

**UNIVERSIDAD DE EL SALVADOR  
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE  
DEPARTAMENTO DE INGENIERIA Y ARQUITECTURA.**



**TRABAJO DE GRADUACIÓN**

**TEMA:**

**“DISEÑO Y CREACIÓN DE UN SISTEMA DE INFORMACIÓN PARA  
LA CLÍNICA DE ODONTOLOGÍA DE LA FACULTAD  
MULTIDISCIPLINARIA DE OCCIDENTE”**

**PARA OPTAR AL GRADO DE:  
INGENIERO DE SISTEMAS INFORMATICOS.**

**PRESENTADO POR:  
MORENO, DENIS ARMANDO  
SALINAS MINEROS, JAIME ALBERTO**

**DOCENTE DIRECTOR:  
ING. ERNESTO ALEXANDER CALDERÓN PERAZA**

**SEPTIEMBRE DE 2008**

**SANTA ANA      EL SALVADOR      CENTRO AMERICA**

**FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE**

**DECANO**

**LICDO. JORGE MAURICIO RIVERA**

**VICEDECANO**

**LICDO. Y MASTER ELADIO EFRAIN ZACARÍAS**

**SECRETARIO DE LA FACULTAD**

**LICDO. VICTOR HUGO MERINO QUEZADA**

**JEFE DEL DEPARTAMENTO DE INGENIERIA Y  
ARQUITECTURA**

**ING. RAÚL ERNESTO MARTÍNEZ BERMUDEZ**

**UNIVERSIDAD DE EL SALVADOR**

**RECTOR**

**LICDO Y MSC RUFINO ANTONIO QUEZADA SÁNCHEZ**

**VICERRECTOR ACADÉMICO**

**ARQ. Y MASTER MIGUEL ANGEL PÉREZ RAMOS**

**VICERRECTOR ADMINISTRATIVO**

**LICDO. Y MATER OSCAR NOE NAVARRETE**

**SECRETARIO GENERAL**

**LICDO. DOUGLAS VLADIMIR ALFARO CHÁVEZ**

**FISCAL GENERAL**

**DR. RENÉ MADECADEL PERLA JIMÉNEZ**



**TRABAJO DE GRADUACIÓN APROBADO POR**

---

**ING. ERNESTO ALEXANDER CALDERÓN PERAZA**

**DOCENTE DIRECTOR**

## CONTENIDO

Págs.

<b>CAPÍTULO I</b>	<b>8</b>
<b>1 GENERALIDADES Y PLANTEAMIENTO DEL PROBLEMA</b>	<b>8</b>
<b>1.1 INTRODUCCIÓN</b>	<b>9</b>
1.1.1 NATURALEZA Y PROPOSITO DE LA CLINICA DE ODONTOLOGIA DE LA FMO.	10
1.1.2 SITUACION ACTUAL DE LA CLINICA	10
<b>1.2 DEFINICION Y PLANTEAMIENTO DEL PROBLEMA.</b>	<b>20</b>
1.2.1 PUNTOS FUNDAMENTALES IDENTIFICADOS EN EL PROBLEMA.	20
<b>1.3 OBJETIVOS</b>	<b>22</b>
<b>1.4 JUSTIFICACIÓN</b>	<b>23</b>
<b>1.5 METODOLOGÍA DE INVESTIGACION</b>	<b>24</b>
<b>MÉTODOS DE RECOPIACIÓN DE INFORMACIÓN</b>	<b>24</b>
<b>1.6 METODOLOGIA DE DESARROLLO</b>	<b>28</b>
<b>INVESTIGACIÓN PRELIMINAR Y PLANIFICACIÓN</b>	<b>30</b>
<b>IDENTIFICACIÓN DE LOS REQUERIMIENTOS</b>	<b>30</b>
<b>DISEÑO DE BASE DE DATOS E INTERFACES</b>	<b>30</b>
<b>CAPÍTULO II</b>	<b>32</b>
<b>2 IDENTIFICACIÓN DE NECESIDADES Y ANÁLISIS DE LOS REQUERIMIENTOS.</b>	<b>32</b>
<b>2.1 DIAGNOSTICO DE LA SITUACION INICIAL DE LA CLINICA DE ODONTOLOGIA.</b>	<b>33</b>
<b>2.2 DESCRIPCION DE LAS AREAS FUNCIONALES INCLUIDAS EN EL SISTEMA FINAL.</b>	<b>34</b>
<b>2.3 DESCRIPCION DE LOS MODULOS NECESARIOS.</b>	<b>37</b>
<b>2.4 ESTUDIOS DE FACTIBILIDAD</b>	<b>41</b>
2.4.1 ESTUDIO DE FACTIBILIDAD TÉCNICA	41
2.4.2 ESTUDIO DE FACTIBILIDAD ECONÓMICA.	43
<b>2.5 ALCANCES DE LOS MÓDULOS DEL SISTEMA.</b>	<b>52</b>
<b>2.6 REQUERIMIENTOS FUNCIONALES</b>	<b>54</b>
2.6.1 REQUERIMIENTOS DE HARDWARE EN PC'S CLIENTES:	54
2.6.2 REQUERIMIENTOS DE SOFTWARE PARA SERVIDOR Y COMPUTADORAS CLIENTES:	55
2.6.3 CONDICIONES RECOMENDADAS DE FUNCIONAMIENTO	55
<b>3 DISEÑO DEL SISTEMA DE INFORMACIÓN</b>	<b>56</b>

<b>3.1</b>	<b>DESCRIPCIÓN GENERAL DEL SISTEMA DE INFORMACIÓN</b>	<b>57</b>
3.1.1	DISEÑO DE LOS MODULOS.	57
3.1.2	DISEÑO DE PROTOTIPOS DE LA APLICACIÓN Y ACCESIBILIDAD DE LOS MODULOS.	57
3.1.3	ARQUITECTURA DEL SISTEMA	63
3.1.4	DISEÑO DE ENTRADAS DE DATOS AL SISTEMA	64
3.1.5	DISEÑO DE SALIDAS DEL SISTEMA	66
3.1.6	FACTIBILIDAD OPERATIVA	67
<b>3.2</b>	<b>DIAGRAMACIÓN DEL SISTEMA</b>	<b>69</b>
3.2.1	DIAGRAMA DE CLASES.	69
3.2.2	DIAGRAMAS DE CASOS DE USO.	71
3.2.3	DIAGRAMAS DE ACTIVIDADES	73
<b>3.3</b>	<b>DISEÑO DE LA BASE DE DATOS</b>	<b>78</b>
3.3.1	DESCRIPCIÓN DEL SISTEMA GESTOR DE BASES DE DATOS	78
3.3.2	ARQUITECTURA DE LA BASE DE DATOS	80
3.3.3	CONECTIVIDAD DE LA BASE DE DATOS	83
3.3.4	SEGURIDAD DE LA BASE DE DATOS	84
3.3.5	MODELO ENTIDAD RELACIÓN	86
3.3.6	DESCRIPCIÓN DE LAS TABLAS PRINCIPALES Y DICCIONARIO DE DATOS DEL SISTEMA	92
<b>CAPÍTULO IV</b>		<b>115</b>
<b>4</b>	<b>DESARROLLO DE LA APLICACIÓN</b>	<b>115</b>
<b>4.1</b>	<b>CODIFICACIÓN</b>	<b>116</b>
<b>4.2</b>	<b>INTEGRACIÓN DE LOS ELEMENTOS Y DEPURACIÓN</b>	<b>175</b>
<b>4.3</b>	<b>ELABORACIÓN DEL PLAN DE IMPLEMENTACIÓN Y DOCUMENTACIÓN</b>	<b>176</b>
<b>4.4</b>	<b>PLAN DE IMPLEMENTACION</b>	<b>176</b>
4.4.1	MIGRACIÓN DE DATOS	178
4.4.2	INSTALACIÓN Y CONFIGURACIÓN DEL PROGRAMA	179
4.4.3	CAPACITACIÓN AL PERSONAL DE LA CLÍNICA	180
<b>CAPÍTULO V</b>		<b>182</b>
<b>5</b>	<b>RESULTADOS ESPERADOS, CONCLUSIONES Y RECOMENDACIONES</b>	<b>182</b>
<b>5.1</b>	<b>RESULTADOS ESPERADOS</b>	<b>183</b>
<b>5.2</b>	<b>CONCLUSIONES</b>	<b>184</b>
<b>5.3</b>	<b>RECOMENDACIONES</b>	<b>184</b>
<b>5.4</b>	<b>ANEXOS</b>	<b>186</b>

## **CAPÍTULO I**

### **1 GENERALIDADES Y PLANTEAMIENTO DEL PROBLEMA**

## 1.1 INTRODUCCIÓN

Las clínicas de odontología de la Universidad de El Salvador, representan un gran beneficio tanto para la comunidad universitaria como para personas particulares que al no poder acceder a los servicios de salud bucal en las instituciones respectivas, se presentan a estas clínicas buscando poder recibir el servicio.

A pesar de que el personal que realiza los diagnósticos de tratamiento son todavía estudiantes universitarios, la satisfacción del servicio recibido es notable en las personas, en gran parte debido a la supervisión que realizan los doctores directores en cada clínica.

Sin embargo el potencial benéfico que tienen estas clínicas muchas veces puede verse aminorado por diversos factores no controlables por el personal de la clínica, tales como escasez de materiales, y otros factores mas relacionados al área operativa de la clínica como por ejemplo la forma de archivar los expedientes de los pacientes.

Es por esta razón que se decide tomar el reto de incorporar a la clínica de la facultad de Occidente una herramienta que venga a colaborarles en la superación de los obstáculos operativos que les causan pérdida de tiempo o de información relevante para la atención a los pacientes.

El presente proyecto surge de la necesidad mostrada por el personal de la clínica de mejorar el manejo que se hace actualmente de los datos y con ello buscar hacer más eficiente la atención de pacientes y las actividades cotidianas de la clínica.

Se busca implementar un sistema de información completo que sirva de base operativa de la atención a pacientes, asignación de citas, manejo de inventarios y la mayoría de actividades diarias, al final de este proyecto se espera haber cumplido con las expectativas generadas en el personal de la clínica y contribuir de alguna manera al mejoramiento de una de las entidades de servicio a la comunidad que la universidad posee.

## **1.1.1 NATURALEZA Y PROPOSITO DE LA CLINICA DE ODONTOLOGIA DE LA FMO.**

### **1.1.1.1 ANTECEDENTES**

La clínica de odontología de la Facultad Multidisciplinaria de Occidente de la Universidad de El Salvador brinda sus servicios desde julio de 1990 y nace a raíz de un objetivo de integración de componentes de Servicio, Docencia e investigación propuesto en el cambio curricular que se gestó en el año 1987

Desde sus inicios la clínica cubría áreas de atención a sus pacientes respecto a odontología preventiva, pero con el pasar de la década ha logrado cubrir muchas más áreas tales como: Diagnóstico, periodoncia, restaurativa, Endodoncia y cirugías. Aquellas áreas que no se atienden por carecer de personal especializado o instrumentación específica, generalmente, se remiten a hospitales como Hospital San Juan de Dios.

Actualmente el área de Prostodoncia y Ortodoncia son, en la mayoría de las ocasiones, remitidos con especialistas reconocidos.

La clínica fue cerrada en ciclo I de 2003, como efecto provocado de un paro de labores en la Universidad de El Salvador. Esta es el único periodo en el que la clínica ha cesado sus labores en todos sus años de funcionamiento. El 7 de febrero del mismo año, la Junta Directiva tomó Acuerdo No. 23 para realizar un análisis exhaustivo en relación al funcionamiento académico administrativo de la clínica de Santa Ana.

Desde sus inicios, la clínica nunca ha contado con la ayuda de una herramienta informática para manipular y registrar la información que brinde reportes necesarios que simplifiquen la toma de decisiones y que agilicen los procesos.

## **1.1.2 SITUACION ACTUAL DE LA CLINICA**

La clínica brinda atención a sus pacientes con la ayuda de entre 10 y 20 estudiantes de la Facultad de Odontología que rotan en sus funciones cada 11 o 10 días. Los registros

de los expedientes de los pacientes se llevan, en parte, en hojas electrónicas en las computadoras existentes en la entidad y en formularios impresos aprobados por las autoridades respectivas de San Salvador.

Entre la variedad de formularios para recopilación y registro de información se encuentran los formularios improvisados, elaborados por el personal encargado del funcionamiento de la clínica, entre el que se encuentra la Dra. Lisseth López, y la Sra. Maritza Sermeño.

Periódicamente EL equipo y medicamento que es utilizado en la clínica se ve evaluado por personal de San Salvador.

La clínica brinda sus servicios, en horario diario de 8:00 a.m. 12:00 a.m. y de 1:00 a 4:00pm. El volumen de pacientes atendidos en la clínica entre los que se encuentran niños provenientes de las escuelas de atención comunitaria, y particulares, pacientes adultos, y estudiantes de la universidad, puede apreciarse en la siguiente información tabular:

**1.1.2.1 CUADRO PACIENTES ATENDIDOS Y TRATAMIENTOS REALIZADOS POR UNA POBLACIÓN DE 72 ESTUDIANTES PROGRAMADOS EN SANTA ANA; AÑO 2007**

<b>SEXO</b>	<b>SANTA ANA</b>
MASCULINO	108
FEMENINO	162

**TRATAMIENTOS REALIZADOS PACIENTES ADULTOS CICLO I-2007**

<b>TRATAMIENTO</b>	<b>CANTIDAD</b>
DIAGNOSTICO	41
RASPAJE	36
ALISADO	8
PROFILAXIS	16
OBT. Cl. I	25
OBT. Cl. II	35
OBT. Cl. III	11
OBT. Cl. IV	2
OBT. Cl. V	4
SS FF	60
TCR MONO	23

TCR MULTI	22
RECONSTRUCCIONES	27
PINES PARAPOST	7
EXODONCIAS	6
BLANQUEAMIENTOS	1
EMERGENCIAS	19
TOMA RX	10
ALARGAMIENTO DE CORONA	1
CEMENTACIÓN	5
<b>TOTAL</b>	<b>359</b>

### TRATAMIENTOS REALIZADOS PACIENTES ADULTOS CICLO II-2007

TRATAMIENTO	CANTIDAD
DIAGNOSTICO	42
RASPAJE	44
ALISADO	6
PROFILAXIS	27
OBT. CL. I	52
OBT. CL. II	30
OBT. CL. III	12
OBT. CL. IV	5
OBT. PREV.	5
OBT. CL. V	4
SS FF	70
TCR MONO	18
TCR MULTI	44
RECONSTRUCCIONES	52
PINES PARAPOST	14
EXODONCIAS	13
EMERGENCIAS	39
TOMA RX	6
CEMENTACIÓN	3
<b>TOTAL</b>	<b>486</b>

### ATENCIÓN A NIÑOS:

SEXO	SANTA ANA
MASCULINO	34
FEMENINO	23

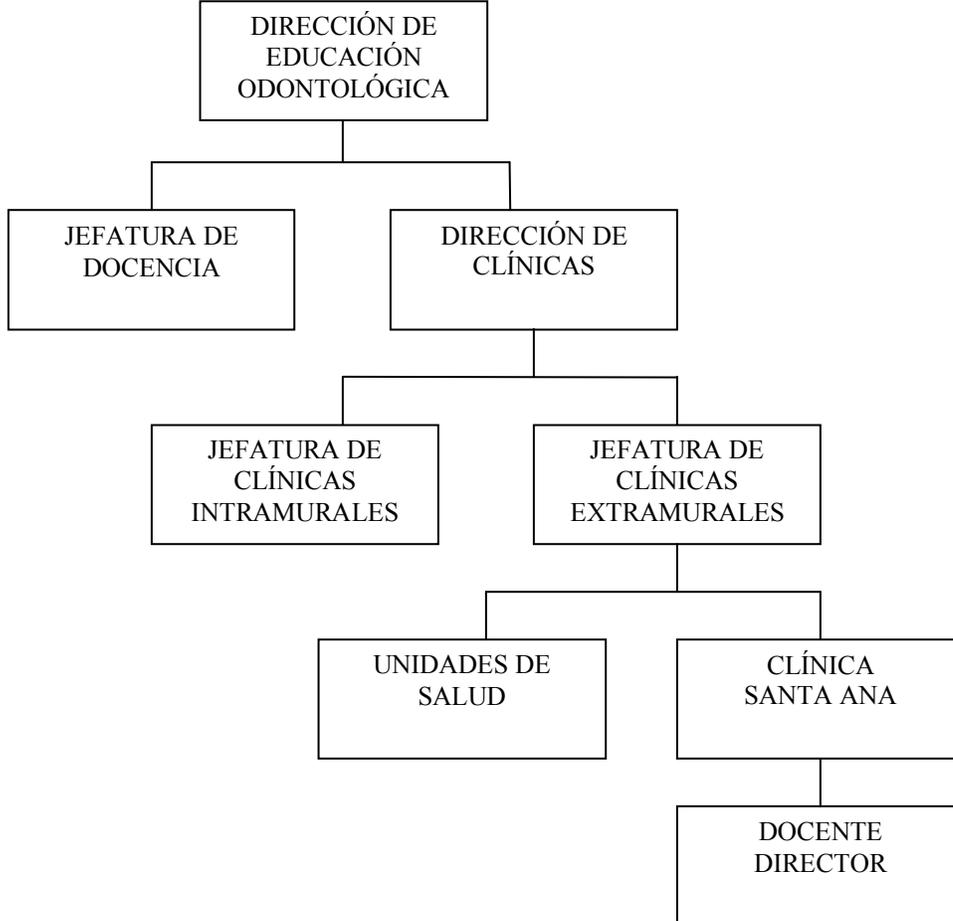
NIÑOS ATENDIDOS = 57

**TRATAMIENTOS REALIZADOS PACIENTES NIÑOS CICLO I-2007**

<b>TRATAMIENTOS</b>	<b>CANTIDAD</b>
DIAGNOSTICO	8
PROFILAXIS	2
OBT. CI. PREV.	5
OBT. CI. I.	2
OBT. CI. II	10
OBT. CI. III	1
SELLANTES FOSAS Y FISURAS	24
PULPOTOMIA	1
PULPECTOMIA	6
TCR MONO	1
TCR MULTI	2
PIN PARAPOST	3
RECONSTRUCCIONES	7
APEXOGENESIS	1
APEXIFICACION	1
EXODONCIAS	10
EMERGENCIAS	2
<b>TOTAL</b>	<b>86</b>

**TRATAMIENTOS REALIZADOS PACIENTES NIÑOS CICLO II-2007**

<b>TRATAMIENTO</b>	<b>CANTIDAD</b>
DIAGNOSTICO	12
PROFILAXIS	9
RASPAJE	1
OBT. CI. I	5
OBT. CI. II	16
OBT. CI. III	5
OBT. CI. V	1
OBT. PREV.	3
SELLANTES FOSAS Y FISURAS	31
PULPOTOMIA	4
PULPECTOMIA	23
TCR MULTI	1
RECONSTRUCCIONES	25
EXODONCIAS	20
EMERGENCIAS	8
TOMA RX	6



TOTAL	170
-------	-----

En cuanto al equipo de cómputo existente en la clínica solo dos computadoras las cuales no se encuentran en red y además se cuenta con conexión a Internet.

### 1.1.2.2 ESTRUCTURA ORGANIZATIVA



## **PRINCIPALES FUNCIONES DE LAS DIVISIONES DE LA ESTRUCTURA ORGANIZATIVA**

### **FUNCIONES DE DIRECCIÓN DE EDUCACIÓN ODONTOLÓGICA**

- Dirigir la Comisión de Currículum
- Promover talleres curriculares para la evaluación de los procesos de investigación – servicio-docencia.
  - Hacer propuestas ante Comisión de Currículum de todo lo relacionado con lo académico, investigación-servicio-docencia.
  - Gestionar la capacitación continua del recurso docente.

- Dirigir las tesis de graduación
- A través de la jefatura de docencia:
  1. Velar por el cumplimiento de los procesos de enseñanza aprendizaje.
  2. Velar por la calidad de la enseñanza
  3. Velar por que se cumplan los procesos de evaluación y contenidos programáticos.
  4. Gestionar la contratación de docentes asesores; en algunas áreas de la carrera deficitarias de recurso humano calificado.

### FUNCIONES DE DIRECCIÓN DE CLÍNICAS

- Planificar las actividades clínicas de los estudiantes en coordinación con jefatura de clínicas y jefes de área, en base a requisito y tiempo, según contenidos programáticos.
- Consultar a los jefes de área si se han reunido para hacer innovaciones a los documentos utilizados en el proceso de enseñanza aprendizaje en la práctica clínica, y que se proceda a su duplicación.
- Promover el funcionamiento de las clínicas a través de la radio, boletines y periódico.
- Establecer estrategias de control en el gasto de materiales, salidas al laboratorio dental, préstamo de equipo e instrumental de uso clínico.
- Definir funciones al personal administrativo de acuerdo a la demanda estudiantil de cada ciclo.
- Velar por que se aplique el arancel en las clínicas escuela de la facultad.
- Hacer las proyecciones del gasto de materiales en base a existencias en arsenales, bodega de suministros, número de estudiantes programados, nivel de estudiante programado y turnos clínicos.
- Promover ante los jefes de área la innovación de los biomateriales para su conocimiento y utilización y gestionar las compras ante junta directiva.
- Miembro nato en el proceso de licitaciones(convocada por la UACI en cada proceso)

- Coordinar capacitaciones en el manejo de biomateriales para los docentes de cada área.
- Solicitar a las áreas sus necesidades de equipamiento e ir abasteciendo según las disponibilidades económicas.
- Solicitar inventario a los arsenales cada ciclo para un mejor control del gasto de material y del equipo que se presta al estudiante.
- Solicitar inventario a la bodega de suministros 1 a 2 veces en el año.
- Analizar el costo de los materiales utilizados en cada tratamiento para determinar su congruencia con el arancel que se aplica.
- Velar por la calidad de los servicios de laboratorio.
- Velar por que el laboratorio asuma el gasto de repetir trabajos que han fracasado por fallas en el proceso.
- Coordinar reuniones con el personal administrativo de clínicas para afinar detalles en el funcionamiento de cada una de las dependencias, según horario de programación clínica y número de estudiantes programados
  - Promover el trabajo en equipo con el personal administrativo, como apoyo a la calidad de los servicios que se prestan al docente estudiantes y pacientes.
  - Promover la armonía del trabajo mediando en situaciones de crisis.
  - Coordinar con los docentes directores de las clínicas extramurales el abastecimiento de biomateriales según sean sus requerimientos.
  - Entregar a los directores de clínicas extramurales; el arancel y el documento base, aprobado por junta directiva y consejo superior universitario para su aplicación en el cobro de los tratamientos y gastos extras de material.
  - Solicitar a las clínicas extramurales al finalizar cada ciclo el informe de los tratamientos realizados. Para introducirlos en la base de datos y al finalizar el año entregar los informes estadísticos a decanato.
  - Apoyar a los directores de clínicas extramurales en la solución de problemas que surgen en la atención a pacientes y comportamiento del estudiante.
  - Presentar ante Comité Técnico Asesor o junta directiva situaciones que esta dirección da por agotadas en su resolución y pasan a ser competencia de esos organismos.

- Coordinar con otras direcciones las necesidades que surgen en las clínicas extramurales para que sean atendidas.
- Velar porque se conozcan el normativo de funcionamiento de las clínicas escuela.

#### FUNCIONES DEL JEFE DE CLÍNICAS EXTRAMURALES

- Coordinar con dirección de Clínicas la programación de clínicas extramurales.
- Gestionar ante coordinación docente la programación de supervisiones a unidades de salud y clínicas extramurales.
  - Supervisar el funcionamiento de las clínicas extramurales.
  - Informar a Dirección de Clínicas sobre anomalías.
  - Informar a recursos humanos sobre necesidades en el funcionamiento del equipo.
- Coordinar el proyecto preventivo en las escuelas a través de los asesores.
- Participar en las supervisiones a unidades de salud y Clínicas extramurales.

#### FUNCIONES DEL DOCENTE DIRECTOR EN LAS CLÍNICAS EXTRAMURALES

##### EN LO ACADÉMICO

- Velar por la calidad de los tratamientos
- Evaluar los proceso en cada tratamiento
- Actuar como operador en caso de que el estudiante tenga dificultades.
- Velar porque se cumplan las normas de bioseguridad en la clínica
- Velar por el orden y la disciplina dentro del recinto de las clínicas.
- Trabajar en armonía respetando la integridad de estudiantes, pacientes y personal administrativo.

##### EN LO ADMINISTRATIVO

- Velar porque se cumpla lo estipulado en el arancel
- Velar por el gasto de materiales
- Al finalizar cada ciclo; solicitar a la arsenalista el inventario de materiales dentales
- Al finalizar cada ciclo; rendir un informe a Dirección de Clínicas de los tratamientos realizados
  - Rendir un informe a Dirección de Clínicas sobre estudiantes que están en deuda por equipo prestado, o gastos excesivos de materiales.
  - Consultar a Dirección de Clínicas sobre dudas en arancel y otros.
  - Informar a esta Dirección sobre necesidades en cuanto a equipo y materiales.
  - Informar a recursos humanos sobre daños al equipo, instalaciones, reparaciones, etc.

