verificar los siguientes aspectos:

El sistema (SAPIECC) deberá satisfacer todos los requisitos escrito en este documento y en todos aquellos diagramas auxiliares a éste. En especial lo referente a la funcionalidad y comportamiento. Además de ir acompañado de un manual de usuario que servirá de apoyo para los usuarios del sistema.

El sistema deberá cumplir con las especificaciones del diseño arquitectónico así como con el plan de pruebas que se especifican en esta fase del proyecto.

Además de las pruebas de código debe satisfacer las pruebas de unidad, integración, validación y de sistema.

El sistema debe de contar con la verificación de todos los documentos que se especificaron al inicio de este capítulo.

VI. DISEÑO

Una vez que se analizan y especifican los requisitos del software, el diseño es la primera actividad para construir y verificar el sistema. A partir de aquí cada acción transforma la información de manera que dé lugar a un software de computadora eficiente y validado.

Sin un diseño apropiado corremos el riesgo de construir un sistema inestable que fallará cuando se lleven a cabo diferentes tipos de cambios; un sistema que puede resultar difícil de comprobar; y cuya calidad no pueda evaluarse hasta muy avanzado el proceso.

Con el objetivo de garantizar que SAPIECC sea un sistema que funcione efectivamente y que además se encuentre documentado para futuras modificaciones a continuación se presentan las diferentes fases de diseño necesarias para lograr una funcionalidad óptima del sistema.

6.1 Diseño de Datos

En capítulos anteriores se llevó a cabo el análisis de la problemática que pretende resolver SAPIECC especificando sus características y restricciones. Sin embargo para empezar a definir una solución informática real es necesario establecer algunos lineamientos que el sistema debe seguir tales como ¿cuáles son los datos que el sistema va a procesar?, ¿cuáles son las características de los datos? y ¿en qué tipo de estructuras se van a almacenar los datos? entre otros detalles. Al definir esto será posible crear un modelo de datos capaz de definir cómo será representada la información con un alto nivel de abstracción. Es decir, el diseño de datos, transforma el modelo del dominio de información que se crea durante el análisis en las estructuras de datos necesarias para implementar el software. Para lograr esto los objetos de datos y las relaciones definidas en el diagrama entidad relación y el contenido de datos detallado que se representa en el diccionario de datos proporcionaran la base de la actividad del diseño de datos.

Si nos remitimos al diagrama entidad relación de SAPIECC que se presentó en la sección anterior y seguimos los pasos para realizar el mapeo del modelo entidad relacional al modelo relacional tenemos los siguientes esquemas.

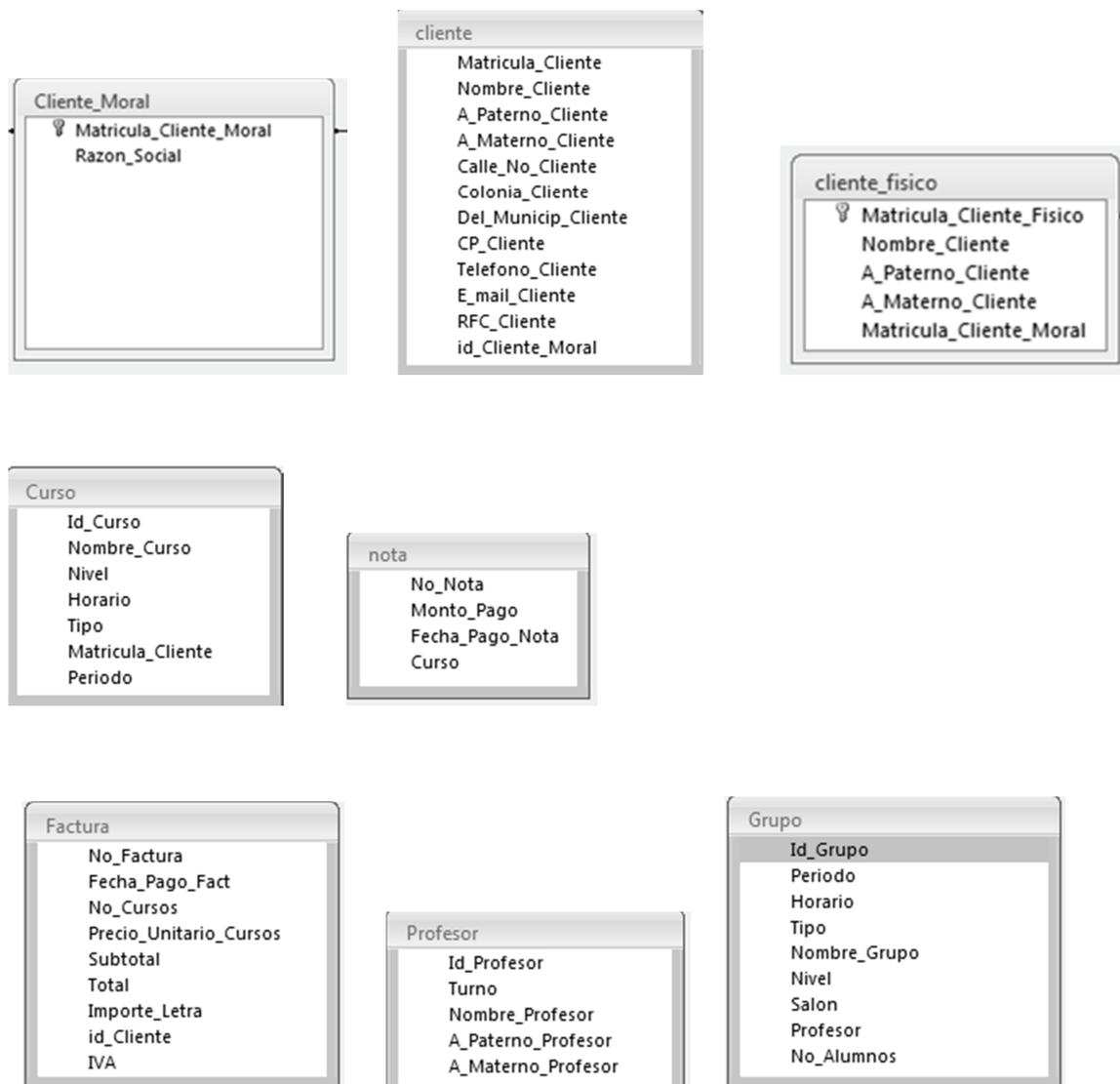


Figura 6.1 Esquemas de SAPIECC

Mientras que el diagrama de relaciones sería el siguiente.

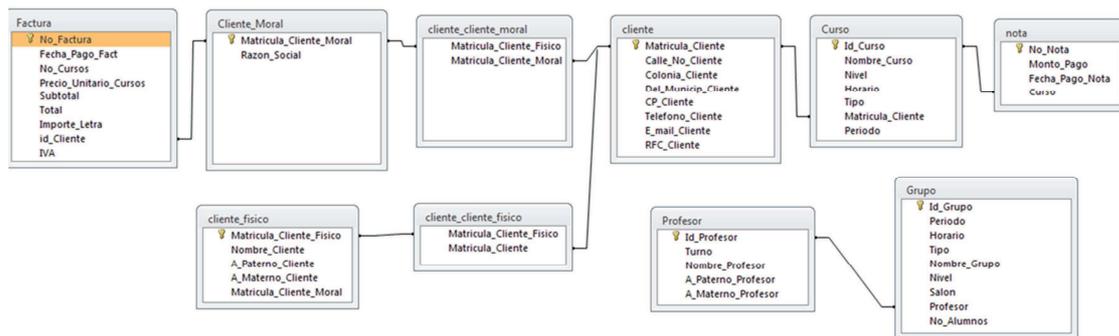


Figura 6.2

A través de estos diagramas podemos observar que, para la implementación del sistema SAPIECC, diseñaremos 8 esquemas relacionados entre sí. La descripción detallada de cada uno de los elementos que aparecen en este diagrama se encuentra en el diccionario de datos (Apéndice 1). Para obtener información adicional acerca del diccionario de datos de SAPIECC consulte el Apéndice 1.

6.2 Diseño Arquitectónico

La arquitectura del software es la representación que capacita al ingeniero del software para analizar la efectividad del diseño para la consecución de los requisitos fijados, considerar las alternativas arquitectónicas en una etapa en la cual hacer cambios en el diseño es relativamente fácil, y reducir los riesgos asociados a la construcción del software.

En seguida se muestra como está constituido el diseño arquitectónico para lograr que el sistema SAPIECC cumpla con sus objetivos definidos a lo largo de los capítulos anteriores.

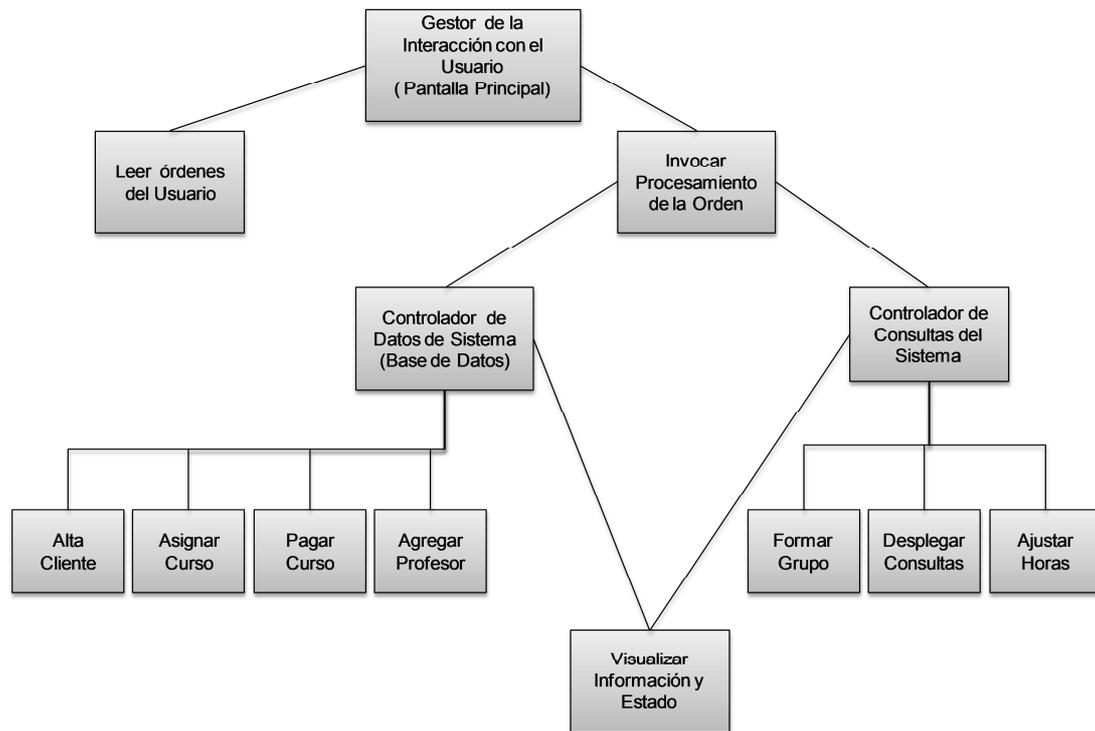


Figura 6.3 Diseño Arquitectónico de SAPIECC

Como podemos observar el sistema SAPIECC consistirá de un Gestor de Interacción con el Usuario (pantalla principal) el cual es capaz de realizar dos tipos de funciones: Leer órdenes o Invocar el Procesamiento de una orden. Dentro del grupo de funciones que invocan el procesamiento de una orden nos encontramos con el controlador de datos del sistema y el controlador de consultas del sistema. El controlador de Datos del Sistema permite al usuario interactuar directamente con la base de datos a través de funciones como alta de clientes, asignación de cursos, pago de cursos y alta de profesores. Mientras que el Controlador de Consultas del Sistema se limita a ejecutar consultas preprogramadas. Todo esto se visualiza en la pantalla del sistema.

La forma en que implementaremos cada uno de los elementos que integran la arquitectura de SAPIECC la definiremos en la sección siguiente.

6.3 Diseño Detallado

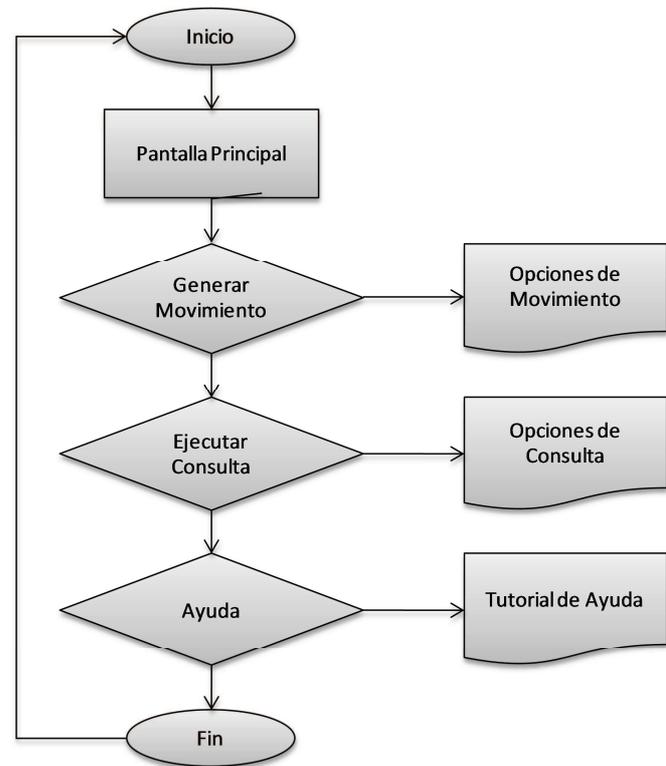
El punto de inicio para el diseño detallado es una estructura arquitectónica a la que se le van a proporcionar detalles algorítmicos, representaciones concretas de datos, interconexiones entre funciones y estructuras de datos, y empaque del producto de programación.

Debemos recordar que algunas de las notaciones para el diseño detallado incluyen herramientas como el pseudocódigo, los diagramas de estructuras de datos y las distribuciones físicas para las representaciones de datos.

En el caso de SAPIECC, si nos remitimos al diseño arquitectónico podemos observar que para diseñar la solución informática funcional es necesario detallar el funcionamiento de cada uno de los módulos que interactúan entre sí. Estos módulos son: Gestor de Interacción con el usuario, Controlador de datos de sistema, Alta cliente, Alta Curso, Pagar Curso, Agregar profesor, Controlador de Consultas del Sistema, Formar Grupo, Consultas y Ajustar Horas. Para definir su operación a continuación se presenta el diagrama de flujo y pseudocódigo para cada una de estas secciones.

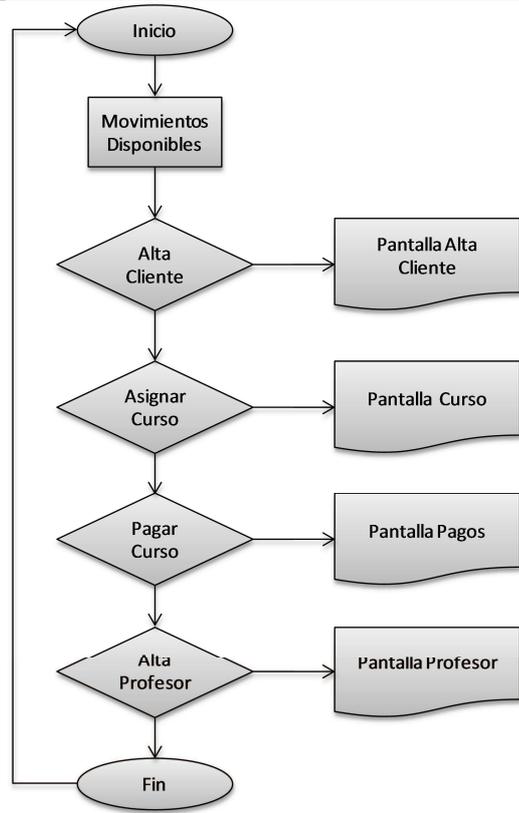
Gestor de la Interacción con el Usuario

INICIAR Pantalla Principal
SI se elige Generar Movimiento ENTONCES
Mostrar Movimientos Disponibles
FIN SI
SI se elige Ejecutar Consulta ENTONCES
Mostrar Opciones de Consulta
FIN SI
SI se elige Ayuda ENTONCES
Mostrar Tutorial de Ayuda
FIN SI
TERMINAR



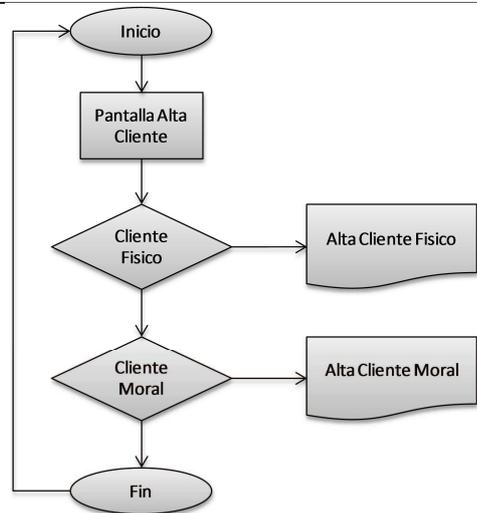
Controlador de Datos de Sistema

INICIAR Pantalla de Movimientos Disponibles
 SI se elige Alta Cliente
 MOSTRAR Opciones Disponibles
 FIN SI
 SI se elige Asignar Curso
 MOSTRAR Opciones Disponibles
 FIN SI
 SI se elige Pagar Curso
 MOSTRAR Opciones Disponibles
 FIN SI
 SI se elige Agregar Profesor
 MOSTRAR Opciones Disponibles
 FIN SI
 TERMINAR



Alta Cliente

INICIAR Pantalla Opciones Cliente
 SI se elige Cliente Físico ENTONCES
 MOSTRAR Opciones de movimiento
 FIN SI
 SI se elige Cliente Moral ENTONCES
 MOSTRAR Opciones de Movimiento
 FIN SI
 TERMINAR



Opciones Alta Cliente Físico

INICIAR Pantalla Opciones Cliente Físico
 SI se elige Alta Cliente ENTONCES

MOSTRAR Formulario para llenar Datos

SI elige Alta Cliente ENTONCES

GUARDA Información en Tabla Cliente

FIN SI

SI elige Borrar ENTONCES

LIMPIA Cajas del Formulario

FIN SI

FIN SI

SI elige Buscar Cliente ENTONCES

MOSTRAR Formulario de Búsqueda

SI elige Buscar ENTONCES

MOSTRAR resultado de la búsqueda requerida

FIN SI

SI elige Borrar ENTONCES

LIMPIA Cajas del Formulario

FIN SI

FIN SI

SI elige Eliminar Cliente ENTONCES

MOSTRAR Formulario de Búsqueda para Eliminación

SI los datos Mostrados para eliminar son correctos ENTONCES

EJECUTAR consulta de eliminación con variable confirmar

MOSTRAR valores del registro eliminado

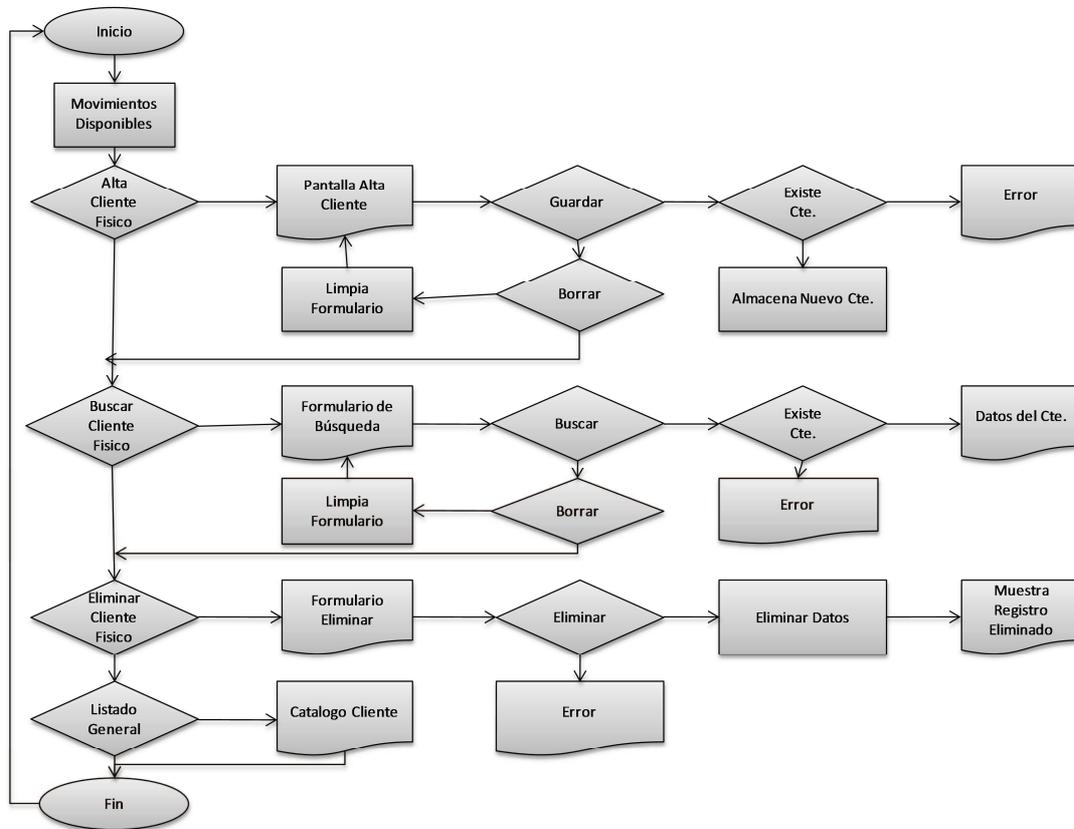
FIN SI

SI elige Listado General Cliente Físico ENTONCES

MOSTRAR valores Tabla Cliente

FIN SI

TERMINAR



Opciones Alta Cliente Moral

INICIAR Pantalla Opciones Cliente Moral

SI se elige Alta Cliente ENTONCES

MOSTRAR Formulario para llenar Datos

SI elige Alta Cliente Moral ENTONCES

GUARDA Información en Tabla Cliente Moral

FIN SI

SI elige Borrar ENTONCES

LIMPIA Cajas del Formulario

FIN SI

FIN SI

SI elige Buscar Cliente Moral ENTONCES

MOSTRAR Formulario de Búsqueda

SI elige Buscar ENTONCES

MOSTRAR resultado de la búsqueda requerida

FIN SI

SI elige Borrar ENTONCES

LIMPIA Cajas del Formulario

FIN SI

FIN SI

SI elige Eliminar Cliente Moral ENTONCES

MOSTRAR Formulario de Búsqueda para Eliminación

SI los datos Mostrados para eliminar son correctos ENTONCES

EJECUTAR consulta de eliminación con variable confirmar

MOSTRAR valores del registro eliminado

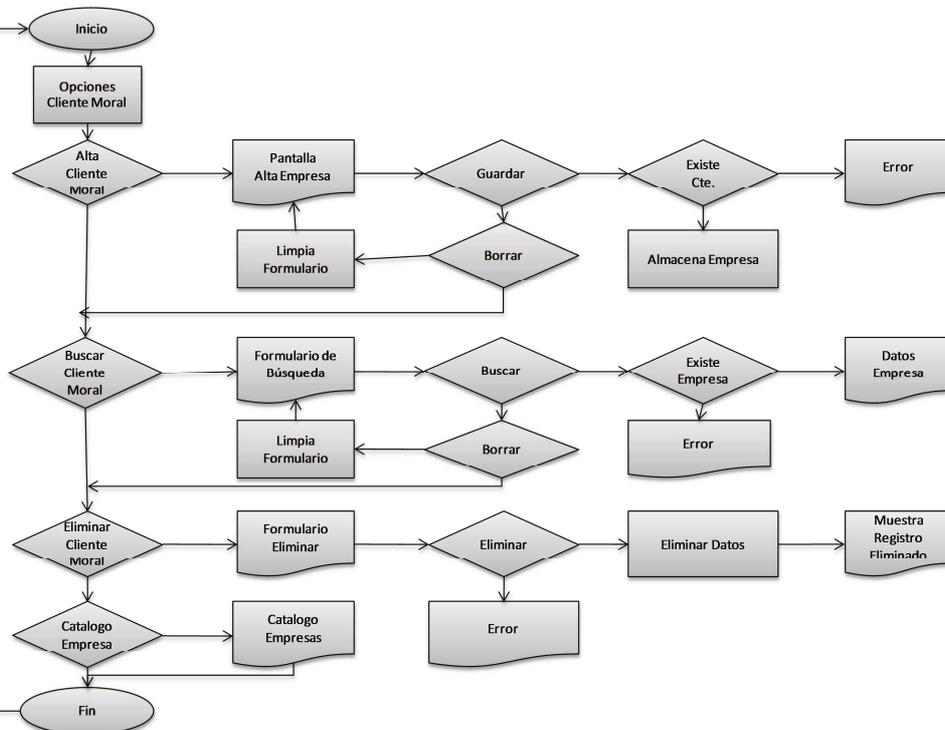
FIN SI

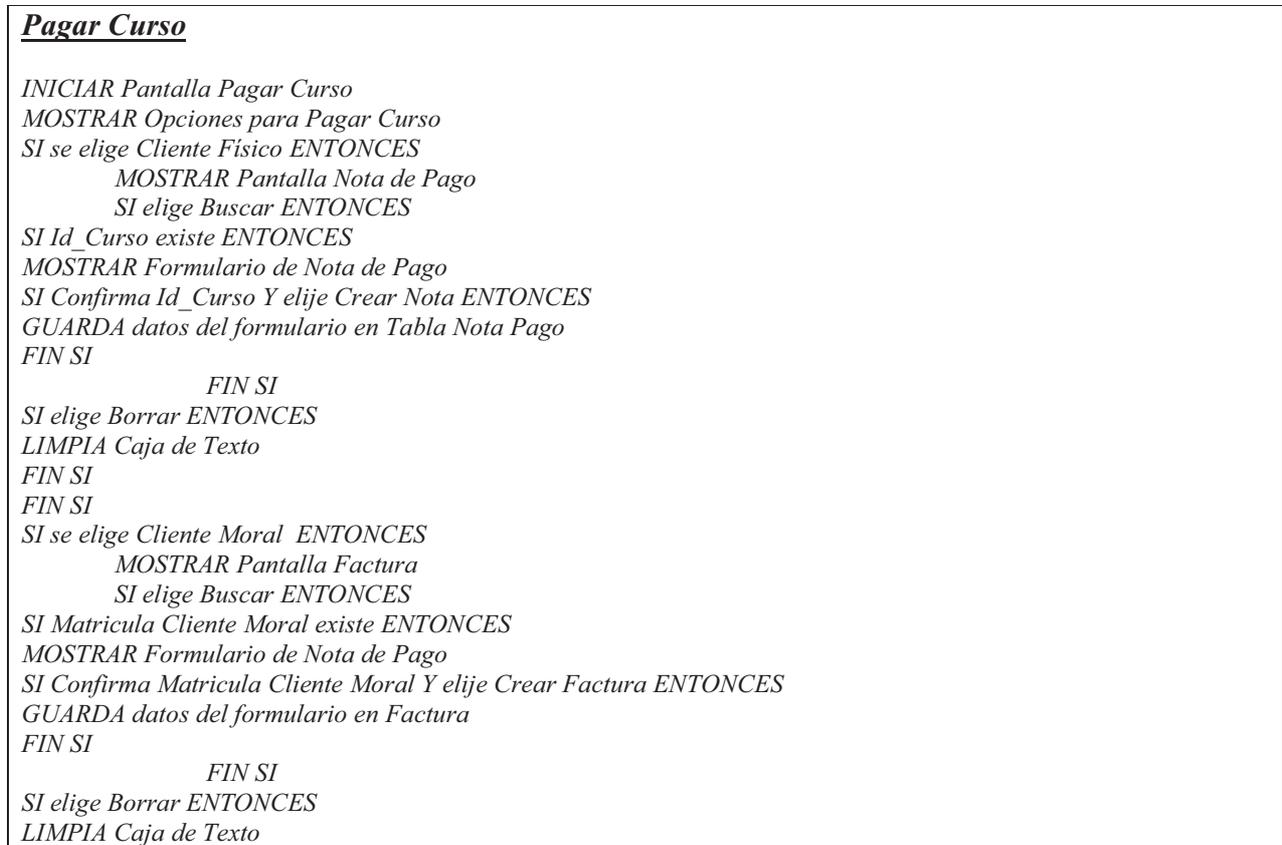
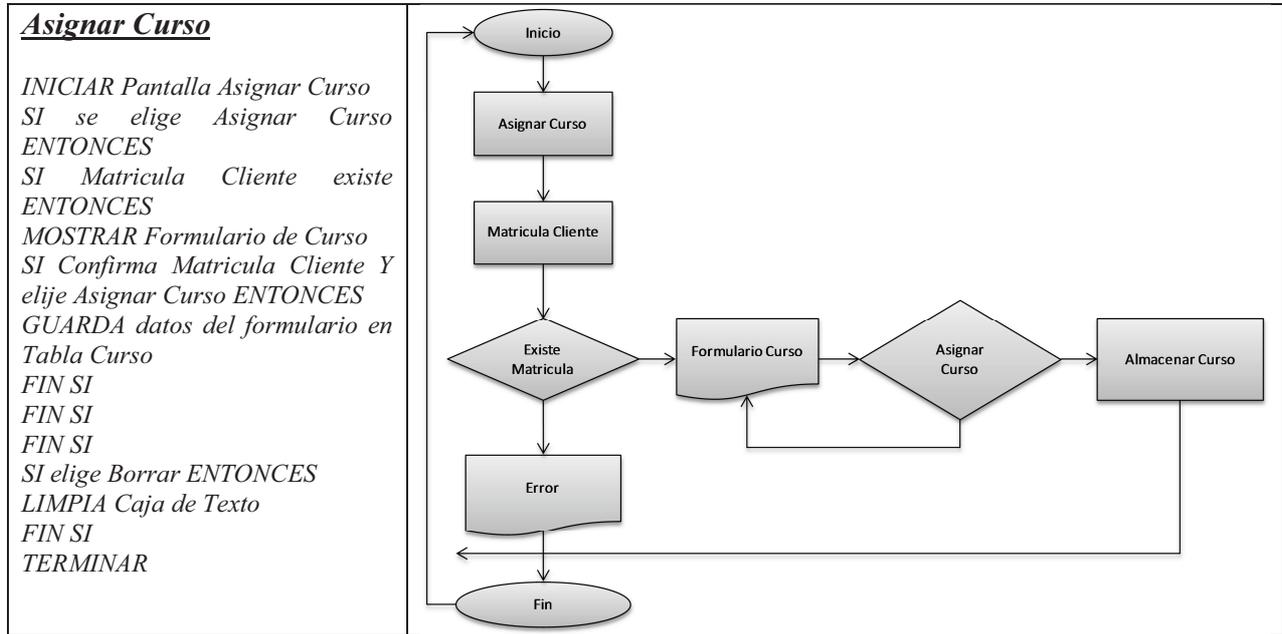
SI elige Catalogo de Empresas ENTONCES

MOSTRAR valores Tabla Cliente Moral

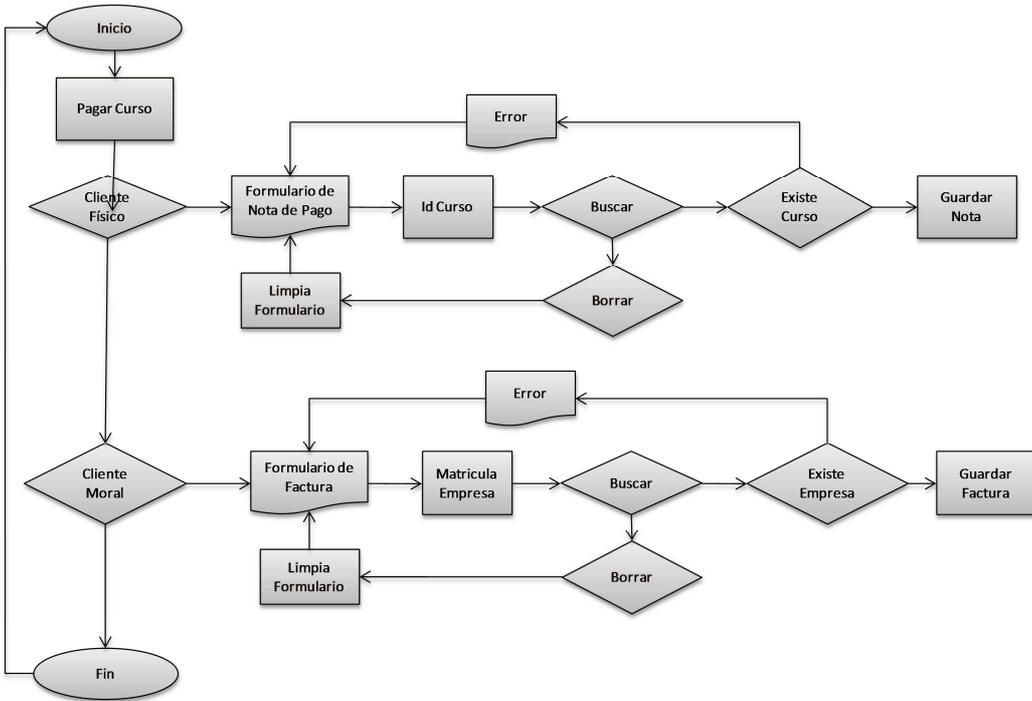
FIN SI

TERMINAR



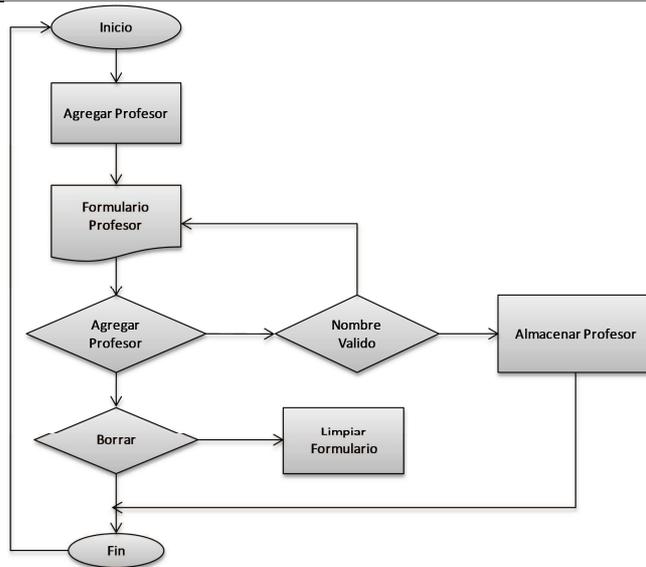


FIN SI
 FIN SI
 FIN SI
 SI elige Borrar ENTONCES
 LIMPIA Caja de Texto
 FIN SI
 TERMINAR



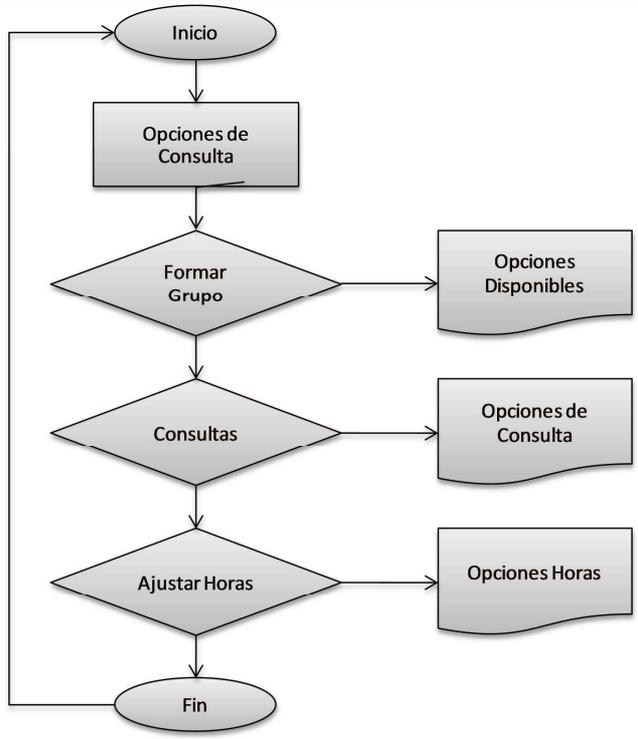
Agregar Profesor

INICIAR Pantalla Agregar Profesor
 SI se elige Agregar Profesor Y Nombre no está vacío ENTONCES GUARDA datos del formulario en Tabla Profesor
 FIN SI
 SI elige Borrar ENTONCES LIMPIA Cajas de Texto del formulario
 FIN SI
 TERMINAR



Controlador de Consultas del Sistema

INICIAR Pantalla Opciones de Consulta
 SI se elige Formar Grupo
 Mostrar Opciones Disponibles
 FIN SI
 SI se elige Consultas
 Mostrar Opciones Disponibles
 FIN SI
 SI se elige Ajustar Horas
 Mostrar Opciones Disponibles
 FIN SI
 TERMINAR