#### FUNCIONES DEL PERSONAL DE COLECTURÍA QUE TAMBIÉN ASUME LAS DE SECRETARIA

- Hacer los cobros apegados al arancel.
- Extender el correspondiente recibo por cada tratamiento que se paga o abono a tratamiento (detallando en el recibo)
  - Aplicar el 50% de descuento solo a niños del programa de escuelas saludables, debiendo presentar el carné, sacar fotocopia del carné para anexar al expediente.
  - Solicitar la papelería correspondiente para el funcionamiento de cada ciclo en Dirección de Clínicas. (Fólder, expedientes, hojas de evaluación de cada una de las áreas, papelería de oficina y otros)
  - Sellar los fólder tamaño oficio y rotularlos según correlativo del expediente/año
  - Rotular los expedientes con nombre y número correlativo por año de acuerdo a lo vendido.
  - Rotular fichas de emergencia en correlativo/año
  - Llevar un archivo de la hoja de compromiso que debe llenar el paciente.

- Llevar en order book, un control en número correlativo de los expedientes y fichas de emergencias vendidas según día y fecha y año.
- Llevar un archivo en orden alfabético de los expedientes.

#### FUNCIONES DE ARSENALISTA

- Revisar el stock de materiales para evitar su vencimiento
- Controlar el préstamo de equipo a través de vales donde el estudiante anota su nombre, cantidad del equipo, nombre del equipo, serie del equipo, fecha de préstamo y firma del bachiller
  - Entregar el material solicitado por el estudiante para la ejecución de los tratamientos según vale de gastos de material.
  - En el caso de los anestésicos manejar el vale solo para anestésicos.

### **1.2 DEFINICION Y PLANTEAMIENTO DEL PROBLEMA.**

El mantenimiento del servicio de la clínica de odontología de la Facultad Multidisciplinaria de Occidente depende del desempeño de su función social, es decir, obtener cifras significativas en cuanto a los tratamientos efectuados y en general, pacientes atendidos. La clínica de Santa Ana es la única que ha mantenido sus servicios de tres clínicas que inicialmente se establecieron a raíz del cambio curricular.

#### **1.2.1 PUNTOS FUNDAMENTALES IDENTIFICADOS EN EL PROBLEMA.**

- Si bien, la cantidad de personas atendidas por la clínica de Santa Ana representa una cifra aceptable, el personal de la clínica ha sostenido que es posible incrementar estos valores en condiciones organizativas mas adecuadas.
- El poco personal administrativo existente en la clínica ralentiza el manejo y registro de información, lo cual genera atrasos en el desempeño tal como la disposición de

un día a la semana para la organización y registro de actividades de la clínica que no se pudieron registrar en los respectivos horarios. Es decir, la mayor parte de este día administrativo, no se atiende pacientes. Lo anterior representa un decremento para la cantidad de pacientes atendidos en el periodo de operaciones.

- Existen formularios improvisados, como ya se ha mencionado, que cumplen con su cometido de no perder la información de los hechos históricos, sin embargo, esta manera de registro y la inexistencia de procesos estandarizados para el tratamiento de esta información dificulta el acceso a la misma y genera problemas de coordinación y administración de los horarios de atención a los pacientes, llegando a situaciones en la que se dificulta la asignación de citas por confusión en los métodos manuales, lo cual ocasiona que no se utilice de manera óptima el tiempo de atención a los pacientes.

- La evaluación de los estudiantes que se desempeñan en la clínica es una función que desempeña el Dr. encargado, función que ejecuta paralelamente a otras funciones administrativas cruciales para el buen funcionamiento de la clínica en general.

- El personal operativo de la clínica en el campo de odontología, no solo se organiza para atender un horario de consulta a pacientes que asisten a la clínica, sino también, actividades que se ejecutan fuera de las instalaciones de la clínica, como en las escuelas. Estas actividades se derivan del programa de prevención comunitaria o PPC, las cuales también se planifican en sus horarios respectivos y de las cuales se lleva un registro detallado.

- Dentro de las funciones del personal director de la clínica, se encuentra la emisión de reportes a las autoridades respectivas, reportes derivados de diferentes áreas administrativas y funcionales de la clínica. La emisión u obtención de tales reportes es una actividad en la que se invierte mucho tiempo y que el resumen de las actividades se elabora a mano, accediendo uno a uno a formularios improvisados. Tal situación puede generar reportes que no sean consistentes con las actividades realmente efectuadas.

## 1.3 OBJETIVOS

### **Objetivo General**

Analizar, diseñar y desarrollar un sistema de información eficiente y eficaz para la clínica de odontología de la Facultad Multidisciplinaria de Occidente de la Universidad de El Salvador, que cubra sus necesidades de información en las principales áreas administrativas y técnicas que se ejecutan en la misma.

### **Objetivos Específicos**

1. Beneficiar a los directores, estudiantes y personal administrativo de la clínica, para que cuenten con una herramienta moderna y disponible en la que fácil y rápidamente puedan acceder a información necesaria en el desarrollo de sus actividades.
2. Desarrollar módulos para cada una de las principales áreas administrativas que se coordinen entre si para proporcionar la mayor comodidad en el registro y acceso a la información.
3. Proveer de un mecanismo de almacenamiento centralizado de información, que permita eliminar la utilización de múltiples fuentes de datos y sus problemas de duplicidad y falta de concordancia.
4. Facilitar la asignación de citas y permitir coordinar de una manera más eficiente las actividades externas a la clínica.
5. Garantizar una seguridad completa, en lo que se refiere a almacenamiento, consulta, modificación y eliminación de la información por parte de los usuarios habilitados para dichas acciones.

## **PRINCIPALES BENEFICIOS QUE REPRESENTA LA HERRAMIENTA INFORMATICA PARA LA CLINICA.**

La disminución de los tiempos dedicados por el personal de la clínica en los aspectos del registro y manipulación de la información representa un aumento de tiempo dedicado para atención y mejoramiento de los servicios comunitarios ofrecidos. Lo anterior representa aumentos en las cifras de pacientes atendidos y tratamientos efectuados por periodo de evaluación

Con la elaboración de una herramienta informática capaz de dar un tratamiento adecuado a la información generada en la clínica, puede obtenerse como resultado un incremento en el tiempo efectivo para atención de pacientes. Y mejor planificación de sus actividades de índole comunitaria en las escuelas.

Lo anterior no significa que el personal de la clínica se desliga a la atención al tratamiento de la información generada en la clínica, sino que con el desarrollo de la herramienta informática, todas las tareas de registro y acceso de información se facilitan, provocando que el empleo de tiempo sea mínimo en relación al tiempo dedicado actualmente.

Es necesario mencionar en este punto, que al menos un día por semana es empleado para organizar la información de recopilada en los formularios utilizados en la clínica a lo largo de una semana de ejecución de labores.

Otro punto importante es que a través de la herramienta se dispone de reportes e información histórica debidamente organizada, que brinde un punto de vista más amplio del desempeño de las actividades ejecutadas en la clínica, representando así, un importante recurso para toma de decisiones.

### **1.4 JUSTIFICACIÓN**

La clínica de odontología de la Facultad Multidisciplinaria es una entidad donde se aplica, evalúa y aprovecha todo el conocimiento de los estudiantes de odontología, mediante el desarrollo del servicio de atención a pacientes; además, se utiliza materiales y equipos que están sujetos a control y evaluación periódica, se registra flujo de efectivo derivado de las operaciones elementales de la clínica. Sin embargo, el número reducido de

personas relativo a la cantidad de tareas que se efectúa, dificulta la ejecución y control de los procesos y tareas, provocando que las actividades no se desarrollen con la eficiencia y eficacia que se desea y planifica.

Dada la delicada labor de coordinación de las actividades efectuadas en la clínica de odontología, se vuelve imprescindible contar con herramientas que favorezcan al proceso de toma de decisiones y planificación de las tareas. La principal herramienta que ayuda a la proceso de toma de decisiones y planificación es la información, obtenida y disponible en forma oportuna y confiable, que sea fiel a los hechos.

Se propone, en este trabajo de grado, el diseño y desarrollo de un sistema informático, conformado por subsistemas que aborden las necesidades de administración de información de las más importantes áreas administrativas de la clínica, para incrementar el nivel de eficiencia de sus operaciones y atención a la comunidad.

## **1.5 METODOLOGÍA DE INVESTIGACION**

### **MÉTODOS DE RECOPIACIÓN DE INFORMACIÓN**

Para el cumplimiento de los objetivos y metas que se han utilizado las siguientes metodologías de investigación:

1. Visitas a la clínica para consultas y observación de los procesos
2. Reuniones grupales con personal administrativo de la clínica.
3. Consulta bibliográfica y de Internet sobre software para odontología.

Con la observación directa se logró tener un acercamiento real a los procesos que se ponen en ejecución cada vez que se realiza una consulta o tratamiento en la clínica.

Pasos realizados:

1. Se eligió una calendarización para visitas de observación.
2. Se anotaron y analizaron todos los pasos que se siguen para una mejor apreciación.

3. Se obtuvo copias de las formas y papelería utilizada.
4. Se revisó la forma actual de almacenamiento de información que se posee.

### **REUNIONES CON EL PERSONAL ADMINISTRATIVO DE LA CLÍNICA.**

Con las reuniones se ordenó toda la información recopilada en las visitas y se creó una estrategia para dar solución a las necesidades de la clínica, con esta metodología se ha integrado el resultado de los demás métodos.

Pasos realizados:

1. Se elaboró un calendario de reuniones mensuales.
2. En las reuniones se analizaron los procedimientos actuales y propuesta de ideas para mejorarlos.
3. Todos los documentos de recopilación de información de pacientes y diagnósticos fueron explicados en detalle por el personal encargado de la clínica.
4. Entrevistas temáticas para mejor conocimiento del problema

### **CONSULTA BIBLIOGRÁFICA Y DE INTERNET SOBRE SOFTWARE PARA ODONTOLOGÍA**

Este recurso ha permitido tener una idea de la arquitectura de sistemas utilizados en esta área y se ha obtenido documentación e ideas importantes.

Pasos realizados:

1. Se ha buscado en Internet información sobre proyectos relacionados y el software que se utiliza actualmente en la medicina dental, esto ha servido para tener un panorama más amplio de lo que se podía incluir en este proyecto de tesis.
2. Se recopilaron ejemplos de pantallas de interfaces de esta clase de software.
3. Se han obtenido ideas importantes que de una u otra forma fueron determinantes en el sistema.

### **BASE TEÓRICA DE ESTA METODOLOGÍA DE INVESTIGACIÓN**

Para seguir esta metodología de investigación se tomaron en cuenta sus bases teóricas, y de esa manera se determinó si se podían aplicar a este proyecto; a continuación se presentan los principales conceptos teóricos.

### **OBSERVACIÓN DE LOS PROCESOS:**

Esto se presenta como uno de los procesos empíricos debido a que su fundamento radica en la percepción directa del objeto de investigación y del problema.

Como observación científica el investigador conoce el problema y el objeto de investigación, estudiando su curso natural, sin alteración de las condiciones naturales, es decir que la observación tiene un aspecto contemplativo.

La observación configura la base de conocimiento de toda ciencia y, a la vez, es el procedimiento empírico más generalizado de conocimiento. Mario Bunge reconoce en el proceso de observación cinco elementos:

1. El objeto de la observación
2. El sujeto u observador
3. Las circunstancias o el ambiente que rodean la observación
4. Los medios de observación
5. El cuerpo de conocimientos de que forma parte la observación

### **Entrevista temática**

La entrevista es una buena elección si:

1. El objeto de estudio no se conoce muy bien; el problema y el objetivo del estudio pueden ser revisados durante el proyecto.

2. El "rango" de las respuestas no puede ser conocido con anticipación. Algunos encuestados pueden presentar puntos de vista que sean nuevos y desconocidos para nosotros.

3. Necesitamos la opción de presentar preguntas adicionales basadas en la información de los encuestados.

4. Las preguntas están relacionadas con el conocimiento tácito o los puntos de vista personales (actitudes, valores, creencias, etc.), de los encuestados.

5. Podemos permitirnos el tiempo suplementario y el coste de entrevistas y viajes.

### **Investigación bibliográfica**

La investigación bibliográfica es aquella etapa de la investigación científica donde se explora qué se ha escrito en la comunidad científica sobre un determinado tema o problema. ¿Qué hay que consultar, y cómo hacerlo?

La investigación científica empírica tiene básicamente cinco etapas. Primero, se definen algunas cuestiones generales como el tema, el problema, el marco teórico a utilizar, etc. Segundo, se procede a hacer una investigación bibliográfica, básicamente para ver qué se ha escrito sobre la cuestión. Tercero, se traza un proyecto. Cuarto, se ejecuta lo proyectado. Quinto, se exponen los resultados, usualmente por escrito.

En la etapa de investigación bibliográfica se pueden considerar principalmente el material referente a las herramientas que se utilizarán para desarrollar el proyecto, y no tanto referente al tema de la clínica.

Esencialmente se utilizó esta etapa para una lectura de manuales y documentos de referencia para trabajar tanto con el servidor de bases de datos y la plataforma de desarrollo que en este caso es Visual Basic .Net.

La investigación bibliográfica no tiene mucha relevancia para el proyecto, no así la investigación en Internet que es la base para este trabajo, es allí donde realmente se encuentra la documentación de mayor utilidad para los involucrados en este proyecto.

## 1.6 METODOLOGIA DE DESARROLLO

### **Ciclo de vida del sistema**

El ciclo de desarrollo del sistema o del software, también conocido como ciclo de vida, se ha utilizado para estructurar las actividades y etapas que se lleva a cabo para generar el producto final o resultado, desde la iniciativa de elaborarlo hasta cuando este es entregado o implementado.

Existen muchas aproximaciones de ciclos de desarrollo de software, cada una con ventajas y desventajas de acuerdo al contexto en el que se apliquen.

Para el desarrollo de este sistema utilizaron como base:

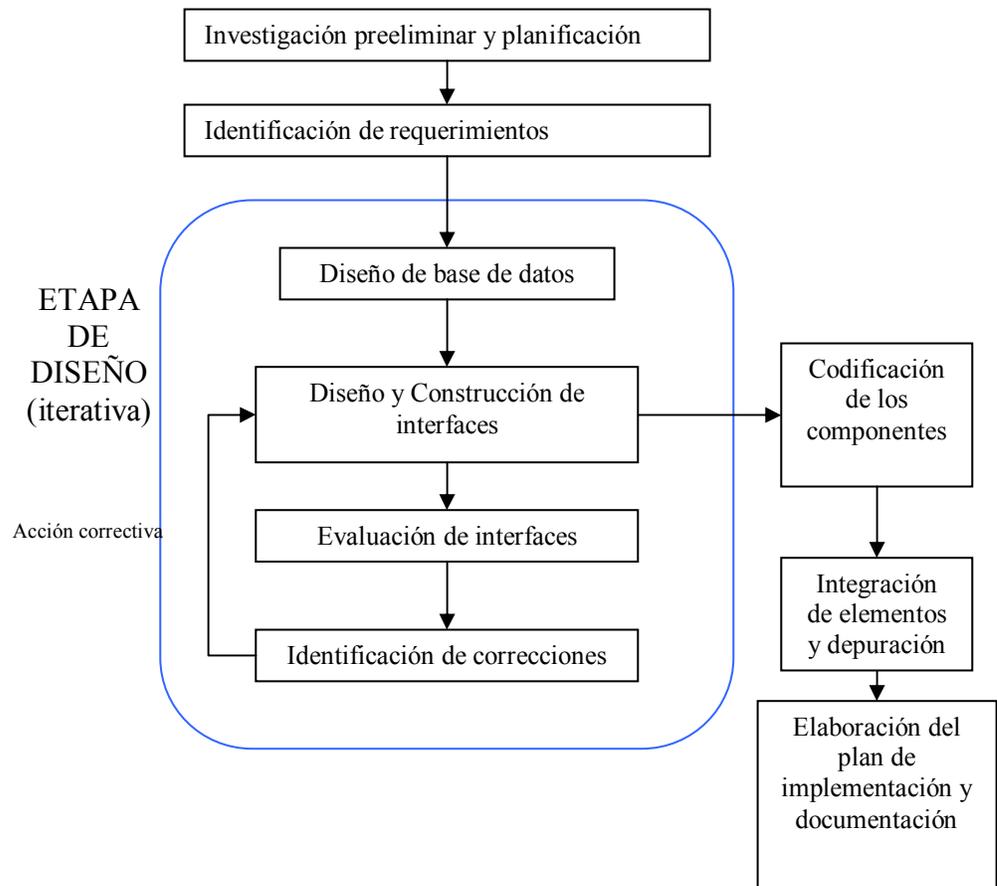
- El clásico desarrollo en cascada
- Híbrido entre RAD y orientado a prototipos.

Ciclo de vida en cascada, porque en el se contemplan básicamente la mayoría de las etapas importantes del desarrollo como lo son: investigación preliminar, requerimientos, diseño, diseño detallado, codificación, integración, implementación, entre otros.

RAD o (Rapid Application Development) : que representa una metodología mediante la cual se elaboran las pantallas del sistema y son revisadas y evaluadas por los usuarios hasta obtener la interfaces finales que posteriormente se codifican. Se consideró este modelo, debido a que dentro de la clínica existen procesos de recopilación de información para los cuales no existe una metodología formal. Mediante la metodología RAD se pretendía, conjuntamente con los usuarios del sistema o personal de la clínica, no solo diseñar las pantallas para su fácil interacción, sino también proponer metodologías formales a través las interfaces diseñadas, aun para aquellas áreas que no cuentan con formatos estandarizados en la clínica para la recopilación.

Orientado a prototipos: este método, meramente pone énfasis a la etapa de especificación de requerimientos a través de construcción de prototipos programados que aproximan al usuario a la idea del sistema final.

Se decidió implementar un híbrido entre RAD y orientado a prototipos, primeramente porque ambos métodos son iterativos; la etapa iterativa en el ciclo de vida propuesto radica en la fase de diseño, en la cual se desarrollará prototipos de interfaces para captura de datos y muestra de información. Las interfaces fueron programadas, y se mostraba una simulación de la navegación a través del sistema sin guardar información. Los grupos de interfaces correspondientes a cada módulo se intentaron presentar debidamente conectados o enlazados para brindar al usuario final la sensación de la funcionalidad que tendrá del sistema final. Tales interfaces fueron sometidas a la respectiva evaluación de parte del personal operativo de la clínica para identificar correcciones o desarrollar propuestas de mejoramiento para el funcionamiento del sistema, todo lo anterior hasta que se llegó a la interfaces existentes en el sistema final.



A continuación se describe cada una de las fases del ciclo de vida que se implementará:

### **INVESTIGACIÓN PRELIMINAR Y PLANIFICACIÓN**

En esta etapa se recopilara toda la información posible de la clínica de odontología de la Facultad, sobre sus procesos y antecedentes, funciones y responsabilidades, con el objeto de adquirir un mejor entendimiento de la entidad. Además, se planificara reuniones y escogerá el método de desarrollo que mejor se adecue a las circunstancias para que la herramienta informática se acople a las necesidades informativas de la clínica. Se determinara, objetivos, premisas y demás lineamientos para la coordinación entre el equipo de desarrollo del software y el personal de la clínica.

### **IDENTIFICACIÓN DE LOS REQUERIMIENTOS**

En esta etapa se determinó una especificación completa y validada de la arquitectura global del sistema de información en lo referente a hardware-software, sus interfaces y el rendimiento del producto final.

### **DISEÑO DE BASE DE DATOS E INTERFACES**

*Del modelo de interfaces:*

Para el formulario principal del sistema se diseñó un formulario con un esquema de navegación a base de nodos circulares relacionados entre sí según el esquema de módulos del sistema.

Este control fue especialmente diseñado para brindar un modelo de navegación diferente y entretenido al usuario del sistema. Sin embargo, también se implementó el tradicional esquema de menús para usuarios que prefieren esta línea de navegación.

La figura muestra un ejemplo de este esquema:

Para algunos módulos se ha utilizado una plantilla de presentación de datos en un formulario utilizando un control de tipo TreeNode, un DataGridView en conjunto con un combobox para paginación de datos, todos contenidos en un control de tipo panel. Lo

anterior es para facilitar el acceso y visualización de la información para tareas básicas como edición y agregación para los módulos que se detallan a continuación:

1. Expedientes
2. Consulta
3. Cuentas por Cobrar y Remesas
4. Planificación de Citas
5. Practicantes
6. Mantenimiento de Equipos
7. Administración de Materiales.

Luego, para el *detalle* de edición y agregación de registros en cada sección se ha implementado formularios con controles según las necesidades de datos para cada módulo.

*De la base de datos:*

La base de datos respeta un esquema relacional manteniendo un comportamiento en *updates* y *deletes* de tipo *cascade*. El modelo relacional permite mantener integridad de la base y evitando dejar registros huérfanos en tablas comunes relacionales. El sistema posee la capacidad de ejecutar actividades de eliminación de registro para usuarios con tales privilegios. No se han programado funciones y procesos especiales del lado de la base de datos, ni triggers manuales; solamente los automáticos ejecutados por las llaves foráneas al momento de eliminaciones y actualizaciones. Lo anterior se solventa con la utilización de Transacciones desde vb.net para mantener la integridad desde los procesos que ejecutan las inserciones y actualizaciones.

## **CAPÍTULO II**

### **2 IDENTIFICACIÓN DE NECESIDADES Y ANÁLISIS DE LOS REQUERIMIENTOS.**

## 2.1 DIAGNOSTICO DE LA SITUACION INICIAL DE LA CLINICA DE ODONTOLOGIA.

De la investigación efectuada en la clínica se pudo identificar puntos críticos y sus causas sobre los que se trabajó para mejorar el desempeño de la misma:

1. **La emisión de reportes:** la emisión de reportes es una de las principales tareas que consume mucho tiempo para los encargados. Lo anterior era producto de la metodología de recolección de la información dentro de la clínica (mixta: parte se recolectaba en papel –formularios- y otra en archivos de hoja electrónica). Para la emisión de los diferentes reportes, los encargados tenían que remitirse a todos los formularios utilizados en el periodo y hacer un recuento a mano de cada una de las áreas de las cuales necesitaban emitir el reporte, por ejemplo: cantidad de pacientes atendidos por un estudiante en particular, cuantos tratamientos y diagnósticos efectuó, etc. Por otra parte, existe una serie de reportes que no se emitían por la falta de tiempo para la agrupación de la información de manera ordenada, pues la información existe. Todos estos inconvenientes también ocasionaban que fuera difícil actualizar los expedientes de los pacientes que no visitan la clínica por más de dos años, para ver si su historial clínico ha cambiado.
2. **Evaluación de los estudiantes:** es notorio que dentro de la clínica se trata de llevar de la manera mas objetiva la calificación o evaluación de los estudiantes, pero esta tarea se dificulta, debido a que no existe registro de los errores o faltas cometidas por los estudiantes ya sea al llenar formularios o en los diagnósticos o tratamientos que este estudiante efectuó. Lo anterior es a consecuencia de que el docente encargado es el responsable de revisar cada una de las actividades que desempeña cada estudiante, y en caso de haber correcciones, el docente encargado la efectúa de manera inmediata, sin haber un registro de la naturaleza de la falta y la correspondencia con el estudiante que la comete, por lo que la evaluación se delimita a la apreciación desarrollada durante todo el periodo del docente encargado Esta tarea es difícil, pues se debe tener en cuenta que son entre 9 y 11 los

estudiantes que se rotan en un periodo aproximado de 2 semanas. Y muchos detalles se escapa de la memoria de los docentes.

3. **Asignación de horario de actividades y consultas:** si bien existe una coordinación entre los horarios de atención al PPC y los horarios de atención para las consultas, han existido, pequeños inconvenientes como el siguiente: resulta difícil asignar horarios para aquellas consultas que se difieren, ya que debido a la demanda creciente de pacientes se hacen asignaciones de hasta 3 meses a partir de la fecha actual.

Generalmente, los horarios de atención al PPC no entran en conflicto con los de atención al paciente en la clínica porque se designa un día a la semana para la atención al PPC, día en el cual, se trata de reducir las consultas en la clínica.

4. **Registro de los pacientes del PPC:** debido a los inconvenientes con el tratamiento de la información, la información registrada para los pacientes del PPC es reducida. Se debe llevar registro de los pacientes del PPC que participan o no en las actividades y actualizar periódicamente los nombres de los pacientes que cambian de grado, se retiran, etc.

## **2.2 DESCRIPCION DE LAS AREAS FUNCIONALES INCLUIDAS EN EL SISTEMA FINAL.**

A continuación se describen las áreas funcionales de la clínica incluidas en el desarrollo el sistema.

1. **Atención a pacientes en la clínica:** en el sistema final se ha incluido un modulo destinado al registro y tratamiento de la información que se genera en la clínica a consecuencia de la atención que se le brinda a los pacientes. Hasta la fecha se registraba todo lo concerniente a las consultas de los pacientes, los tratamientos y diagnósticos efectuados, con la ayuda de formularios y colectores de información (en papel) que, en ocasiones, son de más de 3 hojas. Los pacientes son atendidos, por estudiantes operarios o por el docente encargado. Se efectúa una verificación de

la existencia de un expediente del paciente, en caso de existir uno, se anexa la información que genere la consulta actual al historial del paciente. Si el paciente no tiene expediente se registra y se procede con el diagnóstico y tratamiento. Existe, internamente una clasificación de los pacientes: pacientes adultos y pacientes infantes, para la cual se han elaborado colectores de información respectivos de acuerdo a las diferencias que presentan. Dentro de la clasificación de los pacientes infantes existe otra que corresponde a los infantes que pertenecen al PPC (Programa de Prevención Comunitaria) que efectúa la clínica, y los infantes que no pertenecen a él. Los alumnos o estudiantes que operan en la clínica, recolectan la información y tal información es registrada oficialmente hasta que el docente encargado efectúa la respectiva revisión para chequear que la información (diagnósticos o tratamientos recetados) sean congruentes con los casos atendidos. Posteriormente, los formularios son almacenados en archiveros y los correspondientes a apertura de expediente son también registrados en hojas electrónicas en una de las dos computadoras con las que cuenta la entidad.

2. **Seguimiento de Actividades del PPC:** la clínica desarrolla un Programa de Prevención Comunitaria, comúnmente llamando PPC, el cual consiste en brindar atención especial a niños de diferentes escuelas con respecto a su educación de salud bucal. Se planifica actividades con tales niños, ya sea, enjuagues, desarrollo de charlas, periódicos murales, entre otras. Todas estas actividades se efectúan en la respectiva locación de la escuela. Para cada actividad se nombran responsables de ejecutarla, y se registra la participación de los infantes, dependiendo de la naturaleza de la actividad. Por ejemplo, en la elaboración de fluocolutorios o los rellenos, se debe registrar a los infantes que asistieron a esa actividad. Además se registra quienes efectuaron la actividad, de manera tal que al final del periodo, se posea información sobre todas las actividades realizadas en cada escuela o institución pública y todos los detalles respectivos.
3. **Administración de Horarios:** la clínica administra su horario de manera tal que es posible registrar eventos para el PPC y para la atención local de los pacientes. La asignación de citas en el programa se realizará mediante un formulario en el cual pueda confirmarse la existencia de otras citas en la fecha seleccionada.

4. **Registro de las actividades o tareas realizadas por los estudiantes operarios:** la clínica es, además de una institución de atención a pacientes, una entidad que evalúa el desempeño de los estudiantes que operan internamente. Cada estudiante que opera en la clínica lo hace por un periodo aproximado de dos semanas, tiempo en el cual, es registrada toda actividad que realice, para que, al final de dicho periodo, el docente director de la clínica, posea un informe del desempeño de dicho estudiante y poder asignarle una calificación final para una nota que posteriormente se promediara y será parte de su historial académico. En el programa se registrará su desempeño en el desarrollo de consultas, el manejo de los formularios, capacidad de efectuar diagnósticos, seguimiento de los tratamientos de los pacientes y otras áreas.
5. **Control de material y equipo:** Actualmente se lleva un seguimiento en la evaluación del equipo utilizado en la clínica, así como también control de los inventarios de medicamentos. En el sistema se tendrá un módulo especial de control de materiales o medicamentos, clasificados en grupos. Este módulo será utilizado para crear un registro de productos dentro de la clínica.
6. **Registro de las cuentas por cobrar:** existe dentro de la clínica, un estado de cuenta por cada uno de los pacientes atendidos, como consecuencia de las deudas que adquieren por el pago de sus citas o consultas. Es registrado cada uno de los abonos y cargos a su respectiva cuenta.
7. **Control de usuarios, personas y perfiles:** En este módulo se registran todas las personas existentes en el sistema, clasificándolos como Estudiantes Practicantes, Directores o Administrativos. Según el status de cada persona registrada se ingresan los datos respectivos. Además en esta sección se crean perfiles de usuarios, usuarios de sistema y se hace la asignación respectiva de usuarios a personas.
8. **Elaboración de reportes:** en la clínica también se emiten reportes al final de los periodos o ciclos de operación para reportarse a las autoridades respectivas, los

reportes incluyen reportes de control de los materiales y equipos, pacientes atendidos, desempeño de estudiantes, reportes de actividades del PPC, entre otros. Al final de cada periodo, los reportes se generan como consecuencia de la revisión de todos los formularios utilizados, tarea que representa un esfuerzo adicional.

### **2.3 DESCRIPCION DE LOS MODULOS NECESARIOS.**

Dadas las áreas funcionales de la clínica y los puntos críticos identificados en el diagnóstico anterior, se presenta una descripción de los módulos que se incluyen en el sistema final que atenderán dichas áreas:

#### **Modulo de consultas:**

El modulo de consulta es para registrar toda la información pertinente a las consulta de los pacientes. En este modulo se pretende abordar los siguientes puntos:

1. Autenticación de estudiante o medico que atiende la consulta.
2. acceso a la apertura y registro de expedientes para pacientes nuevos. El anterior representa un modulo por separado, el cual es accedido directamente también desde el módulo de consultas.
3. Acceso a historial de consultas del paciente en cuestión.
4. Acceso a registro de consultas para pacientes adultos e infantes.
5. Representación de diagramas necesarios para registrar la información bucal de la consulta. Tales como.
  - a. Examen periodontal CPITN
  - b. Índice PMA simplificado.
  - c. Placa bacteriana.
6. Captura y registro de radiografías bucales.
7. Toda consulta realizada por un estudiante practicante o alumno, será interpretada como no revisada y pendiente de evaluación por parte de una persona que tenga status de Director en el sistema. Lo anterior brinda mayor detalle en el registro de evaluaciones para el practicante.
8. acceso para la asignación y control de horarios de citas.

La característica principal de éste, estriba en que será capaz, no solo de registrar la información de la consulta, sino también, a través de él, los estudiantes operarios estarán registrando todos sus tratamientos y diagnósticos que efectúen. Además, la sesión de consulta efectuada será oficialmente registrada hasta la respectiva revisión del docente encargado, quedando registradas todas aquellas áreas de corrección, para que automáticamente se registre en la hoja de trabajo del estudiante que atendió la consulta. Lo anterior, aumenta la objetividad a la hora de emitir una calificación para el estudiante.

### **MODULO DE EXPEDIENTES.**

Este es un modulo pequeño, cuya funcionalidad se resume en el ingreso y edición de nuevos registros de clientes o pacientes en la clínica mediante la apertura de un expediente. En este módulo se ingresa tanto pacientes procedentes del Programa de Prevención Comunitaria, como pacientes particulares tales como niños, estudiantes de la universidad o personas ajenas a la universidad.

### **MODULO DE CONTROL DE INVENTARIO.**

Este módulo tiene como propósito registrar entradas y salidas de medicamentos de la bodega principal de la clínica. Registra cada medicamento con información particular del mismo, como lo es: precio de costo, clasificación de medicamento; manteniendo actualizadas las existencias de medicamentos.

Además, este módulo permitirá realizar los ajustes respectivos para las diferencias encontradas entre las existencias calculadas y las existencias reales registradas a partir de inventarios físicos que se realicen de forma periódica.

Como adición funcional a este modulo, se podrá registrar el inventario de mobiliario utilizado en el interior de la clínica y las revisiones que se les realizan de forma periódica. Contando además con el registro de entrada y salida de los mobiliarios con detalles respectivos (fecha, hora, persona responsable, cantidad, etc.).

También en este módulo se establecen los stocks de inventarios mínimo necesario para la elaboración de la respectiva requisición.

## **MÓDULO DE ADMINISTRACIÓN PPC**

En este módulo de planificaciones se programan las diferentes actividades de la clínica con respecto al desarrollo de su Programa de Prevención comunitaria (PPC). Esto consiste en la planificación de actividades que se llevará a cabo en las escuelas que forman parte del programa. En éste registro o planificación de actividades se detalla información referente a:

1. Responsables de la ejecución de la actividad
2. Fecha, hora.
3. Escuela o lugar de desarrollo de la actividad.
4. Alumnos de las escuelas pertenecientes al programa que participan o no en las actividades.
5. otros.

Con respecto a los alumnos de las escuelas que pertenecen al PPC, es necesario llevar información actualizada de cada uno de ellos; especialmente, si realizan o no las actividades del PPC.

Entre algunas actividades que se incluyen en el PPC se pueden mencionar:

1. Enjuagues.
2. Charlas preventivas.
3. Elaboración de periódicos murales con los alumnos de las escuelas.
4. Actividades recreativas que atribuyan a la educación dental de los alumnos.
5. Otros.

## **MÓDULO DE CONTROL PRACTICANTES.**

En este módulo, el docente encargado, puede apreciar el registro de todas las actividades en las cuales cada estudiante ha tomado participación en la emisión de diagnósticos o aplicación de tratamientos, entre otros, de tal manera que sea fácil efectuar una evaluación del desempeño del estudiante y asignarle una calificación. Este módulo esta conectado con los otros:

1. *De consulta*: para registrar un historial de las consultas, diagnósticos y tratamientos efectuados por el estudiante. Además, para registrar las correcciones señaladas por el docente encargado.
2. *De planificación*, pues también se verifican con el respectivo privilegio de evaluar o consultar evaluaciones para todas las actividades del PPC en las cuales el estudiante ha tomado participación.

### **Modulo de configuraciones:**

En este modulo se realizan muchas funciones que permiten el eficaz funcionamiento del sistema entero. Entre ellas tenemos: Adicionar información que aparece en los controles de lista de los diferentes interfaces de los módulos del sistema, tales como las interfaces de consulta, ingreso de expedientes, etc.

### **Módulo de respaldos:**

Básicamente, este módulo es una pantalla simple de ejecución de comando para respaldar la base de datos.

### **Modulo de cuentas por Cobrar.**

Básicamente, en este modulo existe un detalle de las cuentas por cobrar o estado de cuenta de cada uno de los pacientes que frecuenta la clínica. Con detalle de clave del paciente, cargos, abonos y saldos pudientes o actuales.

## **Modulo de reportes.**

En este modulo se han centralizado todos los reportes necesarios en la clínica, reportes de actividades planificadas, medicamento utilizado en el mes, tratamientos efectuado por estudiante, etc.

## **2.4 ESTUDIOS DE FACTIBILIDAD**

### **2.4.1 ESTUDIO DE FACTIBILIDAD TÉCNICA**

#### **Hardware**

En esta sección, se hace referencia a la arquitectura del equipo que puede ser utilizado tanto en el proceso de creación como de utilización del programa de la clínica.

En este caso dada la situación planteada en el capítulo I (véase situación actual), la adquisición de un determinado equipo no es dependiente de la clínica, sino que de autoridades primarias.

Es por eso que se presentará principalmente la factibilidad técnica tanto de software como de Hardware que presenta en este momento la clínica.

Evaluación del hardware actual:

En este punto se ha determinado que las computadoras con las que cuenta la clínica odontológica poseen un nivel aceptable de procesamiento y memoria principal que permitirá la utilización del programa

Las características principales del hardware que posee actualmente la clínica odontológica son las siguientes:

Identificación	Procesamiento	Capacidad	de	Memoria	Conectividad	Otros
	Velocidad	Almacenamiento		RAM	de Red	
	CPU					

Computadora Principal	1.6 GHz	40 GB	256 MB	Tarjeta Ethernet Fast 10/100 Mbps	Impresor HP 3920 Deskjet
-----------------------	---------	-------	--------	---	--------------------------------

## **Software**

### Sistema Operativo

El Sistema operativo instalado en la computadora principal es Windows en la versión XP con Service Pack2.

### **Herramienta de desarrollo**

Existen varias herramientas y lenguajes de programación basadas en distintas tecnologías que podrían utilizarse para el desarrollo del sistema propuesto; dentro de esas podemos mencionar los siguientes:

- JAVA
- Visual C++ .NET
- Visual C# .NET
- Borland Delphi
- Visual Basic .NET
- Visual Fox Pro 9

En el campo de las bases de datos se pueden mencionar también las siguientes opciones:

- MS SQL Server 2005
- PostgreSQL
- MySQL Server 5
- Visual Fox Pro 9

EL lenguaje de programación seleccionado es VB.NET debido a que brinda la facilidad de desarrollar los sistemas utilizando estrictamente la programación orientada a objetos. Además, es uno de los lenguajes de última generación que actualmente se usa para el desarrollo de sistemas. También gracias a al entorno de visual Studio bajo el cual se desarrolla es posible en el futuro robustecer el sistema, no solamente siguiendo la misma estructura de programación vb.net, sino poder seguir otra línea o lenguaje para agregar mayor funcionalidad a los módulos existentes en este sistema o incluir otros módulos desarrollados con otras líneas de programación según las necesidades de la clínica así no pidan. Las nuevas plataformas también están abriendo nuevos panoramas de multiplataforma. Por ejemplo, el proyecto MONO para Linux o software de código abierto permitirá la ejecución de sistemas de desarrollados bajo Windows en Linux, específicamente los desarrollados en .NET.

En cuando al gestor de base de datos MYSQL, es un gestor de base de datos que puede correr en cualquiera de las dos corrientes de sistemas operativos. En este caso, en el futuro se puede instalar el gestor en sistemas que tengan menos vulnerabilidad a problemas de seguridad o virus sin tener mayor problema por el gestor seleccionado. Además MYSQL está en planes de ser ahora proporcionado bajo la calidad de Sun Microsystems que son un equipo altamente reconocido por la fiabilidad de sus productos con lo es JAVA. MYSQL en su versión 5 en adelante será proveído de capacidades que antes no poseía, para mencionar SP, Functions, Triggers, etc. Para el tiempo en que este sistema se ha desarrollado aún tiene ciertas limitantes con respecto a estas áreas pero la actualización del modelo de base de datos, será fácilmente elaborada al cabo de unos meses.

#### **2.4.2 ESTUDIO DE FACTIBILIDAD ECONÓMICA.**

En este estudio de factibilidad se pretende plantear la situación actual de la clínica, contra la implementación real que tuviera este sistema.

Se tratará de demostrar que resulta mucho mas beneficioso, desde el punto de vista financiero, la obtención de un sistema informático para el control de la información dentro de la clínica que continuar haciéndolo manualmente.

En el enfoque presentado sólo se consideran aquellos aspectos que están directamente relacionados con la utilización del tiempo en función al sistema de información implementado. No se consideran los elementos del costo de los medicamentos o equipo técnico utilizado en la consulta (a excepción de las PC), puesto que es un costo que esta fuera del los parámetros controlados por el sistema de información.

A continuación se enumera todos los factores que se consideraron para hacer una comparación entre el modelo utilizado actualmente y el propuesto:

Minutos promedio por consulta con sistema propuesto	26
Pacientes promedio diarios con sistema propuesto	14
Pacientes promedio diarios con sistema actual	10
Minutos promedio por consulta con sistema actual:	30
Ingreso promedio por consulta	\$5.00
Ingreso promedio por minutos en consultas	\$0.17
Tiempo promedio perdido en el acceso a la información por consulta (min.)	10
Costo del equipo de computo actual	\$800.00
Depreciación anual	20.00%
Costo anual de mantenimiento de hardware	\$60.00
Costo de energía eléctrica generado por la PC diariamente	\$1.10
Tiempo promedio perdido en el acceso a la información por consulta (min.) con el sistema propuesto	4
Salario mensual de persona digitador	\$160.00
Salario diario de persona que digita información	\$5.33
Salario por hora de persona que digita información	\$0.67

Los pacientes promedio diarios atendidos en la clínica se presentan en la tabla anterior. Pero es necesario aclarar que este tiempo estimado es el tiempo medido desde que el paciente es atendido por la persona encargada de hacer el registro de la consulta, hasta que es establecida la nueva fecha de consulta o se cobra la cuota establecida, que también se presenta en la tabla.

A continuación se presenta, en base a dicha información, la estimación de los ingresos por consulta utilizando el sistema actual:

<b>ingresos del sistema actual</b>			
	<b>diarios</b>	<b>mensuales</b>	<b>anuales</b>
ingreso promedio por consulta	\$50.00	\$1,000.00	\$12,000.00

Se muestra ahora la lista de egresos o costos en los que se incurre para el mantenimiento del actual sistema:

<b>Costos del sistema actual</b>			
	<b>diarios</b>	<b>mensuales</b>	<b>anuales</b>
<b>Papelería y Útiles</b>	\$1.50	\$30.00	\$360.00
<b>salario persona encargada digitación</b>	\$5.33	\$160.00	\$1,920.00
<b>costo tiempo promedio perdido en el acceso a la información por consulta</b>	\$16.67	\$333.33	\$4,000.00
<b>Energía eléctrica de 1 PC (8horas diarias)</b>	\$1.10	\$22.00	\$264.00
<b>Costos por limpieza y mantenimiento del hardware</b>			\$60.00
<b>costo de depreciación de equipo de computo</b>	\$0.44	\$13.15	\$160.00
	\$25.04	\$558.48	\$6,764.00

El tiempo promedio perdido por consulta se considera un costo, pues representa tiempo que pudo ser invertido en otra tarea de consulta con otros pacientes o actividades afines.

Como tiempo perdido se considera todos aquellos minutos:

1. Los que se consumen cuando se busca al paciente en los expedientes o archivos manualmente, o en las hojas de Excel almacenadas en la computadora.
2. El tiempo que se tarda en establecer una fecha de cita.
3. Tiempo invertido en llenar hojas de control de inventario de medicamentos necesarios para atender al paciente.
4. Tiempo invertido en el registro de la revisión del docente encargado de la clínica.
5. Tiempo invertido en el registro de la hoja de consulta del paciente y la hoja de diagnóstico.
6. Tiempo utilizado para el registro de la obtención de radiografía intraoral.
7. Ocasiones en que se ha movido una cita o consulta por falta de coordinación de los horarios.
8. otros.

Se sabe de las diferencias entre un tipo de consulta y otro. Por tal razón se ha estimado tiempos promedios.

A continuación se presenta el ingreso estimado de la utilización del sistema propuesto, en este estimado se ha tomado en cuenta que el número de pacientes diarios atendidos se incrementa en un 40% aproximadamente, Manteniendo el mismo precio de consulta que el sistema anterior:

<b>Ingresos por sistema propuesto</b>			
	<b>diarios</b>	<b>mensuales</b>	<b>anuales</b>
<b>ingresos por consultas</b>	<b>\$60.67</b>	<b>\$1,213.33</b>	<b>\$14,560.00</b>

Ahora los costos de diseño y desarrollo del software.

<b>Costo de desarrollo del software</b>	
<b>análisis diseño y desarrollo del sistema</b>	<b>\$1500.00</b>
<b>Costo Energía eléctrica por 2 PCS (250horas )</b>	<b>\$175.00</b>

<b>Instalación (Red y PCS)</b>	\$40.00
<b>licencia MySql 5.0</b>	\$0.00
<b>Licencia VB.NET</b>	\$780.00
	\$2,495.00

En la realidad, el costo de elaboración de un software como el de la clínica se amortiza en función a la cantidad de ejemplares vendidos del software. Es decir, el costo de elaboración se recupera con una cantidad determinada de ventas del software; luego, sobrepasada esta cantidad, el ingreso comienza a registrarse. En una tabla de amortización, se puede observar como se cubre el costo con cada venta y como crece el ingreso o utilidad. Generalmente, en la venta del software, lo que se pretenden cubrir en cada venta, es el reembolso del 100 por ciento de la mano de obra, y una parte de los costos por licenciamientos. Por tal razón, el precio de venta del software ha sido estimado como aproximadamente el 63% del costo total de desarrollo del software. El 37% del precio del software lo representa el costo del licenciamiento. Que en este caso de estudio, se plantea como absorbido por los desarrolladores.

<b>Costos con el sistema propuesto</b>			
	<i>diarios</i>	<i>mensuales</i>	<i>anuales</i>
<b>Papelería y Útiles</b>	\$1.50	\$30.00	\$360.00
<b>salario persona encargada digitación</b>	\$5.33	\$160.00	\$1,920.00
<b>costo tiempo promedio perdido en el acceso a la información por consulta</b>	\$9.33	\$186.67	\$2,240.00
<b>Adquisición del software</b>			\$1,500.00
<b>Energía eléctrica de PC (8horas diarias)</b>	\$1.10	\$22.00	\$264.00
<b>Costo por mantenimiento y limpieza del hardware</b>			\$60.00
<b>costo de depreciación de equipo de computo</b>	\$0.44	\$13.15	\$160.00
	\$17.71	\$411.82	\$6,504.00

Se puede observar que todos los costos permanecen constantes, a excepción de los costos de los minutos perdidos en el acceso a información por consulta. Lo anterior es por lo siguiente: El sistema está diseñado para evitar pérdidas de tiempo en la búsqueda de los expedientes en papel, y se realiza de forma automatizada, también se ahorra el tiempo de ir a buscar las citas para un determinado día, porque desde el formulario para nuevas citas puede identificarse las citas programadas haciendo más fácil determinar si es posible o no programar una cita; también se ahorran los minutos perdidos en llenar hojas de inventarios de medicamentos, ya que se hace directamente en el sistema de una manera sencilla que permite tener la hoja de movimientos en menos de un minuto.

Si bien, los cálculos han sido efectuados utilizando como parámetro la cuota actual de consulta de la clínica. La clínica es una entidad sin fines de lucro, por lo que en ambos casos (con el sistema propuesto y el actual) los costos de los sistemas son mayores que los ingresos. Recuérdese que se han excluido los precios de los medicamentos.

Lo que es notable es que los costos del sistema propuesto resultan menores que los costos del sistema actual, inclusive incluyendo el precio total de la aplicación, para ser cancelado en un año.

Una aplicación o sistema como el expuesto para la clínica se puede valorar en 900 dólares incluyendo instalación del software, con acceso a dos usuarios. Para los desarrolladores, el costo de las herramientas con que fue desarrollado el software y el pago por su fuerza de trabajo puede ser recuperado con la venta de tres ejemplares del software y aun pueden ver ganancias. Un sistema empresarial SAE ha sido tomado como base. SAE es un sistema mexicano para control de contabilidad y facturación bajo ambiente Windows. Es un sistema robusto con capacidad de administrar muchas empresas, bodegas, productos y usuarios. El sistema se cotiza a 1000 dólares y su valor incrementa en función a las licencias por usuario y conexiones. En promedio los usuarios se cotizan en 200 dólares cada uno.

La diferencia anual del costo actual con la del costo utilizando el sistema propuesto es mínima.

Es notable que la diferencia en los costos está dada por los minutos perdidos en cada consulta.

El sistema propuesto está diseñado para una red, lo que puede ser utilizado simultáneamente por 2, 3 o más módulos de consulta. En el plan de implementación ideal

del sistema se proponen 4 módulos de consulta simultáneamente, lo que, aumentarían los clientes atendidos diarios y con ello los ingresos y aumentando levemente los costos en las áreas de mantenimiento de hardware, y la depreciación de las maquinas. Lo anterior resulta mucho más rentable que si se tratase de atender muchas personas simultáneamente con el sistema manual, ya que los costos de minutos perdidos por paciente se aumentarían, generando costos mucho mas elevados.

Para ambos casos podemos calcular el valor futuro de los costos al término de un año y el valor futuro de los ingresos al término de un año para ver el valor futuro neto de las utilidades.

VALORES FUTUROS DE INGRESOS Y EGRESOS Y OBTENCION DE VALOR FUTURO DE UTILIDADES

- **SISTEMA PROPUESTO:**



El valor futuro de los egresos correspondiente se calcula utilizando los siguientes datos.

- 11% tasa de interés anual
- 12 Número de egresos
- \$411.82 Importe de los egresos

El valor futuro de los egresos es de: \$5198.73.

El valor futuro de los ingresos correspondiente se calcula utilizando los siguientes datos.

- 11% tasa de interés anual
- 12 Número de egresos

\$1213.33 Importe de los egresos

El valor futuro de los egresos es de: \$15316.92.

Lo anterior genera una utilidad futura al cabo de un año de: \$10118.19.

- **SISTEMA ACTUAL**



El valor futuro de los egresos correspondiente se calcula utilizando los siguientes datos.

- 11% tasa de interés anual
- 12 Número de egresos
- \$558.48 Importe de los egresos

El valor futuro de los egresos es de: \$7050.23.

El valor futuro de los ingresos correspondiente se calcula utilizando los siguientes datos.

- 11% tasa de interés anual
- 12 Número de ingresos
- \$1000 Importe de los egresos

El valor futuro de los ingresos es de: \$12623.87.

Lo anterior genera una utilidad futura al cabo de un año de: \$5573.64.

El valor futuro de la utilidad generada por el sistema propuesto es el doble de lo generado por el sistema actual, por lo tanto es factible económicamente la implementación del sistema informático recomendado.

Se debe tener en cuenta en condiciones ideales de implementación del sistema con muchas mas computadoras (una correspondiente a cada cubículo), el ingreso mensual también aumentaría considerablemente, aunque también el egreso por mantenimiento de computadoras y otros costos. Pero el porcentaje de incremento de los ingresos fuera más del doble que el porcentaje de aumento de egresos.

## **2.5 ALCANCES DE LOS MÓDULOS DEL SISTEMA.**

### **Módulo de consultas.**

1. En este modulo se han integrado todas aquellas secciones mencionadas en el apartado de descripción de módulo del sistema correspondiente a este modulo.
2. En este Módulo se incluye un mecanismo de evaluación para que sea utilizado por del Doctor encargado de las evaluaciones de los estudiantes. El doctor encargado hará las correcciones necesarias al registro de consulta ingresado al sistema por el paciente, y automáticamente todas las evaluaciones serán guardadas para abonar al record del estudiante operario.
3. Se incluye las interfaces necesarias para el almacenamiento de las radiografías previamente escaneadas.
4. El Módulo de consulta es accedido mediante contraseña y login de doctores autorizados.
5. Se pretende mediante este modulo realizar todas las tareas básicas sobre los datos de los pacientes: agregar, editar información.
6. Se incluye interfaces amigables que hacen que la usabilidad de este modulo sea intuitiva.

### **Módulo de expediente:**

1. Se ha mejorado el manejo de los datos de expediente mediante este Módulo.
2. Este modulo se ha diseñado individualmente para que también pueda ser accedido por personas que laboren en la clínica y que desempeñen actividades meramente administrativas. De esta manera es posible que éstas personas puedan registrar información básica de pacientes sin que tengan que acceder al modulo de consultas.
3. Se accede a este modulo también mediante autenticación, ya sea de usuario normal con privilegios de modificación o como autenticación de doctor.
4. Se incluye interfaces amigables que hacen que la usabilidad de este modulo sea intuitiva.