Consultas

INICIAR Pantalla Opciones Consulta
 SI se elige Monto Total de Pagos Realizados ENTONCES
 SELECCIONAR Periodo
 MOSTRAR resultados de la consulta
 FIN SI
 SI se elige No. De Alumnos por Periodo ENTONCES
 SELECCIONAR Periodo
 MOSTRAR resultados de la consulta
 FIN SI
 SI se elige Grupos por Periodo ENTONCES
 SELECCIONAR Periodo
 MOSTRAR resultados de la consulta
 FIN SI
 SI se elige Personas por Grupo ENTONCES
 SELECCIONAR Periodo
 MOSTRAR resultados de la consulta
 FIN SI
 SI se elige Historial Alumnos ENTONCES
 SI Matricula Cliente Existe y Buscar ENTONCES
 MOSTRAR resultados de la consulta
 FIN SI
 FIN SI
 SI se elige Monto Historial Empresas ENTONCES
 SI Matricula Cliente Existe y Buscar ENTONCES

MOSTRAR resultados de la consulta

FIN SI

FIN SI

SI se elige Grupos Asignados por Profeso ENTONCES

SELECCIONAR Profesor

MOSTRAR resultados de la consulta

FIN SI

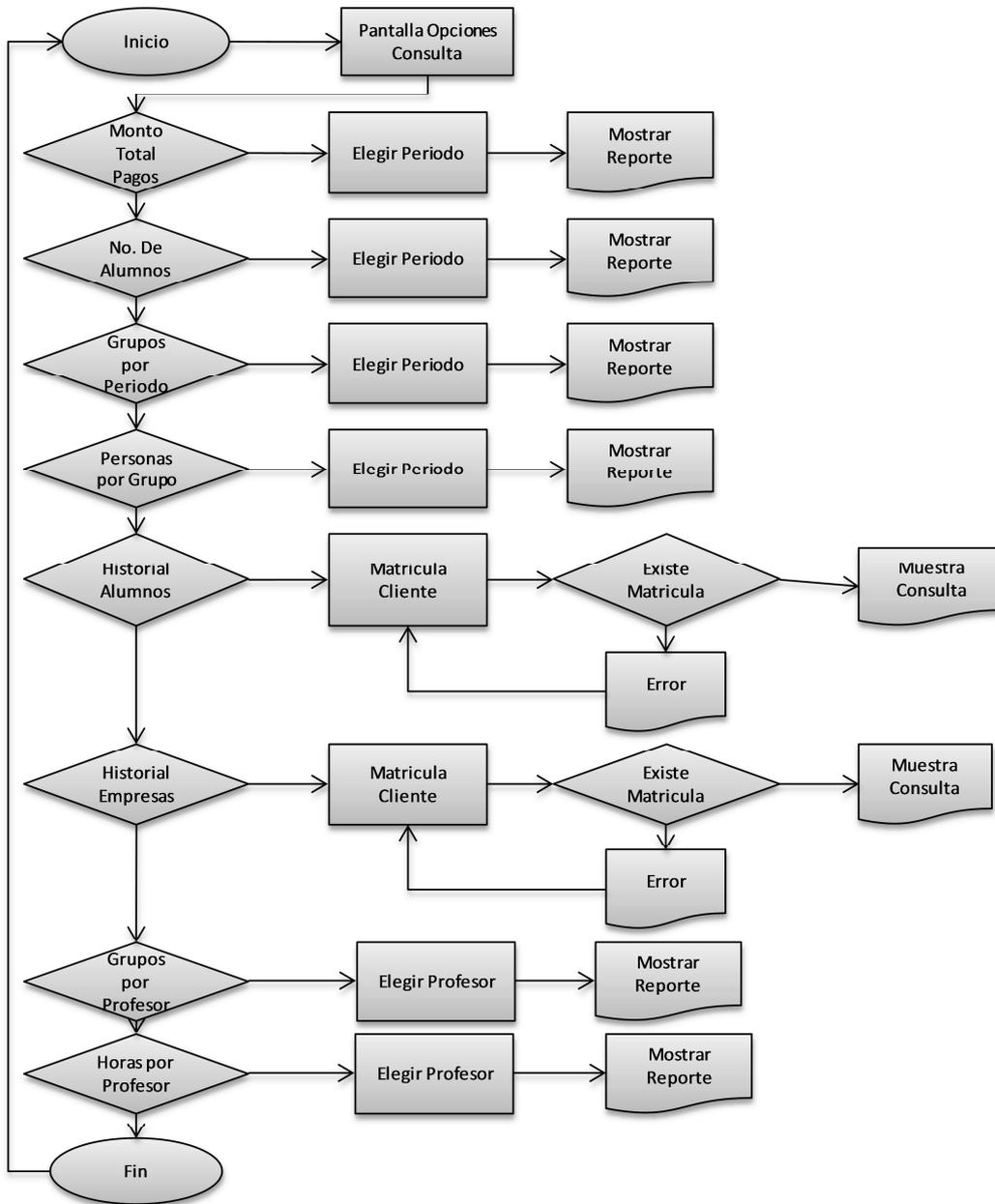
SI se elige Horas Asignadas por Profesor ENTONCES

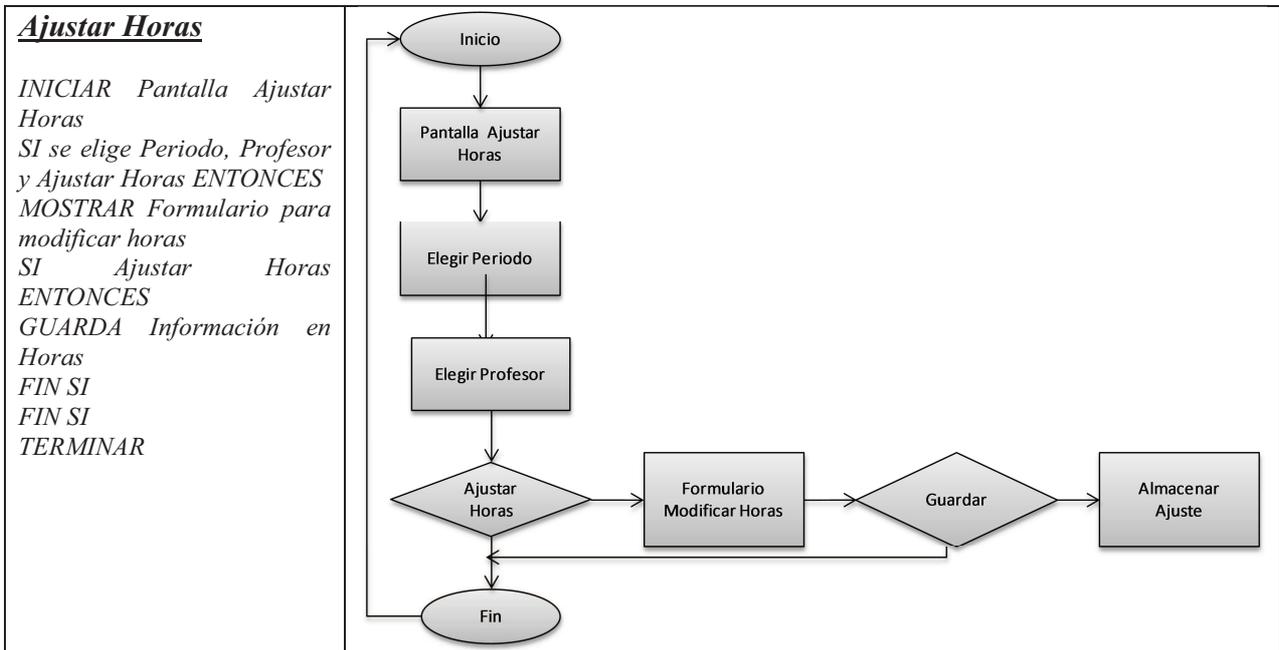
SELECCIONAR Profesor

MOSTRAR resultados de la consulta

FIN SI

TERMINAR





6.4 Instrumentación

Hasta este momento hemos realizado una serie de tareas orientadas a la implementación del sistema SAPIECC, primero llevamos a cabo el análisis completo de los requerimientos del sistema, en seguida, logramos abstraerlo a tal grado que mediante el uso de distintas herramientas desarrollamos un diseño a distintos niveles. Sin embargo es hasta este punto cuando iniciamos con la construcción del software operacional.

El objetivo principal de la instrumentación es escribir el código fuente y la documentación interna de modo que la concordancia del código con sus especificaciones sea fácil de verificar, y que se faciliten la depuración, pruebas y modificaciones. Este objetivo puede alcanzarse haciendo el código fuente tan claro y sencillo como sea posible.

A continuación se presentan algunas secciones del código fuente correspondiente a los principales módulos que integran SAPIECC. Aunque no se presenta de forma completa se pretende dar una visión general de la instrumentación del sistema.

Alta Cliente Moral

```

// Conexión con la base de datos, se debe agregar en cada módulo
<?php
echo "<body>";
include("conex.php");

```

```

$link=Conectarse();
$cons = "SELECT * FROM cliente_moral";
$resultado=mysql_query($cons,$link);

// Número de Empresas almacenadas y definición de la matrícula del cliente a agregar
$num_filas=mysql_num_rows($resultado)+1;
?>

//Almacenamiento de los datos del formulario en las variables
$Razon_Social_Ok=$REQUEST['Razon_Social'];
$Calle_No_Cliente_M_Ok=$REQUEST['Calle_No_Cliente_M'];
$Colonia_Cliente_M_Ok=$REQUEST['Colonia_Cliente_M'];
$Del_MUnicip_Cliente_M_Ok=$REQUEST['Del_Municip_Cliente_M'];
$CP_Cliente_M_Ok=$REQUEST['CP_Cliente_M'];
$Telefono_Cliente_M_Ok=$REQUEST['Telefono_Cliente_M'];
$E_Mail_Cliente_M_Ok=$REQUEST['E_Mail_Cliente_M'];
$RFC_Cliente_M_Ok=$REQUEST['RFC_Cliente_M'];

// Consulta que almacena la empresa
if (!empty($Razon_Social_Ok))
{
    $Consulta_Empresa= "SELECT * FROM cliente_moral WHERE
Razon_Social=$Razon_Social_Ok";
    $resultado3=mysql_query($Consulta_Empresa,$link);
    if($resultado3)
        echo "Razón Social : $Razon_Social_Ok Empresa ya existente";
    else
    {
        $Alta_Empresa="INSERT INTO cliente_moral
VALUES('$num_filas1','$Razon_Social_Ok','$Calle_No_Cliente_M_Ok','$Colonia_Cliente_M_Ok','$Del_Munici
p_Cliente_M_Ok','$CP_Cliente_M_Ok','$Telefono_Cliente_M_Ok','$E_Mail_Cliente_M_Ok','$RFC_Cliente_M
Ok)";
        $resultado2= mysql_query($Alta_Empresa, $link);
    }
}
else
echo "<br><br><br><br><br><b>Debe Almacenar al menos la Razón Social</b>";
?>

```

Buscar Cliente Moral

```

// Seleccionar criterio de búsqueda y valor
<select name="campo">
  <option>Matricula_Cliente_Moral</option>
  <option>Razon_Social</option>
</select>
<input type="text" name="valor" size="50">
<input type="submit" value="Buscar">
<input type="reset" value="Borrar">

//Consulta para realizar la búsqueda
$resultado=mysql_query("SELECT Matricula_Cliente_Moral, Razon_Social FROM cliente_moral where
Scampo='$valor' order by $campo", $link);

```

Eliminar Cliente Moral

```
// Variables que almacenan los datos a eliminar
<h4 align="center"> Para eliminar teclee la matricula de la empresa y de clic en Eliminar </h4>
<input type="text" name="confirmar" size="20">
<input type="submit" value="Eliminar">

//Consulta para eliminar la empresa
$Eliminar="DELETE FROM cliente_moral WHERE Matricula_Cliente_Moral=$confirmar";
$resultado2=mysql_query($Eliminar,$link);
?>
```

Cliente Físico

```
// Datos del Cliente provenientes de un formulario de entrada
$Nombre_Cliente_Ok=$REQUEST['Nombre_Cliente'];
$A_Paterno_Cliente_Ok=$REQUEST['A_Paterno_Cliente'];
$A_Materno_Cliente_Ok=$REQUEST['A_Materno_Cliente'];
$Calle_No_Cliente_Ok=$REQUEST['Calle_No_Cliente'];
$Colonia_Cliente_Ok=$REQUEST['Colonia_Cliente'];
$Del_Municip_Cliente_Ok=$REQUEST['Del_Municip_Cliente'];
$CP_Cliente_Ok=$REQUEST['CP_Cliente'];
$Telefono_Cliente_Ok=$REQUEST['Telefono_Cliente'];
$E_Mail_Cliente_Ok=$REQUEST['E_Mail_Cliente'];
$RFC_Cliente_Ok=$REQUEST['RFC_Cliente'];
$Matricula_Cliente_Moral_Ok=$REQUEST['Matricula_Cliente_Moral'];

//Establecemos si pertenece a una empresa
if (empty($Matricula_Cliente_Moral_Ok))
$Matricula_Cliente_Moral_Ok=0;

//Revisión de que el cliente no exista
$Consulta_Cliente="SELECT * FROM cliente WHERE Nombre_Cliente='$Nombre_Cliente_Ok' AND
A_Paterno_Cliente='$A_Paterno_Cliente_Ok' AND A_Materno_Cliente='$A_Materno_Cliente_Ok'";
$resultado3=mysql_query($Consulta_Cliente,$link);
if(!empty($resultado3))
    echo "Cliente ya existente ";
else    {

//Consulta para agregar cliente
    $Alta_Cliente="INSERT INTO cliente
VALUES('$num_filas1','$Nombre_Cliente_Ok','$A_Paterno_Cliente_Ok','$A_Materno_Cliente_Ok','$Calle_No_
Cliente_Ok','$Colonia_Cliente_Ok','$Del_Municip_Cliente_Ok','$CP_Cliente_Ok','$Telefono_Cliente_Ok','$E_
Mail_Cliente_Ok','$RFC_Cliente_Ok','$Matricula_Cliente_Moral_Ok)";
?>
```

ASIGNAR CURSO

```
//Características del curso
<b>Nombre Curso:</b> <select name="Nombre_Curso">
<option value="1">Introducción
<option value="2">Windows
<option value="3">Word
<option value="4">Excel
```

```
<option value="5">Power Point
<option value="6">Project
<option value="7">Outlook
<option value="8">COI
<option value="9">NOI
<option value="10">SAE
<option value="11">SUA
<option value="12">Corel Draw
<option value="13">Free Hand
<option value="14">Illustrator
<option value="15">Photo Shop
<option value="16">Auto Cad
<option value="17">Flash
<option value="18">HTML
<option value="19">Front Page
<option value="20">Dream Weaver
<option value="21">Publisher
<option value="22">PHP
<option value="23">MySQL
<option value="24">Access
<option value="25">Visual Basic
<option value="26">Visual Basic para Aplicaciones
</select>

<b> Nivel:</b>
<input type="radio" name="Nivel" value="1">Básico
<input type="radio" name="Nivel" value="2">Intermedio
<input type="radio" name="Nivel" value="3">Avanzado

<b>Periodo</b>
<select name="Periodo">
<option value="1">Enero 1
<option value="2">Enero 2
<option value="3">Febrero 1
<option value="4">Febrero 2
<option value="5">Marzo 1
<option value="6">Marzo 2
<option value="7">Abril 1
<option value="8">Abril 2
<option value="9">Mayo 1
<option value="10">Mayo 2
<option value="11">Junio 1
<option value="12">Junio 2
<option value="13">Julio 1
<option value="14">Julio 2
<option value="15">Agosto 1
<option value="16">Agosto 2
<option value="17">Septiembre 1
<option value="18">Septiembre 2
<option value="19">Octubre 1
<option value="20">Octubre 2
<option value="21">Noviembre 1
<option value="22">Noviembre 2
<option value="23">Diciembre 1
<option value="24">Diciembre 2
</select>
```

```

<b>Horario:</b>
<select name="Horario">
<option value="1"> 7:00 a 8:20 a.m.
<option value="2"> 8:30 a 9:50 a.m.
<option value="3"> 10:10 a 11:30 a.m.
<option value="4"> 11:30 a 1:00 p.m.
<option value="5"> 1:30 a 3:00 p.m.
<option value="6"> 3:00 a 4:20 p.m.
<option value="7"> 4:30 a 5:50 p.m.
<option value="8"> 6:10 a 7:30 p.m.
<option value="9"> 7:40 a 9:00 p.m.
<option value="10">-----60 min-----
<option value="11">6:00 a 6:55 p.m.
<option value="12">1:00 a 1:55 p.m.
<option value="13">2:00 a 2:55 p.m.
<option value="14">3:00 a 3:55 p.m.
<option value="15">-----Intensivo-----
<option value="16"> Sabatino AM 8:30 a.m. - 14:30 p.m.
<option value="17"> Sabatino PM 3:00 p.m. - 6:00 p.m.
</select>

<b>Tipo:</b>
<input type="radio" name="Tipo" value="1"> Semi-Intensivo 90 min
<input type="radio" name="Tipo" value="2"> Semi-Intensivo 60 min
<input type="radio" name="Tipo" value="3"> Intensivo
<input type="radio" name="Tipo" value="4"> Sabatino
<input type="radio" name="Tipo" value="5"> Mensual
<input type="radio" name="Tipo" value="6">SNE

<h4 align="center">Para Asignar un Curso teclee la matricula del cliente y de clic en Asignar Curso </h4>
<center><input type="text" name="alta" size="20">
<input type="submit" value="Asignar Curso">
<h1 align="center"> Datos Curso Asignado </h1>
<center>
<?php
echo "<body>";
include ("conex.php");
$link=Conectarse();
$consql= "SELECT * FROM curso";
$resultado1= mysql_query($consql, $link);
$num_filas1=mysql_num_rows($resultado1)+1;
print $num_filas1;
$conconfirmar=$_REQUEST['confirmar'];
$Nombre_Curso=$_REQUEST['Nombre_Curso'];
$Nivel=$_REQUEST['Nivel'];
$Horario=$_REQUEST['Horario'];
$Tipo=$_REQUEST['Tipo'];
$Periodo=$_REQUEST['Periodo'];
$Id_Curso='1';
print " Curso: ";
switch($Nombre_Curso)
{
case "1": print "Introducción";break;
case "2": print "Windows";break;
case "3": print "Word";break;
case "4": print "Excel";break;

```

```

case "5": print "Power Point";break;
case "6": print "Project";break;
case "7": print "Outlook";break;
case "8": print "COI";break;
case "9": print "NOI";break;
case "10": print "SAE";break;
case "11": print "SUA";break;
case "12": print "Corel Draw";break;
case "13": print "Free Hand";break;
case "14": print "Illustrator";break;
case "15": print "Photo Shop";break;
case "16": print "Auto Cad";break;
case "17": print "Flash";break;
case "18": print "HTML";break;
case "19": print "Front Page";break;
case "20": print "Dream Weaver";break;
case "21": print "Publisher";break;
case "22": print "PHP";break;
case "23": print "MySQL";break;
case "24": print "Access";break;
case "25": print "Visual Basic";break;
case "26": print "Visual Basic para Aplicaciones";break;
}
print "<br>";
print "Nivel: ";
switch($Nivel){
case "1": print "Básico";break;
case "2": print "Intermedio";break;
case "3": print "Avanzado";break;}
print "<br>";
print "Horario: ";
switch($Horario)
{
case "1": print "7:00 a 8:20 a.m."; break;
case "2": print "8:30 a 9:50 a.m."; break;
case "3": print "10:10 a 11:30 a.m."; break;
case "4": print "11:30 a 1:00 p.m."; break;
case "5": print "1:30 a 3:00 p.m."; break;
case "6": print "3:00 a 4:20 p.m."; break;
case "7": print "4:30 a 5:50 p.m."; break;
case "8": print "6:10 a 7:30 p.m."; break;
case "9": print "7:40 a 9:00 p.m."; break;
case "10": print "60 min"; break;
case "11": print "6:00 a 6:55 p.m."; break;
case "12": print "1:00 a 1:55 p.m."; break;
case "13": print "2:00 a 2:55 p.m."; break;
case "14": print "3:00 a 3:55 p.m."; break;
case "15": print "Intensivo"; break;
case "16": print "Sabatino AM 8:30 a.m. - 14:30 p.m."; break;
case "17": print "Sabatino PM 3:00 p.m. - 6:00 p.m."; break;
}
print "<br>";
print "Tipo: ";
switch($Tipo)
{
case "1": print "Semi-Intensivo 90 min"; break;

```

```

case "2": print "Semi-Intensivo 60 min"; break;
case "3": print "Intensivo"; break;
case "4": print "Sabatino"; break;
case "5": print "Mensual"; break;
case "6": print "SNE"; break;
}
print "<br>";
print "Periodo: ";
switch($Periodo)
{
case "1": print "Enero1"; break;
case "2": print "Enero2"; break;
case "3": print "Febrero1"; break;
case "4": print "Febrero2"; break;
case "5": print "Marzo 1"; break;
case "6": print "Marzo 2"; break;
case "7": print "Abril 1"; break;
case "8": print "Abril 2"; break;
case "9": print "Mayo 1"; break;
case "10": print "Mayo 2"; break;
case "11": print "Junio 1"; break;
case "12": print "Junio 2"; break;
case "13": print "Julio 1"; break;
case "14": print "Julio 2"; break;
case "15": print "Agosto 1"; break;
case "16": print "Agosto 2"; break;
case "17": print "Septiembre 1"; break;
case "18": print "Septiembre 2"; break;
case "19": print "Octubre 1"; break;
case "20": print "Octubre 2"; break;
case "21": print "Noviembre 1"; break;
case "22": print "Noviembre 2"; break;
case "23": print "Diciembre 1"; break;
case "24": print "Diciembre 2"; break;
}
$Matricula_Cliente_Ok=$_REQUEST['alta'];
print $Matricula_Cliente_Ok;
$Cliente="SELECT * FROM cliente WHERE Matricula_Cliente=$Matricula_Cliente_Ok";
$Curso="SELECT * FROM curso WHERE Matricula_Cliente=$Matricula_Cliente_Ok";
$resultado4=mysql_query($Curso,$link);
$Alta_Curso="INSERT INTO curso VALUES
($num_filas1,$Nombre_Curso,$Nivel,$Horario,$Tipo,$Matricula_Cliente_Ok,$Periodo)";
$resultado2= mysql_query($Alta_Curso, $link);
if($resultado2)
    echo "<br><br><b>Curso Asignado</b>";
?>

```

CREAR NOTA DE PAGO

```

<h4 align="center">Para crear la Nota de Pago teclee el Id. del curso a Pagar y de clic en Crear Nota de Pago
</h4>
<input type="text" name="confirmar" size="20">
<input type="submit" value="Crear Nota de Pago">

```

```
//Revisión de existencia del curso
$confirmar=$_REQUEST['confirmar'];
$Curso="SELECT * FROM curso WHERE Id_Curso=$confirmar";
$resultado2=mysql_query($Curso,$link);

//Datos Complementarios de la nota
Fecha de Pago(AAAA-MM-DD) <input type="text" name="Fecha_Pago" size="15"></td>
Monto a Pagar $ <input type="text" name="Monto_Pago" size="25"></td></tr>
$Fecha_Pago_Ok=$_REQUEST['Fecha_Pago'];
$Monto_Pago_Ok=$_REQUEST['Monto_Pago'];

//Si los datos son correctos, se agrega la nota con la siguiente consulta
$Alta_Nota="INSERT INTO nota_pago
VALUES($num_filas1,$Monto_Pago_Ok,$Fecha_Pago_Ok,$Id_Curso_Ok)";
$resultado2= mysql_query($Alta_Nota, $link);
if($resultado2)
    echo "<br><br><h3>Nota de Pago Almacenada</h3>";
?>
```

CREAR FACTURA

```
<h4 align="center">Para crear la Factura teclee la matricula de la empresa y de clic en Crear Factura </h4>
<input type="text" name="confirmar" size="20">
<input type="submit" value="Crear Factura">

//Asignamos la matricula de la empresa a una variable
$confirmar=$_REQUEST['confirmar'];
$Fecha_Pago_Ok=$_REQUEST['Fecha_Pago'];
$No_Cursos_Ok=$_REQUEST['No_Cursos'];
$Precio_Unitario_Ok=$_REQUEST['Precio_Unitario'];
$Subtotal_Ok=$_REQUEST['Subtotal'];
$Iva_Ok=$_REQUEST['Iva'];
$Total_Ok=$_REQUEST['Total'];
$Importe_Letra_Ok=$_REQUEST['Importe_Letra'];
$Descripcion_Ok=$_REQUEST['Descripcion'];
$Matricula_Empresa_Ok=$_REQUEST['alta'];

//Consulta para almacenar la factura
$Alta_Factura="INSERT INTO factura
VALUES($num_filas1,$Fecha_Pago_Ok,$No_Cursos_Ok,$Precio_Unitario_Ok,$Subtotal_Ok,$Total_Ok,$Imp
orte_Letra_Ok,$Matricula_Empresa_Ok,$Iva_Ok,$Descripcion_Ok)";
$resultado2= mysql_query($Alta_Factura, $link);
if($resultado2)
    echo "<br><br><b>Factura Almacenada</b>";
?>
```

AGREGAR PROFESOR

```
//Elegir características del profesor.
<br>Turno:
<select name="Turno">
<option value="1">Matutino
<option value="2">Vespertino
```

```

<option value="3">Mixto
</select>
<br><br>Nombre Profesor <input type="text" name="Nombre_Profesor" size=35>
<br><br>Apellido Paterno Profesor <input type="text" name="A_Paterno_Profesor" size=35>
<br><br>Apellido Materno Profesor<input type="text" name="A_Materno_Profesor" size=35>

//Consulta para agregar profesor
$Consulta_Profesor= "SELECT * FROM profesor WHERE Nombre_Profesor='$Nombre_Profesor_Ok' AND
A_Paterno_Profesor='$A_Paterno_Profesor_Ok' AND A_Materno_Profesor='$A_Materno_Profesor_Ok'";
$resultado3=mysql_query($Consulta_Cliente,$link);
$Alta_Profesor="INSERT INTO profesor
VALUES('$num_filas1','$Turno_Ok','$Nombre_Profesor_Ok','$A_Paterno_Profesor_Ok','$A_Materno_Profesor_
Ok')";
?>

```

FORMAR GRUPO

```

// Código de Opciones de Periodo (Select de Periodo)
<h2>Elije el periodo a evaluar </h2>
<input type="submit" value="Evaluar">
<?php
include("conex.php");
$periodo=$_REQUEST['Periodo'];
echo "Periodo: ";
// Código para decodificar Periodo (switch)
$scursos="SELECT *,count(*) FROM curso WHERE Periodo='$periodo' group by
'Horario','Tipo','Nivel','Nombre_Curso' ";
$resultado=mysql_query($scursos, $link);
echo "<center>";

// Crear Grupo
<?php
$cons1= "SELECT * FROM grupo";
$resultado1= mysql_query($cons1, $link);
$num_filas1=mysql_num_rows($resultado1)+1;
print "Id. Grupo: ";
print $num_filas1;
$Periodo=$_REQUEST['Periodo'];
$Horario=$_REQUEST['Horario'];
$Tipo=$_REQUEST['Tipo'];
$Nombre_Grupo=$_REQUEST['Nombre_Grupo'];
$Nivel=$_REQUEST['Nivel'];
$Salon=$_REQUEST['Salon'];
$Profesor=$_REQUEST['Profesor'];
$No_Alumnos=$_REQUEST['No_Alumnos'];
$Grupo="SELECT * FROM grupo";
$resultado4=mysql_query($Grupo,$link);
$Alta_Grupo="INSERT INTO grupo
VALUES($num_filas1,$Periodo,$Horario,$Tipo,$Nombre_Grupo,$Nivel,$Salon, $Profesor,$No_Alumnos)";
$resultado2= mysql_query($Alta_Grupo, $link);
echo $resultado2;
if($resultado2)
echo "<br><br><b>Grupo Creado</b>";
?>

```

AJUSTAR HORAS

```
<form action="Horas_P2.php" method="post">
<h2>Selecciona el Periodo de Ajuste </h2> <br>

// Selecciona el Periodo
<h2>Selecciona el Profesor</h2>

//Select Profesor
<input type="submit" value="Buscar">
<input type="reset" value="Borrar">
include("conex.php");
$periodo=$_REQUEST['Periodo'];
echo "Periodo: ";

/Consulta para ajustar horas
$profesor=$_REQUEST['Profesor'];
$horas="SELECT * FROM grupo WHERE Periodo='$periodo' and Profesor='$profesor'";
$resultado=mysql_query($horas, $link);
$num_filas=mysql_num_rows($resultado);
echo "Grupos Encontrados : $num_filas";
$total=$num_filas*15;
echo "Horas Trabajadas : $total";
echo "Horas Trabajadas Ajustadas:";
?>
<input type="text" size="20" name="horas_modificadas">
<input type="submit" Value="Ajustar Horas">
```

CONSULTA MONTO TOTAL DE PAGOS REALIZADOS

```
//Consulta
$Total_Facturas="SELECT count(f.Total) FROM factura f WHERE Periodo=$periodo";
$resultado_facturas=mysql_query($Total_Facturas, $link);
echo $resultado_facturas;
$notas="SELECT * FROM curso inner join nota_pago on curso.Id_Curso=nota_pago.Id_Curso WHERE
((curso.Periodo=$periodo))";
$resultado=mysql_query($notas, $link);
$num_notas=mysql_num_rows($resultado);
echo "No. de Notas del Periodo: $num_notas <br>";
```

CONSULTA ALUMNOS POR PERIODO

```
<h2>Selecciona el Periodo </h2> <br>
//Select Periodo
<input type="submit" value="Buscar">
<?php
include("conex.php");
$periodo=$_REQUEST['Periodo'];
// switch ($periodo)
```

```
$horas="SELECT * FROM curso WHERE Periodo='$periodo'";
$resultado=mysql_query($horas, $link);
// Dibujar la tabla
$num_filas=mysql_num_rows($resultado);
echo "<br><center>";
echo "Total de Alumnos Inscritos : $num_filas";
?>
```

CONSULTA GRUPOS POR PERIODO

```
<h2>Selecciona el Periodo </h2> <br>
//Select Periodo
<input type="submit" value="Buscar">
<?php
include("conex.php");
$periodo=$_REQUEST['Periodo'];
echo "Periodo: ";
//switch ($periodo)
$horas="SELECT * FROM grupo WHERE Periodo='$periodo'";
$resultado=mysql_query($horas, $link);
// Dibujar la tabla
$num_filas=mysql_num_rows($resultado);
echo "<br><center>";
echo "Grupos Encontrados : $num_filas";
?>
```

CONSULTA ALUMNOS POR GRUPO

```
<h2>Selecciona el Periodo </h2>
//Select Periodo
<input type="submit" value="Buscar">
<?php
include("conex.php");
$periodo=$_REQUEST['Periodo'];
//switch ($periodo)
$horas="SELECT * FROM grupo WHERE Periodo='$periodo'";
$resultado=mysql_query($horas, $link);
// Dibujar la tabla
$num_filas=mysql_num_rows($resultado);
echo "<br><center>";
echo "Grupos Encontrados : $num_filas";
?>
```

CONSULTA HISTORIAL DE PAGOS EMPRESAS

```
<form action="Saldo_Empresas_2.php" method="post">
<h2>Matricula Cliente Moral </h2>
<input type="text" name="valor" size="50"><br><br>
<input type="submit" value="Buscar">
<input type="reset" value="Borrar">
<?php
```

```
include("conex.php");
$valor=$_REQUEST['valor'];
$resultado=mysql_query("SELECT Matricula_Cliente_Moral, Razon_Social FROM cliente_moral where
Matricula_Cliente_Moral=$valor", $link);
$cons = "SELECT * FROM factura WHERE Matricula_Cliente_Moral=$valor";
$resultado=mysql_query($cons,$link);
// Dibujar la tabla
$num_facturas=mysql_num_rows($resultado);
echo "No. Facturas: $num_facturas <br>";
?>
```

CONSULTA GRUPOS POR PROFESOR

```
<form action="Grupos_Profesor_2.php" method="post">
<h2>Id. Profesor </h2>
<input type="text" name="valor" size="50"><br><br>
<input type="submit" value="Buscar">
<input type="reset" value="Borrar">
<?php
include("conex.php");
$valor=$_REQUEST['valor'];
echo "Id. Profesor: ";
echo $valor;
$resultado=mysql_query("SELECT * FROM profesor where Id_Profesor='$valor'", $link);
// Recorrido del cursor
$horas="SELECT * FROM grupo WHERE Profesor='$valor'";
$resultado=mysql_query($horas, $link);
// Dibujar la tabla
$num_filas=mysql_num_rows($resultado);
echo "<br><center>";
echo "Grupos Encontrados : $num_filas";
?>
```

CONSULTA HORAS POR PROFESOR

```
<form action="Horas_Profesor_2.php" method="post">
<h2>Id. Profesor </h2>
<input type="text" name="valor" size="50"><br><br>
<input type="submit" value="Buscar">
<input type="reset" value="Borrar">
<?php
include("conex.php");
$valor=$_REQUEST['valor'];
echo "Id. Profesor: ";
echo $valor;
$resultado=mysql_query("SELECT * FROM profesor where Id_Profesor='$valor'", $link);
// Recorrido del cursor
$horas="SELECT * FROM horas WHERE Id_Profesor='$valor'";
$resultado=mysql_query($horas, $link);
// Dibujar la tabla
$num_filas=mysql_num_rows($resultado);
echo "<br><center>";
echo "Grupos Encontrados : $num_filas";
?>
```

6.5 Plan de Pruebas de Aceptación

Una vez generado el código fuente, todo programa debe ser probado para descubrir (y corregir) el máximo de errores posibles antes de su entrega al cliente. El objetivo de tener un plan de pruebas es diseñar una serie de pruebas que tengan una alta probabilidad de encontrar errores. Aquí es donde aplicamos las estrategias de pruebas del software, las cuales facilitan una guía sistemática para diseñar pruebas que comprueben la lógica interna de los componentes del software y verifiquen los dominios de entrada y salida del programa para descubrir errores en la funcionalidad, el comportamiento y el rendimiento.

Es importante recordar que el plan de pruebas especifica los objetivos de las pruebas, los criterios de realización de las pruebas, el plan de integración del sistema, los métodos a utilizarse en módulos particulares y los casos particulares de prueba a utilizarse.

Como podemos recordar, las pruebas en el contexto de ingeniería del software consisten en una serie de cuatro pasos que se llevan a cabo secuencialmente. Inicialmente, la prueba se centra en cada módulo individual asegurando que funciona adecuadamente como una unidad. De ahí el nombre de prueba de unidad. A continuación, se deben ensamblar o integrar los módulos para formar el paquete de software completo. La prueba de integración se dirige a todos los aspectos asociados con el doble problema de verificación y construcción del problema. Después de que el software se ha integrado, se deben comprobar los criterios de validación establecidos durante el análisis de requisitos, la prueba de validación proporciona una seguridad final de que el software satisface todos los requisitos funcionales, de comportamiento y de rendimiento. El software una vez validado, se debe combinar con otros elementos del sistema tales como hardware, gente, bases de datos, etc. La prueba del sistema verifica que cada elemento encaja de forma adecuada y que se alcanza la funcionalidad y el rendimiento del sistema total.

En el caso del sistema SAPIECC para realizar las pruebas de unidad se propone hacer uso de las técnicas de prueba del camino básico y las pruebas de valores al límite para cada uno de los módulos principales que integran el sistema tales como clientes, pagos, profesores, y consultas. Esto con el objetivo de garantizar la funcionalidad de cada uno de los módulos desarrollados.

En lo referente a las prueba de integración, como el objetivo es recoger los módulos probados por la prueba de unidad y así construir una estructura de programa que esté de acuerdo con el diseño y considerando las características de SAPIECC. Se hará uso de las técnicas de prueba de integración ascendente, así como de las pruebas de regresión. Partiendo desde el alta de clientes hacia la asignación de curso, generación de pago, formación de cursos y finalmente las consultas.

Para las pruebas de validación se hará uso del documento de especificación de requisitos y se comparara los resultados que ofrece el sistema con respecto a lo propuesto inicialmente.

Finalmente para las pruebas del sistema se realizaran pruebas de seguridad, de resistencia y de rendimiento.

VII. PRUEBAS Y MANTENIMIENTO

Las pruebas del software son un elemento crítico para la garantía de calidad del software y representan una revisión final de las especificaciones, del diseño y de la codificación.

El término “mantenimiento de software” se usa para describir las actividades de la ingeniería de software que ocurren después de entregar un producto al cliente. La fase de mantenimiento del ciclo de vida del software es el periodo en el que un producto desempeña un trabajo útil.

Las actividades de mantenimiento implican mejorar los productos de software, adaptarlos a nuevos ambientes y corregir problemas. La mejora de los productos de software puede dar como resultado proporcionar nuevas capacidades funcionales, mejorar los despliegues al usuario y los modos de interacción, revalorar los documentos externos y la documentación interna, o revalorar las características del desempeño de un sistema. El mejoramiento y la adaptación del software reinician el desarrollo en la fase de análisis, mientras que la corrección de un problema de software puede reiniciar el ciclo de desarrollo en la fase de análisis, diseño, o en la implementación. Por lo tanto, todas las herramientas y técnicas utilizadas para desarrollar el software son potencialmente útiles para el mantenimiento del software.

A continuación se presentan los resultados de las pruebas efectuadas sobre SAPIECC, así como la descripción de algunas métricas y estrategias para facilitar su mantenimiento.

7.1 Tipos de Pruebas

Una vez terminada la fase de diseño, es necesario verificar que el sistema cumpla con la especificaciones que se definieron al iniciar el proyecto, para lograr esto hacemos uso de diferentes tipos de pruebas.