### **Módulo de control de inventarios.**

1. Con este Módulo se pretende tener actualizados los niveles de existencia de los medicamentos y herramientas que se encuentran en la clínica.
2. El Módulo permite realizar inventarios físicos en cualquier momento.
3. Mediante este modulo se pretende facilitar la administración de la información del inventario para que sea fuente critica de información para su control.

### **Módulo de planificaciones de citas:**

1. En este Módulo se lleva administra la información de la programación de citas.
2. Permite crear modificar y borrar citas de una manera sencilla y permite visualizar las citas por períodos
3. Se incluye interfaces amigables que hacen que la utilización de este modulo sea intuitiva.

### **Módulo de control de estudiantes practicantes de la clínica.**

1. En este Módulo se pretende que se facilite de gran manera la manipulación y administración de la información generada por los estudiantes operarios de la clínica para la evaluación.
2. Se ha planificado que a través de este módulo se filtre toda la información relacionada con cada evaluación estudiante, desde sus proyectos para el PPC hasta los diagnósticos efectuados, tratamientos y otros. Asimismo, desde esta sección será posible ver las evaluaciones y generar evaluaciones pendientes para el estudiante.

### **Módulo de configuraciones:**

1. Esta diseñado con el propósito de dar mantenimiento a los formularios de entrada de datos respectivos a cada Módulo, permitiendo añadir, o borrar elementos de selección en dichas interfaces.
2. Solo se consideran aspectos de selección en combobox y listas con respecto a los objetos de selección mencionados en el párrafo anterior.
3. Se incluye interfaces amigables que hacen que la utilización de este modulo sea intuitiva.

### **Modulo de cuentas por cobrar.**

1. Mediante este modulo, se pretende dar facilidad de administración de la información financiera de los pacientes respecto a sus abonos y deudas, brindando información puntual sobre sus saldos y detalle de movimientos.
2. Se puede visualizar a los pacientes basados en sus saldos.

### **Modulo de reportes.**

3. Con este módulo brinda acceso a una fuente de información general y específica de la clínica referente a cada área o globalmente.
4. Se utiliza una interfaz amigable e intuitiva que permita sacarle mayor provecho por parte del usuario.

## **2.6 REQUERIMIENTOS FUNCIONALES**

### **2.6.1 REQUERIMIENTOS DE HARDWARE EN PC'S CLIENTES:**

<b>Procesamiento</b>	<b>Capacidad de Memoria</b>	<b>Conectividad</b>	<b>Otros</b>
<b>Velocidad CPU</b>	<b>Almacenamiento</b>	<b>RAM</b>	<b>de Red</b>
1.1GHz	15 GB	256 MB	10/100 Impresor para reportes

Cualquier computadora con estas características mínimas puede desempeñar el papel de **Servidor** para el programa de la clínica.

## 2.6.2 REQUERIMIENTOS DE SOFTWARE PARA SERVIDOR Y COMPUTADORAS CLIENTES:

<b>Función a desempeñar</b>	<b>Sistema Operativo</b>	<b>Software de Desarrollo</b>	<b>Servidor de base de datos</b>
Servidor	Windows XP Pro o Linux cualquier versión	.Net Framework 2.0	MySQL Server 5.0 + Mysql connector .Net
Cliente	Windows XP Pro	.Net Framework 2.0	Mysql connector .Net

## 2.6.3 CONDICIONES RECOMENDADAS DE FUNCIONAMIENTO

### RECOMENDACIONES DE HARDWARE PARA UTILIZACIÓN DEL PROGRAMA:

<b>Identificación</b>	<b>Procesamiento Velocidad CPU</b>	<b>Capacidad de Almacenamiento</b>	<b>de Memoria RAM</b>	<b>Conectividad de Red</b>	<b>Otros</b>
Servidor	1.1 GHz en adelante	>=20 GB	256 MB	Tarjeta Ethernet 10/100 Mbps	Impresor HP 3920 Deskjet
Cliente	1.1 GHz en adelante	10 GB	256 MB	Ethernet 10/100 Mbps	

### RECOMENDACIONES DE SOFTWARE PARA UTILIZACIÓN DEL SISTEMA FINAL

1. Microsoft Windows XP Home o Professional Edition o en caso del servidor puede plantearse un sistema operativo linux para albergar la base de datos.
2. Microsoft .Net Framework 2.0
3. MySQL Server 2005 (Instalado solo en la computadora principal)

4. MySQL Connector Net (Instalado en todas las computadoras que ejecutarán el programa)

### **CAPÍTULO III**

## **3 DISEÑO DEL SISTEMA DE INFORMACIÓN**

### **3.1 DESCRIPCIÓN GENERAL DEL SISTEMA DE INFORMACIÓN**

#### **3.1.1 DISEÑO DE LOS MODULOS.**

Los módulos de la aplicación han sido desarrollados en base a un prototipo básico definido para este sistema. Cada módulo tiene su propio sistema de autenticación de usuarios, y estructuralmente la mayoría están compuestos del mismo grupo de pantallas que los otros módulos, desde el punto de vista de la distribución de pantallas y navegabilidad.

##### **Áreas de distinción de los módulos**

La diferencia principal entre cada módulo radica en las pantallas de captura de información para el nuevo registro y la actualización o modificación. Pues la mayoría de los módulos comparten un formulario especial para que ayude a la búsqueda de datos según el módulo en cuestión.

#### **3.1.2 DISEÑO DE PROTOTIPOS DE LA APLICACIÓN Y ACCESIBILIDAD DE LOS MODULOS.**

Para el diseño de las interfaces de los diferentes módulos se tomó como base un prototipo de interfaz que brinda ventajas de usabilidad, distribución de pantallas o formularios y navegabilidad, desde el punto de vista del usuario final y cumple con los requerimientos de funcionalidad para el tratamiento de la información.

Cada módulo está diseñado en función al prototipo y las pantallas principales en orden de aparición se describen en la siguiente sección:

El prototipo escogido para el diseño de los módulos se compone de las siguientes pantallas:

6. Pantalla de autenticación o ingreso a módulo.

7. Pantalla de resumen o presentación de registros.
8. Pantallas de agregación de nuevos registros.
9. Pantallas de modificación de registros existentes.
10. Pantallas de confirmación de eliminación de registros existentes.

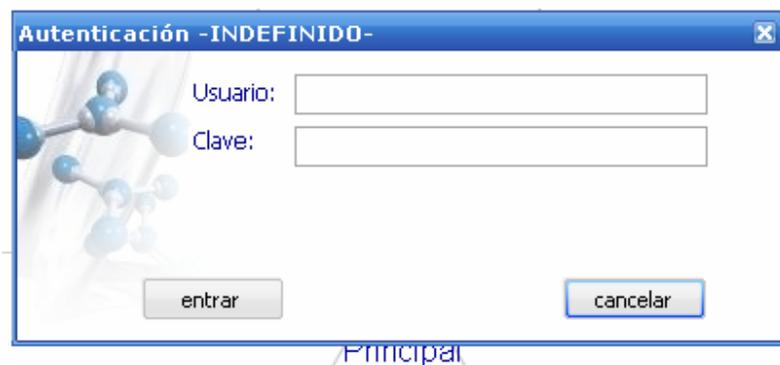
Un esquema general de la distribución de las pantallas se presenta a continuación:

### 3.1.2.1 DESCRIPCIÓN DE LA PANTALLA DE AUTENTICACIÓN O INGRESO AL MÓDULO.

La pantalla de autenticación de cada módulo está destinada a brindar seguridad al acceso de cada módulo permitiendo el ingreso al módulo a los usuarios registrados en el sistema. Cada módulo tiene restricciones de acceso. Básicamente el acceso del usuario a los diferentes módulos es dependiente del perfil del usuario previamente creado.

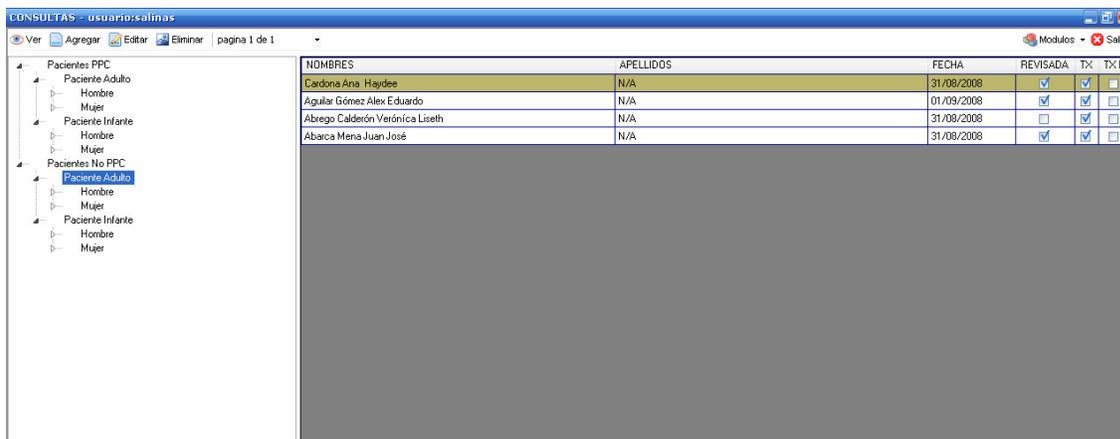
La pantalla está compuesta de los siguientes controles:

1. Textbox: para la captura del nombre de usuario y clave respectiva.
2. Botones: para iniciar el proceso de autenticación según el nombre de usuario y contraseña (aceptar) y para cancelar el proceso de autenticación (cancelar).
3. Etiquetas: que dan indicaciones respectivas respecto a cómo se debe de llenar el formulario de autenticación. También muestran mensajes de error en caso de autenticaciones fallidas, caso en el cual la etiqueta inferior cambia de color a rojo con el mensaje respectivo.



La autenticación exitosa lleva al usuario a la pantalla de presentación o resumen de registros de ese módulo. Dado que la rotación de personal en la clínica es sumamente elevada, el concepto de usuario en este sistema será utilizado de forma similar al concepto de sesión. Es decir, los operarios del sistema (estudiantes) rotan cada 11 días, por lo tanto, será creado en este sistema un número fijo de usuarios los cuales serán asignados a los respectivos estudiantes al momento de tener derechos sobre el uso del sistema.

### 3.1.2.2 DESCRIPCIÓN DE LA PANTALLA DE RESUMEN O PRESENTACION DE REGISTROS



NOMBRES	APELLIDOS	FECHA	REVISADA	TX	TX F
Cardona Ana	Haydee	31/08/2008	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Agular Gómez Alex	Eduardo	01/09/2008	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Abrego Calderón Verónica	Liseth	31/08/2008	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Abarca Mena Juan José		31/08/2008	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

En la pantalla de resumen de registros se muestran todos los registros guardados desde una fecha determinada hasta la actual, los registros dependen del módulo al cual se ha ingresado. Por ejemplo, si se ha ingresado al módulo de consultas, se mostrarán los registros de consultas elaborados basados en el filtro generado por la estructura de árbol de la izquierda. Si por el contrario, se ingresa al modulo de expedientes de pacientes, se mostrarán todos los registros de los expedientes de los pacientes registrados siempre basados en la estructura de árbol.

Los elementos principales de esta pantalla del prototipo de interfaces se enumeran a continuación:

1. Barra de título.
2. La barra de actividades.
3. La rejilla de registros.
4. El árbol de filtro de registros.
5. Paginador de registros

**La barra de título:** muestra el título del módulo al cual se ha ingresado después de la autenticación.

**La barra de actividades:** en esta barra de actividades están las principales actividades que se pueden efectuar sobre los registros de información, tales como: agregar un registro, modificar o actualizar un registro, También contiene los botones que llevan hacia otros módulos, tales como: reportes y configuraciones. Una característica especial es que estos enlaces a estos módulos responden de acuerdo a usuario que ha entrado al modulo estando visibles solo si el usuario autenticado es usuario privilegiado para usar el módulo de reportes o configuraciones. El cuando el usuario utiliza estos enlaces, la autenticación respectiva al ingreso del módulo actual se pasa como un parámetro a los módulos de reportes o configuraciones para que se registre también ahí el ingreso del usuario en cuestión.

**Rejilla o colección de registros:** en esta parte de la pantalla se presenta una colección de los registros guardados según el módulo al cual se haya ingresado. La funcionalidad es muy diversa, ya que a través del árbol de registros, el usuario se puede desplazar entre las diferentes fechas y registros y la rejilla se actualiza en función a la ubicación del usuario. Lo mismo ocurre cuando se ejecutan los filtros, búsquedas, etc.

**El árbol de filtro de registros:** esta en una representación de los registros, el cual permite verlos de manera agrupada por criterios diferentes, tales como: registros del día, semana o mes; registros de determinado grupo de pacientes como pacientes adultos, niños pertenecientes al PPC, niños no pertenecientes al PPC, etc.

**El paginador General:** cuya única finalidad es la de paginar los registros.

El principal objetivo de la pantalla de presentación de los registros es brindar la facilidad al usuario de poder visualizar los registros y las diferentes actividades que se pueden efectuar con ellos de manera simultánea.

Minimizar el tiempo de acceso a un registro a través de la navegación ordenada de los registros.

### **3.1.2.3 DESCRIPCION DE LAS PANTALLAS DE AGREGACION DE NUEVOS REGISTROS**

En las pantallas de agregación de nuevos registros se presenta una serie de formularios para el registro de nuevos registros a la base de datos, desde los diferentes módulos. Cada diseño

de secuencia de pantallas esta en función al modulo al cual se refiera, sea este expedientes, consultas, etc.

Cada uno de los módulos contiene un tipo distinto de diseño en estas pantallas, de acuerdo a los datos que se necesita registrar desde cada uno de ellos.

Pero la convención de controles utilizados es la misma que la descrita en la sección de DISEÑO DE MODULOS.

El acceso a este grupo de pantallas se tiene desde la pantalla de presentación o resumen de registros del módulo correspondiente.

La secuencia de pantallas de agregación para la captura de los datos finaliza de vuelta a la pantalla de presentación o resumen de registros del módulo.

#### **3.1.2.4 DESCRIPCION DE LAS PANTALLAS DE MODIFICACION DE REGISTROS EXISTENTES**

En las pantallas de modificación de registros se presenta una serie de formularios con la información detallada del registro seleccionado en el área de presentación o resumen de registros de la pantalla del mismo nombre, con el fin de dar facilidad a su actualización. Básicamente, estas pantallas corresponden a las mismas de agregación, salvo que se accede a ellas con datos cargados en modalidad para actualización.

El acceso a este grupo de pantallas se tiene desde la pantalla de presentación o resumen de registros del módulo correspondiente.

La secuencia de pantallas de modificación o actualización de los datos del registro seleccionado finaliza de vuelta a la pantalla de presentación o resumen de registros del módulo adecuadamente actualizado.

#### **3.1.2.5 DESCRIPCION DE LA PANTALLA DE ELIMINACIÓN O CANCELACIÓN DE REGISTROS.**

Desde la pantalla de presentación o resumen de registros del módulo correspondiente, es posible seleccionar un registro y marcarlo para su eliminación o cancelación de acuerdo al módulo en el que se está trabajando.

Antes de la eliminación o cancelación, se mostrará al usuario una pantalla de confirmación antes de proceder a realizar la acción.

### **3.1.3 ARQUITECTURA DEL SISTEMA**

#### **3.1.3.1 CENTRALIZADA CON PROCESAMIENTO DISTRIBUIDO**

El sistema desarrollado consiste en una aplicación del tipo cliente servidor, con las características típicas de un programa de tipo local, es decir que la interacción del usuario es tal cual es en un programa instalado localmente y que almacena los datos localmente.

Sin embargo todo será almacenado en un equipo remoto, es decir que será un sistema de red. La arquitectura en sí de este sistema es centralizada sin embargo el procesamiento de los datos es manejado distribuidamente entre todas las computadoras clientes que se conecten al servidor.

A continuación se presentan los pros y contras de este tipo de arquitectura.

Las ventajas de tener un sistema de bases de datos centralizadas son:

1. Requieren poco personal de administración
2. Siempre tiene los datos actualizados

Algunas desventajas son:

1. Al haber grandes volúmenes de información se afecta los tiempos de respuesta.
2. Dependencia de los sistemas de comunicación
3. Requiere que el servidor posea mejores capacidades de procesamiento para tener un buen rendimiento.

Al hablar del procesamiento distribuido se relacionan a los sistemas conectados en red de los que se pueden mencionar las siguientes ventajas:

1. Procesamiento paralelo (varios CPUs ejecutan la aplicación)
2. Permite el uso de interfaces gráficas variadas y versátiles
3. Los sistemas aparecen ante los usuarios como si fuera una única computadora.
4. Facilidad de crecimiento e inclusión de nuevos recursos.
5. Mayor velocidad de ejecución en los programas debido a que no sobrecargan el servidor.

Algunas desventajas son:

1. Se requieren mayores controles al procesar los datos para evitar incoherencia en la información
2. Puede llegar a existir saturación del servidor de la base de datos

Esta arquitectura ha sido considerada como la más idónea para este sistema, debido a los factores de trabajo simultáneo y de integridad de los datos, estos factores son totalmente englobados en este tipo de arquitectura.

### **3.1.4 DISEÑO DE ENTRADAS DE DATOS AL SISTEMA**

En el diseño de entradas al sistema se ha buscado la facilidad en el sistema para que los usuarios ingresen la información, de manera que se haga de la forma más rápida y sencilla posible y que a la vez se registren todos los datos de las operaciones que se están ejecutando.

### **Textbox**

Se utilizaron las cajas de texto como principales controles para el ingreso de datos, las entradas de datos serán validadas en el momento que pierdan el foco; esto para garantizar que se ingresen datos acordes a lo que se almacenará en la base de datos.

### **Listbox y Combobox**

Estos controles se utilizarán para los casos en que se necesiten selecciones por parte del usuario, principalmente cuando los datos que se pueden ingresar ya se encuentran definidos previamente por el personal de la clínica.

Cabe mencionar que la mayoría de estas opciones de estos controles serán configuradas desde el módulo de configuración.

### **Checkbox**

Estos controles solo se utilizarán para la representación de datos booleanos, en los cuales solo existe el True o False.

### **Controles de examinar**

Se utilizarán estos controles principalmente para el área de las radiografías, dada la necesidad de almacenar esa imagen en la base de datos. La otra forma que se tiene previsto utilizar dentro del sistema, es la posibilidad de inclusión de imágenes digitales dentro de la base de datos, es decir que se permitirá elegir las imágenes resultantes del escaneo de las radiografías.

La forma de trabajo para este método de entradas de datos es la siguiente:

1. Se contará con un scanner normal, el cuál utilizarán para escanear las placas de

las radiografías, de esa forma se obtiene un archivo en formato digital con una resolución no mayor a 300 ppp, esto debido al tamaño en disco.

2. El Operador del sistema busca esa imagen desde la pantalla de la consulta y la agrega al registro de radiografías.
3. Una vez agregada se almacenará en la base de datos.

### **3.1.5 DISEÑO DE SALIDAS DEL SISTEMA**

En cualquier sistema en el cual se registren datos, se debe presentar la información ya sea en pantalla, o a través de un reporte impreso.

Las consultas de datos representan una parte vital de este sistema, es por eso que se ha intentado preparar la base de datos con todas las reglas de normalización y de integridad referencial para que en ningún momento se rompa la coherencia de los datos y por lo tanto la información que se presente sea correcta.

#### **3.1.5.1 SALIDAS EN PANTALLA**

La pantalla de resumen y filtro de los módulos presenta la colección de registros estableciendo filtros lógicos predeterminados según el módulo en mención. Todas las salidas en pantalla en su mayoría se realizan utilizando DataGridView.

#### **3.1.5.2 SALIDAS IMPRESAS**

En esta área del sistema se pretende proporcionar diversidad de reportes para impresión que simplifiquen la obtención de los datos.

La presentación de resultados estadísticos es una de las principales tareas tediosas del personal administrativo de la clínica, pero desde el momento en que se implemente este sistema no lo será más.

Los reportes generados en el sistema van desde el listado de pacientes hasta las estadísticas de atención a pacientes en todo el ciclo o incluso todo el año.

A continuación se presenta el listado de los reportes que generarán en el sistema.

#### Referentes a Consultas

- Reporte de expedientes
- Reporte de consultas y tratamientos de un paciente

#### Referentes a Citas

- Detalle de citas por fechas

#### Referente a Finanzas

- Detalle de Estado de cuenta del paciente
- Emisión de recibos por la consulta (dependerá de la disponibilidad de los mismos)

#### Referente a estadísticas

- Consultas atendidas por período de tiempo
- Pacientes atendidos por período de tiempo

#### Referentes a PPC

- Listado de niños por instituciones
- Listado actividades planificadas para atención de PPC

#### SALIDAS POR ARCHIVOS

A través del cristal Reports, es posible tener las siguientes opciones de salida para los reportes: Word, Excel, csv, pdf.

### **3.1.6 FACTIBILIDAD OPERATIVA**

Para determinar la factibilidad operativa de la implementación del sistema, se hará basado en un perfil de usuario del sistema; el perfil de usuario presenta las características y conocimientos básicos que debe tener una persona que va a utilizar el sistema.

Esto se hace para determinar si el personal operario de la clínica es capaz de adaptarse a un entorno de trabajo con gran apoyo en la informática.

### **3.1.6.1 DEFINICIÓN DEL PERFIL**

Como perfil de usuario se establecen las siguientes características básicas que debe cumplir una persona destinada a interactuar con el programa.

- Conocimiento de las partes principales de una computadora (monitor, cpu, teclado, Mouse)
- Conocimiento de los cuidados que se debe tener al manipular una computadora
- Experiencia utilizando un programa de interfaz de ventanas
- Experiencia llenando información en un programa informático.
- Habilidad escribiendo mecanográficamente (No necesario pero útil)

Una forma mas sencilla de determinar la aptitud de los usuarios para interactuar con el sistema puede ser verificar si cumple con los requisitos de la certificación grado digital del Ministerio de Educación; sin embargo con los ítems arriba mencionados es suficiente.

Después de consultar durante los últimos 3 meses con los estudiantes, los doctores y el personal de administración de la clínica sobre sus habilidades para trabajar con computadoras se ha comprobado que se cumple en casi un 100% con las expectativas del proyecto respecto a los usuarios.

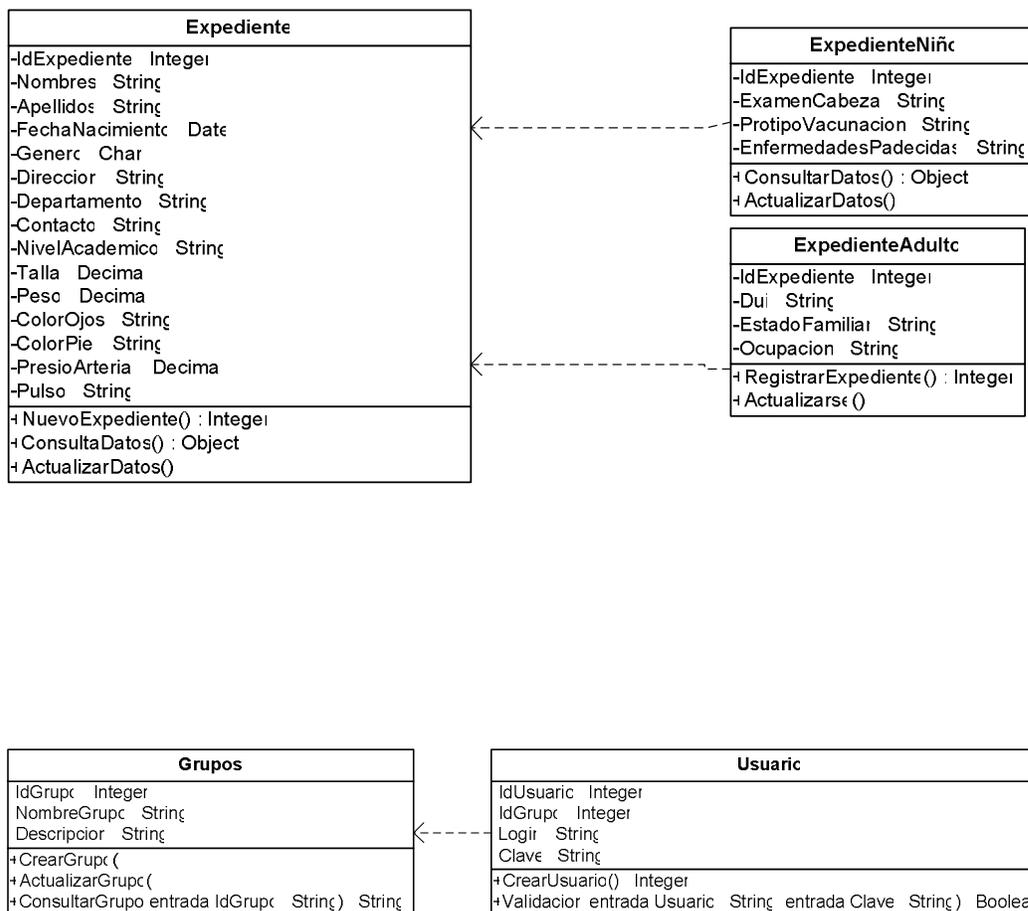
En este sentido se puede dar por factible la implementación de un sistema informático como un elemento central del funcionamiento de la clínica, además que el diseño del

sistema es del tipo user-friendly, o amigable al usuario, por lo cual no se espera tener problemas con su utilización.

### 3.2 DIAGRAMACIÓN DEL SISTEMA

A continuación se presentan los principales diagramas UML que se utilizarán para el desarrollo del sistema.

#### 3.2.1 DIAGRAMA DE CLASES.



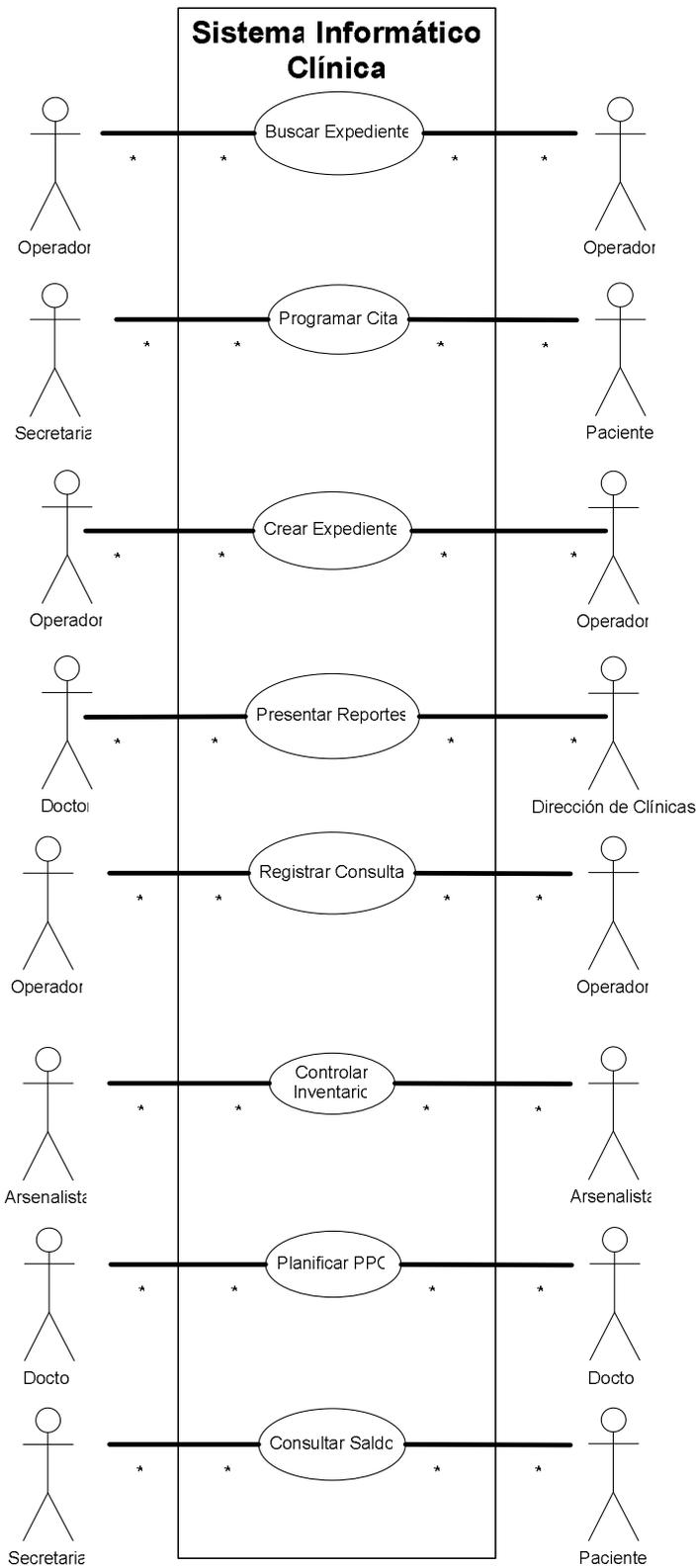
Consulta
-IdConsulta Integer -FechaConsulta Date -HoraConsulta -Motivo String
+ RealizarConsulta(entrada IdExpediente Integer entrada IdUsuario Integer entrada IdOperador Integer) + BuscarConsulta(entrada IdExpediente Integer) : String

Citas
-IdCita Integer -IdExpediente Integer -Fecha Date -HoraCita Date
+ CrearCita(entrada IdExpediente Integer) : Integer + BuscarTiempo() Date

Operadores
-IdOperador Integer -NombresOperador String -ApellidosOperador String -IdUsuario Integer -IdGrupo Integer -FechaNacimiento Date -CicloAcademico String -NumeroCarnet String -InicioRotacion Date -FinRotacion Date -Evaluacion Decimal
+ CrearRegistro() Integer + ActualizarRegistro() + AtenderConsultas()

Doctores
-IdDoctor Integer -NombresDoctor String -ApellidosDoctor String -IdUsuario Integer -IdGrupo Integer -CargoDoctor String
+ CrearRegistro() Integer + ActualizarRegistro() + AprobarTratamiento()

### **3.2.2 DIAGRAMAS DE CASOS DE USO.**



### **3.2.3 DIAGRAMAS DE ACTIVIDADES**

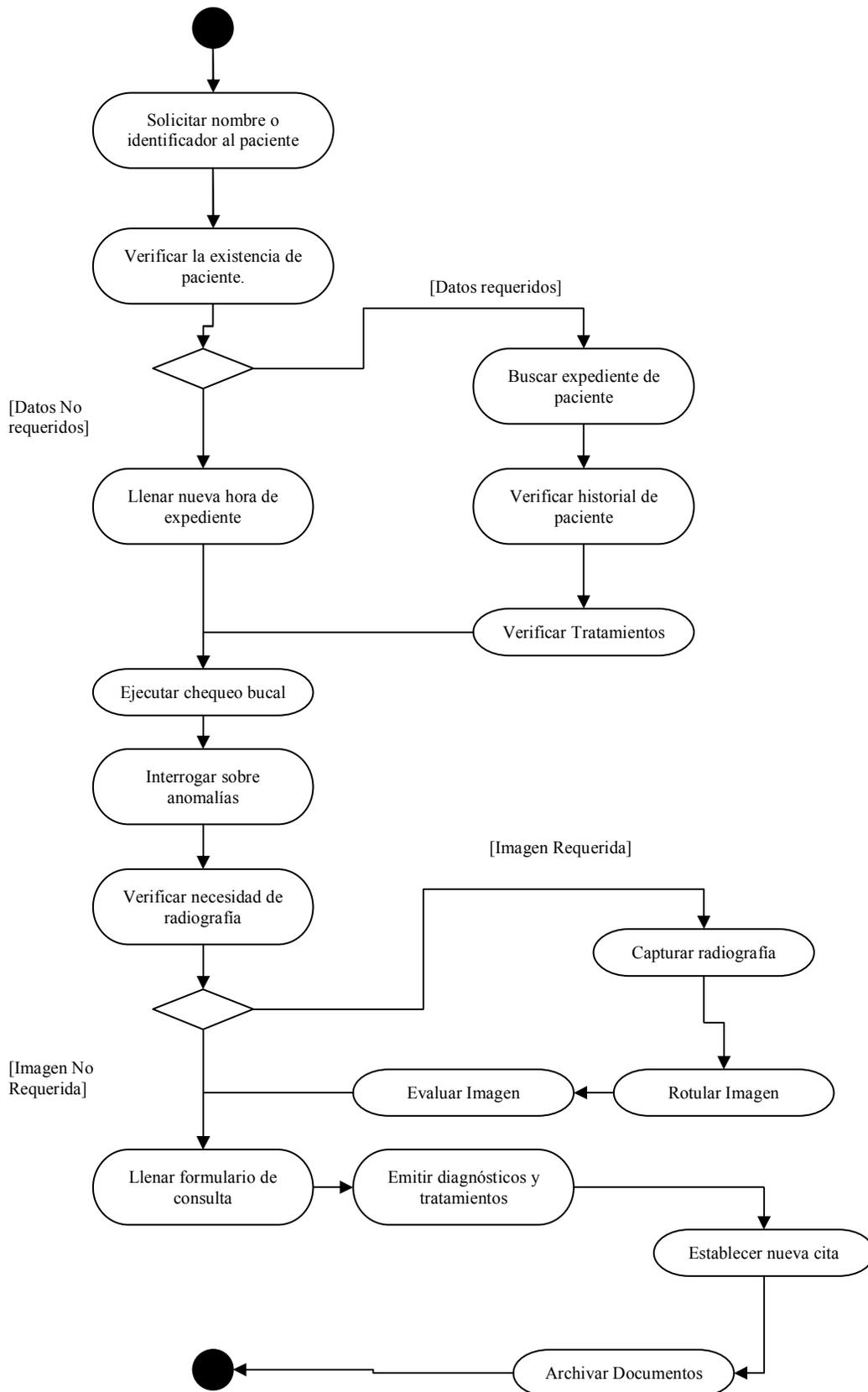
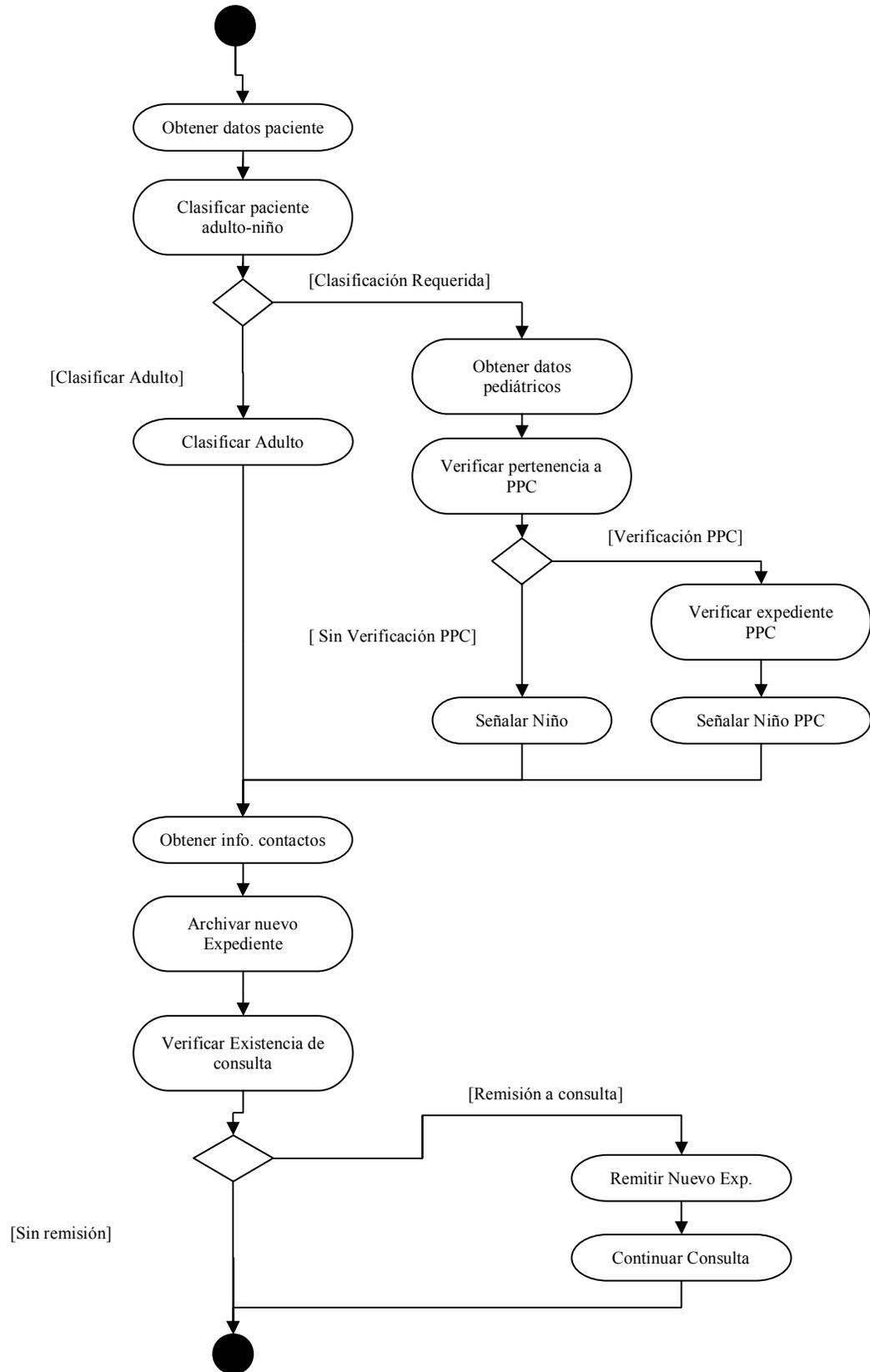
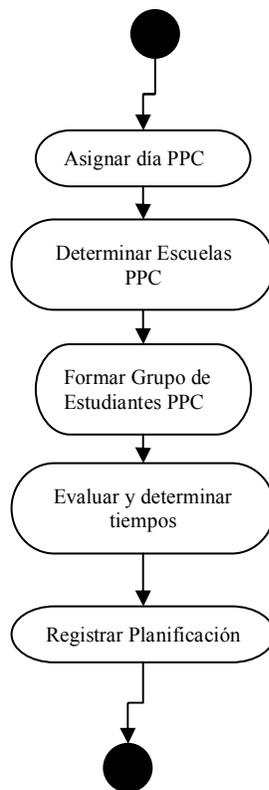


DIAGRAMA DE ACTIVIDADES – EJECUCIÓN DE CONSULTA.

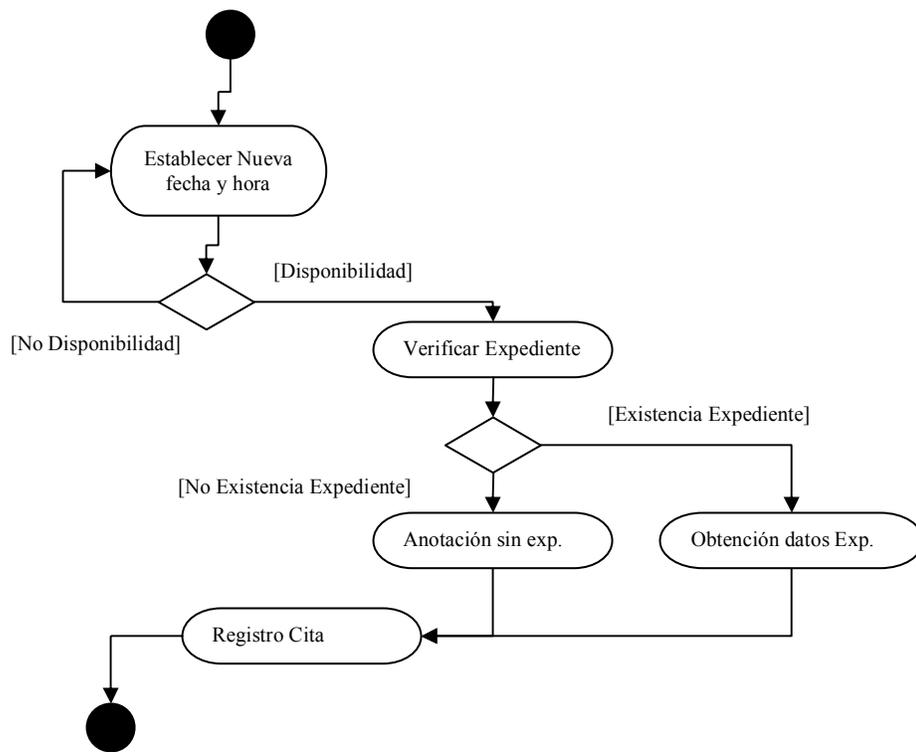
# DIAGRAMA DE ACTIVIDADES – APERTURA DE EXPEDIENTE



## DIAGRAMA DE ACTIVIDADES – ASIGNACION DE ACTIVIDADES PP



# DIAGRAMA DE ACTIVIDADES – ASIGNACION DE NUEVA CITA.



### **3.3 DISEÑO DE LA BASE DE DATOS**

#### **3.3.1 DESCRIPCIÓN DEL SISTEMA GESTOR DE BASES DE DATOS**

##### MySQL 5.0

Para el desarrollo del sistema de la clínica de Odontología, se necesita un sistema de bases de datos que permita el almacenamiento y administración de los datos de forma óptima y segura.

El Sistema Gestor de Bases de datos que se ha decidido utilizar es MySQL Server 5.0, para lo cuál se han analizado todas las ventajas que ofrece y comparado con las necesidades que presenta el desarrollo del sistema.

Al hablar de un Sistema Gestor de Bases de Datos, inmediatamente se puede pensar en nombres como Oracle, SQL Server, y también es muy conocido el nombre de MySQL, aquí se presentará una breve descripción de lo que es MySQL Server 5.0. MySQL Server es un Servidor de Bases de Datos opensource y con una arquitectura relacional. Su desarrollo ha tenido una larga trayectoria como uno de los proyectos más representativos de la comunidad opensource. Actualmente MySQL Server se encuentra en su versión 5.0 que ha representado un salto por parte de este sistema hacia un entorno de bases de datos completo y capaz de entrar en proyectos de carácter crítico.

Las tres prioridades de los creadores de MySQL son: *Confiabilidad*: Denotada por la poca o nula intervención requerida para poder alcanzar los niveles de servicio ininterrumpidos que se requieren. *Desempeño*: Caracterizado por ofrecer un gran desempeño y tiempos de respuesta muy rápidos en cualquier interacción con las bases de datos. *Fácil de utilizar*: La instalación y configuración de MySQL Server no toma más de 15 minutos, y su

mantenimiento y ejecución de tareas administrativas requieren muy poco tiempo de intervención.

Entre los beneficios que presenta la integración de nuestros sistemas con MySQL Server se pueden encontrar:

#### Diseño escalable

- Fácil instalación
- Arquitectura con un motor de almacenamiento único
- Fuerte modelo de seguridad
- Alta disponibilidad permitiendo integración de cluster
- Altos desempeños gracias al uso de memorias caché

Además MySQL 5.0 permite la utilización de transacciones, las cuales son interacciones con estructuras de datos, y que funcionan de la forma “todo o nada”, con lo cual se garantiza la conservación de la integridad de los datos.

Ejemplos de las características para el manejo de transacciones son:

Server Based Data Integrity: Con esta característica se puede rechazar automáticamente fechas erróneas, números inválidos, etc.

Automatic Storage Management: Con esta característica es el mismo servidor el encargado de administrar el espacio para almacenamiento de las bases de datos, además de compactar el espacio en bases de datos que han sufrido eliminación de registros.

Online Backup and Point-in time recovery: Un servidor MySQL es capaz de realizar backups de sus datos en tiempo real sin afectar el funcionamiento de cualquier operación que se pueda realizar en ese momento. Además de ofrecer la restauración de la base de datos en un punto del tiempo con uso de binary logging.

## Data Warehouses

Si se necesitan Almacenes de datos, MySQL es ideal debido a su capacidad de crecimiento, su capacidad para manejar bases de datos de hasta terabytes y por ofrecer en la recuperación y almacenamiento de datos velocidades increíbles.

### Nuevas características a partir de MySQL 5.0

- Posee la capacidad de utilizar procedimientos almacenados
- Triggers o desencadenantes
- Mejoras sustanciales en cuanto a la conectividad con la inclusión de conectores que pueden usarse en el desarrollo de aplicaciones que van a utilizar bases de datos mysql.
- Procesamiento para transacciones distribuidas
- Vistas de datos que permiten una mejor administración de la seguridad y rendimiento en los datos
- Diccionario de datos
- Mejoras importantes en las herramientas de administración gráficas.
- Herramienta para migración de bases de datos ya sea de Oracle o MS SQL Server.

MySQL ha demostrado que es capaz de ser la columna vertebral de cualquier negocio o sistema informático que necesite rendimiento y seguridad en los datos y es por eso que no hay dudas respecto a la efectividad de su implementación en este proyecto.

### **3.3.2 ARQUITECTURA DE LA BASE DE DATOS**

Primordialmente el sistema informático de la clínica será diseñado en base a la arquitectura de bases de datos centralizadas, y con procesamiento distribuido. Esto se refiere a que el repositorio principal de las bases de datos no será otro que el servidor principal, sin embargo el procesamiento de los datos será ejecutado por la aplicación instalada en las computadoras clientes, permitiendo a cada computadora con el sistema instalado manipular los datos según tenga privilegios para hacerlo.

Dentro del diseño de la arquitectura de las bases de datos que se ha realizado el análisis de cómo elegir el modelo de datos más adecuado.

Según el modelo relacional se pueden dividir los sistemas de bases de datos en 4 módulos, cada uno con la intención de satisfacer distintas necesidades, sin embargo para este proyecto solo se han considerado el análisis de dos de ellos como posibles a utilizar; esto se basa en el volumen de datos que se manejarán con el sistema de la clínica.

Para ello se introducen los términos de Procesamiento de transacciones en Línea (OLTP por sus siglas en inglés) y de Almacén de datos Operacionales (ODS por sus siglas en inglés).

Las bases de datos para procesamiento de transacciones en línea, son las que existen en todos los sistemas para el manejo de los datos de las operaciones corrientes de los programas. Es decir que son las bases de datos sobre las que funcionan los sistemas de información. No se debe confundir este tipo de bases de datos con la utilización de transacciones, sino más bien se refiere a que son las bases de datos sobre las cuales descansa el funcionamiento diario del sistema.

Están diseñadas y optimizadas con el propósito de recibir cualquier número de transacciones de usuarios, haciendo inserciones, modificaciones o eliminación de datos existentes.

Toda la información de los sistemas debe ser almacenado o ha sido almacenado en este tipo de bases de datos, que es el caso de las bases de datos históricas.

El principal objetivo de estas bases de datos es la integridad de los datos que se logra gracias a los dos puntos siguientes:

- Almacenamiento de cada porción de datos en un mismo repositorio, de manera que al ser editados cualquier cambio se refleja por igual en cualquier lugar que se utilice.
- Proporcionar soporte de transacciones, de manera que cualquier alteración en múltiples bases de datos tienen que hacerse de forma conjunta, y si una de las transacciones falla todas las demás no deben ser permitidas de aplicarse.

La desventaja de estas bases de datos es que están diseñadas pensando en el rendimiento y facilidad de escritura de los datos, en cambio la obtención de información de utilidad puede ser a veces muy compleja. Esto debido a la normalización de las tablas de estas bases de datos, lo cual permite mayor eficiencia e integridad a la hora de realizar cambios a estas tablas, pero muchas veces es necesario realizar consultas complejas que involucran muchas tablas para obtener un simple registro completo que se presentará en un reporte.

Las bases de datos del tipo Almacén de Datos operacionales, son diseñadas con el fin de tener una base de datos en la cuál se puedan obtener datos útiles para la ejecución de reportes diarios, semanales, mensuales, etc.

Las OLTP poseen una gran cantidad de registros con mucho detalle, y es muy probable que la información se halle distribuida en varias tablas (producto de la normalización), por lo que se debe recorrer todas estas tablas y sus registros para mostrar un registro completo o mostrar totales provenientes de esas tablas.

En los casos donde se generan estos reportes constantemente se puede presentar una saturación y respuesta más lenta del servidor, tomando en cuenta que no solo está realizando ese proceso sino también insertando, modificando y eliminando registros como parte del trabajo normal del sistema.

Allí es donde se utilizan las ODS (Almacén de Datos operacionales) por sus siglas en ingles. Si se utilizaran estas bases de datos como un complemento a las OLTP se podría tener datos ya totalizados, y fáciles de consultar para generar un reporte.

Sin embargo analizando el volumen de datos de la clínica se ha determinado que es bajo y dada la velocidad proporcionada por MySQL Server, la carga de generar reportes directamente de las OLTP no se percibe la necesidad de utilizar las bases de datos ODS.

### **3.3.3 CONECTIVIDAD DE LA BASE DE DATOS**

La conectividad que se posee actualmente para MySQL Server en cuanto al acceso de las aplicaciones, o específicamente al desarrollo de software que utilice éste servidor de bases de datos puede comprender diferentes formas, dependiendo siempre de los lenguajes de desarrollo que se quieran utilizar:

Conectividad a través de drivers ODBC

MySQL provee soporte para conexiones por ODBC usando el MySQL Connector/ODBC.

Conectividad con plataforma .NET a través del MySQL Connector/.NET

El Connector/.NET MySQL permite a los desarrolladores crear fácilmente aplicaciones .NET que requieren seguridad, alto rendimiento y conectividad con MySQL. Este implementa las interfaces requeridas de ADO.NET y se integra dentro de las herramientas de ADO.NET. Los desarrolladores pueden construir aplicaciones usando su mejor opción en lenguajes .NET. El Connector/.NET es un controlador ADO.NET completamente manejado escrito al 100% en C#.

El Connector/Net MySQL incluye soporte completo para:

- Características de MySQL 5.0 (Procedimientos almacenados, Etc.)
- Características de MySQL 4.1 (Sentencias preparadas del lado del servidor, Unicode y Acceso a memoria compartida, etc.)
- Soporte de paquetes grandes para enviar y recibir filas y datos de más de 2 gigabytes de tamaño.

- El protocolo de compresión que permite la compresión de flujos de datos entre el cliente y el servidor.
- Soporte para conectar usando sockets TCP/IP, tuberías, o memoria compartida en Windows.
- Soporte de conectividad usando sockets TCP/IP o sockets Unix en Unix
- Soporte para el framework de código abierto Mono, desarrollado por Novell.
- Completamente manejable, no utiliza las librerías cliente de MySQL.

#### MySQL Connector/J

MySQL proporciona conectividad para aplicaciones clientes desarrolladas en el lenguaje de programación Java vía el JDBC driver, que se llama MySQL Connector/J.

MySQL Connector/J es un driver JDBC-3.0 "Tipo 4", lo que significa que es puramente Java, implementa la versión 3.0 de la especificación JDBC, y se comunica directamente con el servidor MySQL usando el protocolo MySQL.

Considerando que el sistema propuesto se quiere desarrollar utilizando Visual Basic .NET, se tiene la ventaja de contar con el conector específico para esta plataforma, lo cual facilitará la integración de cualquier módulo con las bases de datos de MySQL.