Para comprobar el correcto funcionamiento de SAPIECC seguiremos la metodología, establecida en el plan de pruebas que ya se mencionó anteriormente. Partiremos de pruebas de unidad, en donde se verificara la correcta implementación en código fuente de cada uno de los módulos que integran el sistema. Enseguida se realizarán pruebas de integración que tendrán como foco de atención el diseño y la construcción de la arquitectura de software de SAPIECC, con esto se pretende explicar cómo se entrelazan los módulos que integran el sistema. Posteriormente validaremos que los requisitos establecidos en la fase de análisis se cumplieron a través de pruebas de validación. Finalmente comprobaremos que el sistema funciona de manera correcta en los equipos y bajo el uso de personal capacitado a través de la prueba del sistema.

En las secciones siguientes se describe cada una de las pruebas y los resultados que se obtuvieron al llevarse a cabo bajo el sistema SAPIECC.

7.1.1 Pruebas de Unidad

Para verificar los el sistema SAPIECC partiremos de lo más pequeño hacia lo más grande, esto implica que el primer tipo de prueba será de unidad que, como ya se mencionó centra el proceso de verificación en la menor unidad de diseño: el componente software o módulo.

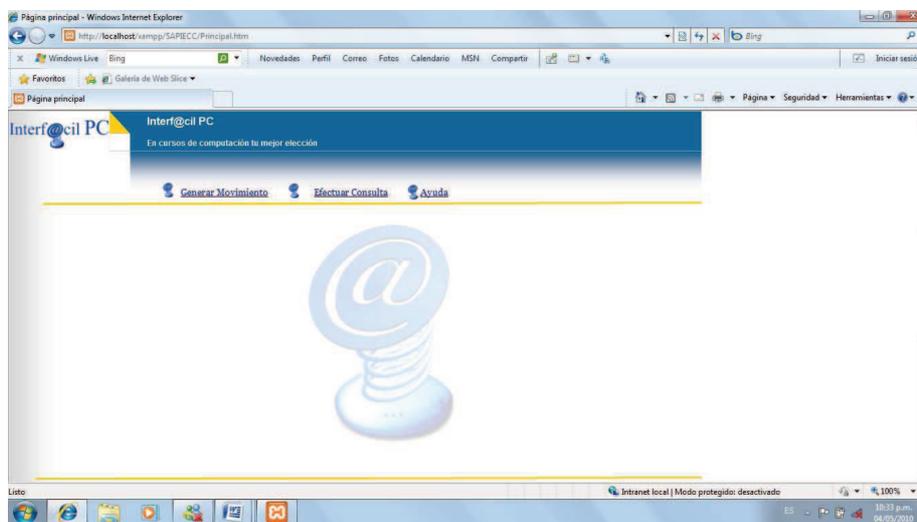
Para realizar una correcta inspección se tomaran en cuenta los siguientes aspectos en cada módulo:

- **Interfaz:** Para asegurar que la información fluye de forma adecuada hacia y desde la unidad de programa que está siendo probada
- **Estructuras de Datos Locales:** Para asegurar que los datos que se mantienen temporalmente conservan su integridad durante todos los pasos de ejecución del algoritmo.
- **Condiciones Límite:** Para asegurar que el módulo funciona correctamente en los límites establecidos como restricciones de procesamiento.
- **Caminos Independientes:** con el fin de asegurar que todas las sentencias del módulo se ejecutan por lo menos una vez.

Enseguida se muestran los resultados de la prueba de unidad para cada módulo de SAPIECC de acuerdo al diseño arquitectónico descrito en el capítulo anterior.

GESTOR DE INTERACCIÓN CON EL USUARIO

- *Interfaz*



- *Estructuras de Datos Locales*
Este módulo en particular no almacena ningún dato por lo que no está relacionado directamente con ninguna estructura, su única función es direccionar al usuario hacia un movimiento o bien hacia una consulta.
- *Condiciones Límite*

Al ser solo una pantalla de bienvenida no maneja valores numéricos de frontera por lo cual puede aparecer cualquier número de veces sin que presente problemas de operación.

- *Caminos Independientes*

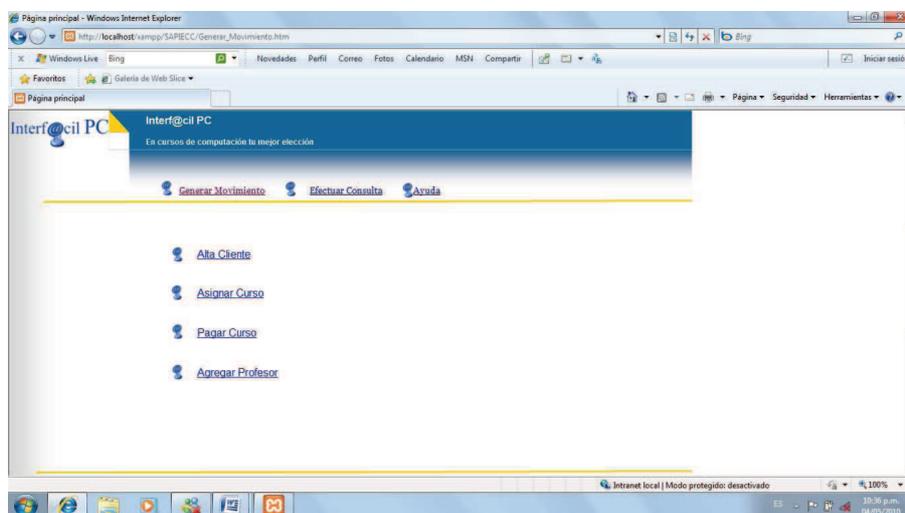
Existen tres opciones a considerar en esta pantalla. Para elegir cada una de ellas solo es necesario dar clic sobre el hipervínculo correspondiente:

OPCIÓN	RESULTADO
Generar Movimiento	Despliega la pantalla correspondiente a los diferentes movimientos que se pueden efectuar en el sistema y que está relacionada en la arquitectura del sistema con el Controlador de Datos del Sistema.
Efectuar Consulta	Despliega la pantalla correspondiente al Controlador de Consultas del Sistema, en donde el usuario puede visualizar las consultas preprogramadas de SAPIECC
Ayuda	Despliega el tutorial de ayuda para el sistema.

CONTROLADOR DE DATOS DE SISTEMA

- *Interfaz*

El Controlador de Datos del Sistema permite al usuario interactuar directamente con la base de datos a través de funciones como Alta de Clientes, Asignación de Cursos, Pago de Cursos y Alta de Profesores. Su apariencia se muestra a continuación, mientras que su descripción detallada se puede consultar en el Diccionario de Datos.



- *Estructuras de Datos Locales*

El controlador de Datos del Sistema en realidad no arroja, ni almacena ningún dato, su función es direccionar hacia los diferentes módulos que si permiten interactuar directamente con la base de datos del sistema

- *Condiciones Límite*

Por no manejar valores iniciales ni finales, no tiene condiciones en el límite su aparición está determinada por la forma en que el usuario navegue a través del sistema.

- *Caminos Independientes*

De acuerdo a la arquitectura de SAPIECC el controlador presenta cuatro caminos posibles, los cuales se pueden elegir dando clic sobre el hipervínculo correspondiente.

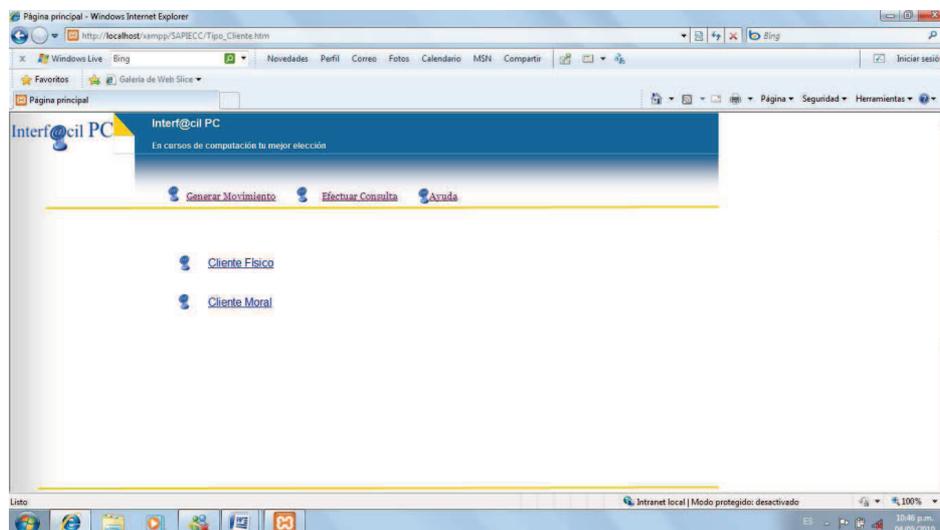
OPCIÓN	RESULTADO
Alta Cliente	Despliega la pantalla correspondiente al alta de clientes.
Asignar Curso	Despliega la pantalla correspondiente a la asignación de cursos
Pagar Curso	Despliega la pantalla que tiene como función llevar el registro de datos.
Asignar Profesor	Despliega la interfaz correspondiente al alta de profesores en el plantel.

Además en la parte superior se cuenta con las mismas opciones que el gestor de Interacción con el usuario, cuyo funcionamiento ya se explico anteriormente.

ALTA CLIENTE

- *Interfaz*

La forma en que el usuario interactúa con este módulo de SAPIECC es a través de la siguiente pantalla.



- *Estructuras de Datos Locales*

De manera directa este módulo no almacena ni arroja ningún resultado solo direcciona de acuerdo a la opción que se elija.

- *Condiciones Límite*

Por no almacenar ni devolver resultados no presenta condiciones en el límite.

- *Caminos Independientes*

Existen dos opciones que se pueden tomar en este módulo las cuales se describen a continuación:

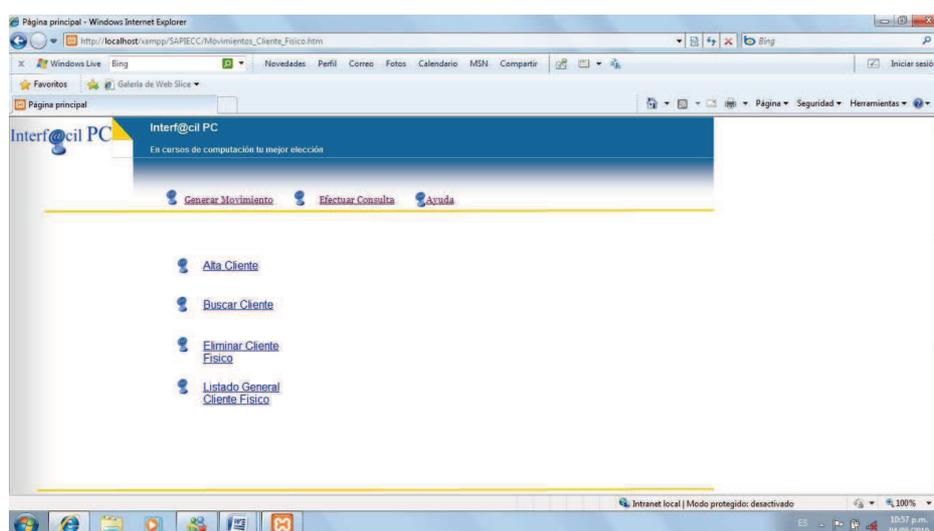
OPCIÓN	RESULTADO
Cliente Físico	Despliega la pantalla correspondiente al Alta de Clientes Físicos.
Cliente Moral	Despliega la pantalla correspondiente al Alta de Clientes Morales o Empresas.

Además en la parte superior se cuenta con las mismas opciones que el Gestor de Interacción con el Usuario, cuyo funcionamiento ya se explicó anteriormente.

OPCIONES ALTA CLIENTE FÍSICO

- *Interfaz*

Las opciones correspondientes al Alta de Clientes físicos tienen la apariencia siguiente:



- *Estructuras de Datos Locales*

En este caso todos los movimientos que realiza este módulo operan directamente sobre la tabla Cliente aunque también se relaciona con la tabla Cliente Moral. Para mayor información se puede consultar el diccionario de datos.

- *Condiciones Límite*

No aplica condiciones en el límite porque directamente en esta pantalla no se maneja un rango de valores.

- *Caminos Independientes*

Existen cuatro opciones que se pueden elegir dentro de este módulo. Enseguida se describe su funcionamiento:

OPCIÓN	RESULTADO
Alta Cliente	Despliega la pantalla correspondiente al movimiento de Alta de Clientes Físicos, desplegando todos los datos necesarios para realizar este.
Buscar Cliente	Despliega la pantalla que permite introducir parámetros para realizar diferentes tipos de búsqueda dentro de la tabla Clientes Físicos.
Eliminar Cliente	Despliega la pantalla destinada a la eliminación de Clientes Físicos.
Listado General de Clientes Físicos	Muestra un reporte con las tabla completa de Clientes Físicos almacenados hasta el momento en la base de datos del sistema.

Enseguida se describen algunas características para cada uno de estos casos.

a) Caso Alta Cliente

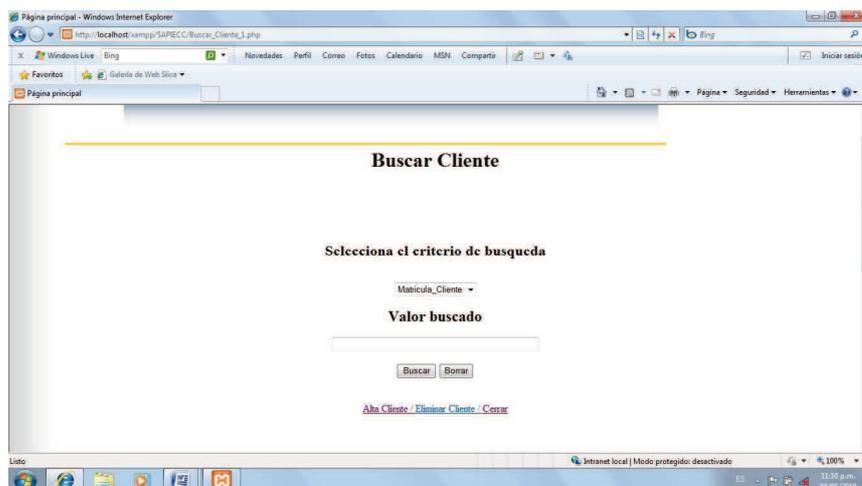
The screenshot shows a web browser window with the URL http://localhost:8080/SAPECC/Alta_Cliente_1.php. The page title is 'Alta Cliente'. Below the title, it says 'Conexión con la base de datos conseguida.' and 'Matricula Cliente: 67'. The form contains the following fields: Nombre, Apellido Paterno, Apellido Materno, Calle y Número, Colonia, Delegación / Municipio, Código Postal, Teléfono, E-mail, RFC, and Matricula Empresa. At the bottom of the form, there are two buttons: 'Alta Cliente' and 'Borrar'.

Los datos que se requieren para generar el alta de un Cliente Físico son: Nombre, Apellido Paterno, Apellido Materno, Calle y Número, Colonia, Delegación / Municipio, Código Postal, Teléfono, E-mail, RFC, Matricula Empresa. Estos datos interaccionan con la tabla Cliente al dar clic en el botón Alta Cliente, presentando un comportamiento como el que se describe a continuación.

DATOS DE ENTRADA	RESULTADO DE LA PRUEBA
<p><u>Caso 1: Información Completa</u></p> <p>Datos de Entrada: <i>Nombre:</i> David <i>Apellido Paterno:</i> Flores <i>Apellido Materno:</i> Lona <i>Calle y Número:</i> Insurgentes 434 <i>Colonia:</i> Nápoles <i>Delegación / Municipio:</i> Benito Juárez <i>Código Postal:</i> 03100 <i>Teléfono:</i> 9180-4142 <i>E-mail:</i> david_lona@hotmail.com</p>	<p>Se almacena en la tabla Clientes el registro siguiente:</p> <p>“David”, “Flores”, “Lona”, “Insurgentes 434”, “Nápoles”, “Benito Juárez”, 03100, 91804142, “david_lona@hotmail.com”, “xxxx”, 5</p>

<p>RFC: xxxx Matricula Empresa: 5</p>	
<p><u>Caso 2: Matricula Empresa Invalida</u></p> <p>Datos de Entrada: <i>Nombre:</i> Carlos <i>Apellido Paterno:</i> Flores <i>Apellido Materno:</i> Flores <i>Calle y Número:</i> Centenario 24 <i>Colonia:</i> Nápoles <i>Delegación / Municipio:</i> Benito Juárez <i>Código Postal:</i> 03100 <i>Teléfono:</i> 9180-4142 <i>E-mail:</i> carlifloflo@hotmail.com RFC: xxxx Matricula Empresa: 234</p>	<p>Mensaje de Error:</p> <p>“Empresa Invalida, consulte el catálogo de Clientes Morales”</p>
<p><u>Caso 3: Matricula Empresa = “0”</u></p> <p>Datos de Entrada: <i>Nombre:</i> Adrian <i>Apellido Paterno:</i> Peña <i>Apellido Materno:</i> Hernández <i>Calle y Número:</i> Luis Saviñon 54 <i>Colonia:</i> Del Valle <i>Delegación / Municipio:</i> Benito Juárez <i>Código Postal:</i> 03100 <i>Teléfono:</i> 9180-4142 <i>E-mail:</i> carlifloflo@hotmail.com RFC: xxxx Matricula Empresa: 0</p>	<p>Se almacena en la tabla Clientes el registro siguiente:</p> <p>“Adrian”,”Peña”,”Hernández”,”Luis Saviñon 54”,”Del Valle”,”Benito Juárez”, 03100, 91804142, “carlifloflo@hotmail.com”, “xxxx”, 0</p> <p>Nota: Los alumnos con matricula de empresa = 0, hacen referencia a aquellos que no provienen de alguna empresa en particular.</p>

b) Caso Buscar Cliente



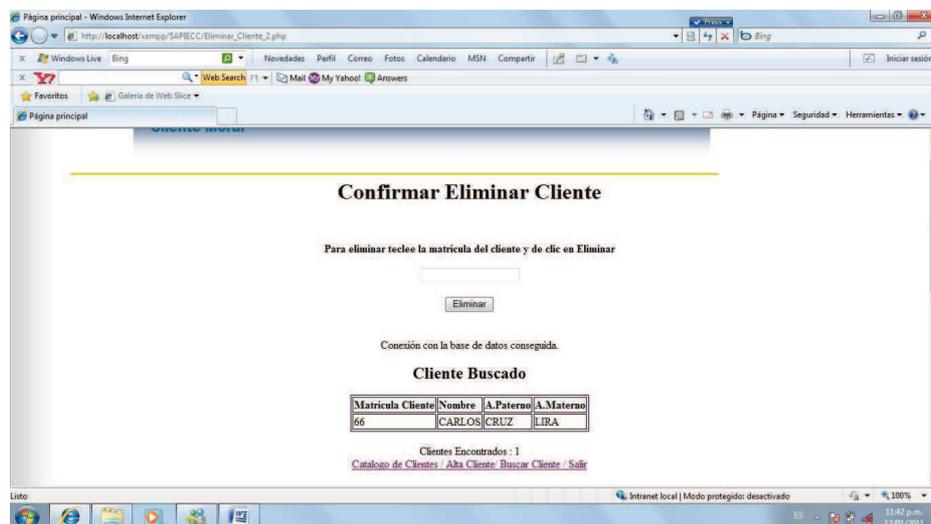
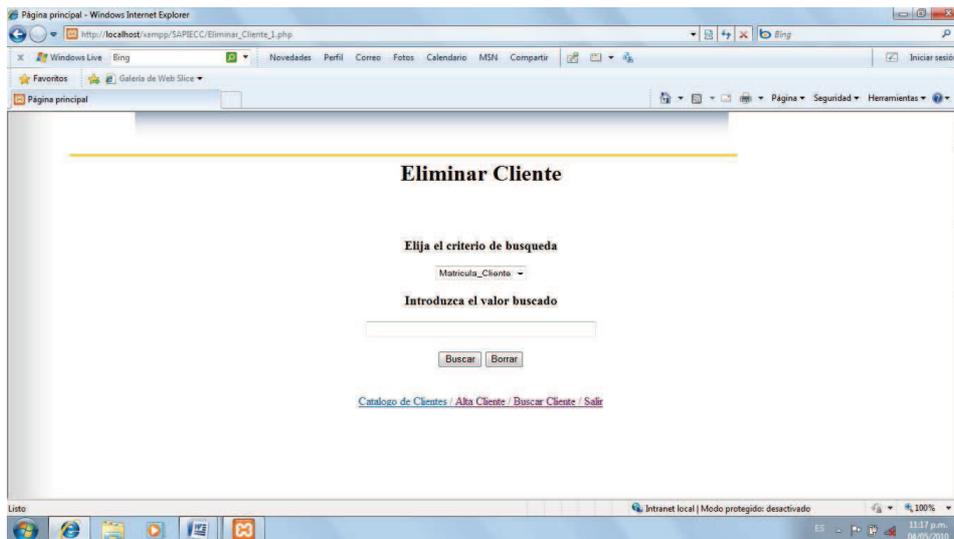
Los datos que se requieren para realizar la búsqueda de un cliente físico son: El criterio de búsqueda y el valor buscado. Existen dos formas de realizar estas búsquedas: por matrícula y por nombre, enseguida se describe el comportamiento de SAPIECC para algunos casos de búsqueda:

DATOS DE ENTRADA	RESULTADO DE LA PRUEBA								
<p><u>Caso 1: Matrícula de Cliente Existente</u></p> <p>Datos de Entrada: <i>Criterio de Búsqueda:</i> Matrícula Cliente <i>Valor Buscado:</i> 2</p>	<p>Pantalla Resultado de Búsqueda indicando una tabla con la matrícula y el nombre del cliente correspondiente.</p> <table border="1"> <thead> <tr> <th>Matricula Cliente</th> <th>Nombre</th> <th>A.Paterno</th> <th>A.Materno</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>MAYRA</td> <td>DANIEL</td> <td>ARGANIS</td> </tr> </tbody> </table>	Matricula Cliente	Nombre	A.Paterno	A.Materno	2	MAYRA	DANIEL	ARGANIS
Matricula Cliente	Nombre	A.Paterno	A.Materno						
2	MAYRA	DANIEL	ARGANIS						
<p><u>Caso 2: Buscar por Matrícula que no existe</u></p> <p>Datos de Entrada: <i>Criterio de Búsqueda:</i> Matrícula Cliente <i>Valor Buscado:</i> 525</p>	<p>Pantalla Resultado de Búsqueda indicando 0 ocurrencias de la búsqueda. Es decir que el cliente no se encuentra dentro de la base de datos.</p>								
<p><u>Caso 3: Buscar por Nombre que Existe</u></p> <p>Datos de Entrada: <i>Criterio de Búsqueda:</i> Nombre Cliente <i>Valor Buscado:</i>“IVAN MEJÍA HERNÁNDEZ”</p>	<p>Pantalla Resultado de Búsqueda indicando el número de ocurrencias de la búsqueda y una tabla con la matrícula y el nombre del cliente correspondiente.</p> <table border="1"> <thead> <tr> <th>Matricula Cliente</th> <th>Nombre</th> <th>A.Paterno</th> <th>A.Materno</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>IVÁN</td> <td>MEJÍA</td> <td>HERNÁNDEZ</td> </tr> </tbody> </table>	Matricula Cliente	Nombre	A.Paterno	A.Materno	3	IVÁN	MEJÍA	HERNÁNDEZ
Matricula Cliente	Nombre	A.Paterno	A.Materno						
3	IVÁN	MEJÍA	HERNÁNDEZ						
<p><u>Caso 4: Buscar por Nombre que no existe</u></p> <p>Datos de Entrada: <i>Criterio de Búsqueda:</i> Nombre Cliente <i>Valor Buscado:</i>“EDGAR LÓPEZ LÓPEZ”</p>	<p>Pantalla Resultado de Búsqueda indicando 0 ocurrencias de la búsqueda. Es decir que el cliente no se encuentra dentro de la base de datos.</p>								

Para conocer a detalle las pantallas correspondientes al movimiento de búsqueda de clientes físicos se puede dirigir a la sección de ayuda, ahí se describe a detalle el funcionamiento de este movimiento.

c) Caso Eliminar Cliente

Para que el sistema pueda eliminar un cliente de la base de datos primero es necesario realizar la búsqueda de la persona que se desea quitar y posteriormente es necesario confirmar que se desea eliminar definitivamente de la base de datos del sistema. Por lo que para los datos que se necesitan son el criterio de búsqueda y el valor buscado.



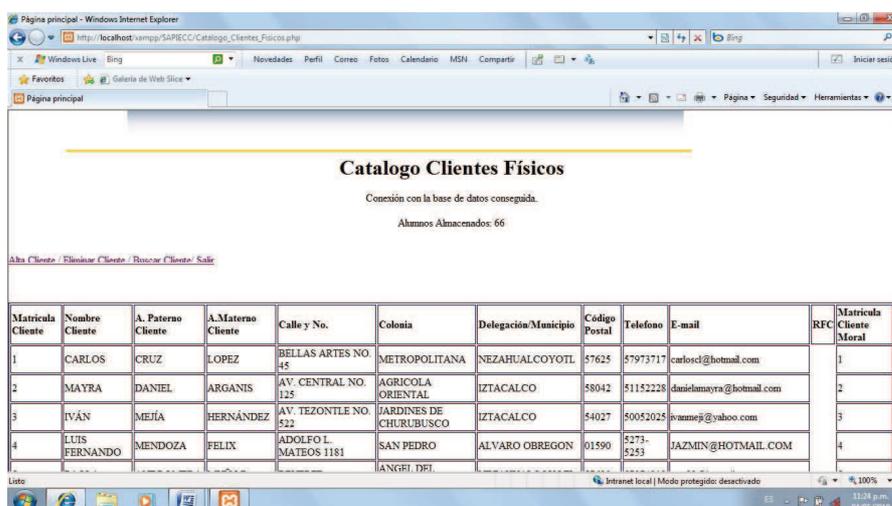
Para realizar la búsqueda se tienen los mismos criterios que el comando búsqueda y una vez encontrado el cliente en la base se tienen los siguientes casos:

DATOS DE ENTRADA	RESULTADO DE LA PRUEBA																								
<p><u>Caso1: Confirmación Correcta</u></p> <p>Datos de Entrada: Criterio de Búsqueda: Matricula Cliente Valor buscado: 58 Confirmación Matricula:58</p>	<p>Pantalla Cliente Eliminado indicando que se ha eliminado el registro con la siguiente información:</p> <table border="1"> <thead> <tr> <th>Matricula Cliente</th> <th>Nombre Cliente</th> <th>A. Paterno Cliente</th> <th>A.Materno Cliente</th> <th>Calle y No.</th> <th>Colonia</th> <th>Delegación/Municipio</th> <th>Código Postal</th> <th>Teléfono</th> <th>E-mail</th> <th>RF C</th> <th>Matricula Cliente Moral</th> </tr> </thead> <tbody> <tr> <td>58</td> <td>PAOLA</td> <td>ANDROMEDA</td> <td>MUÑOZ</td> <td>CALLER</td> <td>COLONIA</td> <td></td> <td>CP</td> <td>X</td> <td>XX</td> <td>X</td> <td>0</td> </tr> </tbody> </table>	Matricula Cliente	Nombre Cliente	A. Paterno Cliente	A.Materno Cliente	Calle y No.	Colonia	Delegación/Municipio	Código Postal	Teléfono	E-mail	RF C	Matricula Cliente Moral	58	PAOLA	ANDROMEDA	MUÑOZ	CALLER	COLONIA		CP	X	XX	X	0
Matricula Cliente	Nombre Cliente	A. Paterno Cliente	A.Materno Cliente	Calle y No.	Colonia	Delegación/Municipio	Código Postal	Teléfono	E-mail	RF C	Matricula Cliente Moral														
58	PAOLA	ANDROMEDA	MUÑOZ	CALLER	COLONIA		CP	X	XX	X	0														

<p><u>Caso 2: Confirmación Incorrecta:</u></p> <p>Datos de Entrada: <i>Criterio de Búsqueda: Matricula Cliente</i> <i>Valor buscado: 9</i> <i>Confirmación Matricula:7</i></p>	<p>Mensaje de error indicando que no se puede eliminar el registro debido a que las claves no coinciden.</p>
<p><u>Caso 3: Búsqueda Invalida</u></p> <p>Datos de Entrada: <i>Criterio de Búsqueda: Matricula Cliente</i> <i>Valor buscado: 525</i> <i>Confirmación Matricula:525</i></p>	<p>Mensaje de error indicando que no se puede eliminar el registro debido el cliente no existe dentro de la base de datos.</p>

Para este comando es válido realizar las búsquedas de acuerdo a otro criterio, sin embargo los resultados de la eliminación son los que ya se describieron independientemente del criterio de búsqueda que se elija.

d) Caso Listado General Cliente Físico

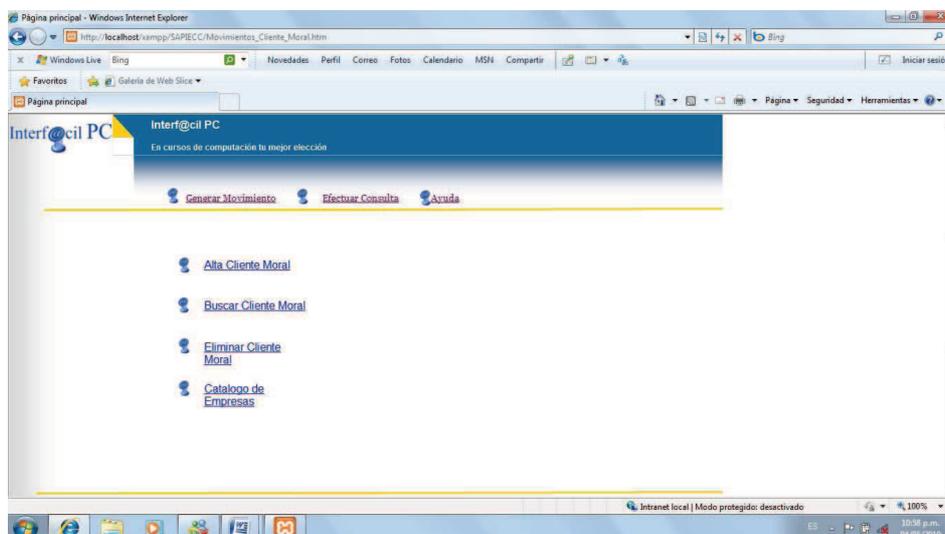


En este caso solo aparece la pantalla con todos los registros integrantes de la tabla clientes físicos. En esta opción al igual que en todas las correspondientes a movimientos cliente físico se cuenta con un menú que permite al usuario desplazarse a través de los distintos movimientos como son altas, búsquedas y eliminaciones.

OPCIONES ALTA CLIENTE MORAL

- *Interfaz*

Las opciones correspondientes al Alta de Cliente Moral tienen la apariencia siguiente:



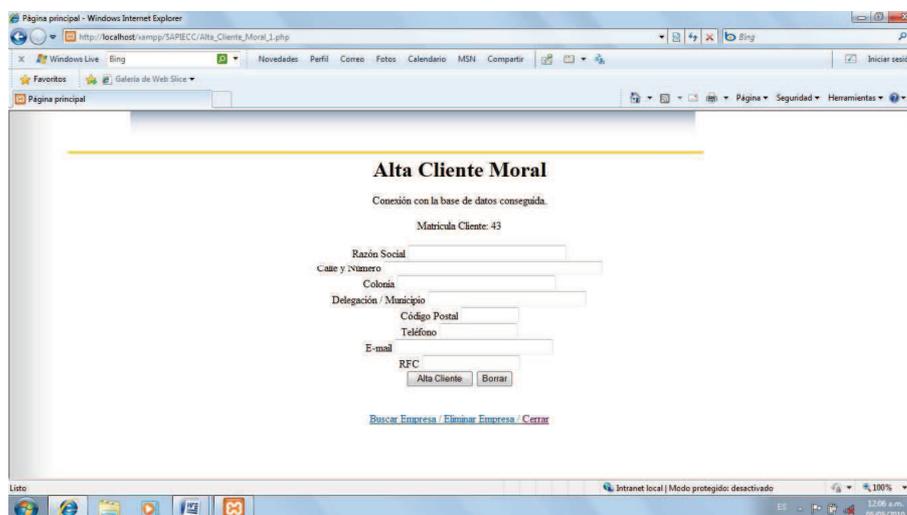
- *Estructuras de datos Locales*
En este caso todos los movimientos que realiza este módulo operan directamente sobre la tabla Cliente Moral. Para mayor información se puede consultar el diccionario de datos.
- *Condiciones Límite*
No aplica condiciones en el límite porque directamente en esta pantalla no se maneja un rango de valores.
- *Caminos Independientes*

Existen cuatro opciones que se pueden elegir dentro de este módulo. Enseguida se describe su funcionamiento:

OPCIÓN	RESULTADO
Alta Cliente Moral	Despliega la pantalla correspondiente al movimiento de Alta de Clientes Morales, desplegando todos los datos necesarios para realizar este.
Buscar Cliente Moral	Despliega la pantalla que permite introducir parámetros para realizar diferentes tipos de búsqueda dentro de la tabla Cliente Moral.
Eliminar Cliente Moral	Despliega la pantalla destinada a la eliminación de Clientes Morales.
Catálogo de Empresas	Muestra un reporte con las tabla completa de Clientes Morales almacenados hasta el momento en la base de datos del sistema.

Enseguida se describen algunas características para cada uno de estos casos.

a) Caso Alta Cliente Moral

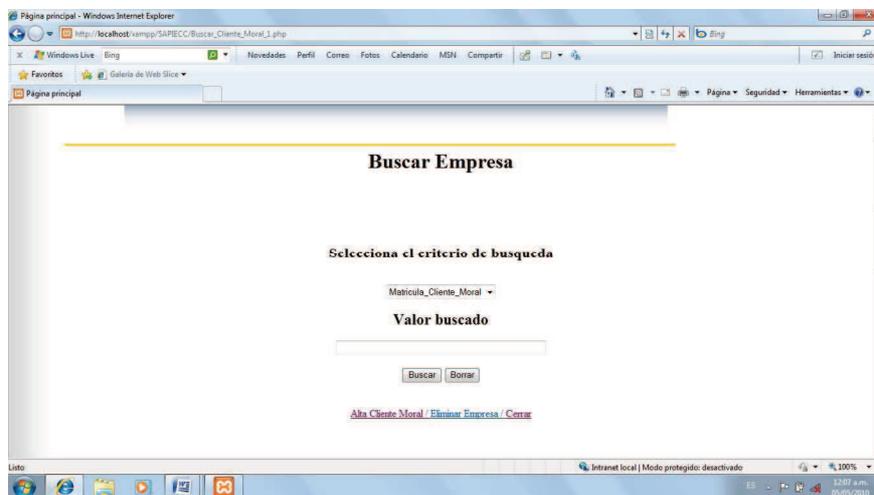


Los datos que se requieren para generar el alta de un Cliente Físico son: Razón Social, Calle y Número, Colonia, Delegación / Municipio, Código Postal, Teléfono, E- mail, RFC. Estos datos interaccionan con la tabla Cliente Moral al dar clic en el botón Alta Cliente, presentando un comportamiento como el que se describe a continuación.

DATOS DE ENTRADA	RESULTADO DE LA PRUEBA
<p><u>Caso 1: Información Completa</u></p> <p>Datos de Entrada: <i>Razón Social:</i> Grupo Ceres <i>Calle y Número:</i> Río Mixcoac 298 <i>Colonia:</i> Del Valle Sur <i>Delegación / Municipio:</i> Benito Juárez <i>Código Postal:</i> 03100 <i>Teléfono:</i> 9514-3465 <i>E-mail:</i> recursoh@ceres.com.mx <i>RFC:</i> CEWSMX0892T</p>	<p>Se almacena en la tabla Cliente Moral el registro siguiente:</p> <p>“97”, “Grupo Ceres”, “Del Valle Sur”, “Benito Juárez”, 03100, 95143465, “recursoh@ceres.com.mx”, “CEWSMX0892T”</p> <p>Nota: El primer valor corresponde a la matricula del cliente moral, este valor es autonumérico generado automáticamente por el sistema.</p>
<p><u>Caso 2: Razón Social Existente</u></p> <p>Datos de Entrada: <i>Razón Social:</i> Grupo Ceres <i>Calle y Número:</i> Río Mixcoac 298 <i>Colonia:</i> Del Valle Sur <i>Delegación / Municipio:</i> Benito Juárez <i>Código Postal:</i> 03100 <i>Teléfono:</i> 9514-3465 <i>E-mail:</i> recursoh@ceres.com.mx <i>RFC:</i> CEWSMX0892T</p>	<p>Mensaje de Error:</p> <p>“Empresa existente, consulte el catálogo de Clientes Morales”</p>
<p><u>Caso 3: RFC Existente</u></p>	<p>Mensaje de Error:</p>

<p>Datos de Entrada: <i>Razón Social:</i> Grupo Ceres <i>Calle y Número:</i> Río Mixcoac 298 <i>Colonia:</i> Del Valle Sur <i>Delegación / Municipio:</i> Benito Juárez <i>Código Postal:</i> 03100 <i>Teléfono:</i> 9514-3465 <i>E-mail:</i> recursoh@ceres.com.mx <i>RFC:</i> CEWSMX0892T</p>	<p>“RFC existente, consulte el catálogo de Clientes Morales”</p>
---	--

b) Caso Buscar Cliente Moral



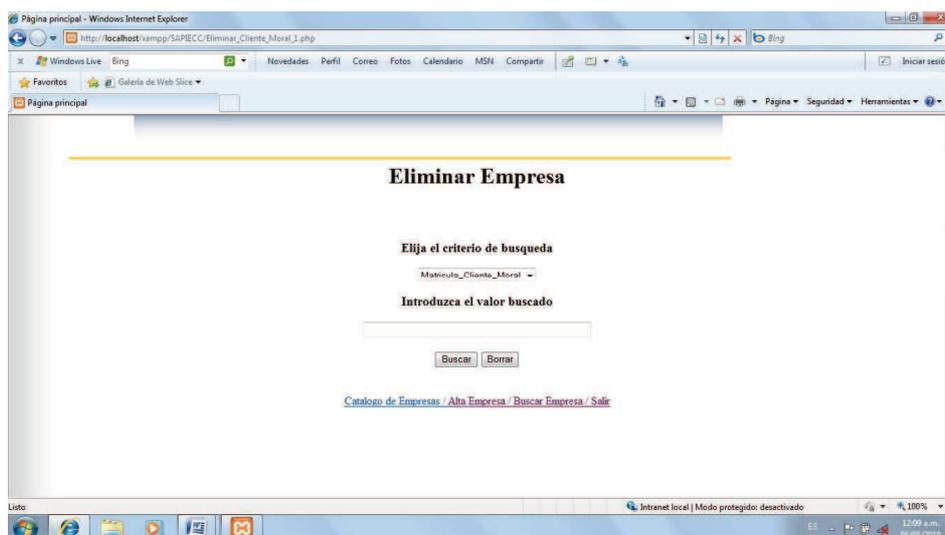
Los datos que se requieren para realizar la búsqueda de un cliente moral son: El criterio de búsqueda y el valor buscado. Existen dos formas de realizar estas búsquedas: por matricula del cliente moral y por Razón Social, enseguida se describe el comportamiento de SAPIECC para algunos casos de búsqueda:

DATOS DE ENTRADA	RESULTADO DE LA PRUEBA				
<p><u>Caso 1: Matricula de Cliente Moral Existente</u></p> <p>Datos de Entrada: <i>Criterio de Búsqueda:</i> Matricula Cliente <i>Valor Buscado:</i> 9</p>	<p>Pantalla Resultado de Búsqueda indicando una tabla con la matricula y el la razón social del cliente correspondiente.</p> <table border="1" data-bbox="818 1621 1343 1711"> <tr> <td>Matricula Cliente Moral</td> <td>Razón Social</td> </tr> <tr> <td>9</td> <td>OFFICE DEPOT</td> </tr> </table>	Matricula Cliente Moral	Razón Social	9	OFFICE DEPOT
Matricula Cliente Moral	Razón Social				
9	OFFICE DEPOT				
<p><u>Caso 2: Buscar por Matricula que no existe</u></p> <p>Datos de Entrada: <i>Criterio de Búsqueda:</i> Matricula Cliente <i>Valor Buscado:</i> 250</p>	<p>Pantalla Resultado de Búsqueda indicando 0 ocurrencias de la búsqueda. Es decir que el cliente moral no se encuentra dentro de la base de datos.</p>				
<p><u>Caso 3: Buscar por Razón Social que Existe</u></p>	<p>Pantalla Resultado de Búsqueda indicando el</p>				

<p>Datos de Entrada: <i>Criterio de Búsqueda:</i> Razón Social <i>Valor Buscado:</i>“JOSKES”</p>	<p>número de ocurrencias de la búsqueda y una tabla con la matrícula y la razón social correspondiente.</p> <table border="1" data-bbox="837 369 1324 459"> <tr> <td>Matricula Cliente Moral</td> <td>Razón Social</td> </tr> <tr> <td>20</td> <td>JOSKES</td> </tr> </table>	Matricula Cliente Moral	Razón Social	20	JOSKES
Matricula Cliente Moral	Razón Social				
20	JOSKES				
<p><u>Caso 4: Buscar por Razón Social que no existe</u> Datos de Entrada: <i>Criterio de Búsqueda:</i> Razón Social <i>Valor Buscado:</i>“INTELEISIS”</p>	<p>Pantalla Resultado de Búsqueda indicando 0 ocurrencias de la búsqueda. Es decir que el cliente moral no se encuentra dentro de la base de datos.</p>				

Para conocer a detalle las pantallas correspondientes al movimiento de búsqueda de clientes morales el usuario se puede dirigir a la sección de ayuda, ahí se describe a detalle el funcionamiento de este movimiento.

c) Caso Eliminar Cliente Moral



Para que el sistema pueda eliminar un cliente moral de la base de datos primero es necesario realizar la búsqueda de la empresa que se desea quitar y posteriormente es necesario confirmar que se desea eliminar definitivamente de la base de datos del sistema.

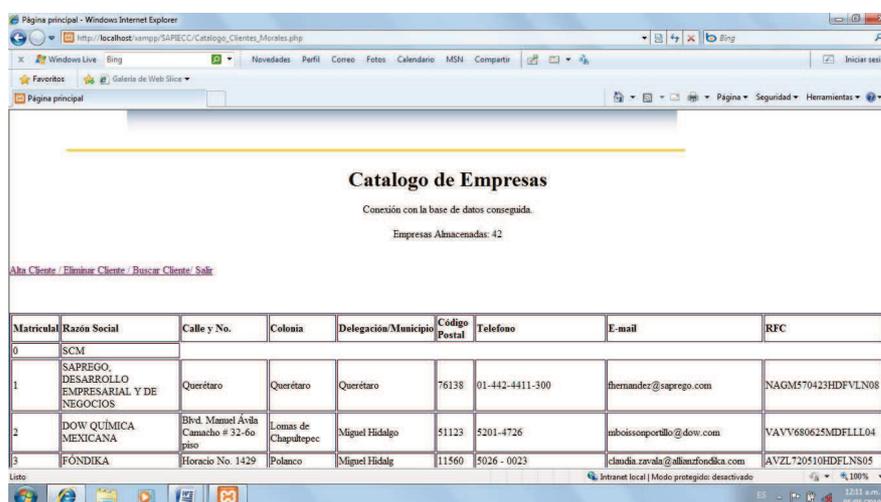
Para realizar la búsqueda se tienen los mismos criterios que el comando búsqueda y una vez encontrada la empresa en la base se tienen los siguientes casos:

DATOS DE ENTRADA	RESULTADO DE LA PRUEBA
<u>Caso1: Confirmación Correcta</u>	Pantalla Cliente Moral eliminado indicando que se ha eliminado el registro con la siguiente información:

<p>Datos de Entrada: <i>Criterio de Búsqueda:</i> Matricula Cliente Moral <i>Valor buscado:</i> 41 <i>Confirmación Matricula:</i>41</p>	<table border="1"> <tr> <td>Razón Social</td> <td>RFC</td> </tr> <tr> <td>CINETECA NACIONAL</td> <td>MAHE740418MGTZGN05</td> </tr> </table>	Razón Social	RFC	CINETECA NACIONAL	MAHE740418MGTZGN05
Razón Social	RFC				
CINETECA NACIONAL	MAHE740418MGTZGN05				
<p><u>Caso 2: Confirmación Incorrecta:</u> Datos de Entrada: <i>Criterio de Búsqueda:</i> Matricula Cliente Moral <i>Valor buscado:</i> 9 <i>Confirmación Matricula:</i>7</p>	<p>Mensaje de error indicando que no se puede eliminar el registro debido a que las claves no coinciden.</p>				
<p><u>Caso 3: Búsqueda Invalida</u> Datos de Entrada: <i>Criterio de Búsqueda:</i> Matricula Cliente Moral <i>Valor buscado:</i> 183 <i>Confirmación Matricula:</i>183</p>	<p>Mensaje de error indicando que no se puede eliminar el registro debido el cliente no existe dentro de la base de datos.</p>				

Para este comando es válido realizar las búsquedas de acuerdo a otro criterio, sin embargo los resultados de la eliminación son los que ya se describieron independientemente del criterio de búsqueda que se elija.

d) Caso Listado General Cliente Moral

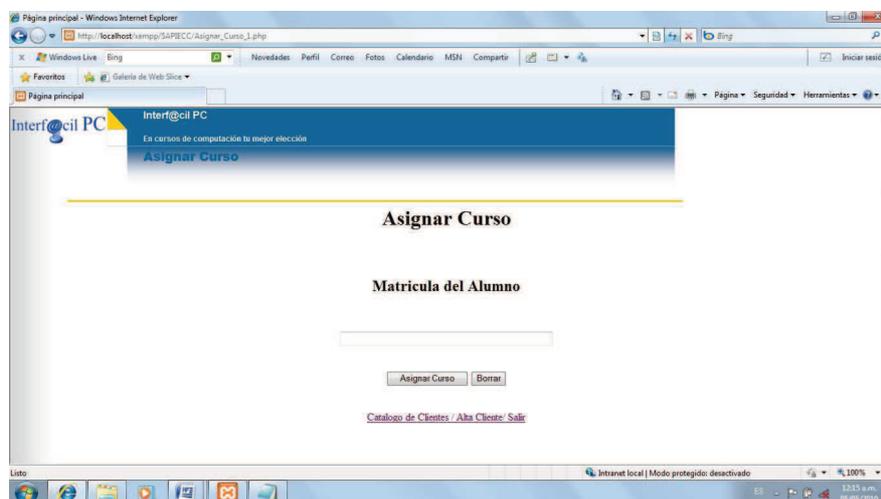


En este caso solo aparece la pantalla con todos los registros integrantes de la tabla clientes morales. En esta opción al igual que en todas las correspondientes a movimientos cliente moral se cuenta con un menú que permite al usuario desplazarse a través de los distintos movimientos como son altas, búsquedas y eliminaciones.

ASIGNAR CURSO

- *Interfaz*

La pantalla destinada a la asignación cursos tiene la apariencia siguiente:



- *Estructuras de Datos Locales*

Este módulo opera directamente sobre la tabla Curso que a su vez se encuentra relacionada con la tabla Cliente y Cliente Moral. Para mayor detalle se puede consultar el diccionario de datos del sistema.

- *Condiciones Límite*

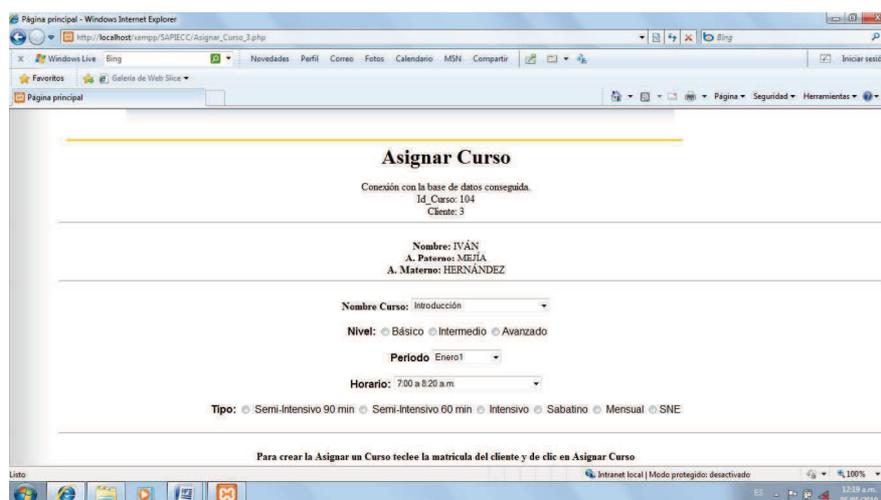
Los límites que se manejan para poder realizar las asignaciones de cursos corresponden a los clientes que existen almacenados dentro de la base de datos del sistema, además de las empresas registradas. Ya que en caso de no existir el sistema presentara las correspondientes pantallas de errores.

- *Caminos Independientes*

De acuerdo al resultado de insertar la matricula del alumno, tenemos dos principales casos de prueba para realizar la asignación de curso:

DATOS DE ENTRADA	RESULTADO DE LA PRUEBA
Matricula de Cliente Valida	Despliega la pantalla con los datos necesarios para la asignación de cursos.
Matricula de Cliente Invalida	Mensaje de error indicando que no se puede asignar el curso debido a que el cliente no existe dentro de la base de datos.

En el caso de que la matricula del cliente sea válida aparecerá la siguiente interfaz:



En esta pantalla aparecen los datos correspondientes a la matricula insertada además de las opciones correspondientes al curso que se desea asignar, una vez indicados estos es necesario realizar una confirmación insertando nuevamente la matricula del cliente para lo cual tenemos dos casos:

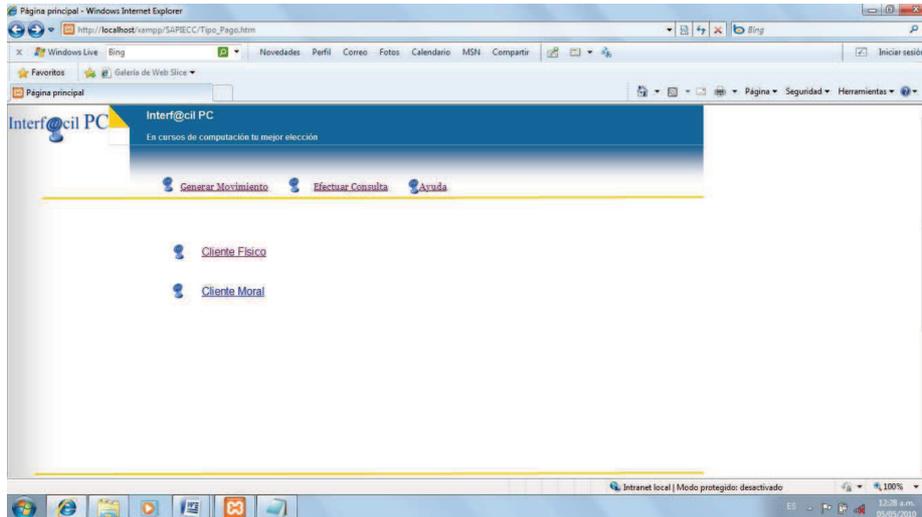
DATOS DE ENTRADA	RESULTADO DE LA PRUEBA														
<p><u>Caso 1: Confirmación Valida</u></p> <p>Datos de Entrada: Matricula Cliente: 3 Curso: Word Nivel: Básico Horario: 7:00 a 8:20 a.m. Tipo: Semi-Intensivo 90 min Periodo: Enero1 Confirmación Matricula: 3</p>	<p>Se almacena en la tabla Curso un registro con la siguiente información:</p> <table border="1"> <thead> <tr> <th>Id_Curso</th> <th>Matricula Cliente</th> <th>Nombre_Curso</th> <th>Nivel</th> <th>Horario</th> <th>Tipo</th> <th>Periodo</th> </tr> </thead> <tbody> <tr> <td>1 04</td> <td>3</td> <td>3</td> <td>1</td> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table> <p>En donde el primer valor corresponde al ID de Curso, el cual es generado automáticamente por el sistema y los valores siguiente corresponden a los valores asociados a las opciones elegidas para el curso.</p>	Id_Curso	Matricula Cliente	Nombre_Curso	Nivel	Horario	Tipo	Periodo	1 04	3	3	1	1	1	1
Id_Curso	Matricula Cliente	Nombre_Curso	Nivel	Horario	Tipo	Periodo									
1 04	3	3	1	1	1	1									
<p><u>Caso 2: Confirmación Invalida</u></p> <p>Datos de Entrada: Matricula Cliente: 5 Curso: Excel Nivel: Avanzado Horario: 7:00 a 8:20 a.m. Tipo: Semi-Intensivo 90 min Periodo: Enero2 Confirmación Matricula: 3</p>	<p>Mensaje de error indicando que la matricula no coincide y dando la opción de volver a colocar la matricula o salir.</p>														

Todas las interfaces de este comando tienen acceso a los comandos de alta cliente, búsqueda cliente y salir (regresar a la página principal). Esto tratando de facilitar al usuario el acceso a estos movimientos, en caso de ser necesarios.

PAGAR CURSO

- *Interfaz*

Existen dos formas de realizar el pago de un curso estas corresponden a notas de pago y facturas. El caso en que se generara cada uno de ellos corresponde al tipo de cliente, por lo que la primera pantalla que aparecerá será la correspondiente a tipo de cliente.



- *Estructuras de Datos Locales*

Este módulo opera directamente sobre las tablas Nota_Pago y Factura.

- *Condiciones Límite*

Los límites de trabajo para este módulo corresponden a los registros existentes en las tablas Cliente, Cliente Moral, Nota_Pago y Factura, ya que en caso de no existir la información necesaria el sistema mostrara los casos de error correspondientes a cada caso.

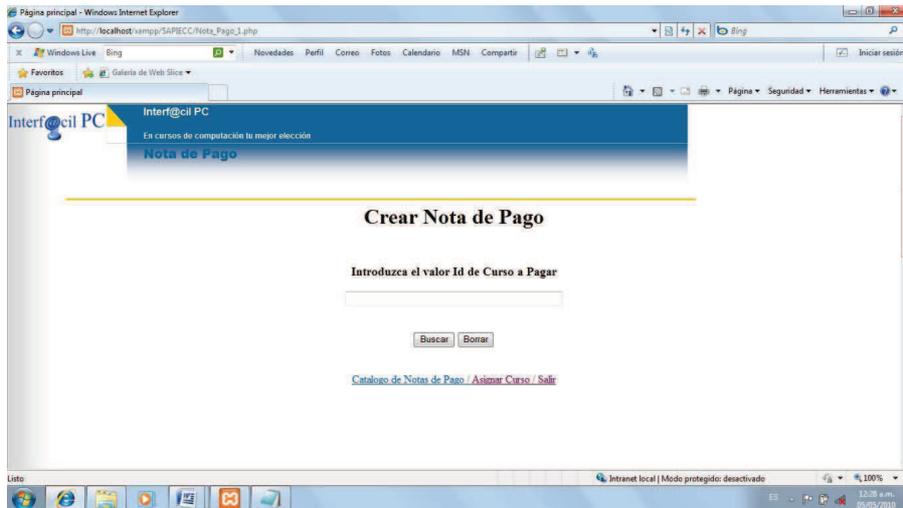
- *Caminos Independientes*

En la pantalla inicial existen principalmente dos caso de prueba:

OPCIÓN	RESULTADO
Cliente Físico	Despliega la pantalla correspondiente a Crear Nota de Pago
Cliente Moral	Despliega la pantalla correspondiente a Crear Factura

a) Crear Nota de Pago

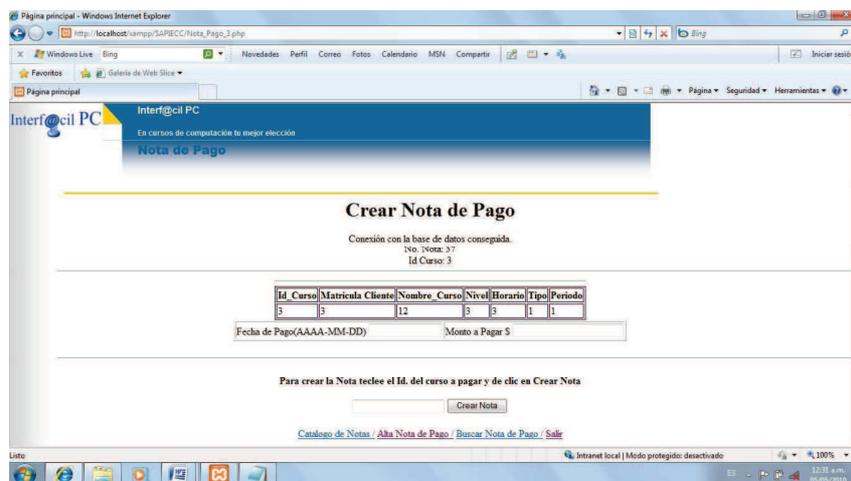
Para crear una nota de pago es necesario introducir el id_Curso que se desea pagar y dar clic en el botón buscar, para esto tenemos dos casos de prueba:



DATOS DE ENTRADA	RESULTADO DE LA PRUEBA
<p><u>Caso 1: Id Curso Valido</u></p> <p>Datos de Entrada: Id_Curso: 3</p>	<p>Muestra la pantalla crear nota de Pago con los datos correspondientes al id_Curso introducido y en donde se coloca además la fecha de pago y el monto a pagar.</p>
<p><u>Caso 2: Id Curso Invalido</u></p> <p>Datos de Entrada: Id_Curso: 930</p>	<p>Muestra pantalla de error indicando que el curso no existe.</p>

Esta pantalla cuenta además con las opciones de catálogo de notas de pago, para consultar todos los pagos que se han realizado y con el comando asignar curso además de la opción salir que regresa a la pantalla principal.

a.1) Id_Curso Valido

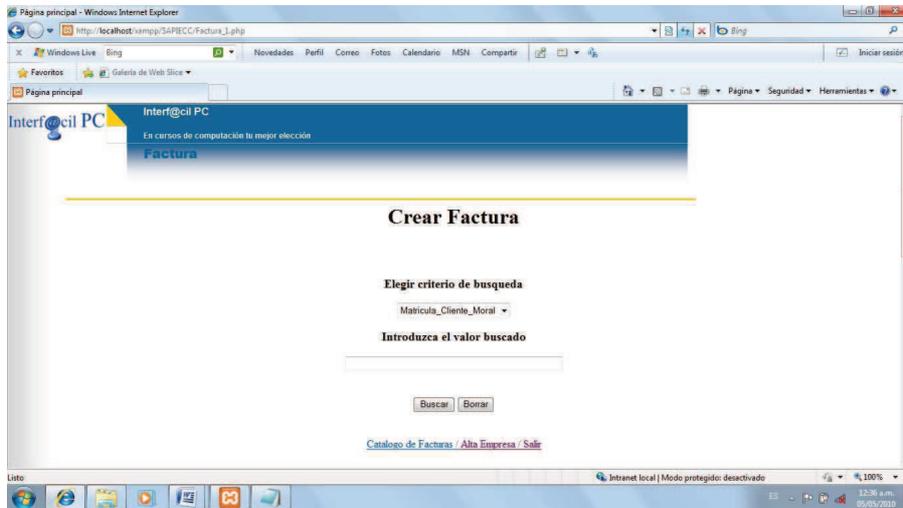


En caso de que la asignación de curso exista aparecerá una pantalla donde se solicita la fecha de pago y el monto a pagar. Además de una confirmación del pago a través del Id_Curso. Para esta opción tenemos los siguientes casos de prueba:

DATOS DE ENTRADA	RESULTADO DE LA PRUEBA								
<p><u>Caso 1: Fecha Correcta, Monto Correcto, Confirmación Correcta</u></p> <p>Datos de Entrada: Id_Curso: 15 Fecha de pago: 2011-01-19 Monto a pagar \$:750 Confirmación id_Curso:15</p>	<p>Aparece un mensaje en pantalla indicando que se almaceno en la tabla Nota_Pago un registro con la siguiente información:</p> <table border="1"> <thead> <tr> <th>No. Nota</th> <th>Monto Pagado</th> <th>Fecha de Pago</th> <th>Id. Curso</th> </tr> </thead> <tbody> <tr> <td>37</td> <td>750</td> <td>2011-01-159</td> <td>15</td> </tr> </tbody> </table> <p>Nota: El número de nota es generado automáticamente por el sistema.</p>	No. Nota	Monto Pagado	Fecha de Pago	Id. Curso	37	750	2011-01-159	15
No. Nota	Monto Pagado	Fecha de Pago	Id. Curso						
37	750	2011-01-159	15						
<p><u>Caso 2: Fecha Incorrecta, Monto Correcto, Confirmación Correcta</u></p> <p>Datos de Entrada: Id_Curso: 3 Fecha de pago: 01-12-1934 Monto a pagar \$:1500 Confirmación id_Curso:3</p>	<p>Mensaje de error indicando que la fecha es incorrecta.</p>								
<p><u>Caso 3: Fecha Correcta, Monto Incorrecto, Confirmación Correcta</u></p> <p>Datos de Entrada: Id_Curso: 5 Fecha de pago: 2011-02-09 Monto a pagar \$:7a50 Confirmación id_Curso:5</p>	<p>Mensaje de error indicando que la cantidad a pagar es incorrecta.</p>								
<p><u>Caso 4: Fecha Correcta, Monto Correcto, Confirmación Incorrecta.</u></p> <p>Datos de Entrada: Id_Curso: 4 Fecha de pago: 2011-01-19 Monto a pagar \$:750 Confirmación id_Curso:7</p>	<p>Mensaje de error indicando que el Id_Curso introducido no coincide con el curso que se desea asignar.</p>								

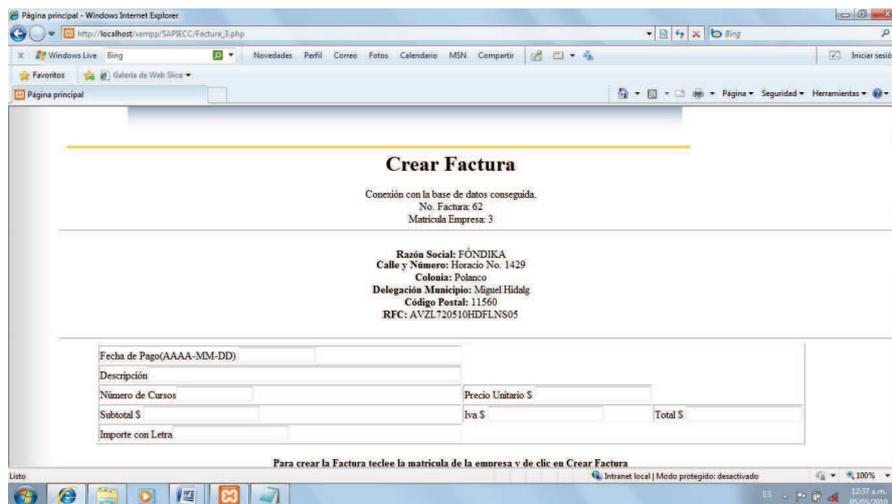
b) Crear Factura

En el caso del que el pago corresponda a un cliente moral o empresa, es necesario que antes de generar la factura se realice la búsqueda del cliente dentro de la base de datos para esto se requerirá introducir el criterio de búsqueda y el valor buscado. La búsqueda se puede realizar mediante la matrícula de la empresa o bien a través de la Razón social.



DATOS DE ENTRADA	RESULTADO DE LA PRUEBA
<p><u>Caso 1: Cliente Moral existente</u></p> <p>Datos de Entrada: Matricula_Cliente_Moral: 3</p>	<p>Muestra la pantalla crear Factura con los datos correspondientes al cliente moral introducido. El resultado es el mismo independientemente del criterio de búsqueda que se decida utilizar.</p>
<p><u>Caso 2: Cliente Moral Inexistente</u></p> <p>Datos de Entrada: Matricula_Cliente_Moral: 432</p>	<p>Muestra pantalla de error indicando que el curso no existe.</p>

b.1) Cliente Moral Existente



En caso de que el cliente moral exista, aparecerá la pantalla para crear la factura, la cual además de mostrar sus datos generales permitirá introducir la fecha de pago, el número de cursos a pagar, su descripción, precio unitario, además del importe con letra. Para este módulo se tienen los siguientes casos:

DATOS DE ENTRADA	RESULTADO DE LA PRUEBA																		
<p><u>Caso 1: Fecha Correcta, No Cursos Correcto, Precio Unitario Correcto Confirmación Correcta</u></p> <p>Datos de Entrada: Matricula Empresa: 3 Fecha de Pago: 2011-01-23 No. Cursos: 3 Precio Unitario: 2000 Confirm. Matricula Empresa: 3</p>	<p>Aparece un mensaje en pantalla indicando que se almaceno en la tabla Facturas un registro con la siguiente información:</p> <table border="1" data-bbox="692 589 1369 719"> <thead> <tr> <th>No. Factura</th> <th>Matricula Empresa</th> <th>Cantidad</th> <th>Descripción</th> <th>Precio Unitario</th> <th>Subtotal</th> <th>Iva</th> <th>Total</th> <th>Importe con Letra</th> </tr> </thead> <tbody> <tr> <td>35</td> <td>3</td> <td>3</td> <td>Cursos</td> <td>2000</td> <td>6000</td> <td>900</td> <td>6900</td> <td>Seis mil pesos</td> </tr> </tbody> </table> <p>Nota: El número de factura es generado automáticamente por el sistema.</p>	No. Factura	Matricula Empresa	Cantidad	Descripción	Precio Unitario	Subtotal	Iva	Total	Importe con Letra	35	3	3	Cursos	2000	6000	900	6900	Seis mil pesos
No. Factura	Matricula Empresa	Cantidad	Descripción	Precio Unitario	Subtotal	Iva	Total	Importe con Letra											
35	3	3	Cursos	2000	6000	900	6900	Seis mil pesos											
<p><u>Caso 2: Fecha Incorrecta, No Cursos Correcto, Precio Unitario Correcto Confirmación Correcta</u></p> <p>Datos de Entrada: Matricula Empresa: 7 Fecha de Pago: 11-12-2010 No. Cursos: 3 Precio Unitario: 2000 Confirm. Matricula Empresa: 7</p>	<p>Mensaje de error indicando que la fecha es incorrecta.</p>																		
<p><u>Caso 3: Fecha Correcta, No Cursos Correcto, Precio Unitario Incorrecto Confirmación Correcta</u></p> <p>Datos de Entrada: Matricula Empresa: 6 Fecha de Pago: 2011-01-23 No. Cursos: 2 Precio Unitario: as2 Confirm. Matricula Empresa: 6</p>	<p>Mensaje de error indicando que el precio unitario de los cursos a pagar es incorrecto.</p>																		
<p><u>Caso 4: Fecha Correcta, No Cursos Incorrecto, Precio Unitario Correcto Confirmación Correcta</u></p> <p>Datos de Entrada: Matricula Empresa: 3 Fecha de Pago: 2011-01-23 No. Cursos: s Precio Unitario: 2000 Confirm. Matricula Empresa: 3</p>	<p>Mensaje de error indicando que número de cursos es incorrecto.</p>																		
<p><u>Caso 5: Fecha Correcta, No Cursos Correcto, Precio Unitario Correcto Confirmación Incorrecta</u></p>	<p>Mensaje de error indicando que la matricula del cliente moral introducida no coincide con la de la empresa que desea generar la factura.</p>																		

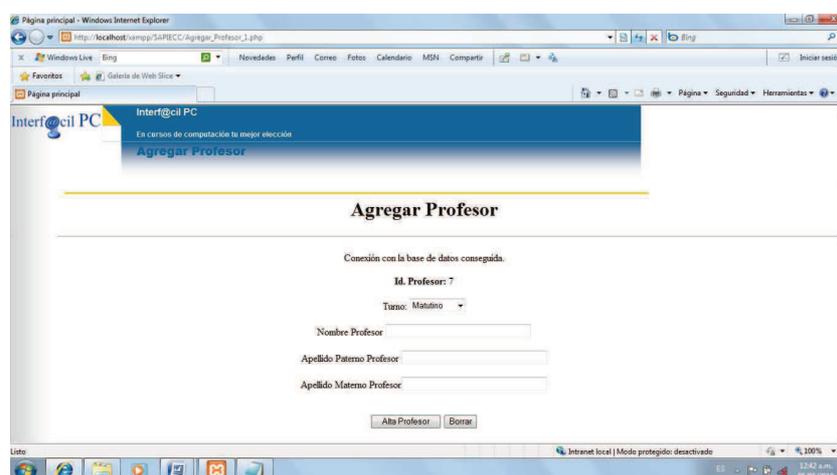
<p>Datos de Entrada: <i>Matricula Empresa:</i> 5 <i>Fecha de Pago:</i> 2011-01-23 <i>No. Cursos:</i> 3 <i>Precio Unitario:</i> 2000 <i>Confirm. Matricula Empresa:</i> 3</p>	
--	--

Es importante mencionar que en el caso de que se haya introducido más de un dato de manera incorrecta aparecerá el mensaje de error correspondiente al primero que detecte el sistema, de acuerdo al orden en que se realizaron estos casos de prueba.

AGREGAR PROFESOR

- *Interfaz*

En caso de que el usuario haya elegido el comando agregar profesor la pantalla que aparecerá será la siguiente.



- *Estructuras de Datos Locales*

Este módulo opera directamente sobre la tabla profesor, los datos que almacena son el Nombre, Apellido Paterno, Apellido Materno y Turno.

- *Condiciones Límite*

Este módulo genera de manera automática el id del profesor, esta es la clave máxima que puede utilizar para realizar cualquier operación dentro del sistema.

- *Caminos Independientes*

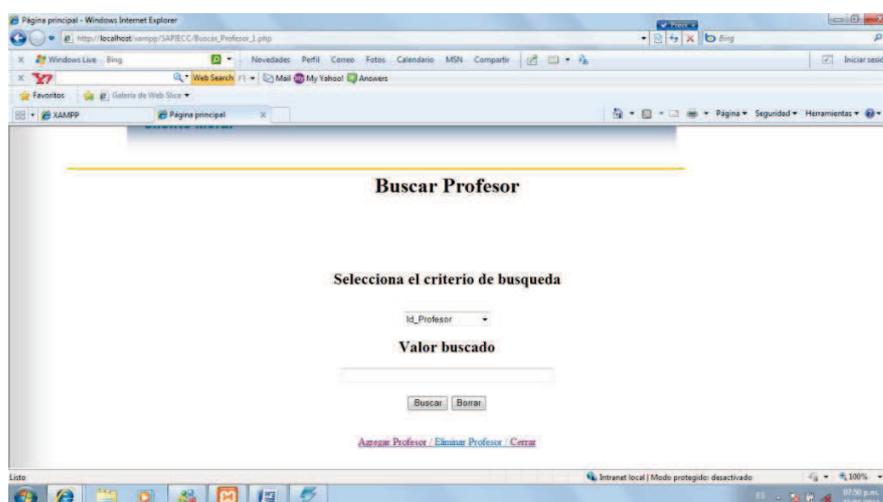
Existen tres opciones que se pueden elegir dentro de este módulo. Enseguida se describe su funcionamiento:

OPCIÓN	RESULTADO
Alta Profesor	Almacena en la tabla profesor los resultados insertados en el formulario correspondiente.

Buscar Cliente	Despliega la pantalla que permite introducir parámetros para realizar diferentes tipos de búsqueda dentro de la tabla Profesor.
Eliminar Profesor	Despliega la pantalla destinada a la eliminación de Profesores.

Enseguida se describen algunas características para cada uno de estos casos.

a) Caso Buscar Profesor

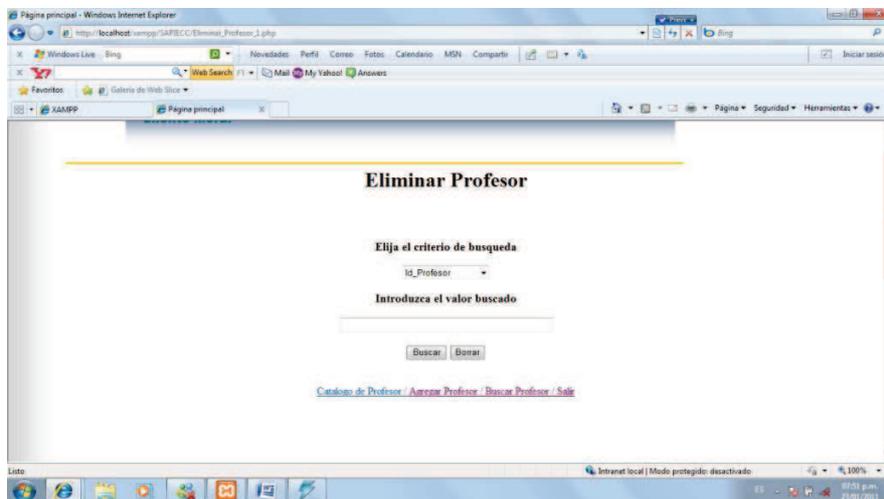


Los datos que se requieren para realizar la búsqueda de un profesor son: El criterio de búsqueda y el valor buscado. Existen dos formas de realizar estas búsquedas: por id_profesor y por nombre, enseguida se describe el comportamiento de SAPIECC para algunos casos de búsqueda:

DATOS DE ENTRADA	RESULTADO DE LA PRUEBA										
<p><u>Caso 1: Id Profesor Existente</u></p> <p>Datos de Entrada: Criterio de Búsqueda: id_Profesor Valor Buscado: 2</p>	<p>Pantalla Resultado de Búsqueda indicando una tabla con el id_Profesor y el nombre del profesor correspondiente.</p> <table border="1"> <thead> <tr> <th>Id. Profesor</th> <th>Turno</th> <th>Nombre</th> <th>A.Paterno</th> <th>A.Materno</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>1</td> <td>Erick</td> <td>Hernández</td> <td>López</td> </tr> </tbody> </table>	Id. Profesor	Turno	Nombre	A.Paterno	A.Materno	2	1	Erick	Hernández	López
Id. Profesor	Turno	Nombre	A.Paterno	A.Materno							
2	1	Erick	Hernández	López							
<p><u>Caso 2: Id Profesor inexistente</u></p> <p>Datos de Entrada: Criterio de Búsqueda: id_Profesor Valor Buscado: 51</p>	<p>Pantalla Resultado de Búsqueda indicando 0 ocurrencias de la búsqueda. Es decir que el profesor no se encuentra dentro de la base de datos.</p>										
<p><u>Caso 3: Buscar por Nombre que Existe</u></p> <p>Datos de Entrada:</p>	<p>Pantalla Resultado de Búsqueda indicando el número de ocurrencias de la búsqueda y una tabla</p>										

<p><i>Criterio de Búsqueda:</i> Nombre Profesor <i>Valor Buscado:</i>“Leticia Santiago Ramírez”</p>	<p>con el id_profesor y el nombre del profesor correspondiente.</p> <table border="1" data-bbox="815 342 1343 450"> <thead> <tr> <th>Id. Profesor</th> <th>Turno</th> <th>Nombre</th> <th>A.Paterno</th> <th>A.Materno</th> </tr> </thead> <tbody> <tr> <td>4</td> <td>3</td> <td>Leticia</td> <td>Santiago</td> <td>Ramírez</td> </tr> </tbody> </table>	Id. Profesor	Turno	Nombre	A.Paterno	A.Materno	4	3	Leticia	Santiago	Ramírez
Id. Profesor	Turno	Nombre	A.Paterno	A.Materno							
4	3	Leticia	Santiago	Ramírez							
<p><u>Caso 4: Buscar por Nombre que no existe</u> Datos de Entrada: <i>Criterio de Búsqueda:</i> Nombre Cliente <i>Valor Buscado:</i>“EDGAR LÓPEZ LÓPEZ”</p>	<p>Pantalla Resultado de Búsqueda indicando 0 ocurrencias de la búsqueda. Es decir que el profesor no se encuentra dentro de la base de datos.</p>										

b) Caso Eliminar Profesor



Para que el sistema pueda eliminar un profesor de la base de datos primero es necesario realizar la búsqueda de la persona que se desea quitar y posteriormente es necesario confirmar que se desea eliminar definitivamente de la base de datos del sistema. Por lo que para los datos que se necesitan son el criterio de búsqueda y el valor buscado.