### **3.3.4 SEGURIDAD DE LA BASE DE DATOS**

Dentro de los principales puntos que se deben analizar y planificar, se encuentra la seguridad de los datos. Cuales serán las medidas a implementar en el sistema para asegurar la integridad y evitar que los datos sean accesibles a cualquier persona.

**AUTENTIFICACIÓN:** Como primera medida de seguridad se encuentra la autenticación de usuarios para permitir el acceso, esto requiere por tanto una buena administración de usuarios y contraseñas para acceso al servidor.

**ASIGNACIÓN DE PRIVILEGIOS:** La asignación de privilegios en MySQL es realizada por el usuario administrador o por otro usuario que tenga privilegios para asignarlos; la tarea de asignación de privilegios debe ser manipulada cuidadosamente ya que es importante asignar sólo aquellos privilegios que serán sumamente necesarios para la realización de las actividades de un determinado usuario.

Como política interna para el desarrollo del sistema, solo existirá un usuario administrador con privilegios para esta tarea.

**RESPALDOS:** Como se pudo ver en la descripción de MySQL, existe la posibilidad de realizar diferentes tipos de respaldos a las bases de datos, la principal forma es mediante Online Backup and Point-in time recovery: Un servidor MySQL es capaz de realizar backups de sus datos en tiempo real sin afectar el funcionamiento de cualquier operación que se pueda realizar en ese momento. Además de ofrecer la restauración de la base de datos en un punto del tiempo con uso de binary logging

Como medida de seguridad en el sistema se tendrá una opción de respaldo de la base de datos para que el personal autorizado de la clínica pueda hacer ese respaldo.

### 3.3.5 MODELO ENTIDAD RELACIÓN

Como ya se mencionó anteriormente el sistema de bases de datos a utilizar posee una arquitectura relacional, estas bases de datos como su nombre lo indica se basan en relaciones, principalmente entre tablas, con lo cual se persigue mantener una integridad en todos los datos almacenados.

El diseño de las tablas se hará utilizando las reglas de normalización que han sido establecidas pensando en la optimización de las bases de datos relacionales, logrando así un óptimo registro de los datos y facilitando su administración.

Además de las reglas de normalización se utilizarán las claves, tanto primarias como foráneas, a continuación se brinda una pequeña descripción de los principales tipos de claves existentes en MySQL Server:

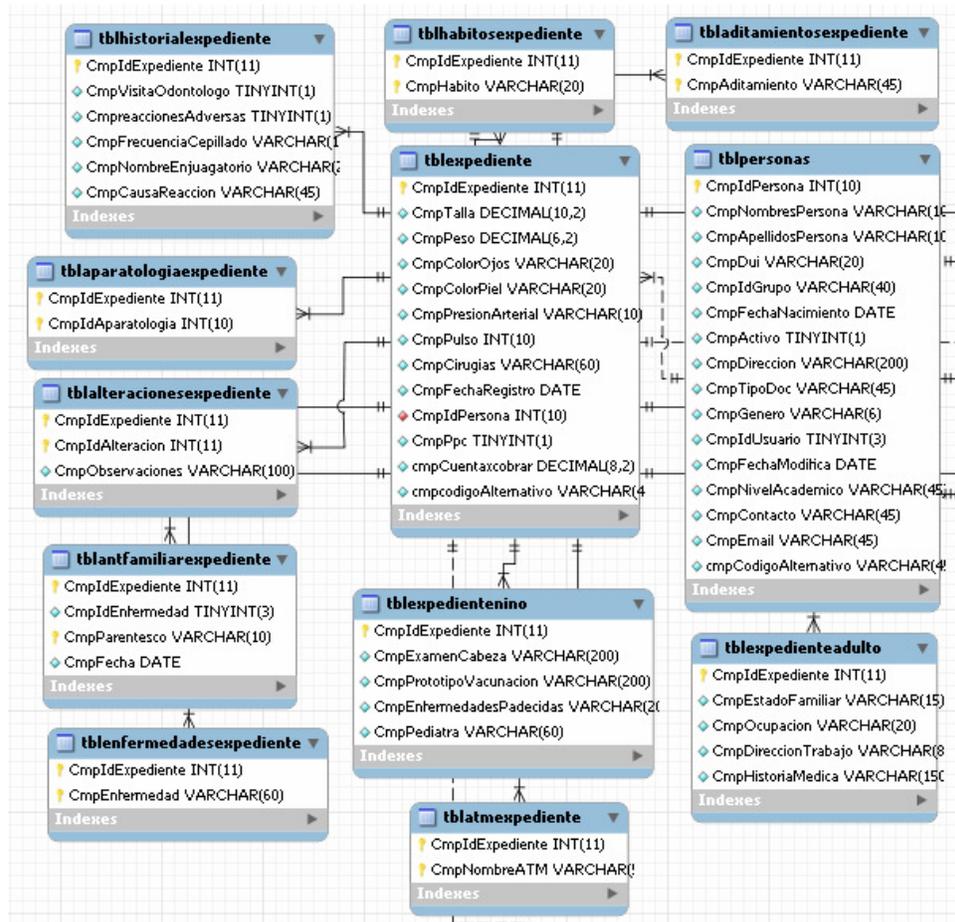
- Clave candidata: Es un campo o una combinación de campos, que identifican de manera única un registro de una tabla.
- Clave primaria: Es una clave candidata que ha sido diseñada para identificar de manera única los registros de una tabla, a través de toda la estructura de la base de datos.
- Clave foránea: Es simplemente un campo en una tabla que se corresponde con la clave primaria de otra tabla.

Integridad referencial: De manera simple, se dice que la integridad referencial significa que cuando un registro en una tabla haga referencia a un registro de otra tabla, el registro correspondiente debe existir.

Las claves foráneas tienen que ver precisamente con la integridad referencial, lo que significa que si una clave foránea contiene un valor, ese valor se refiere a un registro existente en la tabla relacionada.

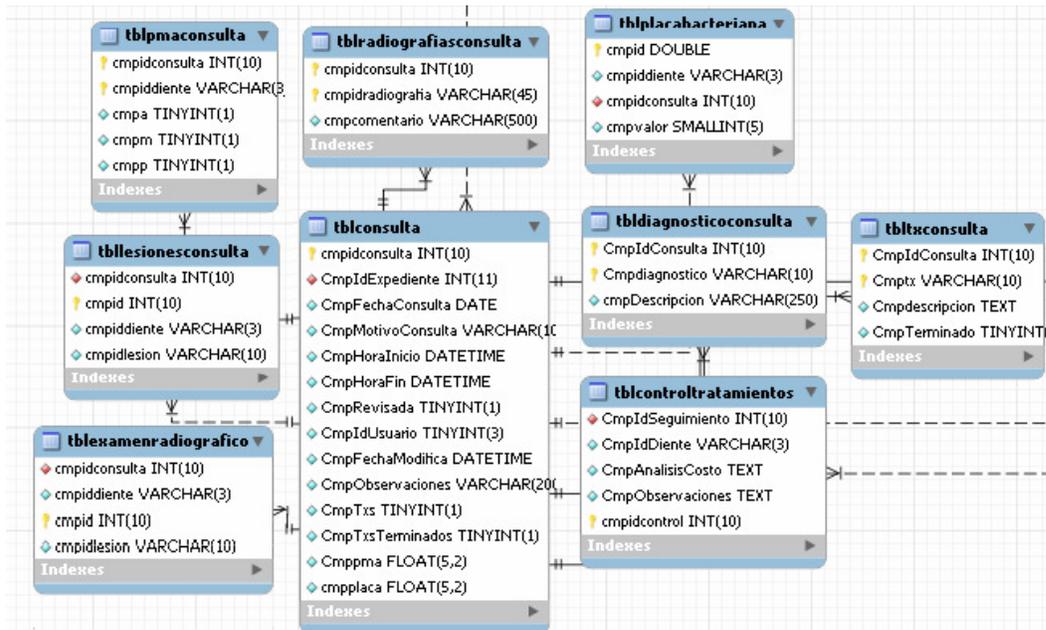
A continuación se presenta los principales diagramas de Entidad-Relación para la base de datos del sistema:

### DIAGRAMA DE EXPEDIENTES PACIENTES



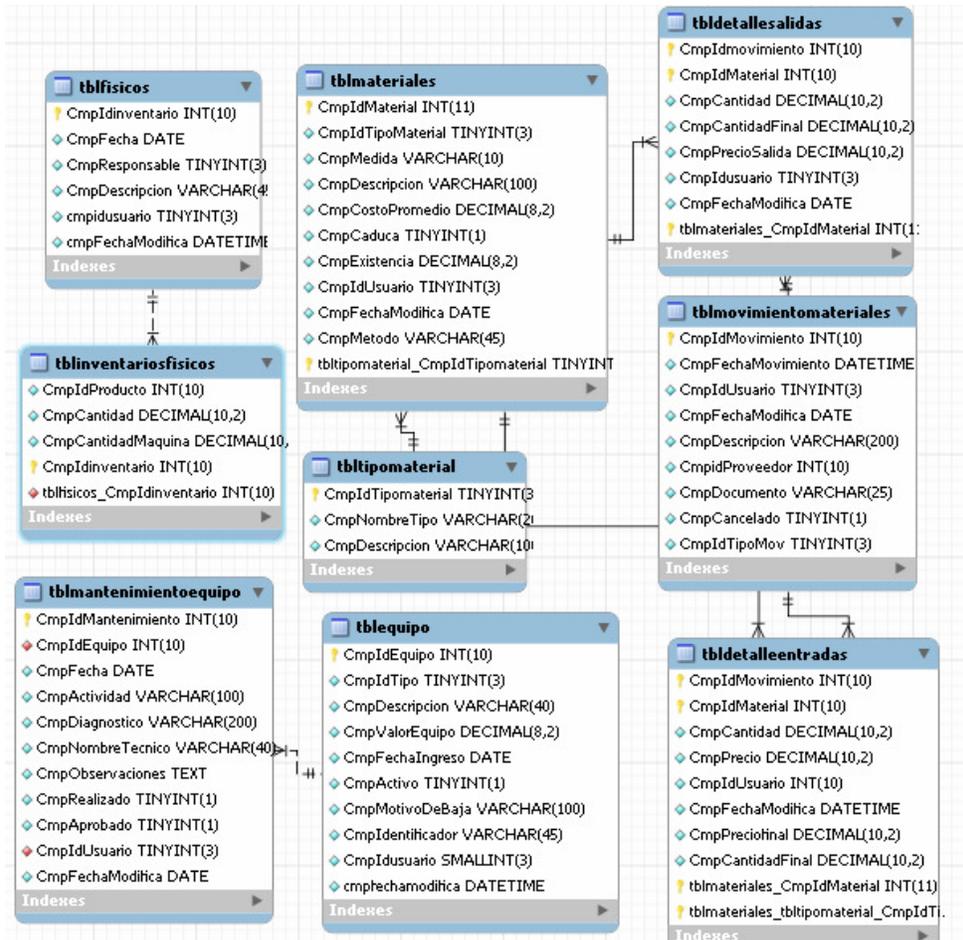
Un área fundamental de la base de datos es todo lo relacionado con los expedientes de los pacientes, en el diagrama anterior se muestran las tablas más importantes relacionadas al almacenamiento de los datos referentes al expediente de cada paciente.

## CONSULTAS



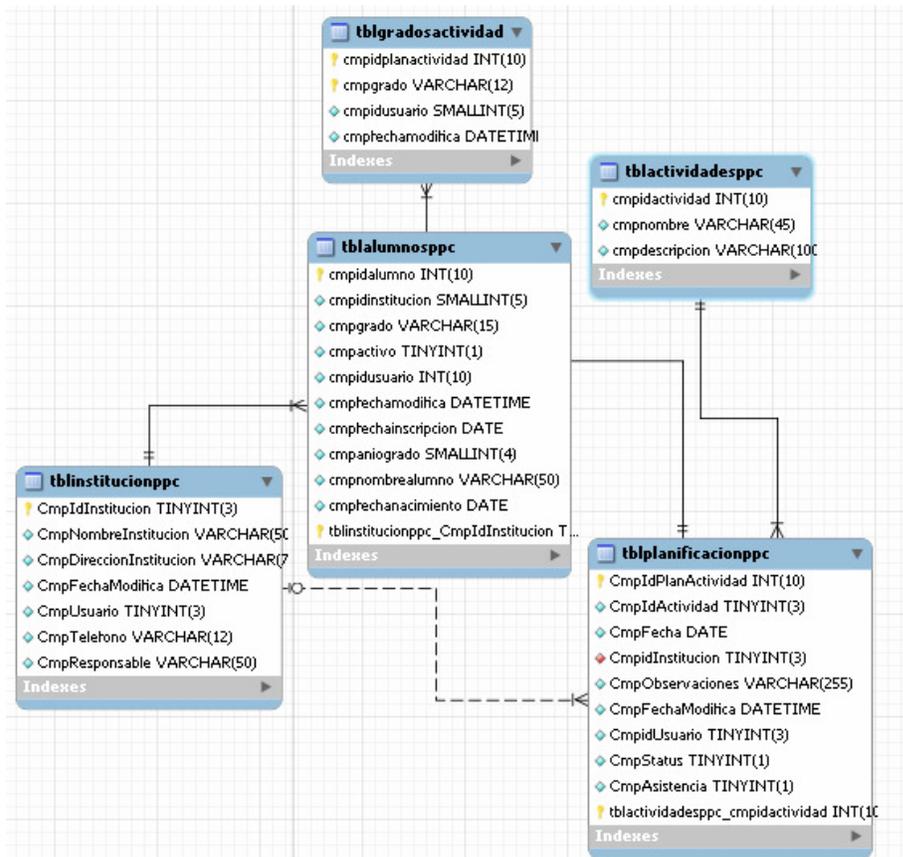
La siguiente área importante para el sistema de la clínica es la referente a las consultas y tratamientos; en el diagrama anterior se puede ver las principales tablas que se relacionan para mantener el registro de todos los datos de cada consulta y cada tratamiento realizado a todos los pacientes.

## INVENTARIOS



La siguiente área importante dentro de la base de datos de este sistema es la referente al inventario de materiales y equipo; en el diagrama anterior se puede visualizar las principales tablas y relaciones existentes en la base de datos para manejar el control de stocks y movimientos de materiales, así como para controlar el equipo existente en la clínica.

## DATOS PPC



Finalmente la cuarta área mas importante dentro de la base de datos es la relacionada al plan de prevención comunitaria, que se realiza externamente a la clínica pero que influye de gran manera en las actividades de la misma.

En el diagrama anterior se muestran las principales tablas, y sus respectivas relaciones que permiten registrar todos estos datos.

### 3.3.6 DESCRIPCIÓN DE LAS TABLAS PRINCIPALES Y DICCIONARIO DE DATOS DEL SISTEMA

Convenciones utilizadas para nombrar a las tablas y campos de las mismas dentro de la base de datos, todos los tipos de datos que se han utilizado han sido tomados de la documentación oficial de MySQL Server.

Nomenclatura: Las tablas del sistema llevan el prefijo “Tbl” como parte del nombre de la tabla, seguida por la(s) palabra(s) que identifique a que se refiere esa tabla.

Los campos de todas las tablas llevan como prefijo “Cmp”, y al igual que los nombres de las tablas le siguen las palabras que identifiquen el contenido para ese campo.

#### Tabla Expediente

Esta tabla contiene los datos principales y estándares de todos los pacientes de la clínica, en esta tabla se asigna el Número de ID que se utilizará como identificador del paciente en todas las tablas del sistema que lo requieran, además contiene la información inicial sobre la dentadura del paciente.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRESICION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdExpediente	int(11)	NO		10	PRI	auto_increment	select,insert,update,references
2	CmpNombres	Varchar(30)	NO	30				select,insert,update,references
3	CmpApellidos	Varchar(30)	NO	30				select,insert,update,references
4	CmpFechaNacimiento	Date	NO					select,insert,update,references
5	CmpGenero	char(1)	NO	1				select,insert,update,references

6	CmpDireccion	Varchar(100)	NO	100				select,insert,update,references
7	CmpDepartamento	Varchar(12)	YES	12				select,insert,update,references
8	CmpContacto	Varchar(30)	YES	30				select,insert,update,references
9	CmpNivelAcademico	Varchar(25)	YES	25				select,insert,update,references
10	CmpTalla	float(4,2)	YES		4			select,insert,update,references
11	CmpPeso	float(6,2)	YES		6			select,insert,update,references
12	CmpColorOjos	Varchar(10)	NO	10				select,insert,update,references
13	CmpColorPiel	Varchar(20)	NO	20				select,insert,update,references
14	CmpPresionArterial	Varchar(10)	YES	10				select,insert,update,references
15	CmpPulso	Varchar(5)	YES	5				select,insert,update,references
16	CmpCirugias	Varchar(60)	YES	60				select,insert,update,references
17	CmpIdUsuario	tinyint(3) unsigned	NO		3			select,insert,update,references
18	CmpFechaRegistro	date	NO					select,insert,update,references
19	CmpFechaModifica	datetime	NO					select,insert,update,referentes
20	CmpEliminado	Bool	NO					select,insert,update,referentes

### Tabla ExpedienteAdulto

En esta tabla se almacenan los datos específicos de un paciente adulto, ej. Lugar de trabajo, estado familiar, etc.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdExpediente	int(11)	NO		10	PRI	auto_increment	select,insert,update,references

2	CmpDui	varchar(10)	NO	10				select,insert,update,referentes
3	CmpEstadoFamiliar	varchar(15)	NO	15				select,insert,update,referentes
4	CmpOcupacion	varchar(20)	YES	20				select,insert,update,referentes
5	CmpDireccionTrabajo	varchar(80)	YES	80				select,insert,update,referentes
6	CmpHistoriaMedica	varchar(150)	YES	150				select,insert,update,referentes
7	CmpEliminado	Bool	NO					select,insert,update,referentes

### Tabla ExpedienteNinos

En esta tabla se almacenan los datos específicos de un paciente menor de edad, además de información relacionada con las enfermedades en período de embarazo de la madre. Ej. Institución educativa, grado, persona responsable, etc.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdExpediente	int(11)	NO		10	PRI	auto_increment	select,insert,update,referentes
2	CmpExamenCabeza	varchar(200)	YES	200				select,insert,update,referentes
3	CmpPrototipoVacunacion	varchar(200)	YES	200				select,insert,update,referentes
4	CmpEnfermedadesPadecidas	varchar(200)	YES	200				select,insert,update,referentes
5	CmpPediatria	varchar(60)	YES	60				select,insert,update,referentes
6	CmpPPC	tinyint(1)	YES		3			select,insert,update,referentes
7	CmpEliminado	Bool	NO					select,insert,update,referentes

### Tabla Enfermedades

En esta tabla se almacenarán las diferentes enfermedades distinguidas según la clasificación previa de la clínica, sobre las cuales se preguntará en antecedentes familiares.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdEnfermedad	tinyint(3) unsigned	NO		3	PRI	auto_increment	select,insert,update,references
2	CmpNombre	varchar(25)	NO	25				select,insert,update,references
3	CmpDescripcion	varchar(60)	YES	60				select,insert,update,references

### Tabla AntecedentesFamiliares

Esta tabla se utilizará para almacenar los diferentes antecedentes familiares de un paciente, tendrá relación con el expediente.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdExpediente	int(11)	NO		10	PRI	auto_increment	select,insert,update,references
2	CmpIdEnfermedad	tinyint(3) unsigned	NO		3	PRI		select,insert,update,references
3	CmpParentesco	varchar(10)	NO	10				select,insert,update,references
4	CmpFecha	date	YES					select,insert,update,references

### Tabla TestDentExpediente

En esta tabla se almacenará la información sobre el estado inicial de cada diente en el examen que se realiza al abrir el expediente.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdExpediente	int(11)	NO		10	PRI	auto_increment	select,insert,update,references
2	CmpIdDiente	tinyint(3) unsigned	NO		3	PRI		select,insert,update,references
3	CmpEstadoActual	varchar(20)	YES	20				select,insert,update,references
4	CmpTxIndicado	varchar(40)	YES	40				select,insert,update,references
5	CmpSecuencia	varchar(40)	YES	40				select,insert,update,references
6	CmpObservaciones	varchar(255)	YES	255				select,insert,update,references

### Tabla HabitosBucales

En esta tabla se tendrán todos los hábitos bucales que puede tener un paciente los cuales han sido definidos por la clínica.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdHabito	tinyint(3) unsigned	NO		3	PRI	auto_increment	select,insert,update,references
2	CmpNombre	varchar(20)	NO	20				select,insert,update,referentes
3	CmpObservaciones	varchar(45)	YES	45				select,insert,update,referentes

### Tabla HabitosExpediente

En esta tabla se almacenarán los diferentes hábitos bucales que tiene cada paciente

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdExpediente	int(11)	NO		10	PRI	auto_increment	select,insert,update,references
2	CmpIdHabito	tinyint(3) unsigned	NO		3	PRI		select,insert,update,references

### Tabla Alteraciones

En esta tabla se almacenarán todas las alteraciones definidas por la clínica como identificables en un examen bucal.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdAlteracion	tinyint(3) unsigned	NO		3	PRI	auto_increment	select,insert,update,references
2	CmpNombre	varchar(20)	NO	20		MUL		select,insert,update,references
3	CmpIdTipo	tinyint(3) unsigned	NO		3	PRI		select,insert,update,references
4	CmpDescripcion	varchar(100)	YES	100				select,insert,update,referentes

### Tabla AlteracionesExpediente

En esta tabla se almacenan todas las alteraciones que se han detectado en el examen bucal del paciente.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdExpediente	int(11)	NO		10	PRI	auto_increment	select,insert,update,references
2	CmpIdAlteracion	tinyint(3) unsigned	NO		3	PRI		select,insert,update,references
3	CmpIdTipo	tinyint(3) unsigned	NO		3	PRI		select,insert,update,references
4	CmpObservaciones	varchar(60)	YES	60				select,insert,update,references

### Tabla OclusionesExpediente

En esta tabla se almacenarán todas las oclusiones detectadas en el examen bucal, dando espacio para observaciones sobre esas oclusiones.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdOclusion	tinyint(3) unsigned	NO		3	PRI		select,insert,update,references
2	CmpIdOpcionOclusion	tinyint(3) unsigned	NO		3	PRI		select,insert,update,references
3	CmpIdExpediente	int(11)	NO		10	PRI		select,insert,update,references
4	CmpObservaciones	varchar(45)	YES	45				select,insert,update,references

### Tabla Consultas

En esta tabla se almacenarán la información general de todas las consultas que se hagan en la clínica, y será la tabla padre para la tabla tratamientos y la tabla diagnósticos.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdConsulta	int(10) unsigned	NO		10	PRI	auto_increment	select,insert,update,references
2	CmpIdExpediente	int(11)	NO		10	PRI		select,insert,update,references
3	CmpFechaConsulta	date	NO					select,insert,update,references
4	CmpMotivoConsulta	varchar(40)	NO	40				select,insert,update,references
5	CmpHoraInicio	datetime	NO					select,insert,update,references
6	CmpHoraFin	datetime	NO					select,insert,update,references
7	CmpRevisada	tinyint(1)	NO		3			select,insert,update,references
8	CmpIdOperador	tinyint(3) unsigned	NO		3			select,insert,update,references
9	CmpIdUsuario	tinyint(3) unsigned	NO		3			select,insert,update,references
10	CmpFechaModifica	datetime	NO					select,insert,update,references
11	CmpObservacionesTx	varchar(45)	YES	45				select,insert,update,references
12	CmpEliminado	Bool	NO					select,insert,update,references

### Tabla DiagnosticoConsulta

En esta tabla se almacena el diagnostico hecho por el estudiante operador, en el momento de abrir el expediente del paciente.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdConsulta	int(10) unsigned	NO		10	PRI	auto_increment	select,insert,update,references
2	CmpIdExpediente	int(11)	NO		10	PRI		select,insert,update,references
3	CmpGeneral	varchar(100)	YES	100				select,insert,update,references
4	CmpAlteracionesSist	varchar(255)	YES	255				select,insert,update,references
5	CmpExComplementarios	varchar(255)	YES	255				select,insert,update,references
6	CmpEstomagtonico	varchar(255)	YES	255				select,insert,update,references
7	CmpDxPeriodontal	varchar(255)	YES	255				select,insert,update,references
8	CmpPulparPeriapical	varchar(255)	YES	255				select,insert,update,referentes
9	CmpDxCaries	varchar(255)	YES	255				select,insert,update,referentes
10	CmpRiesgoCariogeno	varchar(255)	YES	255				select,insert,update,referentes
11	CmpOtrosDx	varchar(255)	YES	255				select,insert,update,referentes
12	CmpApreciacionPsico	varchar(255)	YES	255				select,insert,update,referentes

### Tabla ExTxDientesConsulta

En esta tabla se almacenarán los valores resultantes de los diferentes exámenes que se realizan en los dientes por cada consulta

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdConsulta	int(10) unsigned	NO		10	PRI	auto_increment	select,insert,update,references

							nt	
2	CmpMaxila	tinyint(1)	NO		3	PRI		select,insert,update,references
3	CmpIdExpediente	int(11)	NO		10	PRI		select,insert,update,references
4	CmpConfiguracion	varchar(255)	YES	255				select,insert,update,references
5	CmpObservaciones	varchar(255)	YES	255				select,insert,update,references
6	CmpCPO_Ceo	tinyint(3) unsigned	YES		3			select,insert,update,references

### Tabla Radiografias

En esta tabla se almacenarán las radiografías digitalizadas, se guardará una por consulta.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdRadiografia	int(10) unsigned	NO		10	PRI	auto_increment	select,insert,update,references
2	CmpIdConsulta	int(10) unsigned	NO		10	PRI		select,insert,update,references
3	CmpRadiografia	mediumblob	NO	16777215				select,insert,update,references
4	CmpObservaciones	varchar(150)	YES	150				select,insert,update,references

### Tabla TratamientosRecomendados

En esta tabla se almacena la información complementaria de la apertura del expediente, sobre los tratamientos recomendados a realizar.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
----------	------------------	----------------	------	---------------	--------------------	-------	-------	-------------

ACION	CAMPO	CAMPO		MAXIMO	NUMERIC			
					A			
1	CmpIdTratamientos	mediumint(8) unsigned	NO		7	PRI		select,insert,update,references
2	CmpIdConsulta	int(10) unsigned	NO		10	PRI		select,insert,update,references
3	CmpIdExpediente	int(11)	NO		10	PRI		select,insert,update,references
4	CmpTrataPrioritarios	varchar(50)	YES	50				select,insert,update,references
5	CmpTrataPeriodontal	varchar(50)	YES	50				select,insert,update,references
6	CmpTrataEndodonticos	varchar(50)	YES	50				select,insert,update,references
7	CmpTrataQuirurgicos	varchar(50)	YES	50				select,insert,update,references
8	CmpOperacionDental	varchar(50)	YES	50				select,insert,update,references
9	CmpControlPlaca	varchar(50)	YES	50				select,insert,update,references
10	CmpControlInfeccion	varchar(50)	YES	50				select,insert,update,references
11	CmpEducacionMotivacion	varchar(50)	YES	50				select,insert,update,references
12	CmpFluoterapia	varchar(50)	YES	50				select,insert,update,references
13	CmpSellantesResina	varchar(50)	YES	50				select,insert,update,references
14	CmpControlDieta	varchar(50)	YES	50				select,insert,update,references
15	CmpControlHabitos	varchar(50)	YES	50				select,insert,update,references
16	CmpTerapiasComple mentarias	varchar(50)	YES	50				select,insert,update,references

**Tabla doctores**

En esta tabla se almacenará la información relacionada a los doctores directores de la clínica, ellos serán los encargados de aprobar lo que todos los operarios hacen.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdDoctor	tinyint(3) unsigned	NO		3	PRI	auto_increment	select,insert,update,references
2	CmpNombresDoctor	varchar(30)	NO	30				select,insert,update,references
3	CmpIdGrupo	tinyint(3) unsigned	NO		3	PRI		select,insert,update,references
4	CmpApellidosDoctor	varchar(30)	NO	30				select,insert,update,references
5	CmpIdUsuario	tinyint(3) unsigned	NO		3	PRI		select,insert,update,references
6	CmpCargoDoctor	varchar(30)	NO	30				select,insert,update,references
7	CmpEliminado	Bool	NO					select,insert,update,references

### Tabla operadores

En esta tabla se almacenará la información de todos los operadores que van realizando sus prácticas en la clínica.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdOperador	tinyint(3) unsigned	NO		3	PRI	auto_increment	select,insert,update,references
2	CmpNombresOperador	varchar(30)	NO	30				select,insert,update,references
3	CmpIdGrupo	tinyint(3)	NO		3	PRI		select,insert,update,references

		unsigned						
4	CmpApellidosOperad or	varchar(30)	NO	30				select,insert,update,references
5	CmpFechaNacimiento	date	NO					select,insert,update,references
6	CmpCiclo	varchar(7)	NO	7				select,insert,update,references
7	CmpNumeroCarnet	varchar(10)	NO	10				select,insert,update,references
8	CmpInicioRotacion	date	NO					select,insert,update,references
9	CmpFinRotacion	date	NO					select,insert,update,references
10	CmpIdUsuario	tinyint(3) unsigned	NO			3	PRI	select,insert,update,references
11	CmpEvaluacion	float(5,3)	YES			5		select,insert,update,references

### Tabla dientes

En esta tabla se almacenarán los códigos asignados a los dientes de adultos y niños

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdDiente	tinyint(3) unsigned	NO			3	PRI	auto_increment select,insert,update,references
2	CmpNombreDiente	varchar(3)	YES	3				select,insert,update,references
3	CmpDescripcion	varchar(45)	YES	45				select,insert,update,references

### Tabla Citas

En esta tabla se almacenarán todas las asignaciones de las citas de todos los pacientes o personas que soliciten una cita en la clínica.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdCita	mediumint(8) unsigned	NO		7	PRI	auto_increment	select,insert,update,references
2	CmpIdExpediente	int(11)	NO		10	PRI		select,insert,update,references
3	CmpFecha	date	NO					select,insert,update,references
4	CmpIdTratamiento	tinyint(3) unsigned	NO		3			select,insert,update,references
5	CmpHoraCita	datetime	NO					select,insert,update,references
6	CmpObservacion	varchar(254)	YES	254				select,insert,update,references
7	CmpEliminado	Bool	NO					select,insert,update,references

### Tabla DiasFeriados

Esta tabla se utilizará para almacenar todas las fechas que el personal de la clínica considere que no será posible atender pacientes.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdDia	mediumint(8) unsigned	NO		7	PRI	auto_increment	select,insert,update,references
2	CmpFecha	date	NO					select,insert,update,references
3	CmpHoraInicio	datetime	NO					select,insert,update,references
4	CmpHoraFin	datetime	NO					select,insert,update,references

### Tabla Telefonos

En esta tabla se almacenarán todos los números telefónicos independientemente de quien sea el que posee ese teléfono

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdPersona	int(10) unsigned	NO		10	PRI	auto_increment	select,insert,update,references
2	CmpIdGrupo	int(10) unsigned	NO		10			select,insert,update,references
3	CmpTipoTelefono	varchar(10)	NO	10				select,insert,update,references
4	CmpNumero	varchar(15)	YES	15				select,insert,update,references
5	CmpDescripcion	int(10) unsigned	YES		10			select,insert,update,references

### Tabla Grupos

En esta tabla se almacenarán todos los tipos de grupos de personas o entidades que tendrán posibilidad de almacenar un número telefónico.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdGrupo	tinyint(3) unsigned	NO		3	PRI	auto_increment	select,insert,update,references
2	CmpNombreGrupo	varchar(20)	NO	20		UNI		select,insert,update,references
3	CmpDescripcionGrupo	varchar(50)	YES	50				select,insert,update,references

### Tabla Usuarios

En esta tabla se almacenarán todos los datos de los usuarios del sistema, claves y nivel de privilegios sobre el sistema.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdUsuario	tinyint(3) unsigned	NO		3	PRI	auto_increment	select,insert,update,references
2	CmpIdGrupo	tinyint(3) unsigned	NO		3	PRI		select,insert,update,references
3	CmpLogin	varchar(10)	NO	10		UNI		select,insert,update,references
4	CmpClave	varchar(10)	NO	10				select,insert,update,references
5	CmpGrupo	char(1)	NO	1				select,insert,update,references
6	CmpEliminado	Bool	NO					select,insert,update,references

### Tabla Empleados

En esta tabla se almacenarán los datos de los empleados que laboran en la clínica.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdEmpleado	tinyint(3) unsigned	NO		3	PRI	auto_increment	select,insert,update,references
2	CmpNombresEmpleado	varchar(30)	NO	30				select,insert,update,references
3	CmpIdGrupo	tinyint(3) unsigned	NO		3	PRI		select,insert,update,references
4	CmpApellidosEmpleado	varchar(30)	NO	30				select,insert,update,references

5	CmpDui	varchar(10)	NO	10		UNI		select,insert,update,references
6	CmpIdUsuario	tinyint(3) unsigned	NO		3	PRI		select,insert,update,references
7	CmpEliminado	Bool	NO					select,insert,update,references

### Tabla Equipo

En esta tabla se almacena la información del equipo médico y mobiliario de la clínica, de manera que sirva como un inventario del equipo existente.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdEquipo	tinyint(3) unsigned	NO		3	PRI	auto_increment	select,insert,update,references
2	CmpIdTipo	tinyint(3) unsigned	NO		3	PRI		select,insert,update,references
3	CmpDescripcion	varchar(40)	NO	40				select,insert,update,references
4	CmpValorEquipo	decimal(8,2)	NO		8			select,insert,update,references
5	CmpFechaIngreso	date	NO					select,insert,update,references
6	CmpActivo	tinyint(1)	YES		3			select,insert,update,references
7	CmpEliminado	Bool	NO					select,insert,update,references

### Tabla MantenimientoEquipo

En esta tabla se registrarán todos los mantenimientos realizados al equipo, principalmente al equipo utilizado en las consultas.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdMantenimiento	int(10) unsigned	NO		10	PRI	auto_increment	select,insert,update,referentes
2	CmpIdEquipo	tinyint(3) unsigned	NO		3	PRI		select,insert,update,referentes
3	CmpIdTipo	tinyint(3) unsigned	NO		3	PRI		select,insert,update,referentes
4	CmpFecha	date	NO					select,insert,update,referentes
5	CmpActividad	varchar(30)	NO	30				select,insert,update,referentes
6	CmpDiagnostico	varchar(200)	NO	200				select,insert,update,referentes
7	CmpNombreTecnico	varchar(40)	NO	40				select,insert,update,referentes
8	CmpObservaciones	varchar(100)	NO	100				select,insert,update,referentes
9	CmpEliminado	Bool	NO					select,insert,update,referentes

### Tabla Materiales

En esta tabla se almacenarán todos los datos de los materiales que se utilizan en la clínica, sirviendo como la base del inventario de materiales.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdMaterial	int(10) unsigned	NO		10	PRI	auto_increment	select,insert,update,referentes
2	CmpIdTipoMaterial	tinyint(3) unsigned	NO		3	PRI		select,insert,update,referentes

3	CmpMedida	varchar(10)	NO	10				select,insert,update,referentes
4	CmpDescripcion	varchar(100)	NO	100				select,insert,update,referentes
5	CmpCostoPromedio	decimal(8,2)	YES		8			select,insert,update,referentes
6	CmpCaduca	tinyint(1)	YES		3			select,insert,update,referentes
7	CmpExistencia	decimal(8,2)	YES		8			select,insert,update,referentes
8	CmpIdUsuario	tinyint(3) unsigned	NO		3			select,insert,update,references
9	CmpFechaModifica	date	NO					select,insert,update,referentes
10	CmpEliminado	Bool	NO					select,insert,update,referentes

### Tabla IngresoMateriales

En esta tabla se registrará la información general de todos los ingresos de materiales que se realicen en la clínica.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdIngreso	mediumint(8) unsigned	NO		7	PRI	auto_increment	select,insert,update,references
2	CmpIdTipoIngreso	tinyint(3) unsigned	NO		3	PRI		select,insert,update,references
3	CmpFechaIngreso	date	NO					select,insert,update,references
4	CmpDescripcionIngreso	varchar(100)	NO	100				select,insert,update,references
5	CmpIdGrupo	tinyint(3) unsigned	NO		3	PRI		select,insert,update,references
6	CmpTipoIngreso	tinyint(3)	NO		3			select,insert,update,references

		unsigned						
7	CmpIdUsuario	tinyint(3) unsigned	NO		3	PRI		select,insert,update,references
8	CmpFechaModifica	date	NO					select,insert,update,references
9	CmpIdProveedor	tinyint(3) unsigned	NO		3	PRI		select,insert,update,references
10	CmpEliminado	Bool	NO					select,insert,update,referentes

### Tabla SalidasMaterial

En esta tabla se almacenará la información general de las salidas de materiales que se realicen en la clínica, identificando el motivo de la salida.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpIdSalida	mediumint(8) unsigned	NO		7	PRI	auto_increment	select,insert,update,references
2	CmpIdTipoCliente	tinyint(3) unsigned	NO		3	PRI		select,insert,update,references
3	CmpFechaSalida	date	NO					select,insert,update,references
4	CmpIdGrupo	tinyint(3) unsigned	NO		3	PRI		select,insert,update,references
5	CmpTipoSalida	tinyint(3) unsigned	NO		3			select,insert,update,references
6	CmpIdCliente	mediumint(8) unsigned	YES		7			select,insert,update,references

7	CmpCantidad	decimal(6,2)	YES			6			select,insert,update,references
8	CmpCosto	decimal(6,2)	YES			6			select,insert,update,references
9	CmpIdUsuario	tinyint(3) unsigned	NO			3	PRI		select,insert,update,references
10	CmpFechaModifica	date	NO						select,insert,update,references
11	CmpEliminado	Bool	NO						select,insert,update,references

### Tabla Configuraciones

En esta tabla se almacenarán todas las configuraciones globales del sistema, que servirán para establecer los parámetros con los que trabajará el sistema. Se almacenará desde el formato de fecha hasta el número de citas diarias.

POSICION	NOMBRE DEL CAMPO	TIPO DEL CAMPO	NULL	TAMAÑO MAXIMO	PRECISION NUMERICA	LLAVE	EXTRA	PRIVILEGIOS
1	CmpFormatoFecha	Date	NO					select,insert,update,references
2	CmpConfiguracion1		NO					select,insert,update,references
3	CmpConfiguracion2		NO					select,insert,update,references
4	CmConfig3		NO					select,insert,update,references

### REGLAS DE INTEGRIDAD REFERENCIAL ENTRE LAS TABLAS

ENTIDAD 1	TIPO RELACIÓN	ENTIDAD 2	ACCIÓN ACTUALIZAR	ACCIÓN AL ELIMINAR
TblExpediente	1-M	TblConsulta	Cascada	Restringir

TblExpediente	1-M	TblAntecedenteFamiliar TblTejidosExpediente TblHabitosExpediente TblAlteracionesExpediente TblMovimientoCuenta TblCitas	Cascada	Restringir
TblExpediente	1-1	TblHistoriaOdontologia TblExpedienteAdulto TblExpedienteNino TblOclusionExpediente TblTestDentExpediente TblAtmExpediente	Cascada	Restringir
TblGruposUsuarios	1-M	TblUsuarios	Cascada	Restringir
TblUsuario	1-M	Todas las tablas	Cascada	Restringir
TblConsulta	1-1	TblExTxDientesConsulta	Cascada	Restringir
TblConsulta	1-M	TblTxConsulta TblDiagnosticoConsulta TblLigaLesionConsulta TblPmaPlacaConsulta TblLesionOclusalConsulta TblLesionPeriaConsulta TblReabsorcionesConsulta TblExRadioConsulta	Cascada	Restringir

Se debe aclarar que en esta base de datos no se permitirá la eliminación de registros en casi todas las tablas, es por eso que en las reglas de Integridad Referencial se ha optado por utilizar la acción de Restrict o restringir y en lugar de eliminarlos físicamente de la tabla se utilizará un campo de tipo boolean que servirá para indicar si ha sido marcado como eliminado o no.

**CAPÍTULO IV**  
**4 DESARROLLO DE LA APLICACIÓN**

## 4.1 CODIFICACIÓN

Para la programación del sistema se utilizó clases diseñadas y desarrolladas a base de controles, tales como:

El formulario summ para hacer las interfaces de cada módulo dotado de paginación de registros y cuadrícula:

```
Imports MySql.Data.MySqlClient
```

```
Imports MySql.Data.MySqlClient
```

```
Public Class summary
```

```
Inherits System.Windows.Forms.Form  
Friend WithEvents panel As System.Windows.Forms.SplitContainer  
Friend WithEvents arbol As System.Windows.Forms.TreeView  
Friend WithEvents rejilla As System.Windows.Forms.DataGridView  
Friend WithEvents barra As System.Windows.Forms.ToolStrip  
Friend WithEvents tsbAgregar As System.Windows.Forms.ToolStripButton  
Friend WithEvents tsbModificar As System.Windows.Forms.ToolStripButton  
Friend WithEvents tsbEliminar As System.Windows.Forms.ToolStripButton  
Friend WithEvents tsbVer As System.Windows.Forms.ToolStripButton  
Friend WithEvents tsbSalir As System.Windows.Forms.ToolStripButton  
Friend WithEvents tsbddModulos As  
System.Windows.Forms.ToolStripDropDownButton  
Friend WithEvents ToolStripSeparator1 As System.Windows.Forms.ToolStripSeparator  
Private objModulos As New AccesoModulos()  
Private cadenaConexion As New stringDeConexion  
Private conexionResumen As MySqlConnection 'conexion a la bd'  
Private comandoResumen As New MySqlCommand() 'variable para los comandos'  
Private adaptador As New MySqlDataAdapter() 'variable para leer...  
Private tabla As New DataTable()  
Private lioperacion As Byte = 0  
Private actualModulo As Byte = 0  
Friend WithEvents tsbevaluaciones As System.Windows.Forms.ToolStripButton  
Private frmPadre As Principal  
Private filtroArbol As String = ""  
Friend WithEvents treeImages As System.Windows.Forms.ImageList  
Private components As System.ComponentModel.IContainer  
Friend WithEvents cbbPaginacion As System.Windows.Forms.ToolStripComboBox  
Private limite As Integer = 50  
Private meses As New Hashtable
```

```

Private ciclos As New Hashtable
Private totalRegistros As Integer = 0

#Region "propiedades publicas"
Public ReadOnly Property miModulo()
    Get
        Return Me.actualModulo
    End Get
End Property
#End Region

#Region "metodos privados"
Private Sub configurarBarra()

    Try
        'para configurar los botones que llevan a los diferentes modulos.
        If (Me.tsbddModulos.HasDropDownItems()) Then
            Me.tsbddModulos.DropDownItems.Clear()
        End If
        Dim h As Integer
        For h = 1 To objModulos.contarModulos
            Me.tsbddModulos.DropDownItems.Add(Me.objModulos.moduloToString(h))
            If (h = Me.objModulos.Modulo) Then
                Me.tsbddModulos.DropDownItems.Item(h - 1).Visible = False
            End If
            If (Me.objModulos.AccesoDeModulo(h) > 1) Then
                Me.tsbddModulos.DropDownItems.Item(h - 1).Enabled = True
            Else
                Me.tsbddModulos.DropDownItems.Item(h - 1).Enabled = False
            End If
        Next

        Me.tsbAgregar.Enabled = Me.objModulos.agregaModulo
        Me.tsbVer.Enabled = Me.objModulos.leeModulo
        Me.tsbEliminar.Enabled = Me.objModulos.eliminamodulo
        Me.tsbModificar.Enabled = Me.objModulos.modifcaModulo

        Select Case Me.objModulos.Modulo
            Case 5 'Modulo de estudiantes practicantes...
                Me.tsbevaluaciones.Visible = False
                Me.tsbEliminar.Visible = False
                Me.tsbAgregar.Visible = False
                Me.tsbModificar.Visible = False
                Me.tsbVer.Text = "Evaluación"
            Case Else
                Me.tsbEliminar.Visible = True
                Me.tsbAgregar.Visible = True
        End Select
    End Try
End Sub

```

```

        Me.tsbModificar.Visible = True
        Me.tsbVer.Text = "Ver"
        Me.tsbevaluaciones.Visible = False
    End Select
Catch ex As Exception
    logError(Me.objModulos, ex)
End Try

End Sub
Private Sub crearConsulta()
    Try
        Select Case objModulos.Modulo
        Case 1 'consulta para expedientes
            Dim sql As String = ""
            comandoResumen.CommandText = "SELECT a.cmpidpersona, "
            comandoResumen.CommandText += " a.cmpnombrespersona, "
            comandoResumen.CommandText += " a.cmpapellidospersona, "
            comandoResumen.CommandText += " a.cmpfechanacimiento, "
            comandoResumen.CommandText += " b.cmpfecharegistro, "
            comandoResumen.CommandText += " b.cmpidexpediente "
            sql += " FROM tblexpediente b LEFT JOIN tblpersonas a
ON(a.cmpidpersona=b.cmpidpersona) "
            sql += " WHERE b.cmpidpersona > 0 "
            If Me.filtroArbol.Length > 0 Then
                Dim filtros As String()
                filtros = Me.filtroArbol.Split("")
                If filtros(0).Equals("Pacientes PPC") Then
                    sql += " AND b.cmpppc = 1 "
                Else
                    sql += " AND b.cmpppc = 0 "
                End If
                Select Case filtros.Length
                Case 2
                    sql += " AND a.cmpidgrupo = " + filtros(1) + ""
                Case 3
                    sql += " AND a.cmpidgrupo = " + filtros(1) + ""
                    sql += " AND a.cmpgenero = " + filtros(2) + ""
                Case 4
                    sql += " AND a.cmpidgrupo = " + filtros(1) + ""
                    sql += " AND a.cmpgenero = " + filtros(2) + ""
                    sql += " AND (a.cmpnombrespersona like " + filtros(3) + "%' OR
a.cmpapellidospersona like " + filtros(3) + "%'"
                End Select
            End If
        End Sub

```

```

        comandoResumen.CommandText += sql + " ORDER BY
a.cmpapellidospersona ASC LIMIT " & IIf(Me.cbbPaginacion.Items.Count > 0,
(Me.cbbPaginacion.SelectedIndex * paginadorGeneral).ToString, "0") & ", " &
paginadorGeneral
    Case 2 'consulta para consulta..
        comandoResumen.CommandText = "SELECT "
        comandoResumen.CommandText += " c.cmpidconsulta, "
        comandoResumen.CommandText += " p.cmpnombrespersona, "
        comandoResumen.CommandText += " p.cmpapellidospersona, "
        comandoResumen.CommandText += " c.cmpfechaconsulta, "
        comandoResumen.CommandText += " e.cmpidexpediente, "
        comandoResumen.CommandText += " c.cmprevisada, "
        comandoResumen.CommandText += " c.cmptxs, "
        comandoResumen.CommandText += " c.cmptxsterminados "
        comandoResumen.CommandText += " FROM tblconsulta c LEFT JOIN
tblexpediente e ON(e.CmpIdExpediente=c.CmpIdExpediente) "
        comandoResumen.CommandText += " LEFT JOIN tblpersonas p
ON(p.CmpIdPersona=e.CmpIdPersona)"
        comandoResumen.CommandText += " WHERE c.cmpidconsulta > 0 "
        If Me.filtroArbol.Length > 0 Then
            Dim filtros As String()
            filtros = Me.filtroArbol.Split("\")
            If filtros(0).Equals("Pacientes PPC") Then
                comandoResumen.CommandText += " AND e.cmpppc = 1 "
            Else
                comandoResumen.CommandText += " AND e.cmpppc = 0 "
            End If
            Select Case filtros.Length
                Case 2 'filtro por grupo...
                    comandoResumen.CommandText += " AND p.cmpidgrupo = "" +
filtros(1) + """"
                Case 3 'filtro por sexo...
                    comandoResumen.CommandText += " AND p.cmpidgrupo = "" +
filtros(1) + """"
                    comandoResumen.CommandText += " AND p.cmpgenero = "" +
filtros(2) + """"
                Case 4 'filtro por ano...
                    comandoResumen.CommandText += " AND p.cmpidgrupo = "" +
filtros(1) + """"
                    comandoResumen.CommandText += " AND p.cmpgenero = "" +
filtros(2) + """"
                    comandoResumen.CommandText += " AND
DATE_FORMAT(c.cmpfechaconsulta, '%Y') = "" + filtros(3) + """"
                Case 5 'filtro por mes...
                    Dim m As Integer
                    For Each i As Integer In Me.meses.Keys
                        If Me.meses(i).Equals(filtros(4)) Then

```

```

        m = i
    End If
Next
comandoResumen.CommandText += " AND p.cmpidgrupo = "" +
filtros(1) + ""
comandoResumen.CommandText += " AND p.cmpgenero = "" +
filtros(2) + ""
comandoResumen.CommandText += " AND
DATE_FORMAT(c.cmpfechaconsulta, '%Y') = " + filtros(3) + ""
comandoResumen.CommandText += " AND
DATE_FORMAT(c.cmpfechaconsulta, '%c') = " + m.ToString + ""
End Select

End If
comandoResumen.CommandText += " ORDER BY p.cmpapellidospersona
DESC LIMIT " & IIf(Me.cbbPaginacion.Items.Count > 0,
(Me.cbbPaginacion.SelectedIndex * paginadorGeneral).ToString, "0") & ", " &
paginadorGeneral

Case 3 'cuentas por cobrar..
comandoResumen.CommandText = "SELECT a.cmpidpersona, "
comandoResumen.CommandText += " a.cmpnombrespersona, "
comandoResumen.CommandText += " a.cmpapellidospersona, "
comandoResumen.CommandText += " a.cmpfechanacimiento, "
comandoResumen.CommandText += " b.cmpfecharegistro, "
comandoResumen.CommandText += " b.cmpidexpediente "
comandoResumen.CommandText += " FROM tblexpediente b LEFT JOIN
tblpersonas a ON(a.cmpidpersona=b.cmpidpersona) "
comandoResumen.CommandText += " WHERE b.cmpidpersona > 0 "
If Me.filtroArbol.Length > 0 Then
    Dim filtros As String()
    filtros = Me.filtroArbol.Split("\")
    Select Case filtros(0)
        Case "CUENTAS"
            Select Case filtros.Length
                Case 2
                    If filtros(1).Equals("Paciente PPC") Then
                        comandoResumen.CommandText += " AND b.cmppppc = 1 "
                    Else
                        comandoResumen.CommandText += " AND b.cmppppc = 0 "
                    End If
                Case 3
                    comandoResumen.CommandText += " AND a.cmpidgrupo = "" +
filtros(2) + ""
                Case 4

```

comandoResumen.CommandText += " AND a.cmpidgrupo = " +  
filtros(2) + ""

comandoResumen.CommandText += " AND a.cmpgenero = " +  
filtros(3) + ""