Para realizar la búsqueda se tienen los mismos criterios que el comando búsqueda y una vez encontrado el profesor en la base se tienen los siguientes casos:

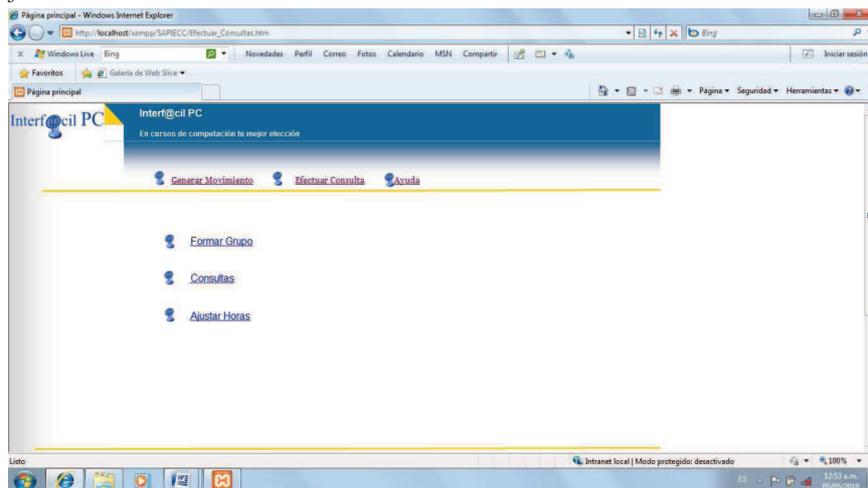
DATOS DE ENTRADA	RESULTADO DE LA PRUEBA										
<p><u>Caso1: Confirmación Correcta</u> Datos de Entrada: <i>Criterio de Búsqueda:</i> Id_Profesor <i>Valor buscado:</i> 6 <i>Confirmación Id_Profesor:</i>6</p>	<p>Pantalla Profesor Eliminado indicando que se ha eliminado el registro con la siguiente información:</p> <table border="1" data-bbox="667 1809 1372 1895"> <thead> <tr> <th>Id_Profesor</th> <th>Turno</th> <th>Nombre</th> <th>A.Paterno</th> <th>A.Materno</th> </tr> </thead> <tbody> <tr> <td>6</td> <td>2</td> <td>Gustavo</td> <td>García</td> <td>Ortiz</td> </tr> </tbody> </table>	Id_Profesor	Turno	Nombre	A.Paterno	A.Materno	6	2	Gustavo	García	Ortiz
Id_Profesor	Turno	Nombre	A.Paterno	A.Materno							
6	2	Gustavo	García	Ortiz							
<p><u>Caso 2: Confirmación Incorrecta:</u></p>	<p>Mensaje de error indicando que no se puede eliminar el registro debido a que las claves no coinciden.</p>										

<p>Datos de Entrada: <i>Criterio de Búsqueda:</i> Matricula <i>Cliente</i> <i>Valor buscado:</i> 9 <i>Confirmación Matricula:</i>7</p>	
<p><u>Caso 3: Búsqueda Invalida</u> Datos de Entrada: <i>Criterio de Búsqueda:</i> Id_Profesor <i>Valor buscado:</i> 525 <i>Confirmación Matricula:</i>525</p>	<p>Mensaje de error indicando que no se puede eliminar el registro debido el cliente no existe dentro de la base de datos.</p>

Para este comando es válido realizar las búsquedas de acuerdo a otro criterio, sin embargo los resultados de la eliminación son los que ya se describieron independientemente del criterio de búsqueda que se elija.

CONTROLADOR DE CONSULTAS DEL SISTEMA

- *Interfaz*



- *Estructuras de Datos Locales*

Este módulo en particular no está relacionado directamente con ninguna estructura del sistema, su principal función es direccionar al usuario hacia la consulta que desea realizar.

- *Condiciones Límite*

Debido a que este modulo no maneja valores numéricos ni contenidos dentro de la base de datos no presenta condiciones de límite. Su aparición dependerá de las necesidades del usuario del sistema.

- *Caminos Independientes*

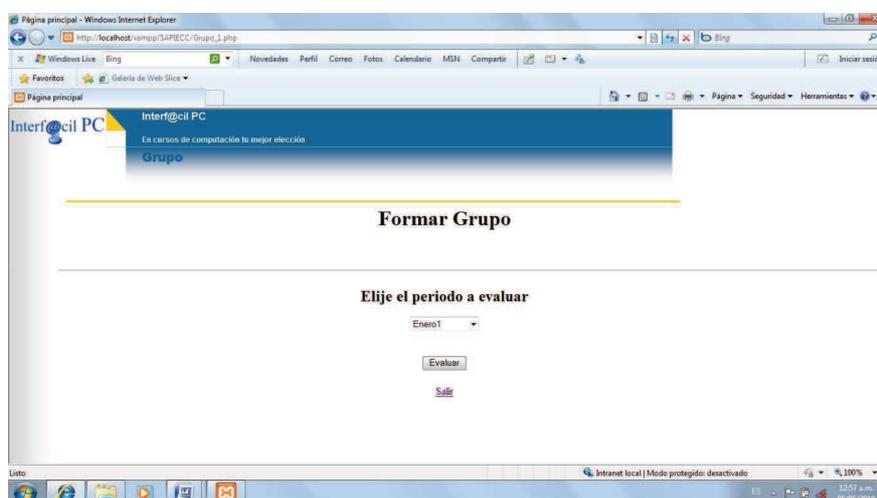
Existen tres opciones a considerar en esta pantalla. Para elegir cada una de ellas solo es necesario dar clic sobre el hipervínculo correspondiente:

OPCIÓN	RESULTADO
Formar Grupo	Despliegue la pantalla donde el usuario puede formar un grupo.
Consultas	Despliega la pantalla donde el usuario puede visualizar las consultas preprogramadas de SAPIECC
Ajustar Horas	Despliega el módulo destinado al ajuste de horas.

FORMAR GRUPO

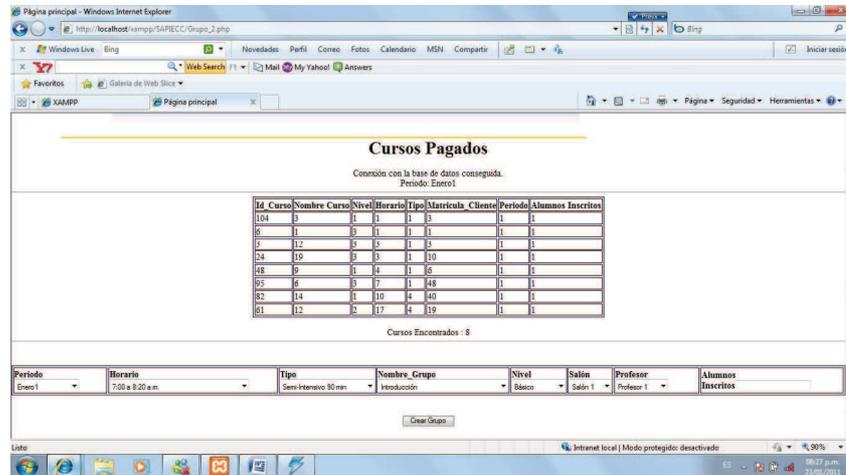
- *Interfaz*

El modulo que permite al usuario del sistema formar los grupos de cada periodo tiene la apariencia siguiente:



- *Estructuras de Datos Locales*
Este modulo opera directamente sobre la tabla grupo, sin embargo para su correcto funcionamiento toma datos de otras tablas como Cursos y clientes. Para mayor detalle se puede consultar el diccionario de datos.
- *Condiciones Límite*
Los límites de este modulo están definidos por la existencia de datos dentro de la tabla Grupo y en la tabla curso, ya que en caso de que estos no existan aparecerán los respectivos casos de error.
- *Caminos Independientes*
Una vez insertado el periodo a evaluar aparecerá una pantalla en la que se indicara el total de cursos pagados durante el periodo así como el número de alumnos inscritos en cada grupo. Si este es mayor o igual a tres o bien es de tipo cuatro

(Intensivo) el usuario deberá confirmar los datos en la sección inferior y dar clic en crear grupo. Esto almacenara los datos respectivos en la tabla grupo.

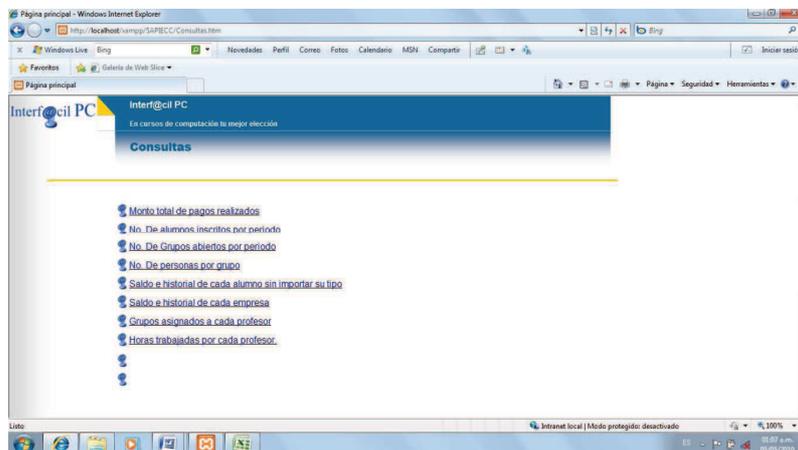


Este modulo presenta además las opciones para realizar un análisis nuevo y para regresar a la pantalla principal.

CONSULTAS

- *Interfaz*

El modulo que nos permite ejecutar las consultas programadas en SAPIECC tiene la apariencia siguiente:



- *Estructuras de Datos Locales*

Este modulo opera sobre todas las tablas y consultas de SAPIECC, sin embargo no realiza ninguna modificación sobre ellas, sino solo muestra su información en pantalla.

- *Condiciones Límite*

Debido a que este modulo no interactúa con datos no presenta condiciones en el límite como tal, sin embargo la información que se despliega en pantalla estará definida por los datos contenidos en la base de datos del sistema.

- *Caminos Independientes*

Existen tres opciones a considerar en esta pantalla. Para elegir cada una de ellas solo es necesario dar clic sobre el hipervínculo correspondiente:

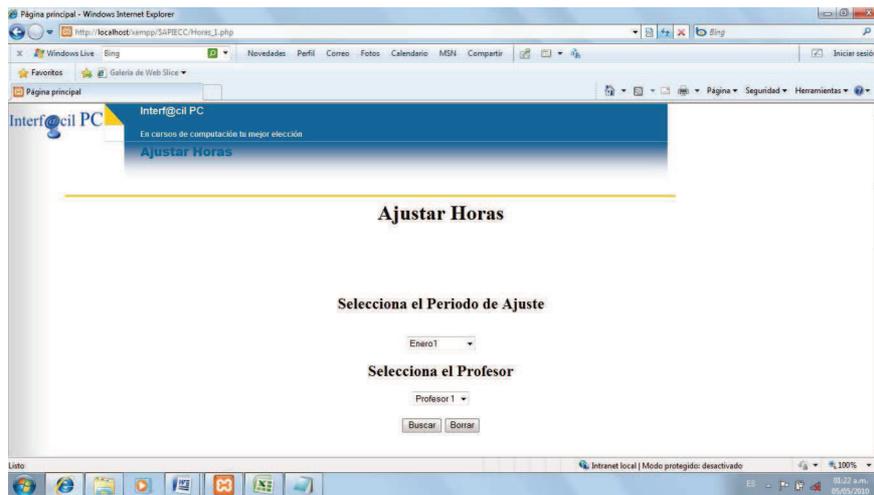
OPCION	RESULTADO
Monto Total de Pagos Realizados	Muestra el reporte de los pagos realizados durante el periodo elegido divididos en Notas de Pago y Facturas. Incluyendo información correspondiente a cada uno de estos movimientos
No. De Alumnos Inscritos por Periodo	Despliega el reporte de los alumnos inscritos durante el periodo elegido.
No. De Grupos Abiertos por Periodo	Despliega el reporte con toda la información referente a los grupos abiertos durante el periodo elegido por el usuario.
No. De Personas por Grupo	Despliega el reporte con toda la información referente a los grupos abiertos durante el periodo elegido por el usuario. Incluyendo el número de alumnos inscritos en cada grupo.
Historial Alumnos	Despliega el reporte correspondiente a los pagos realizados por el alumno correspondiente a la matrícula insertada por el usuario del sistema.
Historial Empresas	Despliega el reporte correspondiente a los pagos realizados por la empresa correspondiente a la matrícula insertada por el usuario del sistema.
Grupos Asignados a cada Profesor	Despliega el reporte correspondiente a los grupos asignados al profesor correspondiente al Id insertado por el usuario del sistema.
Horas Trabajadas por Profesor	Despliega el reporte correspondiente a los grupos asignados y las horas trabajadas por el profesor correspondiente al Id insertado por el usuario del sistema.

Todos estos movimientos tienen las opciones para realizar un nuevo análisis y para regresar a la pantalla principal.

AJUSTAR HORAS

- *Interfaz*

El modulo destinado al ajuste de horas trabajadas por cada profesor tiene la apariencia siguiente:



- *Estructuras de Datos Locales*

Este modulo opera sobre la tabla Horas, sin embargo los datos contenidos ahí dependen de la información generada a través de otros módulos. Para mayor detalle se puede recurrir al diccionario de datos del sistema.

- *Condiciones Límite*

Los límites para el funcionamiento de este módulo están definidos por los profesores existentes, los grupos asignados y el periodo de trabajo, ya que no se pueden ajustar horas si todavía no se realiza la asignación de los cursos.

- *Caminos Independientes*

Dado que este modulo calcula de manera automática las horas trabajadas correspondientes al profesor y periodo elegido tenemos dos casos de prueba.

DATOS DE ENTRADA	RESULTADO DE LA PRUEBA										
<p><u>Caso 1: Horas trabajadas Correctas</u></p> <p><i>Datos de Entrada:</i> Periodo: 1 Profesor: 1 Horas Trabajadas Ajustadas: 15 ó ""</p>	<p>El registro consultado se mantiene igual en la tabla horas.</p> <table border="1"> <thead> <tr> <th>Id_Horas</th> <th>Id_Profesor</th> <th>Periodo</th> <th>Horas_Contabilizadas</th> <th>Horas_Ajustadas</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>1</td> <td>15</td> <td>15</td> </tr> </tbody> </table>	Id_Horas	Id_Profesor	Periodo	Horas_Contabilizadas	Horas_Ajustadas	1	1	1	15	15
Id_Horas	Id_Profesor	Periodo	Horas_Contabilizadas	Horas_Ajustadas							
1	1	1	15	15							
<p><u>Caso 2: Ajustar Horas</u></p> <p><i>Datos de Entrada:</i> Periodo: 1 Profesor: 2</p>	<p>El registro se modifica con los nuevos datos.</p> <table border="1"> <thead> <tr> <th>Id_Horas</th> <th>Id_Profesor</th> <th>Periodo</th> <th>Horas_Contabilizadas</th> <th>Horas_Ajustadas</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>2</td> <td>1</td> <td>15</td> <td>12</td> </tr> </tbody> </table>	Id_Horas	Id_Profesor	Periodo	Horas_Contabilizadas	Horas_Ajustadas	1	2	1	15	12
Id_Horas	Id_Profesor	Periodo	Horas_Contabilizadas	Horas_Ajustadas							
1	2	1	15	12							

Es importante mencionar que este modulo cuenta además con opciones para realizar nuevos ajustes y para regresar a la pantalla principal.

Con esto se finaliza las pruebas de unidad para cada uno de los módulos de SAPIECC y podemos concluir que los resultados obtenidos coinciden con los resultados esperados de acuerdo al planteamiento del problema y a su propuesta de solución.

7.1.2 Pruebas de Integración

Algunas personas piensan que si todos los módulos de un sistema de software funcionan bien por separado deberían seguir así al momento de trabajar juntos. Sin embargo, no siempre es así.

La prueba de integración es una técnica sistemática para construir la estructura del programa mientras que al mismo tiempo se llevan a cabo pruebas para detectar errores asociados con la interacción. El objetivo es tomar los módulos probados mediante la prueba de la unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño.

Partiendo del diseño arquitectónico de SAPIECC aplicaremos la prueba de integración descendente para verificar su correcto funcionamiento. Es decir debido comenzando por el modulo de control principal integraremos los módulos individuales moviéndose hacia abajo por la jerarquía de control y consideraremos la profundidad en la que se encuentra cada modulo.

Los pasos que seguiremos para el proceso de integración son los siguientes:

1. Se usa el módulo de control principal como controlador de la prueba, disponiendo de resguardos para todos los módulos directamente subordinados al módulo de control principal.
2. Dependiendo del enfoque de integración elegido (es decir, primero-en-profundidad o primero-en-anchura) se van sustituyendo uno a uno los resguardos subordinados por los módulos reales.
3. Se llevan a cabo pruebas cada vez que se integra un nuevo módulo.
4. Tras terminar cada conjunto de pruebas, se reemplaza otro resguardo con el módulo real.
5. Se hace la prueba de regresión para asegurarse de que no se han introducido errores nuevos.

El proceso continúa desde el paso 2 hasta que se haya construido la estructura del programa entero.

La estrategia que se seguirá para revisar el correcto funcionamiento de SAPIECC será la siguiente: Se partirá de la pantalla principal (Gestor de Interacción con el Usuario) para dar de alta una empresa (Alta Cliente), la cual dará de alta 2 alumnos. Cada uno de ellos tomara un curso distinto. Sin embargo uno de ellos realizara su pago de forma individual requiriendo una Nota de Pago, mientras que el otro se necesitara una factura. Ambos tendrán profesores distintos, y uno de ellos requerirá ajuste en sus horas debido a que faltó durante dos días. Finalmente se revisara el historial del los alumno y de la empresa así como de los grupos de los cuales formaron parte. Enseguida se muestran los resultados de las pruebas:

COMANDO A EJECUTAR	MÓDULO INICIAL	MÓDULO AGREGADO	DATOS DE ENTRADA	RESULTADO DE LA PRUEBA	RESULTADO DE LA PRUEBA DE REGRESION
Alta Empresa	Gestor de Interacción con el usuario (Pantalla Inicial)	Alta Cliente (Moral)	<p>Razón Social: NEXTRAONE Calle y No: Horacio 1429 Colonia: Polanco Delegación/Municipio: Miguel Hidalgo Código Postal: 11560 Teléfono: 5026 - 0023 E-mail: claudia.za@nextiraone.com RFC: AVZL720510HDFLNS05</p>	<p>En la tabla Cliente Moral se almacena un registro con la siguiente información: (43,'NEXTRAONE','Horacio 1429','Polanco','11560','50260023','claudia.za@nextiraone.com','AVZL720510HDFLNS05')</p>	<p>Al dar clic en cerrar el sistema nuevamente regresa al gestor de interacción con el usuario. Al dar clic en el controlador de datos del sistema, se regresa a la pantalla principal. El catálogo de Empresas muestra la información almacenada. Al realizar la búsqueda por ambos criterios se muestra la información correcta y completa de la empresa.</p>
Alta Cliente	Gestor de Interacción con el usuario (Pantalla Inicial)	Alta Cliente (Físico)	<p>Nombre: Oliver A. Paterno : García A. Materno : Torres Calle y Número: Tacuba 902 Colonia: San Joaquín / Municipio: Cuajimalpa Código Postal: 05678 Teléfono: 92146712 E-Mail: oliver@nextiraone.com RFC: x Matrícula Empresa: 43 Nombre: Jaime A. Paterno : Rivas A. Materno : López Calle y Número: Amores 56 Colonia: Del Valle Delegación / Municipio: Benito Juárez</p>	<p>En la tabla cliente físico se almacenan los registros con la siguiente información: (65,'Oliver','García','Torres','Tacuba 902','San Joaquín','Cuajimalpa','05678','92146712','oliver@nextiraone.com','x','43') (66,'Jaime','Rivas','López','Amores 56','Del Valle','Benito Juárez','03100','23456789','jaime_rl@hotmail.com','x','0')</p>	<p>Al dar clic en cerrar el sistema nuevamente regresa al gestor de interacción con el usuario. Al dar clic en el controlador de datos del sistema, se regresa a la pantalla principal sin problema. El catálogo de clientes físicos muestra la información almacenada. Al realizar la búsqueda por ambos criterios se muestra la información correcta y completa para ambos registros. Al introducir la clave del cliente moral no indico error comprobando que este existe y</p>

Ajustar Horas	Controlador de Consultas del Sistema	Ajustar Horas	Salón: Salón 1 Profesor: Profesor 1 No de Alumnos: 4 Periodo: Enero 1 Profesor: 3 Horas Trabajadas Ajustadas: 12	El registro correspondiente se modifica de: (9, '3', '1', '15', '15') A los siguientes valores: (9, '3', '1', '15', '12')	modificaciones todos los módulos derivados del Controlador de datos del sistema. Al revisar nuevamente el Controlador de consultas de datos del sistema, además el comando formar grupo y agregar profesor con todas sus opciones correspondientes estos no presentaron modificaciones en su comportamiento.
Historial Alumno	Controlador de Consultas del Sistema	Desplegar consultas (Historial Cliente)	Matricula_Cliente: 66	Despliega una pantalla con toda la información correspondiente al cliente 66, es decir muestra la información exacta que se introdujo al crear la nota de pago.	Al revisar el módulo de Clientes Físicos y Morales, estos no presentan modificación en su operación, así como tampoco los módulos de pagos y los relacionados al controlador de consultas del sistema
Historial Empresa	Controlador de Consultas del Sistema	Desplegar consultas (Historial Empresas)	Matricula_Cliente Moral: 43	Despliega una pantalla con toda la información correspondiente a la empresa 43, es decir muestra la información exacta que se introdujo al crear la factura, en la prueba destinada a esto.	Al revisar el módulo de Clientes Físicos y Morales, estos no presentan modificación en su operación, así como tampoco los módulos de pagos y los relacionados al controlador de consultas del sistema.
Alumnos Inscritos	Controlador de Consultas del Sistema	Desplegar consultas (Alumnos Inscritos)	Periodo: 1	Despliega la pantalla correspondiente a todos los alumnos inscritos durante ese periodo.	Revisando todos los módulos anteriores no se encontró modificación alguna.
Pagos Realizados	Controlador de Consultas del Sistema	Desplegar consultas (Pagos Realizados)	Periodo: 1	Despliega la pantalla correspondiente a los pagos realizados durante el periodo elegido, indicando el total para cada tipo de pago y el total general obtenido en el periodo.	Al volver a ejecutar cada una de las funciones de los módulos anteriores, en este caso así como en el resto de las consultas el sistema no presenta modificación alguna en su comportamiento.
Consultar Ayuda	Gestor Interacción con el usuario	Ayuda	-	Despliega la ayuda del sistema.	Ninguno de los módulos anteriores presenta modificaciones.

De acuerdo a estos datos podemos decir que los resultados de la prueba de integración para el sistema SAPIECC son satisfactorios, ya que al ir integrando cada uno de los módulos individuales que integran el sistema este continuo con su flujo de trabajo normal, devolviendo los resultados esperados para cada caso que se puso a prueba en esta sección.

7.1.3 Pruebas de Validación

Tras la culminación de la prueba de integración, el software está completamente ensamblado como un paquete, se han encontrado y corregido los errores de interfaz y puede comenzar una serie final de pruebas del software: la prueba de validación. La validación puede definirse de muchas formas, pero una simple definición es que la validación se consigue cuando el software funciona de acuerdo con la Especificación de Requisitos del Software definida por el cliente en la etapa de análisis. La especificación contiene una sección denominada- «Criterios de validación ». La información contenida en esa sección forma la base del enfoque de la prueba de validación.

En seguida se muestra los resultados de comparar la Especificación de Requisitos de SAPIECC con respecto al sistema final obtenido.

Resumen y panorama del producto

ESPECIFICACIÓN DE REQUISITOS DEL SISTEMA SAPIECC	RESULTADOS REALES
Implementar un sistema integral que no necesite de un conjunto de aplicaciones externas para su manejo y control.	SAPIECC es un sistema informático que no necesita auxilio de programas de aplicación adicionales, ya que toda la información que genera y manipula se encuentra dentro de su propia base de datos. Mientras que el usuario interactúa con ella a través de su propia interfaz.
Reducir tiempos de respuesta para altas de diferentes movimientos.	Debido a que todos los movimientos del sistema se realizan a través de interfaces propias, el tiempo requerido para realizar una operación es menor y más preciso ya que de forma automática se genera la actualización en todos los sitios que sean necesarios conservando la integridad de los datos.
Generación automática de reportes.	Debido a que inicialmente se plantearon los datos que se consultan constantemente. Todas las consultas definidas en la especificación de requisitos del sistema, estas consultas se programaron para que automáticamente se puedan observar e imprimir en forma de reportes.
Reducción de errores.	SAPIECC almacena toda su información en una sola base de datos. Esto implica que la

	<p>redundancia de información se redujo con respecto a la forma anterior. Además al respetar la integridad de los datos se disminuyeron considerablemente los errores.</p>
--	--

Ambientes de desarrollo, operación y mantenimiento

ESPECIFICACIÓN DE REQUISITOS DEL SISTEMA SAPIECC	RESULTADOS REALES
<p>El sistema (SAPIECC), será desarrollado por una sola persona quien seguirá una metodología en espiral para llevar a cabo el proceso de producción del software y quien hará uso principalmente de dos herramientas de software: un lenguaje de programación para páginas dinámicas (PHP) y un sistema de gestión para bases de datos (MySQL). Se pretende que el ambiente de desarrollo y operación sea el sistema operativo Windows pero la posibilidad de que el ambiente de operación sea algún otro se encuentra abierta.</p>	<p>SAPIECC fue desarrollado utilizando el lenguaje de programación PHP enlazado con una base de datos MySQL. Y actualmente el sistema opera correctamente bajo el sistema operativo Windows 7.</p>
<p>En un principio se pretende que el sistema se encuentre instalado en una computadora personal del departamento, por lo cual el sistema no tiene características y restricciones de aplicaciones compartidas o para redes, pero se pueden implementar posteriormente.</p>	<p>Para la realización de estas pruebas el sistema se instalo en una computadora personal que se encuentra dentro de la red y que cuenta con servicio de internet y servicios compartidos como impresora y escáner. Esto no afecto el comportamiento normal del sistema.</p>
<p>El mantenimiento preventivo del sistema se llevará a cabo dos veces al año, mientras que el correctivo se implementará cada vez que el sistema lo requiera.</p>	<p>Se tiene acordado que el sistema recibirá mantenimiento preventivo 2 veces al año. Este consistirá en la revisión de la base y de los módulos en general. Además de su actualización en caso de ser necesario.</p>

Interfaces externas y flujo de datos

ESPECIFICACIÓN DE REQUISITOS DEL SISTEMA SAPIECC	RESULTADOS REALES
<p>El sistema para la Administración de Pagos Internos y Externos de un Centro de Capacitación Computacional (SAPIECC) tendrá dos formas de salida de la información que maneja y este será a través de resultados desplegados en la pantalla o mediante reportes impresos.</p>	<p>Todo movimiento generado en SAPIECC, se puede visualizar de manera automática en pantalla. Además todas las consultas y catálogos se pueden imprimir.</p>

En el caso de los reportes impresos estos siempre serán en tamaño carta y con un encabezado que contenga el nombre del centro, su dirección, y el título del reporte.	Los parámetros de impresión de todas las consultas y catálogos de SAPIECC se encuentran configurados en tamaño carta y el nombre del reporte.
El flujo de datos que maneja el sistema se puede observar a detalle en el diagrama de flujo de datos.	El funcionamiento general del sistema cumple con el orden definido en el diagrama de flujo del sistema, además la secuencia seguida para las pruebas de integración se realizaron tomando este como referencia.
El comportamiento que tendrá el sistema se encuentra desglosado en el diagrama de transiciones del sistema.	La operación general de SAPIECC cumple con los estados y entradas definidas en el diagrama de transiciones del sistema.
La información detallada de los datos que se utilizaran en el sistema y que se muestran en el diagrama entidad-relación se encuentra almacenada en el diccionario de datos.	Todos los datos definidos en el diagrama entidad relación y en los esquemas de la base de datos se encuentran descritos a detalle en el diccionario de datos del sistema.

Requisitos funcionales

ESPECIFICACIÓN DE REQUISITOS DEL SISTEMA SAPIECC	RESULTADOS REALES
El sistema contará con un módulo de alta para los alumnos, el cual será capaz de distinguir entre clientes físicos y morales y será el mecanismo de entrada de todos los datos que corresponden a cada uno de estos.	Cuando el usuario necesita generar el movimiento de alta de clientes, el usuario debe definir si se trata de un cliente físico ó un cliente moral. Esto definirá el formulario de datos que aparecerá en la pantalla de alta.
El sistema contará también con un módulo de cursos, el cual una vez dado de alta un cliente permitirá que se le asigne un curso de acuerdo a sus necesidades.	El sistema cuenta con un modulo para asignar cursos. El único requisito obligatorio es que este se encuentre dado de alta en el sistema.
Para llevar un control de los pagos del centro se diseñará un módulo de pagos el cual permitirá emitir facturas o notas de pago según sean las necesidades del cliente. Este módulo además de llevar un control de dichos documentos permitirá emitir los comprobantes necesarios (facturas o notas de pago).	Una vez asignado un curso SAPIECC permite emitir el comprobante de pago necesario. Nota de pago si es un cliente físico y facturas si es un cliente moral. Estos comprobantes se pueden imprimir y siempre se almacenan en la base de datos del sistema.
Se implementará un módulo de profesores, el cual tendrá como principal función asignar un maestro a cada grupo que se abra en un determinado periodo, horario y asignatura. Así como llevar un control de las horas	SAPIECC cuenta con un módulo para el alta de profesores. Cada grupo que se forme debe tener asignado un solo profesor. Al cuál se le contabilizan las horas trabajadas de acuerdo al tipo de curso. En caso de requerir un ajuste se

trabajadas por cada profesor.	cuenta con un módulo especial para ello.
<p>El sistema contará con un módulo de consultas, el cual permitirá observar en pantalla consultas tales como:</p> <ul style="list-style-type: none"> • Monto total de pagos realizados • No. De alumnos inscritos por periodo • No. De Grupos abiertos por periodo. • No. De personas por grupo • Saldo e historial de cada alumno sin importar su tipo. • Saldo e historial de cada empresa • Grupos asignados a cada profesor • Horas trabajadas por cada profesor. • Total de pagos de alumnos regulares • Total facturado a empresas 	<p>El sistema permite visualizar en pantalla las siguientes consultas:</p> <ul style="list-style-type: none"> • Monto Total de Pagos Realizados • No. De Alumnos Inscritos por Periodo • No. De Grupos Abiertos por Periodo • No. De Personas por Grupo • Historial Alumnos • Historial Empresas • Grupos Asignados a cada Profesor • Horas Trabajadas por Profesor
Finalmente el sistema (SAPIECC) contará con un modulo de Ayuda y Ajustes, el cual tendrá las capacidad de desplegar un tutorial para brindar ayuda o bien desplegar un módulo de ajustes para modificar las horas trabajadas por cada profesor.	El sistema cuenta con un módulo para ajustar las horas trabajadas por cada profesor además de un tutorial de ayuda que se despliega en pantalla.

Subconjuntos iniciales y prioridades de instrumentación

ESPECIFICACIÓN DE REQUISITOS DEL SISTEMA SAPIECC	RESULTADOS REALES
La jerarquía que se deberá seguir para el desarrollo e implementación de los módulos es la siguiente: alta, cursos, pagos, profesores, consultas y finalmente ayuda y ajustes.	La secuencia que se siguió para la implementación de los módulos fue la siguiente: alta clientes, cursos, pagos, alta profesores, consultas y finalmente ayuda y ajustes. Esto debido a la forma en que se relacionan los datos.

Modificaciones y mejoras previsibles

ESPECIFICACIÓN DE REQUISITOS DEL SISTEMA SAPIECC	RESULTADOS REALES
Se espera que en un futuro el sistema (SAPIECC) se adapte a equipos en red para que se pueda manipular desde otros equipos o bien a distancia.	Aunque el sistema radica en una computadora local, la base de datos se encuentra en el servidor local de ese equipo. Sin embargo una vez aprobado se pretende colocar en el servidor de aplicaciones del centro.
SAPIECC permitirá la adición de módulos sin afectar su comportamiento.	Debido a la forma en que se desarrollo el sistema se pueden adicionar más módulos, consultas y movimientos sin alterar su estructura interna

Criterios de aceptación

ESPECIFICACIÓN DE REQUISITOS DEL SISTEMA SAPIECC	RESULTADOS REALES
El sistema (SAPIECC) deberá satisfacer todos los requisitos escrito en este documento y en todos aquellos diagramas auxiliares a éste.	El sistema cumple con todos los requerimientos definidos en el documento de especificaciones técnicas, además de los descritos en los diferentes diagramas de la fase de análisis.
El sistema deberá ir acompañado de un manual de usuario que servirá de apoyo para los usuarios del sistema.	El sistema cuenta con un manual de usuario o tutorial. El cual se encuentra disponible en el módulo de ayuda y puede ser impreso si el usuario lo considera necesario.
El sistema deberá cumplir con las especificaciones del diseño arquitectónico así como con el plan de pruebas que se especifican en esta fase del proyecto.	El sistema cumple con las especificaciones de diseño de datos, diseño arquitectónico y detallado definidos en la fase de diseño. Además las pruebas se realizaron de acuerdo a lo establecido en el plan de pruebas del sistema.
Además de las pruebas de código SAPIECC debe satisfacer las pruebas de unidad, integración, validación y de sistema.	De acuerdo con los resultados obtenidos en este capítulo el sistema cumple con las pruebas de unidad, integración y validación. Mientras que actualmente se encuentra bajo pruebas de sistema.

Finalmente, considerando que es imposible que un desarrollador de software pueda prever cómo utilizará el usuario realmente el programa. Se llevaran a cabo pruebas beta para monitorear el uso de SAPIECC. La prueba beta, se lleva a cabo por los usuarios finales del software en los lugares de trabajo de los clientes. A diferencia de la prueba alfa, el desarrollador no está presente normalmente. El cliente registrara todos los problemas (reales o imaginarios) que encuentra durante la prueba beta e informara a intervalos regulares al desarrollador. Como resultado de los problemas informados durante la prueba beta, el desarrollador del software llevara a cabo modificaciones y así preparara una versión del producto de software para toda la clase de clientes.

7.1.4 Pruebas de Sistema

La prueba del sistema, está constituida por una serie de pruebas diferentes cuyo propósito primordial es ejercitar profundamente el sistema basado en computadora. Aunque cada prueba tiene un propósito diferente, todas trabajan para verificar que se han integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas.

Prueba de recuperación

La prueba de recuperación es una prueba del sistema que fuerza el fallo del software de muchas formas y verifica que la recuperación se lleva a cabo apropiadamente. Si la recuperación es automática (llevada a cabo por el propio sistema) hay que evaluar la corrección de la inicialización, de los mecanismos de recuperación del estado del sistema, de la recuperación de datos y del proceso de re arranque. Si la recuperación requiere la intervención humana, hay que evaluar los tiempos, medios de reparación (TMR) para determinar si están dentro de unos límites aceptables.

Para revisar la recuperación de SAPIECC, se corrompió intencionalmente la base de datos de del sistema, modificando algunos datos. Esto provoco el fallo inmediato en los resultados del sistema. Lo cual se soluciono mediante el la sustitución del respaldo de la base de datos. Por lo que es necesario que se programen respaldos periódicos para evitar perdida de información.

Prueba de resistencia (Stress)

La prueba de resistencia ejecuta un sistema de forma que demande recursos en cantidad, frecuencia o volúmenes anormales .Esencialmente, el responsable de la prueba intenta romper el programa. Una variante de la prueba de resistencia es una técnica denominada prueba de sensibilidad, la cual intenta descubrir combinaciones de datos dentro de una clase de entrada valida que pueda producir inestabilidad o un proceso incorrecto.

Para el caso de SAPIECC se realizaron pruebas de resistencia, dejando el sistema trabajando durante doce horas seguidas. En las cuales se realizaron de manera constante consultas y movimientos de diferente tipo. Esto con el objetivo de comprobar que el sistema trabaja sin contratiempos y demoras después de ese periodo de trabajo.

7.2 Administración de la Configuración

La administración de la configuración está relacionada con el rastreo y control de los productos de trabajo que constituyen un producto de software. Cuando un producto de software pasa la revisión de las metas iniciales, se pone bajo el control de configuración y cualesquiera cambios subsecuentes al producto pueden efectuarse sólo mediante un acuerdo formal entre el cliente y el desarrollador. Se dice que un producto de trabajo puesto bajo el control de configuración se estandariza.

Durante el mantenimiento del software, se requiere un plan de administración de la configuración y herramientas para dicha administración de manera que se pueda seguir la pista y controlar las distintas versiones de los productos de trabajo que constituyen un producto de software.

Bases de datos para la administración de la configuración.

Las herramientas de software para apoyar la administración de la configuración incluyen las bases de datos para esta administración y los sistemas de biblioteca para el control de las versiones. Una base de datos para el control de la configuración puede proporcionar información concerniente a la estructura del producto, el número de revisión en curso, el estado en curso y la historia de las solicitudes de cambios para cada versión del producto.

Versiones del producto: 1

Versiones de productos distribuidas en las instalaciones del centro: Editor de PHP 5 y MySQL Server 5.

Documentos disponibles para cada versión del sistema: Manual de Usuario e Instalación.

Configuración de hardware requerida: Ninguna.

Sin embargo conforme SAPIECC vaya transformándose y se realicen diferentes versiones esta base de datos se debe ir actualizando y debe ser capaz de responder a preguntas como: ¿Cuántas versiones del sistema existe?, ¿cómo difieren las versiones?, ¿qué versiones de cuáles productos están distribuidas en qué instalaciones?, ¿qué documentos están disponibles para cada versión de cada producto?, ¿cuál es la historia de la revisión de cada componente de cada versión de cada producto?, ¿cuándo estará disponible la siguiente revisión de un componente dado de una determinada versión de un producto?, ¿qué configuración de hardware se requiere para operar una versión específica de un producto?, ¿cuáles revisiones constituyen una versión específica de un producto?, ¿qué versiones se afectan con una revisión de un componente dado?, ¿cuáles versiones se afectan con un informe de error específico?, ¿qué errores se corrigieron en una revisión específica?, ¿cuáles son las fallas del producto que han sido informadas?, ¿qué componentes son funcionalmente similares en cuáles versiones?, ¿algún usuario emplea todavía una versión antigua dada?, ¿cuál es la configuración de una versión dada en una determinada fecha?

Bibliotecas para el control de versiones

Una biblioteca para el control de versiones puede ser parte de una base de datos para la administración de la configuración, o se puede usar como una herramienta independiente. Un sistema de biblioteca para el control de versión no es una base de datos y no contiene la información requerida para responder preguntas como las antes formuladas. Entre las entidades en una biblioteca para el control de versiones pueden encontrarse el código fuente, el código objeto localizable, los comandos para el control de trabajos, los archivos de datos y los documentos de apoyo. Cada entidad en esta biblioteca debe tener una tarjeta de identidad que muestre el número de versión, la fecha, la hora y la identidad del programador. Las operaciones que se realizarán en una biblioteca son creación de la biblioteca, adición y borrado de componentes, preparación de copias de respaldo, edición de archivos, listado de estadísticas de resumen, así como compilación y ensamblado de versiones específicas del sistema.

7.3 Métricas Técnicas del Software

Las métricas Técnicas del software se pueden definir como medidas de atributos internos del producto las cuales proporcionan al desarrollador de software una indicación en tiempo real de la eficacia del análisis, del diseño y de la estructura del código, la efectividad de los casos de prueba, y la calidad global del software a construir. Se han propuesto cientos de métricas para el software pero no todas proporcionan un soporte práctico para el desarrollador de software. Algunas demandan mediciones que son demasiado complejas, mientras que otras violan las nociones básicas intuitivas de lo que realmente es el software de calidad.

En seguida se muestran los resultados de ciertas métricas que se aplicaron al sistema SAPIECC durante distintas etapas del proceso de software.

7.3.1 Métricas del Modelo de Análisis

Durante la etapa de análisis definimos cuales serían las características y funciones que debería cumplir SAPIECC de acuerdo con las especificaciones del cliente, esto quedo plasmado en el documento de Especificaciones Técnicas del Sistema, el cual posteriormente fue sometido a pruebas para verificar su cumplimiento.

Ahora que se ha comprobado que se cumplió con lo establecido, es importante demostrar que la creación del modelo de análisis cuenta con calidad, para lo cual haremos uso de métricas del modelo de análisis. Las métricas que utilizaremos para examinar el modelo de análisis de SAPIECC serán: la Métrica de Punto de Función y la Métrica de la calidad de la especificación, ambas con el objetivo de predecir el tamaño del sistema resultante, pero a través de métodos distintos. En seguida describen a detalle los resultados obtenidos.

Métrica de Punto de Función

La métrica de punto de función se usa como medio para predecir el tamaño del sistema que se va a obtener de un modelo de análisis. Para lo cual se utiliza una medida de la funcionalidad entregada por la aplicación como un valor de normalización.

El punto de función se calcula completando la tabla inferior. En la cual se determinan cinco características de dominios de información y se proporcionan las cuentas en la posición apropiada de la tabla.

Para obtener las cuentas necesarias haremos uso del diagrama de flujo del sistema, con lo cual obtenemos los siguientes resultados:

- Número de entradas del usuario: 2 (Movimiento ó Consulta)
- Número de salidas del usuario: 2 (Pantalla e Impresora)
- Número de consultas del usuario: 3 (Grupos, Desplegar Consulta, Ajustar Horas)
- Número de Archivos: 1 (Base de Datos del Sistema)

- Número de Interfaces Externas: 7

Enseguida procedemos al cálculo del factor de complejidad simple, mediante el llenado de la tabla:

<i>Parámetros de medición</i>	<i>Cuenta</i>	<i>Simple</i>	<i>Medio</i>	<i>Complejo</i>	
Número de entradas de usuario	2	• 3	4	6	=6
Número de salidas de usuario	2	• 4	5	7	=8
Número de peticiones de usuario	3	• 3	4	6	=9
Número de archivos	1	• 7	10	15	=7
Numero de interfaces externas	7	• 5	7	10	=35
Cuenta Total					65

Para calcular el punto de función, se utiliza la relación siguiente:

$$PF = \text{cuenta_total} * [0.65 + 0.01 * \Sigma(Fi)]$$

En donde cuenta total es la suma de todas las entradas obtenidas en la tabla anterior.

Fi (i=1 a 14) son valores de la complejidad , según las respuestas a las siguientes preguntas: Cada una de las preguntas es respondida usando una escala con rangos desde 0 (no importante o aplicable) hasta 5(absolutamente esencial). Los valores constantes de la ecuación y los factores de peso que se aplican a las cuentas de los dominios de información se determinan empíricamente.

- 1.- ¿Requiere el sistema copias de seguridad y de recuperación fiables? 5
- 2.- ¿Se requiere comunicación de datos? 5
- 3.- ¿Existen funciones de procesamiento distribuido? 3
- 4.- ¿Es crítico el rendimiento? 0
- 5.- ¿Se ejecutará el sistema en un entorno operativo existente y fuertemente utilizado? 5
- 6.- ¿Requiere el sistema entrada de datos interactiva? 5
- 7.- ¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones? 0
- 8.- ¿Se actualizan los archivos maestros de forma interactiva? 5
- 9.- ¿Son complejas las entradas, las salidas, los archivos o las peticiones? 0
- 10.- ¿Es complejo el procesamiento interno? 0
- 11.- ¿Se ha diseñado el código para ser reutilizable? 5
- 12.- ¿Están incluidas en el diseño la conversión y la instalación? 5
- 13.- ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones? 5

14.- ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario? 5

Por lo tanto para SAPIECC, considerando la cuenta de un sistema simple y las respuestas anteriores tenemos:

$$PF = \text{cuenta_total} * [0.65 + 0.01 * \Sigma (Fi)]$$

$$PF = 65 * [0.65 + 0.01 * (5040)]$$

$$PF = 3328.25$$

De acuerdo a lo anterior podemos decir que un PF supone 3329 líneas de código y que en un esfuerzo de un mes-persona se producen 30PF. Esto permite al desarrollador del sistema prever que la duración del proyecto si es desarrollado por una sola persona tendrá una duración aproximada de 11 meses.

Métrica de la calidad de la especificación

Para valorar la calidad del modelo de análisis y la correspondiente especificación de requisitos: Obtendremos las siguientes métricas para SAPIECC: especificidad, completión y grado de validación de los requisitos.

Para determinar la especificidad (ausencia de ambigüedad) de los requisitos, utilizaremos una métrica basada en la consistencia de la interpretación de los revisores para cada requisito:

$$Q_1 = n_{ui} / n_r$$

Donde n_{ui} es el número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas y n_r es el número total de requisitos en una especificación.

En el caso concreto de SAPIECC la especificación del sistema consta de 25 requisitos de los cuales según el revisor tuvieron resultados idénticos a los requeridos en la especificación 22 requisitos por lo tanto tenemos que su especificidad es :

$$Q_1 = 22 / 25$$

$$Q_1 = 0.88$$

Este valor es cercano a 1 que indicaría una especificidad completa. Por lo que podemos decir que SAPIECC cuenta con un grado aceptable de especificidad.

Mientras que la completión de los requisitos funcionales pueden determinarse calculando la relación:

$$Q_2 = u_n / (n_i * n_s)$$

Donde u_n es el número de requisitos únicos de función, n_i es el número de entradas (estímulos) definidos o implicados por la especificación y n_s es el número de estados especificados. La relación Q_2 mide el porcentaje de funciones necesarias que se han especificado para un sistema.

Para el caso de SAPIECC si nos remitimos al diagrama de transiciones del sistema podemos observar que tenemos 2 entradas y 14 estados especificados, mientras que la especificación de requisitos del sistema cuenta con 6 requisitos únicos de función. Por tanto la compleción de requisitos está dada por:

$$Q_2 = 6 / (2 * 14)$$

$$Q_2 = 0.2142$$

Finalmente con el objetivo de presentar una métrica global completa debemos considerar el grado de validación de los requisitos:

$$Q_3 = n_c / (n_c + n_{nv})$$

Donde n_c es el número de requisitos que se han validado como correctos y n_{nv} es el número de requisitos que no se han validado todavía.

SAPIECC cuenta con 24 requisitos validados y 1 requisito no validados hasta el momento (Consultas completas). Por tanto tenemos que

$$Q_3 = 24 / (24 + 1)$$

$$Q_3 = 0.96$$

Este valor cercano a la unidad indica que casi todos los requisitos del documento de especificación del sistema fueron validados satisfactoriamente.

7.3.2 Métricas del Modelo de Diseño

Es inconcebible que el diseño de un nuevo avión, un nuevo chip de computadora o un nuevo edificio de oficinas se realizara sin definir las medidas del diseño, sin determinar las métricas para varios aspectos de la calidad del diseño y usarlas para guiar su evolución. Y sin embargo, el diseño de sistemas complejos basados en computadora a menudo consigue proseguir sin virtualmente ninguna medición.

Métricas del Diseño Arquitectónico

Las métricas de diseño de alto nivel se concentran en las características de la arquitectura del programa con especial énfasis en la estructura arquitectónica y en la eficiencia de los módulos.

Aplicaremos tres medidas de la complejidad de diseño de software: complejidad estructural, complejidad de datos y complejidad del sistema.

La *complejidad estructural*, $S(i)$, de un módulo i se define de la siguiente manera:

$$S(i) = f_{out}^2(i)$$

Donde $f_{out}(i)$ es la expansión del módulo i .

Si aplicamos esto sobre el módulo controlador de datos de sistema y el módulo controlador de consultas de SAPIECC. Los cuales según la arquitectura del sistema se expanden en 3 y 4 módulos respectivamente tenemos los siguientes datos:

<u>Controlador de Datos del Sistema</u>	<u>Controlador de Consultas del sistema</u>
$S(i) = f_{out}^2(i)$ $S(i) = (4)^2$ $S(i) = 16$	$S(i) = f_{out}^2(i)$ $S(i) = (3)^2$ $S(i) = 9$
Lo cual indica que el modulo controlador de Datos del Sistema podría expandirse hasta 16 módulos sin afectar su estructura y funcionamiento actual.	Lo cual indica que el modulo controlador de Consultas del Sistema podría expandirse hasta 9 módulos sin afectar su estructura y funcionamiento actual.

La complejidad de datos, $D(i)$, proporciona una indicación de la complejidad en la interfaz interna de un módulo i y se define como:

$$D(i) = v(i) / [f_{out}(i) + 1]$$

donde $v(i)$ es el número de variables de entrada y salida que entran y salen del módulo i .

Si aplicamos esto sobre el módulo controlador de datos de sistema y el módulo controlador de consultas de SAPIECC y definiendo las variables de entrada y salida de la arquitectura propia del sistema tenemos los siguientes datos:

<u>Controlador de Datos del Sistema</u>	<u>Controlador de Consultas del sistema</u>
$D(i) = v(i) / [f_{out}(i) + 1]$ $D(i) = 2 / [4 + 1]$ $D(i) = 0.4$	$D(i) = v(i) / [f_{out}(i) + 1]$ $D(i) = 2 / [3 + 1]$ $D(i) = 0.5$
Este valor al ser menor a 1 indica que los datos tienen una complejidad aceptable. Es decir los datos están modelados de tal forma que no presentan modificaciones considerables al ser operados en el módulo.	Este valor nos indica que los datos de entrada no sufren modificaciones complejas al pasar a través del módulo controlador de consultas del sistema.

A medida que crecen los valores de complejidad, la complejidad arquitectónica o global del sistema también aumenta. Esto lleva a una mayor probabilidad de que aumente el esfuerzo necesario para la integración y las pruebas.

Existen varias métricas de morfología simples (por ejemplo, forma) que permiten comparar diferentes arquitecturas de programa mediante un conjunto de dimensiones directas. Si nos referimos al diseño arquitectónico de SAPIECC podemos definir las siguientes métricas:

Se pueden definir las siguientes métricas:

- tamaño = $n + a$, donde n es el número de nodos (módulos) y a es el número de arcos (líneas de control). Para la arquitectura de SAPIECC : tamaño = $13 + 13 = 26$
- profundidad= el camino más largo desde el nodo raíz (parte más alta) a un nodo hoja. Para la arquitectura de SAPIECC, la profundidad = 4.
- anchura = máximo número de nodos de cualquier nivel de la arquitectura. Para la arquitectura de SAPIECC, la anchura = 7.
- Relación arco-a-nodo, $r = a / n$, que mide la densidad de conectividad de la arquitectura y puede proporcionar una sencilla indicación del acoplamiento de la arquitectura. Para el caso de SAPIECC $r = 13 / 13 = 1$.

7.3.3 Métricas del Código Fuente

La ciencia del software asigna leyes cuantitativas al desarrollo del software de computadora, usando un conjunto de medidas primitivas que pueden obtenerse una vez que se ha generado o estimado el código después de completar el diseño. Estas medidas se listan a continuación.

n_1 : el número de operadores diferentes que aparecen en el programa

n_2 : el número de operandos diferentes que aparecen en el programa

N_1 : el número total de veces que aparece el operador

N_2 : el número total de veces que aparece el operando

Los operadores incluyen todas las construcciones del flujo de control, condiciones y operaciones matemáticas. Los operandos agrupan todas las variables y constantes del programa.

Esta métrica usa las medidas primitivas para desarrollar expresiones para la longitud global del programa; volumen mínimo potencial para un algoritmo; el volumen real (número de bits requeridos para especificar un programa); el nivel del programa (una medida de la complejidad del software); nivel del lenguaje (una constante para un lenguaje dado); y otras características tales como esfuerzo de desarrollo, tiempo de desarrollo e incluso el número esperado de fallos en el software.

La longitud N se puede estimar como:

$$N = n_1 \log_2 n_1 + n_2 \log_2 n_2$$

Para el caso de SAPIECC, si nos referimos a la sección de instrumentación del módulo ajuste de horas podemos observar que aparece el operador de multiplicación de se aplica sobre las filas contabilizadas de la consulta. Este operador aparece 2 veces en el código con lo cual tenemos lo siguiente:

$$N = n_1 \log_2 n_1 + n_2 \log_2 n_2$$

$$N = \log_2 1 + \log_2 2$$

$$N=1$$

y el volumen de programa se puede definir como:

$$v = N \log_2 (n_1 + n_2)$$

Es decir

$$v = N \log_2 (n_1 + n_2)$$

$$v = 1 \log_2 (1 + 1) = 1$$

Por lo tanto para realizar el ajuste de horas necesitamos una instrucción que realice la multiplicación de los grupos abiertos durante un periodo a un profesor. El tamaño dependerá del tipo de datos del resultado que como en este caso es doble requiere un byte para su almacenamiento en el sistema.

Se debería tener en cuenta que V variará con el lenguaje de programación y representa el volumen de información (en bits) necesarios para especificar un programa.

7.3.4 Métricas para las Pruebas

Generalmente las métricas para las pruebas están orientadas en el proceso de las pruebas y no en las características técnicas de las mismas. En general, los responsables de las pruebas deben fiarse de las métricas de análisis, diseño y código para que se les guíe en el diseño y ejecución de los casos de prueba.

Al final del capítulo anterior se definió cual era el plan del pruebas a realizar sobre el sistema SAPIECC. Mientras que al inicio de este capítulo se reportaron los resultados de dichas pruebas indicando en cada sección si el funcionamiento correspondía con el esperado en las fases de análisis y diseño e indicando la calidad de la especificación del sistema con respecto al sistema final.

7.3.5 Métricas del Mantenimiento

El estándar IEEE 982.1-1988 sugiere un índice de madurez del software (IMS) que proporciona una indicación de la estabilidad de un producto software, basada en los cambios que ocurren con cada versión del producto. Para su obtención se determina la siguiente información:

M_T = número de módulos en la versión actual

F_c = número de módulos en la versión actual que se han cambiado

F_a = número de módulos en la versión actual que se han añadido

F_d = número de módulos de la versión anterior que se han borrado de la versión actual.

El índice de madurez del software se calcula de la

$$IMS = [M_T - (F_a + F_c + F_d)] / M_T$$

A medida que el IMS se aproxima a 1 el producto se empieza a estabilizar. EL IMS puede emplearse también como métrica para la planificación de las actividades de mantenimiento del software. El tiempo medio para producir una versión de un producto software puede correlacionarse con el IMS desarrollándose modelos empíricos para el mantenimiento.

Para realizar una prueba con SAPIECC se propone añadir un módulo para el control de asistencia de los clientes. De acuerdo, a lo anterior para calcular su índice de madurez tenemos lo siguiente:

$$M_T = 10$$

$$F_c = 1 \text{ (Grupo, Cliente Físico)}$$

$$F_a = 1 \text{ (Módulo de Asistencia por periodo)}$$

$$F_d = 0$$

El índice de madurez del software se calcula de la

$$IMS = [10 - (1 + 2 + 0)] / 10$$

$$IMS = 70 \%$$

En este caso podríamos decir que para esta modificación el sistema presenta un 70 % de estabilidad. Lo cual indica que es adaptable a la adición de este nuevo módulo funcional.

CONCLUSIONES

Este proyecto permitió realizar el análisis de la situación en la que se encontraba un Centro de Capacitación Computacional respecto a la logística y administración de los cursos que imparte y en base a ello proponer una solución informática a través de un sistema llamado SAPIECC. El desarrollo de este sistema se realizó considerando un enfoque de ingeniería de software lo cual permitió que el sistema se concluyera en tiempo y de acuerdo a las especificaciones acordadas al inicio del proyecto. Además de facilitar la documentación a lo largo de las diferentes fases de desarrollo del sistema.

Se comprobó que aunque existe una tendencia a elegir la primera propuesta de solución, cuando es necesario desarrollar un producto de programación, como es el caso de SAPIECC, el análisis de diferentes propuestas de solución nos brinda diferentes perspectivas y opciones para resolver un problema independientemente de las herramientas o la metodología final que se va a seguir para enfrentarlo ya que más que elegir el camino para solucionar algo evaluamos la forma en que podemos seguir ese camino. Observamos que una vez definidos los aspectos generales de la propuesta de solución para desarrollar SAPIECC, era importante analizar que herramientas permitían lograr los objetivos planteados y en base a un comparativo con diferentes opciones, considerando ventajas y desventajas, decidir cual nos funcionaba mejor. En este caso considerando tanto un lenguaje de programación (PHP) como un sistema para la administración de bases de datos (MYSQL). Además comprobamos que aunque muchas veces los programas con licencia comercial son los más amigables, el costo de las licencias y la compatibilidad con los equipos son una gran limitante al momento de desarrollar proyectos de programación.

Se pusieron en práctica los diferentes tipos de diagramas para el análisis de un problema y se observó que es muy importante comprender el funcionamiento del sistema desde el inicio ya que si existen deficiencias o errores, estos aparecen al momento de trasladar los diagramas a un diseño propio del sistema informático.

Además se observó que si no se va documentando el sistema desde un inicio es difícil establecer los tipos de pruebas aplicables a un sistema ya que de acuerdo a las características generales de este a través de las diferentes fases del proceso de ingeniería de software se van definiendo las pruebas necesarias para el sistema. Las cuales requieren un periodo de tiempo considerable ya que evalúan cada parte del sistema y no siempre arrojan resultados agradables ya que en el caso de aparecer un error nos lleva a realizar cambios en la implementación del sistema.

Se comprobó también que las métricas son una herramienta muy útil al momento de desarrollar sistemas ya que además de garantizar la calidad de un producto de software, como es el caso de SAPIECC ayudan a realizar planeaciones más exactas de costos y tiempos para realizar un sistema informático.

Finalmente se comprobó que aunque el sistema ya se encuentre funcionando en una institución, ya sea a modo de prueba o bien de forma definitiva, en realidad siempre continua a prueba por lo que es importante contar con un manual de usuario y referencia, así como un monitoreo constante de la aplicación.

APENDICES

DICCIONARIO DE DATOS DE SAPIECC

Nombre:	Cliente
Donde se Usa:	Modulo Alta Cliente, Modulo de Ayuda y Ajustes, Generación de Notas de Pago, Asignación de Cursos
Derivado de:	-
Descripción de Contenido:	Entidad que almacena los datos personales correspondientes a todos los clientes del centro
Subelementos:	Matricula_Cliente, Nombre_Cliente, A_Paterno_Cliente, A_Materno_Cliente, Calle_No_Cliente, Del_Municip_Cliente, CP_Cliente, Telefono_Cliente, E-Mail_Cliente, RFC_Cliente
Información Adicional:	Si el cliente pertenece a una empresa se agrega en esta tabla la matricula del cliente moral correspondiente. En caso contrario se conserva en cero.

Nombre:	Cliente_Moral
Donde se Usa:	Alta Cliente Moral, Ayuda y Ajustes, Pago de Facturas
Derivado de:	-
Descripción de Contenido:	Almacena información fiscal y general de todas las empresas que tienen contrato con el centro
Subelementos:	Matricula_Cliente_Moral, Razon_Social, Calle_No_Cliente_M, Colonia_Cliente_M, Del_Municip_Cliente_M, CP_Cliente_M, Telefono_Cliente_M, E-Mail_Cliente_M, RFC_Cliente_M
Información Adicional:	Matricula_Cliente_Moral=0 hace referencia a que se trata de un Cliente Físico.

Nombre:	Curso
Donde se Usa:	Asignación de Cursos, Generación de Comprobantes de Pago, Formación de Grupos, Ayuda y Ajustes
Derivado de:	-
Descripción de Contenido:	Contiene un listado de todos los cursos que se imparten en el centro así como de las distintas modalidades en que estos se pueden impartir
Subelementos:	Id_Curso, Nombre_Curso, Nivel, Horario, Tipo, Periodo
Información Adicional:	-

Nombre:	Nota_Pago
Donde se Usa:	Creación de Notas de Pago para clientes físicos

Derivado de:	-
Descripción de Contenido:	Almacena todas las Notas de Pago que sirven como comprobante de los clientes físicos del centro
Subelementos:	No_Nota_Pago, Monto_Pago, Fecha_Pago
Información Adicional:	Las notas de pago no son comprobantes fiscales sino solo para llevar un control interno de los pagos realizados por los alumnos regulares del centro.

Nombre:	Factura
Donde se Usa:	Creación e impresión de comprobantes fiscales para clientes morales (Facturas)
Derivado de:	-
Descripción de Contenido:	Almacena los pagos realizados por clientes morales
Subelementos:	No_Factura, Fecha_Pago, No_Cursos, Precio_Unitario, Subtotal, Total, Importe Letra, IVA.
Información Adicional:	La información almacenada corresponde con los comprobantes fiscales del centro.

Nombre:	Grupo
Donde se Usa:	Asignación de grupos de cada periodo en los distintos horarios
Derivado de:	Se crean al existir la relación de Clientes con Cursos en un determinado periodo y horario. Es decir cuando algún cliente demanda un curso
Descripción de Contenido:	Contiene toda la información que describe a cada uno de los grupos del centro.
Subelementos:	Id_Grupo, Periodo, Horario, Tipo, Nombre_Grupo, Nivel, Salon, No_Alumnos
Información Adicional:	Solo existira un grupo si existe demanda mayor a 3 alumnos exceptuando los casos en que se trata de un curso personal. Por lo que el número de grupos variara en cada periodo .

Nombre:	Horas
Donde se Usa:	Modulo de Ayuda y ajustes
Derivado de:	Grupos asignados a cada profesor en cada periodo
Descripción de Contenido:	Contiene la información de los grupos asignados a cada profesor en un determinado periodo, considerando las horas trabajadas que esto genera y permitiendo la modificación de este número para facilitar el cálculo de la nómina
Subelementos:	Id_Horas, Id_Profesor, Periodo, Horas_Contabilizadas, Horas_Ajustadas
Información Adicional:	Las razones para modificar el número de horas debe incluir el ausentismo de parte del profesor o bien la sustitución de un compañero y esto debe

	estar justificado físicamente por el coordinador del centro.
--	--

Nombre:	Profesor
Donde se Usa:	Asignación de Profesores a cada uno de los grupos, Modulo de ayuda y ajustes y en el conteo de horas de cada profesor.
Derivado de:	-
Descripción de Contenido:	Almacena los datos generales de cada uno de los profesores del centro, así como la clave que los identifica
Subelementos:	Id_Profesor, Turno, Nombre_Profesor, A_Paterno_Profesor, A_Materno_Profesor
Información Adicional:	-

Nombre:	Maricula_Cliente
Donde se Usa:	Tabla Cliente, Tabla Curso, Consultas Especiales
Derivado de:	Tabla Cliente
Descripción de Contenido:	Clave Unica que identifica a cada uno de los clientes del centro
Subelementos:	-
Información Adicional:	Es de tipo autonumerico personalizado. Es decir se incrementa automáticamente conforme se agrega un nuevo cliente.

Nombre:	Nombre_Cliente
Donde se Usa:	Tabla Cliente
Derivado de:	Tabla Cliente
Descripción de Contenido:	Almacena nombre completo cada uno de los clientes del centro
Subelementos:	-
Información Adicional:	Tipo texto longitud menor a 80 caracteres.

Nombre:	A_Paterno_Cliente
Donde se Usa:	Tabla Cliente
Derivado de:	Tabla Cliente
Descripción de Contenido:	Almacena el apellido paterno de cada uno de los clientes del centro
Subelementos:	-

Información Adicional:	Texto longitud menor a 30.
-------------------------------	----------------------------

Nombre:	A_Materno_Cliente
Donde se Usa:	Tabla Cliente
Derivado de:	Tabla Cliente
Descripción de Contenido:	Contiener el Apellido Materno de cada cliente del centro
Subelementos:	-
Información Adicional:	Tipo texto con longitud menor a 30.

Nombre:	Calle_No_Cliente
Donde se Usa:	Tabla Cliente
Derivado de:	Tabla Cliente
Descripción de Contenido:	Hace Referencia a la calle y número correspondiente a la dirección de cada cliente del centro
Subelementos:	-
Información Adicional:	Alfanumerico , longitud menor a 50.

Nombre:	Colonia_Cliente
Donde se Usa:	Tabla Cliente
Derivado de:	Tabla Cliente
Descripción de Contenido:	Corresponde a la colonia perteneciente al domicilio de cada cliente del centro
Subelementos:	-
Información Adicional:	Texto longitud menor a 30.

Nombre:	Del_Municip_Cliente
Donde se Usa:	Tabla Cliente
Derivado de:	Tabla Cliente
Descripción de Contenido:	Almacena la delegación o municipio de cada cliente fisico del centro.

Subelementos:	-
Información Adicional:	Texto longitud menor a 30.

Nombre:	CP_Cliente
Donde se Usa:	Tabla Cliente
Derivado de:	Tabla Cliente
Descripción de Contenido:	Guarda el Código Postal referente al domicilio de cada uno de los clientes del centro
Subelementos:	-
Información Adicional:	Alfanumerico longitud igual a 5.

Nombre:	Telefono_Cliente
Donde se Usa:	Tabla Cliente
Derivado de:	Tabla Cliente
Descripción de Contenido:	Corresponde al número telefónico de cada uno de los clientes del centro.
Subelementos:	-
Información Adicional:	Texto longitud menor o igual a 12. Nota: Solo se deben incluir dígitos en este campo.

Nombre:	E_Mail_Cliente
Donde se Usa:	Tabla Cliente
Derivado de:	Tabla Cliente
Descripción de Contenido:	Hace referencia al correo electrónico de cada uno de los clientes físicos inscritos en el centro de capacitación
Subelementos:	-
Información Adicional:	Texto longitud menor a 50. Este es un dato opcional.

Nombre:	RFC_Cliente
Donde se Usa:	Tabla Cliente
Derivado de:	Tabla Cliente

Descripción de Contenido:	Almacena el Registro Federal de Contribuyentes de cada uno de los clientes del centro de capacitación.
Subelementos:	-
Información Adicional:	Texto longitud menor a 15. Este dato es opcional.

Nombre:	Matricula_Cliente_Moral
Donde se Usa:	Tabla Cliente Moral, Tabla Cliente, Tabla Facturas, Consultas Especiales
Derivado de:	Tabla Cliente Moral
Descripción de Contenido:	Almacena una matricula única para cada una de las empresas que establecen algún contrato con el centro de capacitación
Subelementos:	-
Información Adicional:	Tipo Autonumerico Personalizado. Matricula_Cliente_Moral = 0 indica que se trata de un cliente físico

Nombre:	Razon_Social
Donde se Usa:	Tabla Cliente Moral
Derivado de:	Tabla Cliente Moral
Descripción de Contenido:	Corresponde a la Razón Social que utiliza cada cliente moral del centro de capacitación.
Subelementos:	-
Información Adicional:	Texto longitud menor a 75 caracteres. Este campo es obligatorio para todos los clientes morales.

Nombre:	Calle_No_Cliente_M
Donde se Usa:	Tabla Cliente Moral
Derivado de:	Tabla Cliente Moral
Descripción de Contenido:	Almacena la calle y el número correspondiente a la dirección fiscal de cada una de las empresas que establecen algún contrato el centro.
Subelementos:	-
Información Adicional:	Texto longitud menor o igual a 80 caracteres. Campo obligatorio, ya que se requiere para efectos de facturación.

Nombre:	Colonia_Cliente_M
Donde se Usa:	Tabla Cliente Moral

Derivado de:	Tabla Cliente Moral
Descripción de Contenido:	Hace referencia a la colonia en donde se ubica el domicilio fiscal de cada cliente moral con el cual establece algún contrato.
Subelementos:	-
Información Adicional:	Texto longitud menor a 30. Campo obligatorio para efectos de facturación.

Nombre:	Del_Municip_Cliente_M
Donde se Usa:	Tabla Cliente Moral
Derivado de:	Tabla Cliente Moral
Descripción de Contenido:	Almacena la delagación o municipio correspondiente al domicilio de cada uno de los clientes morales del centro
Subelementos:	-
Información Adicional:	Texto longitud menor o igual a 30 caracteres. Campo obligatorio para efectos de facturación.

Nombre:	CP_Cliente_M
Donde se Usa:	Tabla Cliente Moral
Derivado de:	Tabla Cliente Moral
Descripción de Contenido:	Contiene el Código Postal correspondiente a la dirección fiscal de cada cliente moral
Subelementos:	
Información Adicional:	Tipo Texto longitud igual a 5 caracteres.

Nombre:	Telefono_Cliente_M
Donde se Usa:	Tabla Cliente Moral
Derivado de:	Tabla Cliente Moral
Descripción de Contenido:	Almacena el teléfono de la empresa que establece un contrato con el centro
Subelementos:	-
Información Adicional:	Texto con longitud menor de 15 caracteres.

Nombre:	E-Mail_Cliente_M
----------------	------------------

Donde se Usa:	Tabla Cliente Moral
Derivado de:	Tabla Cliente Moral
Descripción de Contenido:	Almacena el correo electrónico de contacto para cada empresa del centro
Subelementos:	-
Información Adicional:	Datos tipo texto con longitud menor o igual a 50 caracteres.

Nombre:	RFC_Cliente_M
Donde se Usa:	Tabla Cliente Moral, Facturas y Consultas Especiales
Derivado de:	Tabla Cliente Moral
Descripción de Contenido:	Almacena el Registro Federal de Contribuyentes de cada empresa o cliente moral.
Subelementos:	-
Información Adicional:	Datos tipo Texto con longitud menor o igual a 15 caracteres.

Nombre:	Id_Curso
Donde se Usa:	Tabla Curso, Notas de Pago, Formación de Grupos y Consultas Especiales
Derivado de:	Tabla Curso
Descripción de Contenido:	Almacena la clave única de cada curso asignado en el centro.
Subelementos:	-
Información Adicional:	Datos tipo autonúmerico personalizado. Se genera de manera automática e incremental cada que se asigna un nuevo curso en el centro.

Nombre:	Nombre_Curso
Donde se Usa:	Tabla Curso, Asignación de Grupos y Consultas Especiales
Derivado de:	Tabla Curso
Descripción de Contenido:	Nombre de cada uno de los cursos que se imparten en el centro.
Subelementos:	-
Información Adicional:	Cada nombre esta relacionado con un número entero de acuerdo a la siguiente lista: 1.Introducción; 2. Windows; 3. Word; 4. Excel ; 5. Power Point; 6.

	Project; 7. Outlook; 8. COI ; 9. NOI; 10. SAE; 11. SUA; 12. Corel Draw; 13. Free Hand; 14. Ilustrator
--	---

Nombre:	Nivel
Donde se Usa:	Tabla Curso, Formación de Grupos y Consultas Especiales
Derivado de:	Tabla Curso
Descripción de Contenido:	Almacena el nivel correspondiente al curso que se esta dando de alta
Subelementos:	Las opciones que maneja este campo son: 1.- Básico 2.- Intermedio 3.- Avanzado
Información Adicional:	Datos tipo entero condicionado, ya que solo se puede elegir una de las opciones disponibles y visibles para el usuario en forma de texto pero almacenadas de forma numérica de acuerdo a como se menciona anteriormente.

Nombre:	Horario
Donde se Usa:	Tabla Curso, Asignación de Grupos
Derivado de:	Tabla Curso
Descripción de Contenido:	Almacena el horario en que se impartira el curso que se esta dando de alta
Subelementos:	La opciones que se manejan para los horarios disponibles en el centro son las siguientes: 1.- 7:00 a 8:20 a.m. 2.- 8:30 a 9:50 a.m. 3.- 10:10 a 11:30 a.m. 4.- 11:30 a 1:00 p.m. 5.- 1:30 a 3:00 p.m. 6.- 3:00 a 4:20 p.m. 7.- 4:30 a 5:50 p.m.
Información Adicional:	Datos tipo entero. Dependiendo del horario que elija el cliente se almacenara el numero entero que haga referencia al horario elegido.

Nombre:	Tipo
Donde se Usa:	Tabla Curso, Formación de Grupos
Derivado de:	Tabla Curso
Descripción de Contenido:	Tipo de Curso que se asigna a cada uno de los clientes del centro
Subelementos:	El tipo de curso que se puede asignar es tipo numérico de acuerdo a la lista siguiente:

	<p>1.- Semi-Intensivo 90 min 2.- Semi-Intensivo 60 min 3.- Intensivo 4.- Sabatino 5.- Mensual 6.-SNE</p>
Información Adicional:	El usuario observa los datos tipo lista o texto pero en la base se almacena el valor numérico asociado a la opción elegida al dar de alta el curso.

Nombre:	Periodo
Donde se Usa:	Tabla Curso, Formación de Grupos, Nota de Pago, Facturas, Consultas Especiales
Derivado de:	Tabla Curso
Descripción de Contenido:	Almacena el Periodo correspondiente a las fechas en que cada cliente tomara el curso de su interes.
Subelementos:	<p>Cada periodo cuenta con una clave numérica de acuerdo al siguiente orden:</p> <p>1.-Enero1 2.-Enero2 3.-Febrero1 4.-Febrero2 5.-Marzo 1 6.- Marzo 2 7.-Abril 1 8.- Abril 2 9.-Mayo 1 10.-Mayo 2 11.-Junio 1 12.-Junio 2 13.-Julio 1 14.-Julio 2 15.-Ago</p>
Información Adicional:	Las opciones de este campo se encuentran visibles mediante una lista de opciones tipo texto pero que se almacenan de forma numerica referenciando la elección del usuario con el número que le corresponde.

Nombre:	No_Nota
Donde se Usa:	Tabla Nota Pago, Consultas Especiales
Derivado de:	Tabla Nota de Pago
Descripción de Contenido:	Almacena el número de nota correspondiente a cada pago realizado por algún cliente físico.
Subelementos:	-
Información Adicional:	Tipo autonumerico.

Nombre:	Monto_Pago
----------------	------------

Donde se Usa:	Tabla Nota de pago, Consultas Especiales
Derivado de:	Tabla Nota Pago
Descripción de Contenido:	Almacena el importe pagado en cada nota de pago generada por algún cliente
Subelementos:	-
Información Adicional:	Datos tipo moneda.

Nombre:	Fecha_Pago
Donde se Usa:	Tabla Nota de Pago
Derivado de:	Tabla Nota de Pago
Descripción de Contenido:	Contiene la fecha en que se realizo el pago correspondiente a cada nota de pago.
Subelementos:	-
Información Adicional:	Contiene datos tipo fecha.

Nombre:	No_Factura
Donde se Usa:	Tabla Factura, Consultas Especiales
Derivado de:	Tabla Factura
Descripción de Contenido:	Campo que muestra el número de factura que se brindara a algún cliente moral
Subelementos:	-
Información Adicional:	Datos tipo autonumerco que que se genera automáticamente al crear una nueva factura.

Nombre:	Fecha_Pago_F
Donde se Usa:	Tabla Factura
Derivado de:	Tabla factura
Descripción de Contenido:	Corresponde a la fecha en que se expide cada factura a los distintos clientes morales.
Subelementos:	-
Información Adicional:	Datos tipo fecha.

Nombre:	No_Cursos
Donde se Usa:	Tabla Factura
Derivado de:	Tabla Factura
Descripción de Contenido:	Campo que contiene el número de cursos pagados en cada factura expedida a algún cliente moral.
Subelementos:	-
Información Adicional:	Contiene datos tipo entero.

Nombre:	Precio_Unitario
Donde se Usa:	Tabla Factura
Derivado de:	Tabla Factura
Descripción de Contenido:	Campo que almacena el precio unitario correspondiente a cada uno de los cursos que se desglosan en cada factura.
Subelementos:	-
Información Adicional:	Almacena datos tipo moneda.

Nombre:	Subtotal
Donde se Usa:	Tabla Factura
Derivado de:	Tabla Factura
Descripción de Contenido:	Campo que contiene el subtotal correspondiente a cada factura.
Subelementos:	-
Información Adicional:	Datos numéricos tipo moneda

Nombre:	Total
Donde se Usa:	Tabla Factura
Derivado de:	Tabla Factura
Descripción de Contenido:	Campo que almacena el total facturado a cada cliente moral del centro.
Subelementos:	-
Información Adicional:	Contiene datos numéricos tipo moneda.

Nombre:	Importe_Letra
Donde se Usa:	Tabla Factura
Derivado de:	Tabla Factura
Descripción de Contenido:	Almacena el importe con letra correspondiente a la factura expedida a cada cliente moral.
Subelementos:	-
Información Adicional:	Datos tipo texto con longitud menor o igual a 80 caracteres.

Nombre:	IVA
Donde se Usa:	Tabla Factura
Derivado de:	Tabla Factura
Descripción de Contenido:	Guarda el valor correspondiente al iva de cada factura. Es decir el subtotal multiplicado por 0.16.
Subelementos:	-
Información Adicional:	Dato numérico tipo moneda.

Nombre:	Id_Profesor
Donde se Usa:	TablaProfesor, Formación de Grupos, Ajuste de horas y Consultas Especiales.
Derivado de:	Tabla Profesor
Descripción de Contenido:	Almacena la clave única de cada profesor del centro
Subelementos:	-
Información Adicional:	Autonúmerico personalizado.

Nombre:	Turno
Donde se Usa:	Tabla Profesor
Derivado de:	Tabla Profesor
Descripción de Contenido:	Almacena el turno en que se encuentra disponible cada profesor. Existen tres opciones.
Subelementos:	1.-Matutino 2.-Vespertino

	3.- Mixto
Información Adicional:	Numerico Entero. Se debe elegir el número correspondiente al tipo de disponibilidad más adecuado y este se almacena de forma numérica.

Nombre:	Nombre_Profesor
Donde se Usa:	Tabla Profesor, Ajuste de Horas, Consultas Especiales
Derivado de:	Tabla Profesor
Descripción de Contenido:	Almacena el nombre de cada profesor que imparte clases en el centro de capacitación
Subelementos:	-
Información Adicional:	Tipo texto con longitud menor o igual a 30.

Nombre:	A_Paterno_Profesor
Donde se Usa:	Tabla Profesor, Ajuste de Horas, Consultas Especiales
Derivado de:	Tabla Profesor
Descripción de Contenido:	Campo que contiene el apellido paterno de cada profesor.
Subelementos:	-
Información Adicional:	Texto con longitu menor o igual a 30 caracteres.

Nombre:	A_Materno_Profesor
Donde se Usa:	Tabla Profesor, Ajuste de Horas, Consultas Especiales
Derivado de:	Tabla Profesor
Descripción de Contenido:	Almacena el apellido materno de cada profesor del centro.
Subelementos:	-
Información Adicional:	Datos tipo texto con longitud menor o igual a 30 caracteres.

Nombre:	Id_Grupo
Donde se Usa:	Tabla Grupo, Consultas Especiales
Derivado de:	Tabla Grupo
Descripción de Contenido:	Almacena la clave única asignada a cada grupo que se forma en el centro.

Subelementos:	-
Información Adicional:	Datos tipo autonumerico que se genera automaticamente al generar cada nuevo grupo.

Nombre:	Id_Horas
Donde se Usa:	Tabla Horas
Derivado de:	Tabla Horas
Descripción de Contenido:	Clave correspondiente al registro de las horas trabajadas por cada profesor en cada periodo. Es asignado de manera automática por el sistema-
Subelementos:	-
Información Adicional:	Autonumerico personalizado.

Nombre:	Horas Contabilizadas
Donde se Usa:	Tabla Horas, Ajustar Horas
Derivado de:	Tabla Horas
Descripción de Contenido:	Almacena las horas trabajadas por cada profesor, de manera predeterminada corresponde al número de grupos asignados por periodo por 15.
Subelementos:	-
Información Adicional:	Dato numérico tipo entero.

Nombre:	Horas_Ajustadas
Donde se Usa:	Tabla Horas, Ajustar Horas
Derivado de:	Tabla Horas
Descripción de Contenido:	Almacena las horas ajustadas de cada profesor, de manera predeterminada es igual al valor de las horas contabilizadas. Sin embargo cambia de acuerdo a las consideraciones especiales del coordinador.
Subelementos:	-
Información Adicional:	Número tipo entero

MANUAL DE USUARIO

1. Instalación
2. Pantalla principal
3. Generar movimiento
 - 3.1. Alta cliente
 - 3.1.1. Opciones alta cliente físico
 - 3.1.1.1. Alta cliente físico
 - 3.1.1.2. Buscar cliente físico
 - 3.1.1.3. Eliminar cliente físico
 - 3.1.1.4. Listado general cliente físico
 - 3.1.2. Opciones cliente moral
 - 3.1.2.1. Alta cliente moral
 - 3.1.2.2. Buscar cliente moral
 - 3.1.2.3. Eliminar cliente moral
 - 3.1.2.4. Listado general cliente moral
 - 3.2. Asignar curso
 - 3.3. Pagar curso
 - 3.3.1. Crear nota de pago
 - 3.3.2. Crear factura
 - 3.4. Agregar profesor
 - 3.4.1. Agregar profesor
 - 3.4.2. Buscar profesor
 - 3.4.3. Eliminar profesor
4. Consultas del sistema
 - 4.1. Formar grupo
 - 4.2. Consultas
 - 4.3. Ajustar horas
5. Ayuda

1. INSTALACIÓN

Para poder instalar el programa SAPIECC es necesario que el equipo en donde se pretende que resida el sistema, cuente con un compilador de PHP, una base de Datos de MYSQL y un servidor virtual. En caso de no contar con esto es necesario ejecutar el archivo XAMPP que se encuentra en el disco de instalación. Este archivo instala de manera automática el compilador, la base de datos MYSQL y un servidor apache.

El disco de instalación contiene una carpeta llamada SAPIECC, la cual contiene todos los archivos correspondientes a la interfaces y rutinas del sistema. Para que el sistema funcione correctamente es necesario copiar la carpeta completa en la ruta siguiente:

C:\xampp\xampp\htdocs

El disco además contiene el archivo correspondiente a la estructura de la base de datos del sistema, denominado SAPIECC.data, Para que funcione correctamente en conjunto con el sistema es necesario copiarlo en la siguiente carpeta:

C:\Program Files (x86)\MySQL\MySQL Server 5.0\data

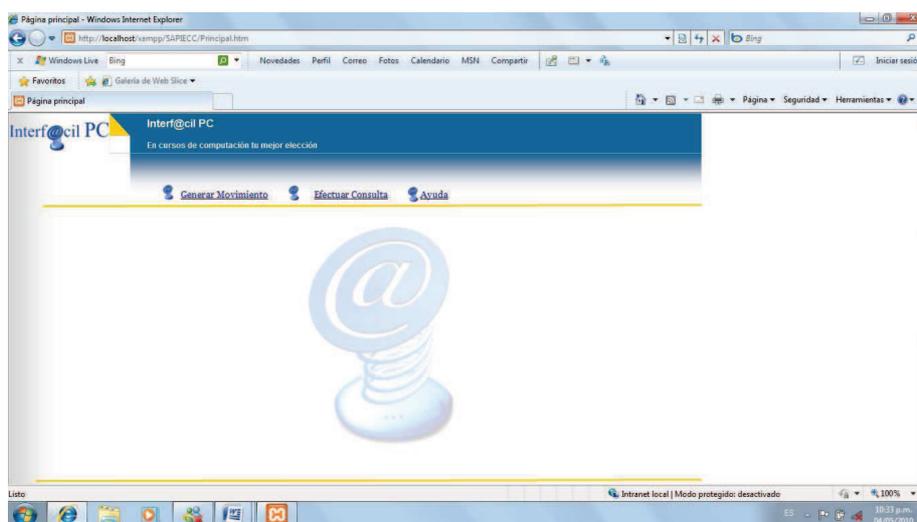
Una vez realizada la copia de estos archivos ya se cuenta con la instalación básica del sistema.

2. PANTALLA PRINCIPAL

Para empezar a utilizar el sistema SAPIECC es necesario seguir los siguientes pasos:

1. En el botón Inicio dar clic en todos los programas
2. Elegir Internet Explorer.
3. En la barra de direcciones escribir la siguiente ruta:
<http://localhost/xampp/SAPIECC/Principal.htm>

Aparecerá la siguiente pantalla:



Existen tres opciones a considerar en esta pantalla. Para elegir cada una de ellas solo es necesario dar clic sobre el hipervínculo correspondiente:

OPCIÓN	RESULTADO
Generar Movimiento	Despliega la pantalla correspondiente a los diferentes movimientos que se pueden efectuar en el sistema.
Efectuar Consulta	Despliega en la pantalla las consultas programadas de SAPIECC.
Ayuda	Despliega el tutorial de ayuda para el sistema.

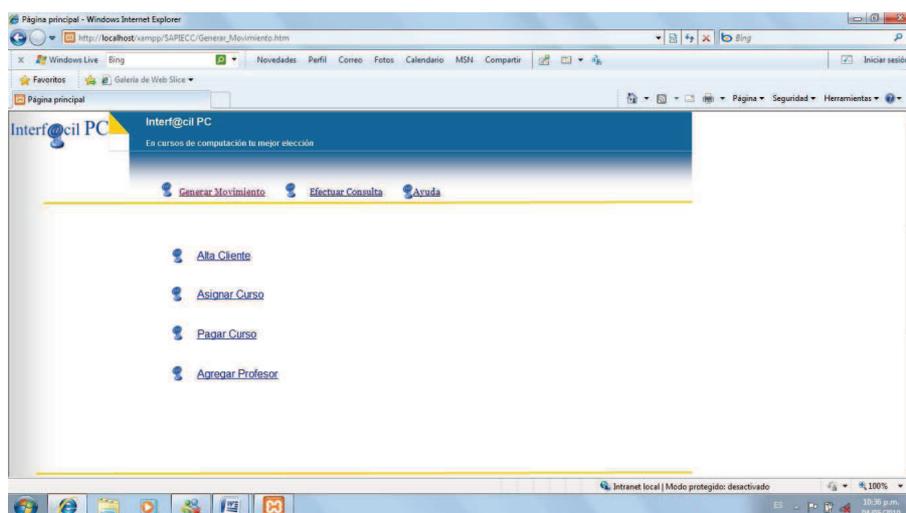
Nota: Para almacenar la ruta correspondiente a la pantalla principal del sistema SAPIECC es necesario seguir los pasos siguientes:

1. En el Explorador de Internet, dar clic en el menú Herramientas.
2. Dar clic en Opciones de Internet.
3. En la pestaña General elegir la opción Usar Actual.
4. Dar clic en Aceptar.

Esto producirá que la próxima vez que se abra el explorador de Internet aparezca de manera predeterminada la página principal de SAPIECC. Y en caso de desaparecer solo será necesario dar clic en el botón home del explorador.

4. GENERAR MOVIMIENTO

Para realizar algún movimiento dentro del sistema SAPIECC es necesario acceder a la pantalla principal del sistema y dar clic sobre la opción Generar Movimiento.



Aparecerá una pantalla con los movimientos disponibles dentro del sistema. Para acceder a alguno de ellos solo es necesario dar clic en la opción correspondiente:

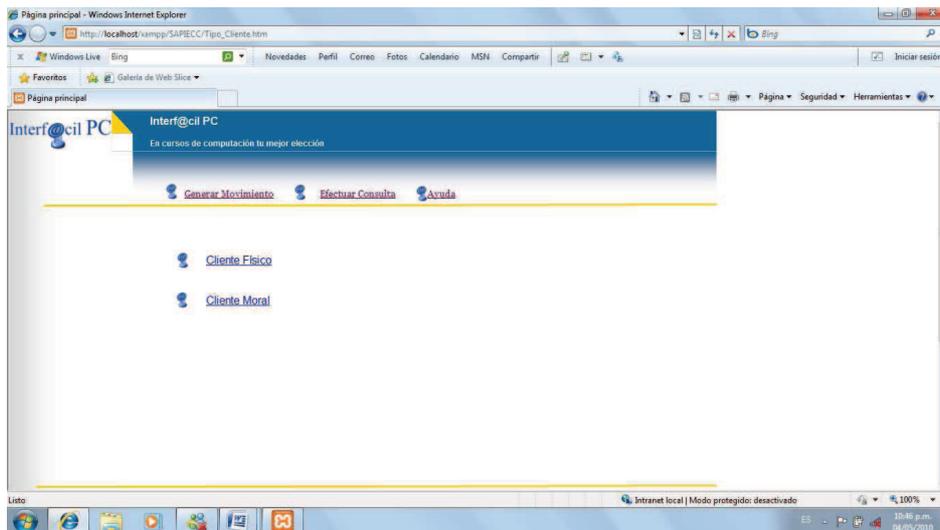
OPCIÓN	RESULTADO
Alta Cliente	Despliega la pantalla correspondiente al alta de clientes.
Asignar Curso	Despliega la pantalla correspondiente a la asignación de cursos
Pagar Curso	Despliega la pantalla que tiene como función administrar los pagos del centro.
Asignar Profesor	Despliega la interfaz correspondiente al alta de nuevos profesores en el plantel.

3.1. ALTA CLIENTE

Para dar de alta un cliente en el sistema se deben seguir los siguientes pasos:

1. En la pantalla principal, dar clic en Generar Movimiento.
2. Dar clic en Alta Cliente.

Aparecerá la siguiente pantalla.

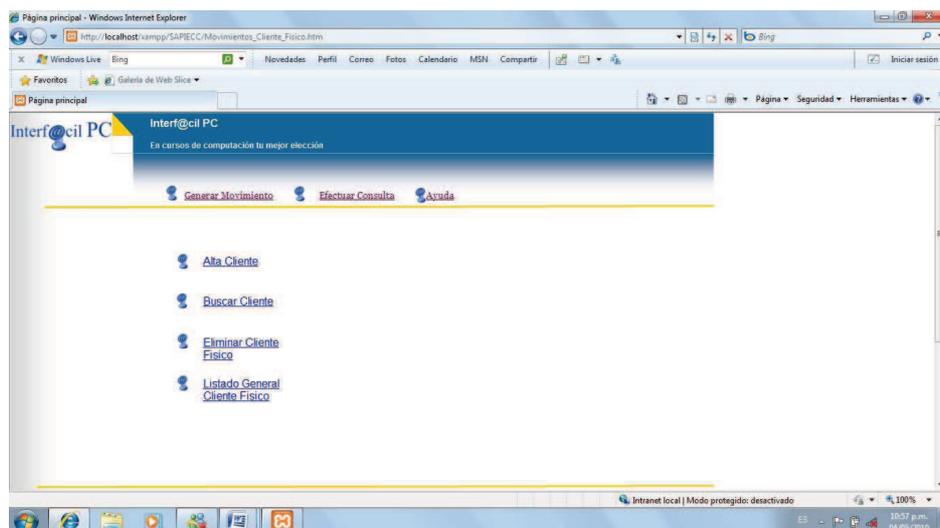


Existen dos opciones que se pueden tomar en esta pantalla las cuales se describen a continuación:

OPCIÓN	RESULTADO
Cliente Físico	Despliega la pantalla correspondiente al Alta de Clientes Físicos.
Cliente Moral	Despliega la pantalla correspondiente al Alta de Clientes Morales o Empresas.

3.1.1 OPCIONES ALTA CLIENTE FÍSICO

En caso de elegir la opción correspondiente a Cliente Físico aparecerá la pantalla con las opciones disponibles para los clientes de este tipo.



Enseguida se describe el funcionamiento de cada una de estas opciones:

OPCIÓN	RESULTADO
Alta Cliente	Despliega la pantalla correspondiente al movimiento de Alta de Clientes Físicos, desplegando todos los datos necesarios para realizar este.
Buscar Cliente	Despliega la pantalla que permite introducir parámetros para realizar diferentes tipos de búsqueda dentro de la tabla Clientes Físicos.
Eliminar Cliente	Despliega la pantalla destinada a la eliminación de Clientes Físicos.
Listado General de Clientes Físicos	Muestra un reporte con las tabla completa de Clientes Físicos almacenados hasta el momento en la base de datos del sistema.

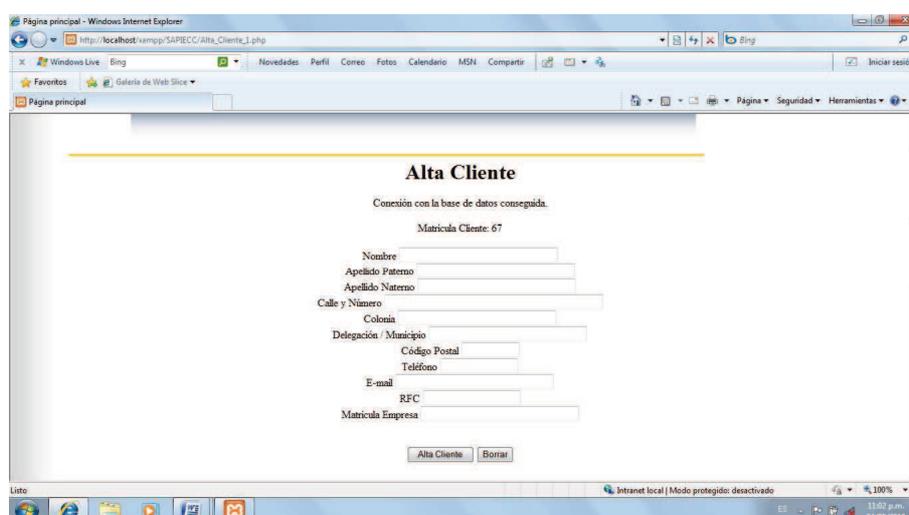
A continuación se describen algunas características para cada uno de estos casos.

3.1.1.1 ALTA CLIENTE FÍSICO

Los datos que se requieren para generar el alta de un Cliente Físico son: Nombre, Apellido Paterno, Apellido Materno, Calle y Número, Colonia, Delegación / Municipio, Código Postal, Teléfono, E-mail, RFC, Matricula Empresa.

Para dar de alta un cliente en el sistema se deben seguir los siguientes pasos:

1. En la pantalla principal, dar clic en Generar Movimiento.
2. Dar clic en Cliente Físico.
3. Dar clic en Alta Cliente.
4. Introducir la información correspondiente.
5. Dar clic en Alta Cliente



Aparecerá una pantalla con los datos almacenados y con las opciones correspondientes a los diferentes tipos de movimientos para clientes físicos.

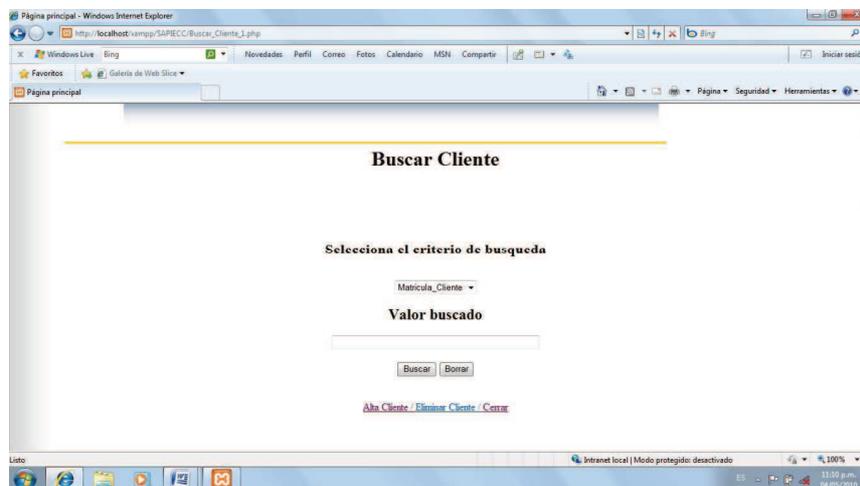
Nota: En caso de que el cliente físico no pertenezca a ninguna empresa es necesario colocar cero en el campo matrícula empresa.

La opción borrar, limpia el contenido de cada una de las cajas de texto del formulario alta cliente físico.

3.1.1.2 BUSCAR CLIENTE FÍSICO

Para buscar a un cliente en el sistema se deben seguir los siguientes pasos:

1. En la pantalla principal, dar clic en Generar Movimiento.
2. Dar clic en Cliente Físico.
3. Dar clic en Buscar Cliente.
4. Elegir el criterio de búsqueda (Matricula ó Nombre).
5. Introducir el valor buscado.
6. Dar clic en buscar.



Aparecerá una pantalla con los datos correspondientes al registro buscado. Ó bien el mensaje indicando que no se encuentra el registro buscado en el sistema.

La opción borrar, limpia el contenido de cada una de las cajas de texto del formulario. Mientras que las opciones de Alta Cliente y eliminar cliente llevan a las pantallas respectivas de cada movimiento. Y la opción de cerrar lleva a la pantalla principal del sistema.

3.1.1.3 ELIMINAR CLIENTE FÍSICO

Para que el sistema pueda eliminar un cliente de la base de datos primero es necesario realizar la búsqueda de la persona que se desea quitar y posteriormente es necesario confirmar que se desea eliminar definitivamente de la base de datos del sistema.

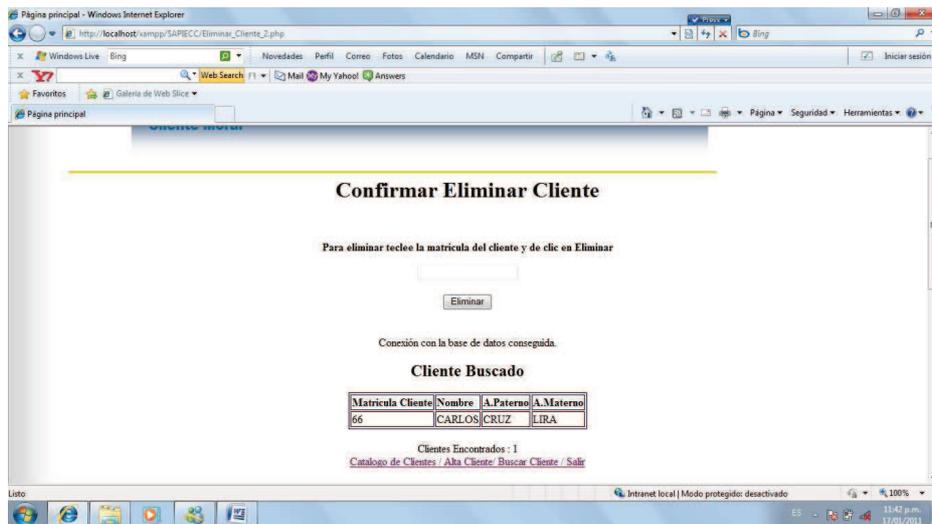
Para eliminar a un cliente del sistema se deben seguir los siguientes pasos:

1. En la pantalla principal, dar clic en Generar Movimiento.
2. Dar clic en Cliente Físico.
3. Dar clic en Eliminar Cliente.
4. Elegir el criterio de búsqueda (Matricula ó Nombre).
5. Introducir el valor buscado.
6. Dar clic en buscar.



Aparecerá una pantalla con los datos correspondientes al registro buscado. Ó bien el mensaje indicando que no se encuentra el registro buscado en el sistema. En caso de no existir la pantalla muestra las opciones para realizar los diferentes movimientos correspondientes a cliente físico. Mientras que en el caso de ser correctos es necesario volver a introducir la matricula del cliente para confirmar que se desea eliminar del sistema.

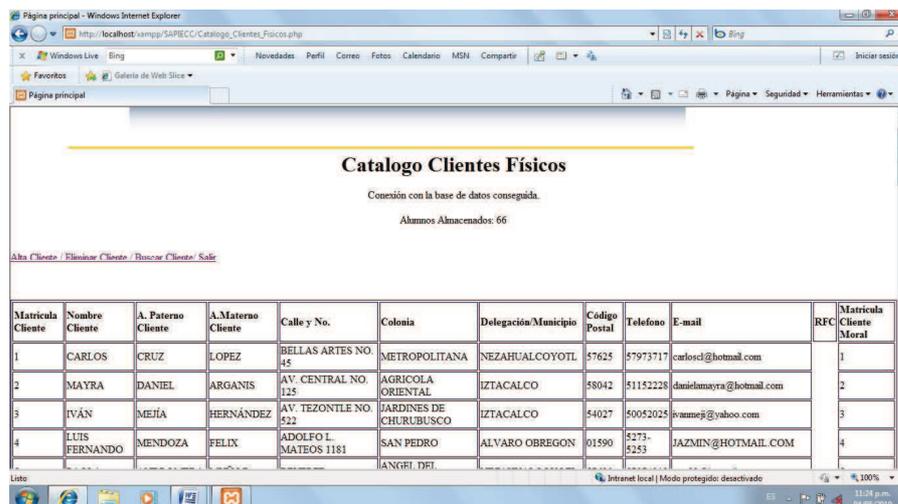
Si la matricula coincide se elimina el registro del sistema y muestra en pantalla los datos del registro que se acaba de eliminar.



3.1.1.4 LISTADO GENERAL FÍSICO

Para desplegar en pantalla el registro de todos los clientes físicos almacenados en el sistema es necesario seguir los siguientes pasos:

1. En la pantalla principal, dar clic en Generar Movimiento.
2. Dar clic en Cliente Físico.
3. Dar clic en Listado General Cliente Físico.

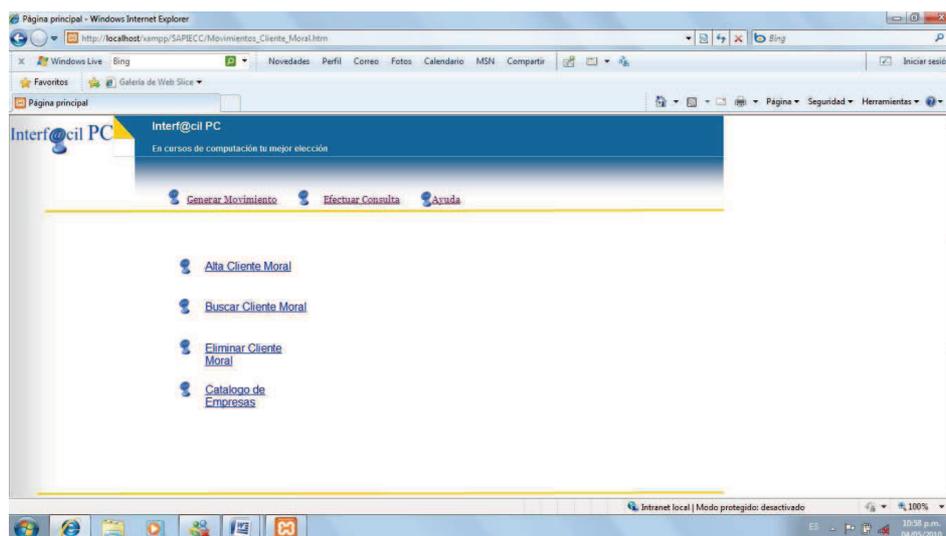


En este caso solo aparece la pantalla con todos los registros integrantes de la tabla clientes físicos. En esta opción al igual que en todas las correspondientes a movimientos cliente físico se cuenta con

un menú que permite al usuario desplazarse a través de los distintos movimientos como son altas, búsquedas y eliminaciones.

3.1.2 OPCIONES CLIENTE MORAL

Las opciones correspondientes al Alta de Cliente Moral tienen la apariencia siguiente:



Existen cuatro opciones que se pueden elegir dentro de esta pantalla. Enseguida se describe su funcionamiento:

OPCIÓN	RESULTADO
Alta Cliente Moral	Despliega la pantalla correspondiente al movimiento de Alta de Clientes Morales, desplegando todos los datos necesarios para realizar este.
Buscar Cliente Moral	Despliega la pantalla que permite introducir parámetros para realizar diferentes tipos de búsqueda dentro de la tabla Cliente Moral.
Eliminar Cliente Moral	Despliega la pantalla destinada a la eliminación de Clientes Morales.
Catalogo de Empresas	Muestra un reporte con las tabla completa de Clientes Morales almacenados hasta el momento en la base de datos del sistema.

3.1.2.1 ALTA CLIENTE MORAL

Los datos que se requieren para generar el alta de un Cliente Físico son: Razón Social, Calle y Número, Colonia, Delegación / Municipio, Código Postal, Teléfono, E- mail, RFC.

Para dar de alta un cliente moral en el sistema se deben seguir los siguientes pasos:

1. En la pantalla principal, dar clic en Generar Movimiento.
2. Dar clic en Cliente Moral.
3. Dar clic en Alta Cliente.
4. Introducir la información correspondiente.
5. Dar clic en Alta Cliente

The screenshot displays a web browser window titled 'Página principal - Windows Internet Explorer'. The address bar shows the URL 'http://localhost/vampp/SAPRECC/Alta_Cliente_Moral_1.php'. The main content area of the browser displays a form titled 'Alta Cliente Moral'. Above the form, it states 'Conexión con la base de datos conseguida.' and 'Matrícula Cliente: 43'. The form contains several input fields: 'Razón Social', 'Calle y Número', 'Colonia', 'Delegación / Municipio', 'Código Postal', 'Teléfono', 'E-mail', and 'RFC'. Below the 'RFC' field, there are two buttons: 'Alta Cliente' and 'Borrar'. At the bottom of the form, there are three links: 'Buscar Empresa', 'Eliminar Empresa', and 'Cerrar'. The browser's taskbar at the bottom shows the system tray with the date and time '12:08 a.m. 05/02/2010'.

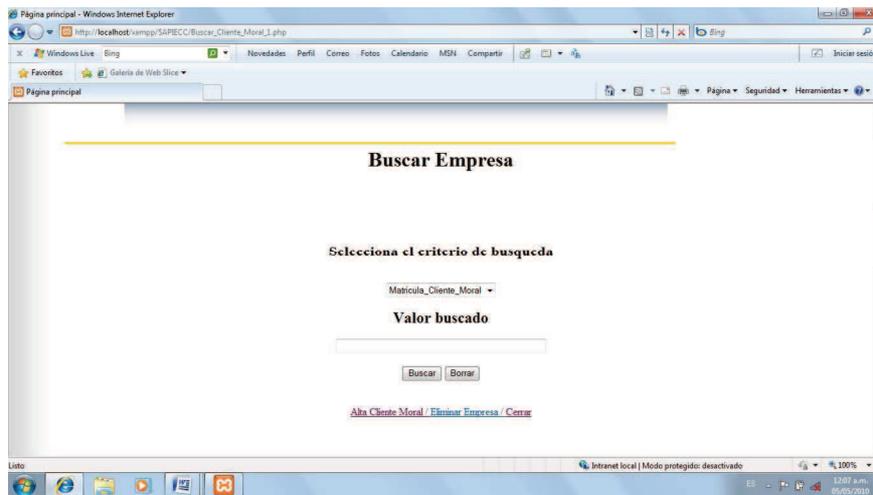
Aparecerá una pantalla con los datos almacenados y con las opciones correspondientes a los diferentes tipos de movimientos para clientes morales.

La opción borrar, limpia el contenido de cada una de las cajas de texto del formulario.

3.1.2.2 BUSCAR CLIENTE MORAL

Para buscar a un cliente moral en el sistema se deben seguir los siguientes pasos:

1. En la pantalla principal, dar clic en Generar Movimiento.
2. Dar clic en Cliente Moral.
3. Dar clic en Buscar Cliente.
4. Elegir el criterio de búsqueda (Matrícula ó Razón Social).
5. Introducir el valor buscado.
6. Dar clic en buscar.



Aparecerá una pantalla con los datos correspondientes al registro buscado. Ó bien el mensaje indicando que no se encuentra el registro buscado en el sistema.

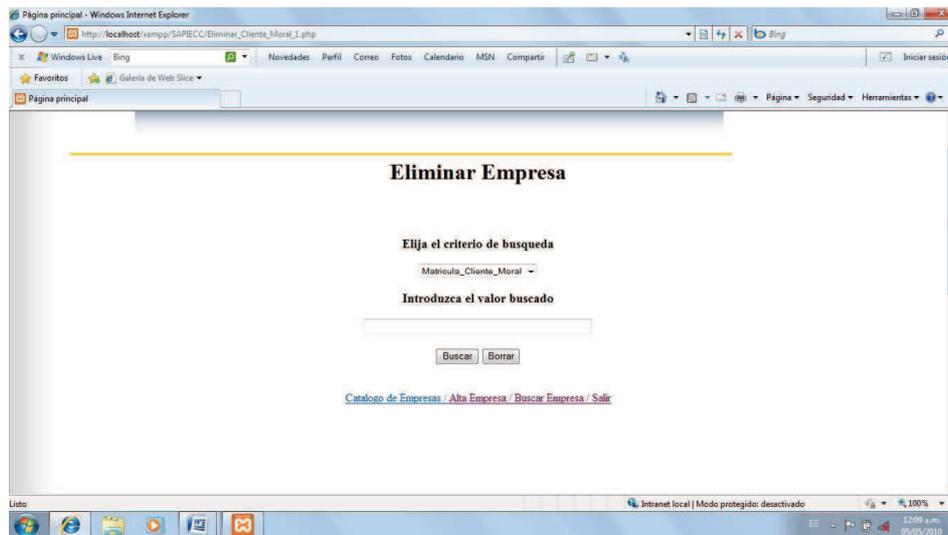
La opción borrar, limpia el contenido de cada una de las cajas de texto del formulario. Mientras que las opciones de Alta Cliente Moral y Eliminar Cliente Moral llevan a las pantallas respectivas de cada movimiento. Y la opción de cerrar lleva a la pantalla principal del sistema.

3.1.2.3 ELIMINAR CLIENTE MORAL

Para que el sistema pueda eliminar un cliente moral de la base de datos primero es necesario realizar la búsqueda de la empresa que se desea quitar y posteriormente es necesario confirmar que se desea eliminar definitivamente de la base de datos del sistema.

Para eliminar a un cliente del sistema se deben seguir los siguientes pasos:

1. En la pantalla principal, dar clic en Generar Movimiento.
2. Dar clic en Cliente Moral.
3. Dar clic en Eliminar Cliente.
4. Elegir el criterio de búsqueda (Matricula ó Razón Social).
5. Introducir el valor buscado.
6. Dar clic en buscar.



Aparecerá una pantalla con los datos correspondientes al registro buscado. Ó bien el mensaje indicando que no se encuentra el registro buscado en el sistema. En caso de no existir la pantalla muestra las opciones para realizar los diferentes movimientos correspondientes a cliente moral. Mientras que en el caso de ser correctos es necesario volver a introducir la matricula del cliente para confirmar que se desea eliminar del sistema.

Si la matricula coincide se elimina el registro del sistema y muestra en pantalla los datos del registro que se acaba de eliminar.

3.1.2.4 LISTADO GENERAL CLIENTE MORAL

Para desplegar en pantalla el registro de todas las empresas almacenadas en el sistema es necesario seguir los siguientes pasos:

1. En la pantalla principal, dar clic en Generar Movimiento.
2. Dar clic en Cliente Moral.
3. Dar clic en Catalogo de Empresas.

Catalogo de Empresas
Conexión con la base de datos conseguida.
Empresas Almacenadas: 42

[Alta Cliente](#) / [Eliminar Cliente](#) / [Buscar Cliente](#) / [Salir](#)

Matricula	Razón Social	Calle y No.	Colonia	Delegación/Municipio	Código Postal	Telefono	E-mail	RFC
0	SCM							
1	SAPREGO, DESARROLLO EMPRESARIAL Y DE NEGOCIOS	Querétaro	Querétaro	Querétaro	76138	01-442-4411-300	lherandez@saprego.com	NAGM570423HDFVLN08
2	DOW QUÍMICA MEXICANA	Bvtd. Mamel Ávila Casacho # 32-6o piso	Lomas de Chapultepec	Miguel Hidalgo	51123	5201-4726	mbcoissopereillo@dow.com	VAVY680625MDFLL04
3	FONDIKA	Horacio No. 1429	Polanco	Miguel Hidalg	11560	5026 - 0023	claudia.zavala@allianzfondika.com	AVZL720510HDFLNS05

En este caso aparecerá en pantalla la tabla correspondiente a todos los registros integrantes de la tabla clientes morales. En esta opción al igual que en todas las correspondientes a movimientos cliente moral se cuenta con un menú que permite al usuario desplazarse a través de los distintos movimientos como son altas, búsquedas y eliminaciones.

3.2 ASIGNAR CURSO

Para asignar un curso a un cliente del sistema se deben seguir los siguientes pasos:

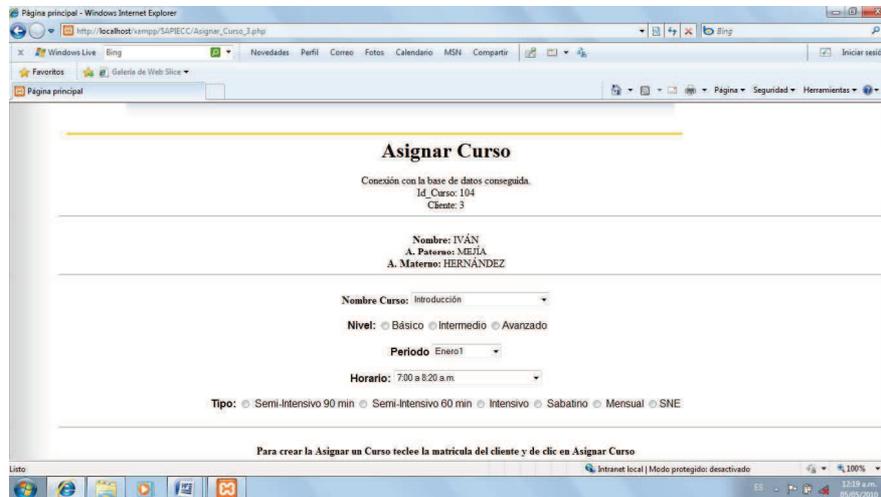
1. En la pantalla principal, dar clic en Generar Movimiento.
2. Dar clic en Asignar Curso.
3. Introducir la matricula correspondiente al alumno al cual se le desea asignar el curso.
4. Dar clic en asignar curso.

Asignar Curso

Matricula del Alumno

[Catalogo de Clientes](#) / [Alta Cliente](#) / [Salir](#)

En el caso de que la matricula del cliente se valida aparecerá la siguiente interfaz:



En esta pantalla aparecen los datos correspondientes a la matricula insertada además de las opciones correspondientes al curso que se desea asignar, una vez indicados estos es necesario realizar una confirmación insertando nuevamente la matricula del cliente. Al introducir esta de forma correcta aparecerá una pantalla con los datos almacenados referentes al curso.

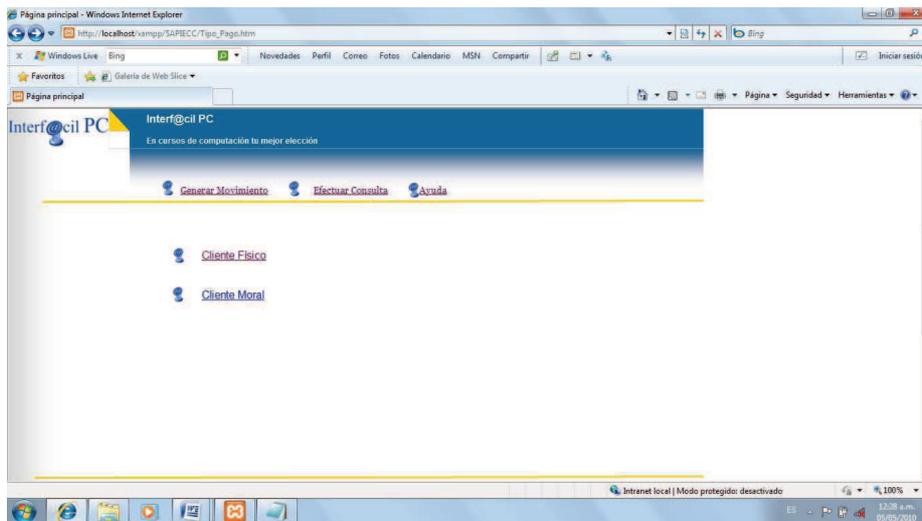
Todas las interfaces de este comando tienen acceso a los comandos de alta cliente, búsqueda cliente y salir (regresar a la página principal). Esto tratando de facilitar al usuario el acceso a estos movimientos, en caso de ser necesarios.

3.3 PAGAR CURSO

Existen dos formas de realizar el pago de un curso estas corresponden a notas de pago y facturas. El caso en que se generara cada uno de ellos corresponde al tipo de cliente, por lo que la primera pantalla que aparecerá será la correspondiente a tipo de cliente.

Para realizar el pago de un curso se deben seguir los siguientes pasos:

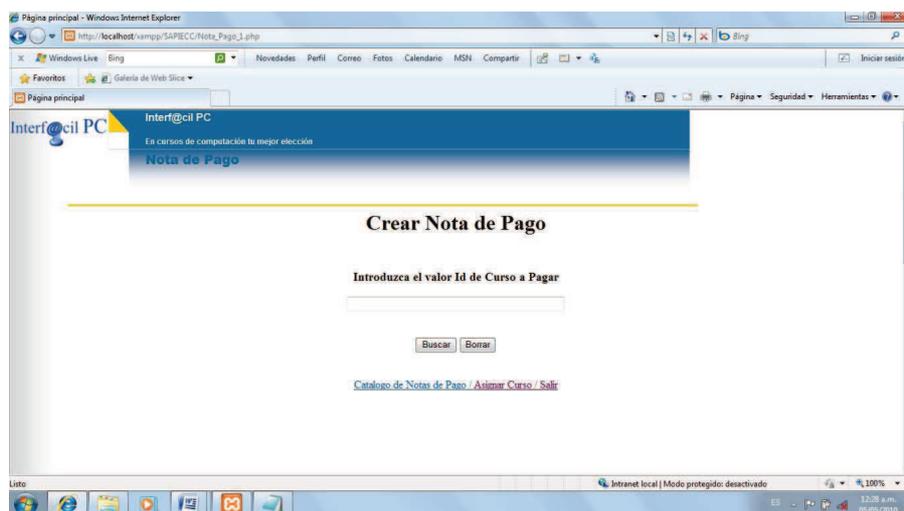
1. En la pantalla principal, dar clic en Generar Movimiento.
2. Dar clic en Pagar Curso.
3. Elegir el tipo de cliente que va a realizar el pago.



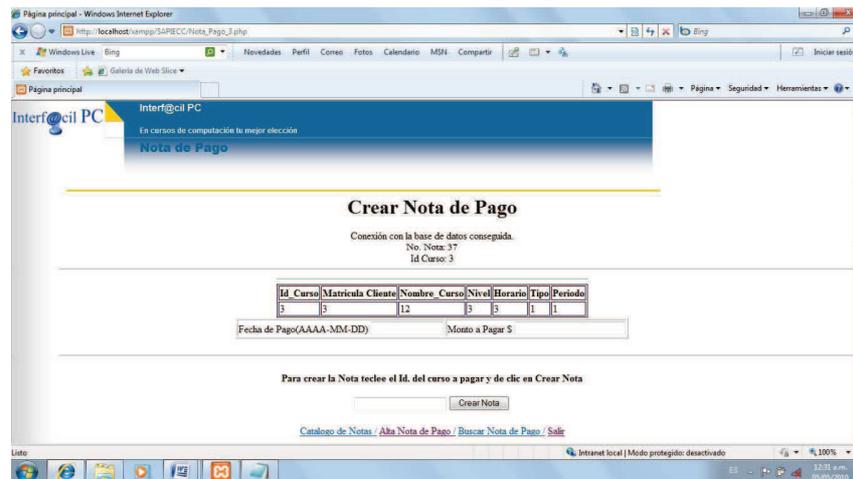
3.3.1 CREAR NOTA DE PAGO

Para generar la nota de pago correspondiente a un cliente físico se deben seguir los siguientes pasos:

1. En la pantalla principal, dar clic en Generar Movimiento.
2. Dar clic en Pagar Curso.
3. Elegir Cliente Físico.
4. Introducir el Id_Curso correspondiente al curso que se desea pagar.
5. Dar clic en buscar.



En caso de existir aparecerá una pantalla con los datos correspondientes al curso que se desea pagar. En caso contrario aparecerá una pantalla indicando que el curso no existe.



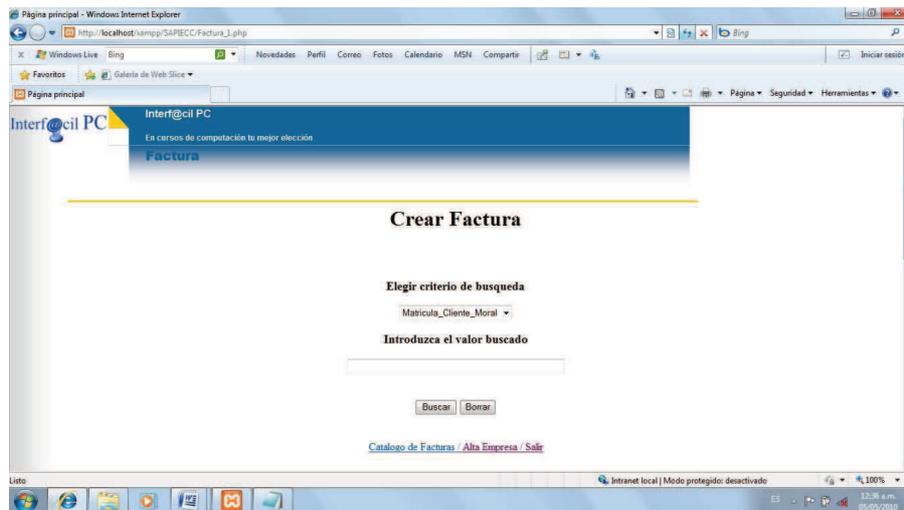
Como ya se menciono en caso de que la asignación de curso exista aparecerá una pantalla donde se solicita la fecha de pago y el monto a pagar. Además de una confirmación del pago a través del Id_Curso. Si el id_curso coincide se genera la nota de pago y se muestra en pantalla los datos de la nota de pago que se acaba de generar. Mientras que si no se introduce correctamente se indica esto en pantalla.

3.3.2 CREAR FACTURA

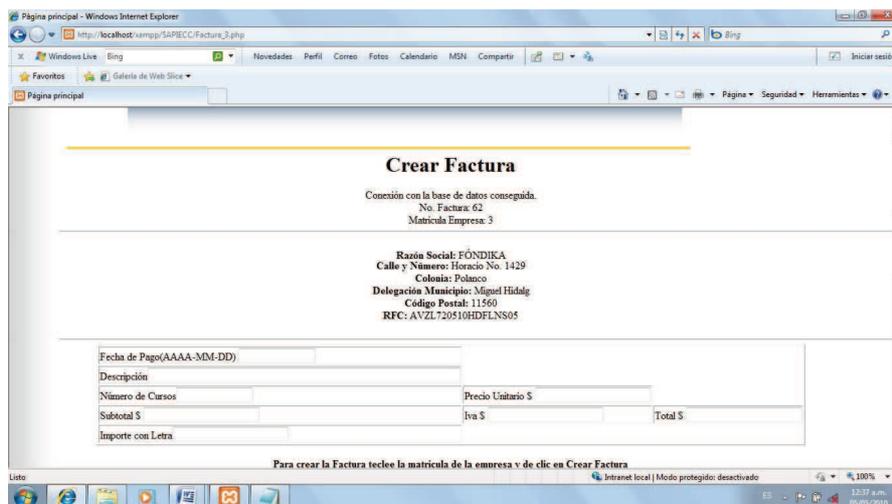
En el caso del que el pago corresponda a un cliente moral o empresa, es necesario que antes de generar la factura se realice la búsqueda del cliente dentro de la base de datos para esto se requerirá introducir el criterio de búsqueda y el valor buscado. La búsqueda se puede realizar mediante la matricula de la empresa o bien a través de la Razón social.

Para genera la factura correspondiente al pago de algún un cliente moral se deben seguir los siguientes pasos:

1. En la pantalla principal, dar clic en Generar Movimiento.
2. Dar clic en Pagar Curso.
3. Elegir Cliente Moral.
4. Elegir el criterio de búsqueda (Matricula ó Razón Social).
5. Introducir el valor buscado.
6. Dar clic en buscar.



En caso de que el cliente moral exista, aparecerá la pantalla para crear la factura, la cual además de mostrar sus datos generales permitirá introducir la Fecha de Pago, el número de cursos a pagar, su descripción, precio unitario, además del importe con letra.



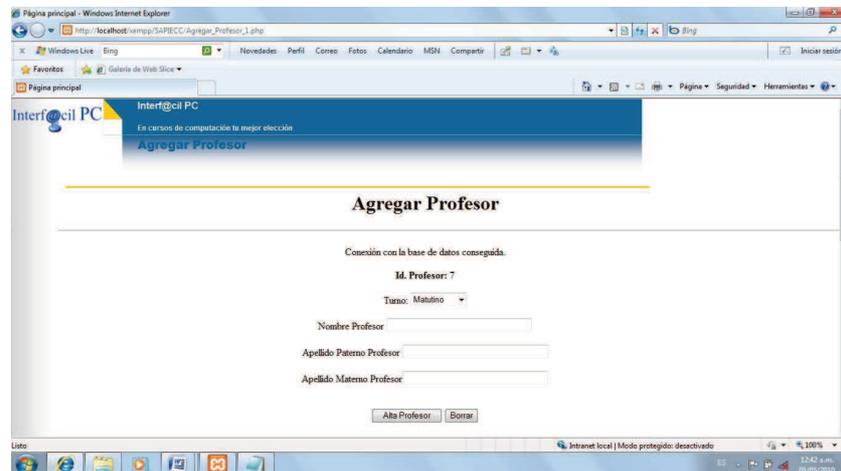
Una vez indicados estos datos es necesario realizar una confirmación insertando nuevamente la matricula del cliente moral. Al introducir esta de forma correcta aparecerá una pantalla con los datos almacenados referentes a la factura.

Es importante mencionar que en el caso de que se haya introducido más de un dato de manera incorrecta aparecerá el mensaje de error correspondiente al primero que detecte el sistema.

3.4.1 AGREGAR PROFESOR

Para genera dar de alta un profesor en el sistema se deben seguir los siguientes pasos:

1. En la pantalla principal, dar clic en Generar Movimiento.
2. Dar clic en Agregar Profesor.



Existen tres opciones que se pueden elegir en esta pantalla. Enseguida se describe su funcionamiento:

OPCIÓN	RESULTADO
Alta Profesor	Almacena en la tabla profesor los resultados insertados en el formulario correspondiente.
Buscar Profesor	Despliega la pantalla que permite introducir parámetros para realizar diferentes tipos de búsqueda dentro de la tabla Profesor.
Eliminar Profesor	Despliega la pantalla destinada a la eliminación de Profesores.

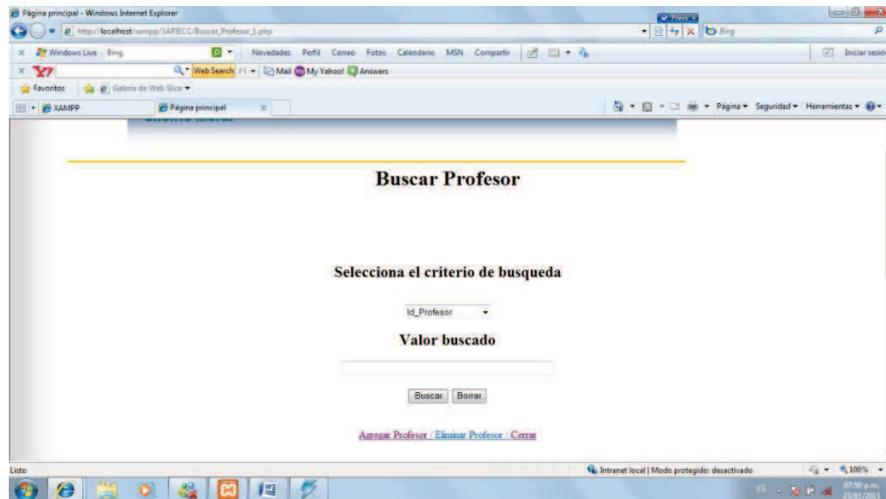
Enseguida se describen algunas características para cada uno de estos casos.

3.4.2 BUSCAR PROFESOR

Para realizar la búsqueda de un profesor en el sistema se deben seguir los siguientes pasos:

1. En la pantalla principal, dar clic en Generar Movimiento.
2. Dar clic en Agregar Profesor.
3. Dar clic en buscar Profesor
4. Elegir el criterio de búsqueda (Id ó Nombre).

5. Introducir el valor buscado.
6. Dar clic en buscar.



Aparecerá una pantalla con los datos correspondientes al registro buscado. Ó bien el mensaje indicando que no se encuentra el registro buscado en el sistema.

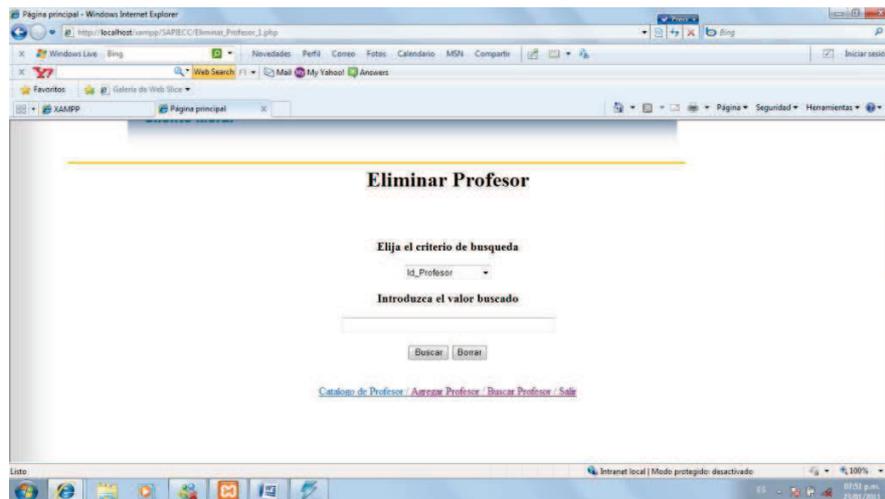
La opción borrar, limpia el contenido de cada una de las cajas de texto del formulario. Mientras que las opciones de Alta Profesor y Eliminar Profesor llevan a las pantallas respectivas de cada movimiento. Y la opción de cerrar lleva a la pantalla principal del sistema.

3.4.3 ELIMINAR PROFESOR

Para que el sistema pueda eliminar un profesor de la base de datos primero es necesario realizar la búsqueda de la persona que se desea quitar y posteriormente es necesario confirmar que se desea eliminar definitivamente de la base de datos del sistema. Por lo que para los datos que se necesitan son el criterio de búsqueda y el valor buscado.

Para eliminar un profesor en el sistema se deben seguir los siguientes pasos:

1. En la pantalla principal, dar clic en Generar Movimiento.
2. Dar clic en Profesor.
3. Dar clic en Eliminar Profesor
4. Elegir el criterio de búsqueda (Id ó Nombre).
5. Introducir el valor buscado.
6. Dar clic en buscar.

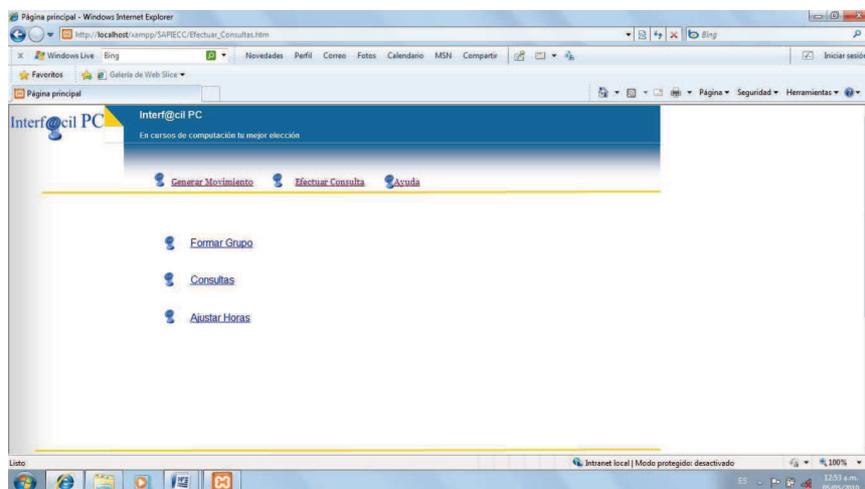


Aparecerá una pantalla con los datos correspondientes al registro buscado. Ó bien el mensaje indicando que no se encuentra el registro buscado en el sistema. En caso de no existir la pantalla muestra las opciones para realizar los diferentes movimientos correspondientes a profesores. Mientras que en el caso de ser correctos es necesario volver a introducir la matricula del profesor para confirmar que se desea eliminar del sistema.

Si la matricula coincide se elimina el registro del sistema y muestra en pantalla los datos del registro que se acaba de eliminar.

4 CONSULTAS DEL SISTEMA

Para ejecutar alguna de las consultas que se encuentran programadas en el sistema es necesario que en la pantalla principal del sistema se elija la opción Efectuar Consulta.



Existen tres opciones a considerar en esta pantalla. Para elegir cada una de ellas solo es necesario dar clic sobre el hipervínculo correspondiente:

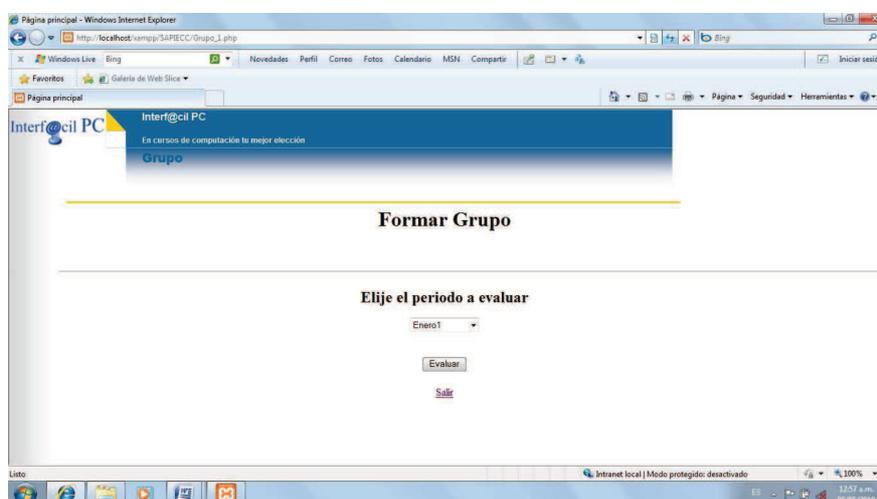
OPCIÓN	RESULTADO
Formar Grupo	Despliegue la pantalla donde el usuario puede formar un grupo.
Consultas	Despliega la pantalla donde el usuario puede visualizar las consultas programadas de SAPIECC.
Ajustar Horas	Despliega el módulo destinado al ajuste de horas.

4.1 FORMAR GRUPO

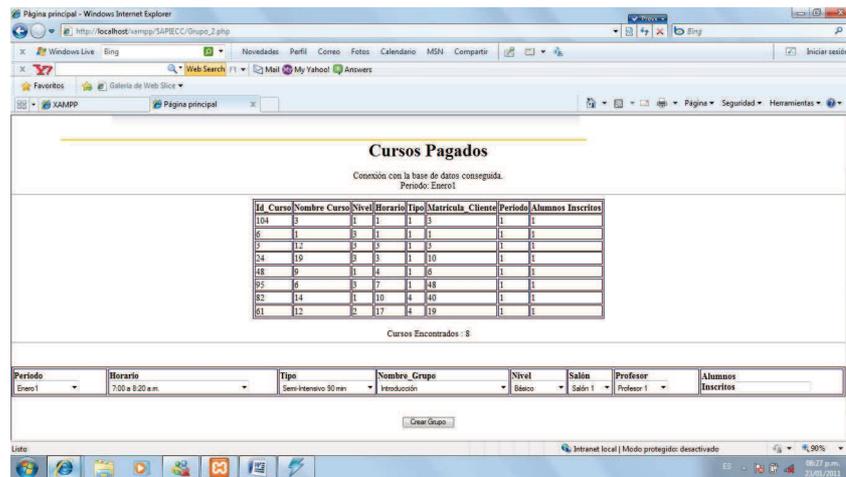
Este comando permite al usuario del sistema formar los grupos de cada periodo.

Para formar un grupo se deben seguir los siguientes pasos:

1. En la pantalla principal, dar clic en Efectuar Consulta.
2. Dar clic en Formar Grupo.
3. Elegir el periodo a evaluar.
4. Dar clic en evaluar.



Aparecerá una pantalla en la que se indicara el total de cursos pagados durante el periodo así como el número de alumnos inscritos en cada grupo. Si este es mayor o igual a tres o bien es de tipo cuatro (Intensivo) el usuario deberá confirmar los datos en la sección inferior y dar clic en crear grupo. Esto almacenara los datos respectivos en la tabla grupo.

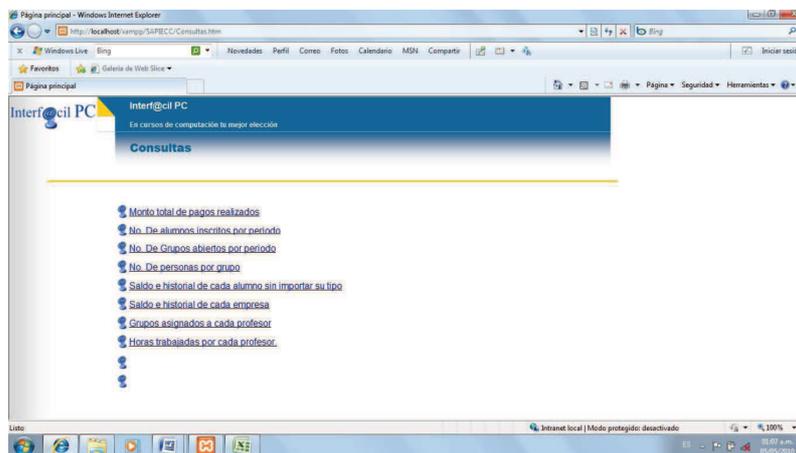


Este comando presenta además las opciones para realizar un análisis nuevo y para regresar a la pantalla principal.

4.2 CONSULTAS

Para formar un ejecutar una consulta se deben seguir los siguientes pasos:

1. En la pantalla principal, dar clic en Efectuar Consulta.
2. Dar clic en Consultas.
3. Elegir la consulta que se desea ejecutar.



Existen varias opciones a considerar en esta pantalla. Para elegir cada una de ellas solo es necesario dar clic sobre el hipervínculo correspondiente:

OPCION	RESULTADO
Monto Total de Pagos Realizados	Muestra el reporte de los pagos realizados durante

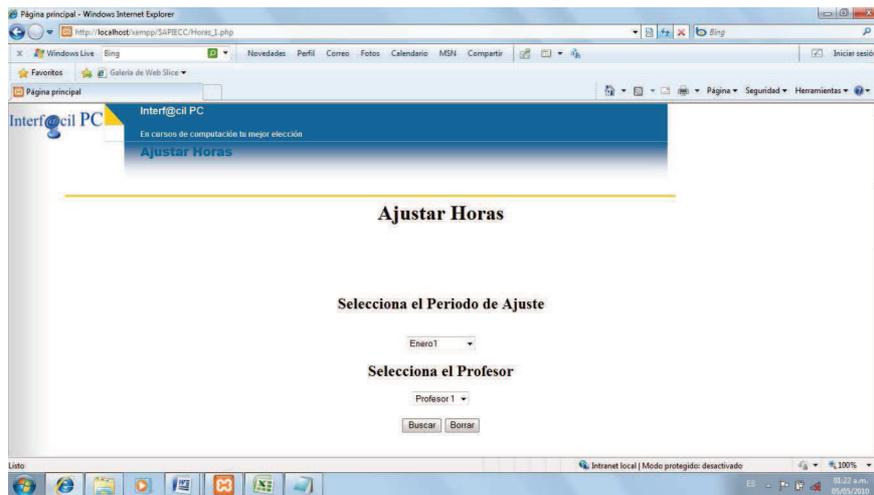
	el periodo elegido divididos en Notas de Pago y Facturas. Incluyendo información correspondiente a cada uno de estos movimientos
No. De Alumnos Inscritos por Periodo	Despliega el reporte de los alumnos inscritos durante el periodo elegido.
No. De Grupos Abiertos por Periodo	Despliega el reporte con toda la información referente a los grupos abiertos durante el periodo elegido por el usuario.
No. De Personas por Grupo	Despliega el reporte con toda la información referente a los grupos abiertos durante el periodo elegido por el usuario. Incluyendo el número de alumnos inscritos en cada grupo.
Historial Alumnos	Despliega el reporte correspondiente a los pagos realizados por el alumno correspondiente a la matrícula insertada por el usuario del sistema.
Historial Empresas	Despliega el reporte correspondiente a los pagos realizados por la empresa correspondiente a la matrícula insertada por el usuario del sistema.
Grupos Asignados a cada Profesor	Despliega el reporte correspondiente a los grupos asignados al profesor correspondiente al Id insertado por el usuario del sistema.
Horas Trabajadas por Profesor	Despliega el reporte correspondiente a los grupos asignados y las horas trabajadas por el profesor correspondiente al Id insertado por el usuario del sistema.

Todos estos movimientos tienen las opciones para realizar un nuevo análisis y para regresar a la pantalla principal.

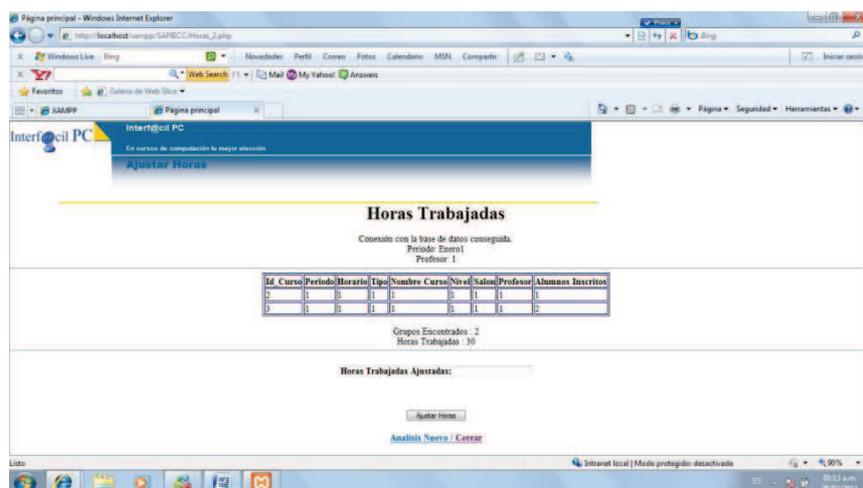
4.3 AJUSTAR HORAS

Para formar un realizar el ajuste de horas trabajadas por un profesor durante un periodo determinado se deben seguir los siguientes pasos:

1. En la pantalla principal, dar clic en Efectuar Consulta.
2. Dar clic en Ajustar Horas.
3. Elegir el periodo de Ajuste
4. Elegir el profesor
5. Dar clic en buscar.



Aparecerá una pantalla en donde muestra los cursos impartidos por el profesor en el periodo elegido. Si se desea modificar el número de horas trabajadas solo es necesario introducir la cantidad en el campo horas ajustadas y dar clic en el botón ajustar horas.

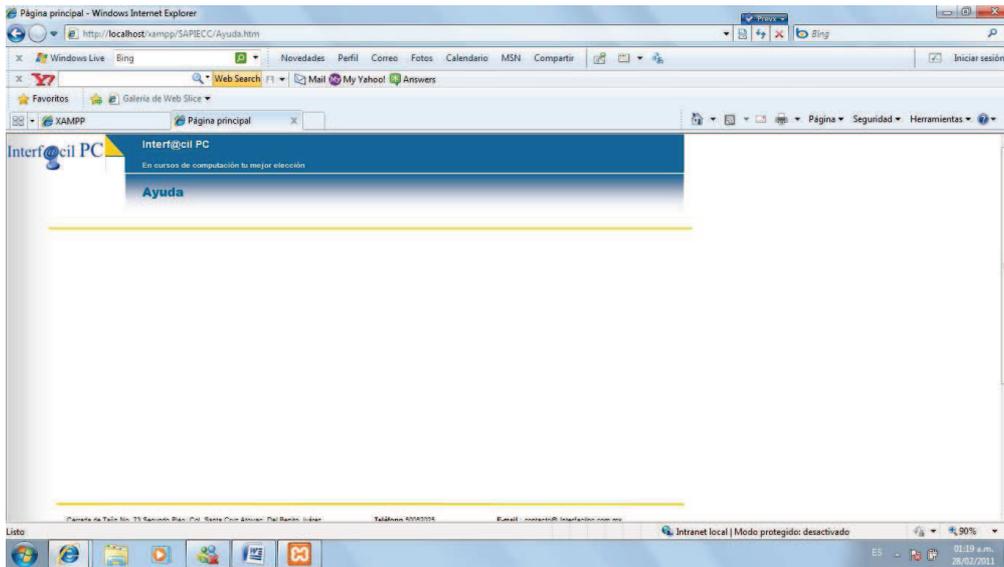


Es importante mencionar que este modulo cuenta además con opciones para realizar nuevos ajustes y para regresar a la pantalla principal.

5 AYUDA

Para formar acceder al módulo de ayuda del sistema deben seguir los siguientes pasos:

1. En la pantalla principal, dar clic Ayuda.
2. Dar clic en el tema que se desea consultar.



BIBLIOGRAFÍA

Pressman, Roger S., “Ingeniería de Software, Un Enfoque Práctico”, McGraw Hill, Quinta Edición, E.E.U.U. 1992.

Chezzi, Carlo; Jazayen, Mehde y Mandrioli, Dino, “Fundamentos de Ingeniería de Software”, Prentice Hall, E.E.U.U , 1991.

Yourdon, Edward, “Ingeniería de Software Clásica”, Yourdon Press, E.E.U.U., 1979.

López, Quijado José, “Domine PHP y MySQL, Programación Dinámica en el Lado del Servidor”, Alfaomega, Primera Edición, México, 2007.

Gil, Rubio Francisco Javier; Tejedor Cerbel Jorge, “ Creación de Sitios Web con PHP4”, McGraw Hill, Primera Edición, España 2001.

Meloni, Julie C., “PHP Fast & Easy Web Development”, Premier Press, Segunda Edición, E.E.U.U. 2002.

Date, C. J. “Introducción a los Sistemas de Bases de Datos”, Addison Wesley, E.E.U.U., Primera Edición 1993.

Korta Henry F, Silvertihalz, “Fundamentos de Bases de Datos”, McGraw Hill.

Minera, José Francisco, “PHP y MySQL: Manual de Programación USERS .CODE “, MP Ediciones S.A., Primera Edición, Argentina 2005.

Fuentes electrónicas.

<http://www.webtaller.com/construccion/lenguajes/php/lecciones/conexion-con-mysql-php.php>

<http://www.elwebmaster.com/editorial/taller-de-php-mysql>

<http://www.wikepeando.com/index.php/archives/410>

http://www.cemev.gob.mx/software_libre/html/libre.html

<http://www.softwarelibre.cl/drupal//files/32693.pdf>

http://es.wikipedia.org/wiki/Programaci%C3%B3n_estructurada

http://es.wikipedia.org/wiki/Programaci%C3%B3n_modular

http://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos

http://es.wikipedia.org/wiki/Programaci%C3%B3n_declarativa

http://es.wikipedia.org/wiki/Consola_de_sistema

http://es.wikipedia.org/wiki/Interfaz_gr%C3%A1fica_de_usuario

http://es.wikipedia.org/wiki/Interfaz_de_usuario

<http://www.tucumanlinux.com.ar/porque/diferencias.html>

<http://www.aulaclie.es/articulos/licencias.html>

http://es.wikipedia.org/wiki/C%C3%B3digo_libre

http://es.wikipedia.org/wiki/Software_no_libre

http://es.wikipedia.org/wiki/Microsoft_Access

http://es.wikipedia.org/wiki/Microsoft_SQL_Server

http://es.wikipedia.org/wiki/Apache_Derby

http://es.wikipedia.org/wiki/Berkeley_DB

<http://es.wikipedia.org/wiki/MySQL>

<http://es.wikipedia.org/wiki/SQLite>

<http://es.wikipedia.org/wiki/PHP>

<http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>

<http://www.todoexpertos.com/categorias/tecnologia-e-internet/bases-de-datos/oracle/respuestas/14706/vetajas-y-desventajas>