Case 5

comandoResumen.CommandText += " AND a.cmpidgrupo = " +  
filtros(2) + ""

comandoResumen.CommandText += " AND a.cmpgenero = " +  
filtros(3) + ""

comandoResumen.CommandText += " AND  
(a.cmpnombrespersona like " + filtros(4) + "%' OR a.cmpapellidospersona like " +  
filtros(4) + "%')"

End Select

Case "SALDOS AZULES"

comandoResumen.CommandText += " AND (((SELECT  
SUM(cmpvalorabono)-SUM(cmpvalorcargo) FROM tblmovimientocuenta WHERE  
cmpidexpediente = b.cmpidexpediente AND cmpcancelada = 0) > 0) OR (SELECT  
SUM(cmpvalorabono)-SUM(cmpvalorcargo) FROM tblmovimientocuenta WHERE  
cmpidexpediente = b.cmpidexpediente AND cmpcancelada = 0)is null )"

Select Case filtros.Length

Case 2

If filtros(1).Equals("Paciente PPC") Then

comandoResumen.CommandText += " AND b.cmpppc = 1 "

Else

comandoResumen.CommandText += " AND b.cmpppc = 0 "

End If

Case 3

If filtros(1).Equals("Paciente PPC") Then

comandoResumen.CommandText += " AND b.cmpppc = 1 "

Else

comandoResumen.CommandText += " AND b.cmpppc = 0 "

End If

comandoResumen.CommandText += " AND a.cmpidgrupo = " +  
filtros(2) + ""

Case 4

If filtros(1).Equals("Paciente PPC") Then

comandoResumen.CommandText += " AND b.cmpppc = 1 "

Else

comandoResumen.CommandText += " AND b.cmpppc = 0 "

End If

comandoResumen.CommandText += " AND a.cmpidgrupo = " +  
filtros(2) + ""

comandoResumen.CommandText += " AND a.cmpgenero = " +  
filtros(3) + ""

```

Case 5
  If filtros(1).Equals("Paciente PPC") Then
    comandoResumen.CommandText += " AND b.cmpppc = 1 "
  Else
    comandoResumen.CommandText += " AND b.cmpppc = 0 "
  End If
  comandoResumen.CommandText += " AND a.cmpidgrupo = '" +
filtros(2) + "'"
  comandoResumen.CommandText += " AND a.cmpgenero = '" +
filtros(3) + "'"
  comandoResumen.CommandText += " AND
(a.cmpnombrespersona like '" + filtros(4) + "%' OR a.cmpapellidospersona like '" +
filtros(4) + "%')
  End Select
Case "SALDOS ROJOS"
  comandoResumen.CommandText += " AND ((SELECT
SUM(cmpvalorabono)-SUM(cmpvalorcargo) FROM tblmovimientocuenta WHERE
cmpidexpediente = b.cmpidexpediente AND cmpcancelada = 0) < 0 )"
  Select Case filtros.Length
    Case 2
      If filtros(1).Equals("Paciente PPC") Then
        comandoResumen.CommandText += " AND b.cmpppc = 1 "
      Else
        comandoResumen.CommandText += " AND b.cmpppc = 0 "
      End If
    Case 3
      If filtros(1).Equals("Paciente PPC") Then
        comandoResumen.CommandText += " AND b.cmpppc = 1 "
      Else
        comandoResumen.CommandText += " AND b.cmpppc = 0 "
      End If
      comandoResumen.CommandText += " AND a.cmpidgrupo = '" +
filtros(2) + "'"
    Case 4
      If filtros(1).Equals("Paciente PPC") Then
        comandoResumen.CommandText += " AND b.cmpppc = 1 "
      Else
        comandoResumen.CommandText += " AND b.cmpppc = 0 "
      End If
      comandoResumen.CommandText += " AND a.cmpidgrupo = '" +
filtros(2) + "'"
      comandoResumen.CommandText += " AND a.cmpgenero = '" +
filtros(3) + "'"

```

```

Case 5
  If filtros(1).Equals("Paciente PPC") Then
    comandoResumen.CommandText += " AND b.cmppppc = 1 "
  Else
    comandoResumen.CommandText += " AND b.cmppppc = 0 "
  End If
  comandoResumen.CommandText += " AND a.cmpidgrupo = '" +
filtros(2) + "'"
  comandoResumen.CommandText += " AND a.cmpgenero = '" +
filtros(3) + "'"
  comandoResumen.CommandText += " AND
(a.cmpnombrespersona like '" + filtros(4) + "%' OR a.cmpapellidospersona like '" +
filtros(4) + "%'"
    End Select
  End Select

End If
  comandoResumen.CommandText += " ORDER BY a.cmpapellidospersona
ASC, a.cmpnombrespersona ASC LIMIT " & IIf(Me.cbbPaginacion.Items.Count > 0,
(Me.cbbPaginacion.SelectedIndex * paginadorGeneral).ToString, "0") & ", " &
paginadorGeneral

```

```

Case 4 'planificacion de citas..
  Me.comandoResumen.CommandText = " SELECT c.cmpidcita,
c.cmpidexpediente, "
  Me.comandoResumen.CommandText += " date_format(c.cmphoracita,
'%Y/%M/%d %r'), "
  Me.comandoResumen.CommandText += " p.cmpnombrespersona,
p.cmpapellidospersona, "
  Me.comandoResumen.CommandText += " CONCAT(k.cmpnombrespersona,'
', k.cmpapellidospersona),"
  Me.comandoResumen.CommandText += " IF(c.cmphoracita < now(), 0, 1)"
  Me.comandoResumen.CommandText = " SELECT c.cmpidcita,
c.cmpidexpediente, p.cmpnombrespersona, cmpapellidospersona,
date_format(cmphoracita, 'yyyy/mm/dd %r') "
  Me.comandoResumen.CommandText += " FROM tblcitas c LEFT JOIN
tblexpediente e ON(c.cmpidexpediente = e.cmpidexpediente) "
  Me.comandoResumen.CommandText += " LEFT JOIN tblpersonas p
ON(p.cmpidpersona = e.cmpidpersona) "
  Me.comandoResumen.CommandText += " LEFT JOIN tblpersonas k
ON(k.cmpidpersona = c.cmpidpersona) "
  Me.comandoResumen.CommandText += " WHERE c.cmpidcita > 0 "

If Me.filtroArbol.Length > 0 Then
  If Me.arbol.SelectedNode.Text.Equals("HOY") Then
    Dim hoy As New Date

```

```

        Me.comandoResumen.CommandText += " AND
date_format(c.cmphoracita, '%Y/%m/%d') = " + Format(hoy.Today,
"yyyy/MM/dd").ToString + " "
    Else
        Dim filtros As String()
        filtros = Me.filtroArbol.Split("\")
        Select Case filtros.Length
            Case 1
                comandoResumen.CommandText += " AND
DATE_FORMAT(c.cmphoracita, '%Y') = " + Me.arbol.SelectedNode.Tag + " "
            Case 2
                comandoResumen.CommandText += " AND
DATE_FORMAT(c.cmphoracita, '%Y') = " + Me.arbol.SelectedNode.Parent.Text + " "
                comandoResumen.CommandText += " AND
DATE_FORMAT(c.cmphoracita, '%c') = " + Me.arbol.SelectedNode.Tag + " "
            End Select
        End If
    End If
    comandoResumen.CommandText += " ORDER BY c.cmphoracita DESC
LIMIT " & IIf(Me.cbbPaginacion.Items.Count > 0, (Me.cbbPaginacion.SelectedIndex *
paginadorGeneral).ToString, "0") & ", " & paginadorGeneral
    Case 5 ' practicantes
        Me.comandoResumen.CommandText = " SELECT a.cmpidpersona,
a.cmpnombrespersona, a.cmpapellidospersona, b.cmpnumerocarnet, "
        Me.comandoResumen.CommandText += " date_format(b.cmpiniciorotacion,
'%Y/%M/%d') "
        Me.comandoResumen.CommandText += " FROM tblpersonas a LEFT JOIN
tbloperadores b ON(a.cmpidpersona = b.cmpidpersona) "
        Me.comandoResumen.CommandText += " WHERE b.cmpidpersona > 0 AND
a.cmpidgrupo = 'Estudiante Practicante'"
        If Me.filtroArbol.Length > 0 Then
            Dim filtros As String()
            filtros = Me.filtroArbol.Split("\")
            Select Case filtros.Length
                Case 1
                    comandoResumen.CommandText += " AND
DATE_FORMAT(b.cmpiniciorotacion, '%Y') = " + Me.arbol.SelectedNode.Tag + " "
                Case 2
                    comandoResumen.CommandText += " AND
DATE_FORMAT(b.cmpiniciorotacion, '%Y') = " + Me.arbol.SelectedNode.Parent.Text + "
"
                    comandoResumen.CommandText += " AND UPPER(b.cmpciclo) = "
+ UCase(Me.arbol.SelectedNode.Tag) + " "
                End Select
            End If
        comandoResumen.CommandText += " ORDER BY b.cmpiniciorotacion
DESC LIMIT " & IIf(Me.cbbPaginacion.Items.Count > 0,

```

(Me.cbbPaginacion.SelectedIndex \* paginadorGeneral).ToString, "0") & ", " &  
paginadorGeneral

Case 6 'equipos e instrumentos...

```
comandoResumen.CommandText = "SELECT a.cmpidmantenimiento, "  
comandoResumen.CommandText += " a.cmpidequipo, "  
comandoResumen.CommandText += " a.cmpfecha, "  
comandoResumen.CommandText += " b.cmpdescripcion, "  
comandoResumen.CommandText += " a.cmpactividad "  
comandoResumen.CommandText += " FROM tblmantenimientoequipo a  
LEFT JOIN tblequipo b ON(a.cmpidequipo=b.cmpidequipo) "  
comandoResumen.CommandText += " LEFT JOIN tbltipoequipo c  
ON(b.cmpidtipo=c.cmpidtipo) "  
comandoResumen.CommandText += " WHERE a.cmpidmantenimiento > 0 "  
If Me.filtroArbol.Length > 0 Then  
    Dim filtros As String()  
    filtros = Me.filtroArbol.Split("\")  
    Select Case filtros(0)  
        Case "REALIZADOS"  
            comandoResumen.CommandText += "AND a.cmprealizado = 1 "  
            Select Case filtros.Length  
                Case 2  
                    comandoResumen.CommandText += "AND a.cmpaprobado = " +  
Me.arbol.SelectedNode.Tag  
                Case 3  
                    comandoResumen.CommandText += " AND a.cmpaprobado = " +  
Me.arbol.SelectedNode.Parent.Tag  
                    comandoResumen.CommandText += " AND  
DATE_FORMAT(a.cmpfecha, '%Y') = " + filtros(2) + " "  
                Case 4  
                    comandoResumen.CommandText += " AND a.cmpaprobado = " +  
Me.arbol.SelectedNode.Parent.Parent.Tag  
                    comandoResumen.CommandText += " AND  
DATE_FORMAT(a.cmpfecha, '%Y') = " + Me.arbol.SelectedNode.Parent.Text + " "  
                    comandoResumen.CommandText += " AND  
DATE_FORMAT(a.cmpfecha, '%c') = " + Me.arbol.SelectedNode.Tag + " "  
                Case 5  
                    comandoResumen.CommandText += " AND a.cmpaprobado = " +  
Me.arbol.SelectedNode.Parent.Parent.Parent.Tag  
                    comandoResumen.CommandText += " AND  
DATE_FORMAT(a.cmpfecha, '%Y') = " + Me.arbol.SelectedNode.Parent.Parent.Text + " "  
                    comandoResumen.CommandText += " AND  
DATE_FORMAT(a.cmpfecha, '%c') = " + Me.arbol.SelectedNode.Parent.Tag + " "  
                    comandoResumen.CommandText += " AND c.cmpidtipo = " +  
Me.arbol.SelectedNode.Tag + " "  
            End Select  
        Case "NO REALIZADOS"
```

```

comandoResumen.CommandText += "AND a.cmprealizado = 0 "
Select Case filtros.Length
Case 2
comandoResumen.CommandText += " AND
DATE_FORMAT(a.cmpfecha, '%Y') = " + filtros(1) + " "
Case 3
comandoResumen.CommandText += " AND
DATE_FORMAT(a.cmpfecha, '%Y') = " + filtros(1) + " "
comandoResumen.CommandText += " AND
DATE_FORMAT(a.cmpfecha, '%c') = " + Me.arbol.SelectedNode.Tag + " "
End Select
End Select
End If
comandoResumen.CommandText += " ORDER BY a.cmpfecha DESC LIMIT
" & IIf(Me.cbbPaginacion.Items.Count > 0, (Me.cbbPaginacion.SelectedIndex *
paginadorGeneral).ToString, "0") & ", " & paginadorGeneral
Case 7 ' medicamentos
Me.comandoResumen.CommandText = "SELECT @iid:=a.cmpidmovimiento,
"
comandoResumen.CommandText += " a.cmpfechamovimiento, "
comandoResumen.CommandText += " a.cmpdocumento, "
comandoResumen.CommandText += " a.cmpcancelado, "
comandoResumen.CommandText += " a.cmpidtipomov "
comandoResumen.CommandText += " FROM tblmovimientomateriales a "
comandoResumen.CommandText += " "
comandoResumen.CommandText += " WHERE a.cmpidmovimiento > 0 "
If Me.filtroArbol.Length > 0 Then

Dim filtros As String()
filtros = Me.filtroArbol.Split("\")
Select Case filtros.Length
Case 1 'filtro por tipo de movimiento...
comandoResumen.CommandText += " AND a.cmpidtipomov = '" +
Me.arbol.SelectedNode.Tag + "'"
Case 2 'filtro por año..
comandoResumen.CommandText += " AND a.cmpidtipomov = '" +
Me.arbol.SelectedNode.Parent.Tag + "'"
comandoResumen.CommandText += " AND
DATE_FORMAT(a.cmpfechamovimiento, '%Y') = " + Me.arbol.SelectedNode.Text + " "
Case 3 'filtro por mes
comandoResumen.CommandText += " AND a.cmpidtipomov = '" +
Me.arbol.SelectedNode.Parent.Parent.Tag + "'"
comandoResumen.CommandText += " AND
DATE_FORMAT(a.cmpfechamovimiento, '%Y') = " + Me.arbol.SelectedNode.Parent.Text
+ " "
comandoResumen.CommandText += " AND
DATE_FORMAT(a.cmpfechamovimiento, '%c') = " + Me.arbol.SelectedNode.Tag + " "

```

```

        Case 4 'filtro por tipo de materiales..
            comandoResumen.CommandText += " AND a.cmpidtipomov = "" +
Me.arbol.SelectedNode.Parent.Parent.Parent.Tag + ""
            comandoResumen.CommandText += " AND
DATE_FORMAT(a.cmpfechamovimiento, '%Y') = " +
Me.arbol.SelectedNode.Parent.Parent.Text + " "
            comandoResumen.CommandText += " AND
DATE_FORMAT(a.cmpfechamovimiento, '%c') = " + Me.arbol.SelectedNode.Parent.Tag
+ " AND ("
                comandoResumen.CommandText += " (SELECT COUNT(*) FROM
tbldetalle salidas WHERE cmpidmovimiento=a.cmpidmovimiento AND "
                comandoResumen.CommandText += " cmpidmaterial IN (SELECT
tmp.cmpidmaterial FROM tblmateriales tmp WHERE tmp.cmpidtipomaterial=" +
Me.arbol.SelectedNode.Tag + ") ) > 0 OR "
                comandoResumen.CommandText += " (SELECT COUNT(*) FROM
tbldetalle entradas WHERE cmpidmovimiento=a.cmpidmovimiento AND "
                comandoResumen.CommandText += " cmpidmaterial IN (SELECT
tmp.cmpidmaterial FROM tblmateriales tmp WHERE tmp.cmpidtipomaterial=" +
Me.arbol.SelectedNode.Tag + ") ) > 0 )"
            End Select
        End If
        comandoResumen.CommandText += " ORDER BY a.cmpfechamovimiento
DESC LIMIT " & IIf(Me.cbbPaginacion.Items.Count > 0,
(Me.cbbPaginacion.SelectedIndex * paginadorGeneral).ToString, "0") & ", " &
paginadorGeneral
        Case 8 'Datos de inventarios...
        End Select
        Me.llenarGrid()
        Catch ex As Exception
            logError(Me.objModulos, ex)
        End Try

    End Sub
    Private Sub llenarGrid()
        Try

            Me.rejilla.Columns.Clear()
            Dim tabla As New DataTable
            Dim adaptador As New MySqlDataAdapter
            adaptador.SelectCommand = comandoResumen
            tabla.Clear()
            tabla.Columns.Clear()
            adaptador.Fill(tabla)
            'revisar propiedad del grid---
            Dim clmnchk As New System.Windows.Forms.DataGridViewCheckBoxColumn
            clmnchk.TrueValue = 1
            clmnchk.FalseValue = 0

```

```

    clmnchk.ValueType = GetType(System.Byte)
    clmnchk.Name = "revisaCheck"
    Me.rejilla.DataSource = tabla
    Select Case Me.objModulos.Modulo
    Case 1 'expedientes
        Me.rejilla.Columns(0).HeaderText = "ID"
        Me.rejilla.Columns(0).Name = "ID"
        Me.rejilla.Columns("ID").Visible = False
        Me.rejilla.Columns(1).HeaderText = "NOMBRE"
        Me.rejilla.Columns(1).Name = "NOMBRE"
        Me.rejilla.Columns("NOMBRE").AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill
        Me.rejilla.Columns(2).HeaderText = "APELLIDOS"
        Me.rejilla.Columns(2).Name = "APELLIDOS"
        Me.rejilla.Columns("APELLIDOS").AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill
        Me.rejilla.Columns(3).HeaderText = "NACIMIENTO"
        Me.rejilla.Columns(3).Name = "NACIMIENTO"
        Me.rejilla.Columns("NACIMIENTO").AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill
        Me.rejilla.Columns(4).HeaderText = "INGRESO"
        Me.rejilla.Columns(4).Name = "INGRESO"
        Me.rejilla.Columns("INGRESO").AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill
        Me.rejilla.Columns(5).Name = "idexpediente"
        Me.rejilla.Columns("idexpediente").Visible = False
        Me.rejilla.Columns(5).HeaderText = "Expediente"
    Case 2 'consultas
        Me.rejilla.Columns(0).HeaderText = "ID"
        Me.rejilla.Columns(0).Name = "ID"
        Me.rejilla.Columns("ID").Visible = False
        Me.rejilla.Columns(1).HeaderText = "NOMBRES"
        Me.rejilla.Columns(1).Name = "NOMBRES"
        Me.rejilla.Columns("NOMBRES").AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill
        Me.rejilla.Columns(2).HeaderText = "APELLIDOS"
        Me.rejilla.Columns(2).Name = "APELLIDOS"
        Me.rejilla.Columns("APELLIDOS").AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill
        Me.rejilla.Columns(3).Name = "FECHA"
        Me.rejilla.Columns("FECHA").HeaderText = "FECHA"
        Me.rejilla.Columns(4).Name = "IDEXP"
        Me.rejilla.Columns("IDEXP").Visible = False
        Me.rejilla.Columns(5).Name = "REVISADA"
        Me.rejilla.Columns("REVISADA").Visible = False
        Me.rejilla.Columns(6).Name = "TXS"
        Me.rejilla.Columns("TXS").Visible = False

```

```

Me.rejilla.Columns(7).Name = "TXS Final"
Me.rejilla.Columns("TXS Final").Visible = False
Me.rejilla.Columns.Add(clmnchk.Clone)
Me.rejilla.Columns("revisaCheck").DataPropertyName =
tabla.Columns(5).ToString
Me.rejilla.Columns("revisaCheck").HeaderText = "REVISADA"
Me.rejilla.Columns("revisaCheck").AutoSizeMode =
DataGridViewAutoSizeColumnMode.ColumnHeader
Dim clmntx As New
System.Windows.Forms.DataGridViewCheckBoxColumn
clmntx = clmnchk.Clone
clmntx.Name = "TX"
clmntx.HeaderText = "TX"
Me.rejilla.Columns.Add(clmntx)
Me.rejilla.Columns("TX").DataPropertyName = tabla.Columns(6).ToString
Me.rejilla.Columns("TX").AutoSizeMode =
DataGridViewAutoSizeColumnMode.ColumnHeader
Dim clmntxf As New
System.Windows.Forms.DataGridViewCheckBoxColumn
clmntxf = clmnchk.Clone
clmntxf.Name = "TXF"
clmntxf.HeaderText = "TX F"
Me.rejilla.Columns.Add(clmntxf)
Me.rejilla.Columns("TXF").DataPropertyName = tabla.Columns(7).ToString
Me.rejilla.Columns("TXF").AutoSizeMode =
DataGridViewAutoSizeColumnMode.ColumnHeader
Case 3 ' cuentas por cobrar
Me.rejilla.Columns(0).HeaderText = "ID"
Me.rejilla.Columns(0).Name = "ID"
Me.rejilla.Columns("ID").Visible = False
Me.rejilla.Columns(1).HeaderText = "NOMBRE"
Me.rejilla.Columns(1).Name = "NOMBRE"
Me.rejilla.Columns("NOMBRE").AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill
Me.rejilla.Columns(2).HeaderText = "APELLIDOS"
Me.rejilla.Columns(2).Name = "APELLIDOS"
Me.rejilla.Columns("APELLIDOS").AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill
Me.rejilla.Columns(3).HeaderText = "NACIMIENTO"
Me.rejilla.Columns(3).Name = "NACIMIENTO"
Me.rejilla.Columns("NACIMIENTO").AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill
Me.rejilla.Columns(4).HeaderText = "INGRESO"
Me.rejilla.Columns(4).Name = "INGRESO"
Me.rejilla.Columns("INGRESO").AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill
Me.rejilla.Columns(5).Name = "idexpediente"

```

```

Me.rejilla.Columns("idexpediente").Visible = False
Me.rejilla.Columns(5).HeaderText = "Expediente"
Case 4 'planificaciones
Me.rejilla.Columns(0).HeaderText = "IDCITA"
Me.rejilla.Columns(0).Name = "IDCITA"
Me.rejilla.Columns("IDCITA").Visible = False
Me.rejilla.Columns(1).HeaderText = "IDEXP"
Me.rejilla.Columns(1).Name = "IDEXP"
Me.rejilla.Columns("IDEXP").Visible = False
Me.rejilla.Columns(2).HeaderText = "FECHA"
Me.rejilla.Columns(2).Name = "FECHA"
Me.rejilla.Columns("FECHA").AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill
Me.rejilla.Columns(3).HeaderText = "NOMBRES"
Me.rejilla.Columns(3).Name = "NOMBRES"
Me.rejilla.Columns("NOMBRES").AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill
Me.rejilla.Columns(4).HeaderText = "APELLIDOS"
Me.rejilla.Columns(4).Name = "APELLIDOS"
Me.rejilla.Columns("APELLIDOS").AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill
Me.rejilla.Columns(5).HeaderText = "DENTISTA"
Me.rejilla.Columns(5).Name = "DENTISTA"
Me.rejilla.Columns("DENTISTA").AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill
Me.rejilla.Columns(6).HeaderText = "EDITABLE"
Me.rejilla.Columns(6).Name = "EDITABLE"
Me.rejilla.Columns("EDITABLE").Visible = False
Case 5 'practicantes
Me.rejilla.Columns(0).HeaderText = "IDM"
Me.rejilla.Columns(0).Name = "IDOPERADOR"
Me.rejilla.Columns("IDOPERADOR").Visible = True
Me.rejilla.Columns(1).HeaderText = "NOMBRES"
Me.rejilla.Columns(1).Name = "NOMBRES"
Me.rejilla.Columns("NOMBRES").AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill
Me.rejilla.Columns(2).HeaderText = "APELLIDOS"
Me.rejilla.Columns(2).Name = "APELLIDOS"
Me.rejilla.Columns("APELLIDOS").AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill
Me.rejilla.Columns(3).HeaderText = "CARNET"
Me.rejilla.Columns(3).Name = "CARNET"
Me.rejilla.Columns("CARNET").AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill
Me.rejilla.Columns(4).HeaderText = "INICIO ROT"
Me.rejilla.Columns(4).Name = "INICIO ROT"

```

```
Me.rejilla.Columns("INICIO ROT").AutoSizeMode =  
DataGridViewAutoSizeColumnMode.Fill
```

#### Case 6

```
Me.rejilla.Columns(0).HeaderText = "IDM"  
Me.rejilla.Columns(0).Name = "IDM"  
Me.rejilla.Columns("IDM").Visible = False  
Me.rejilla.Columns(1).HeaderText = "IDE"  
Me.rejilla.Columns(1).Name = "IDE"  
Me.rejilla.Columns("IDE").Visible = False  
Me.rejilla.Columns(2).HeaderText = "FECHA"  
Me.rejilla.Columns(2).Name = "FECHA"  
Me.rejilla.Columns("FECHA").AutoSizeMode =  
DataGridViewAutoSizeColumnMode.Fill  
Me.rejilla.Columns(3).HeaderText = "EQUIPO"  
Me.rejilla.Columns(3).Name = "EQUIPO"  
Me.rejilla.Columns("EQUIPO").AutoSizeMode =  
DataGridViewAutoSizeColumnMode.Fill  
Me.rejilla.Columns(4).HeaderText = "ACTIVIDAD"  
Me.rejilla.Columns(4).Name = "ACTIVIDAD"  
Me.rejilla.Columns("ACTIVIDAD").AutoSizeMode =  
DataGridViewAutoSizeColumnMode.Fill
```

#### Case 7

```
Me.rejilla.Columns(0).HeaderText = "ID"  
Me.rejilla.Columns(0).Visible = False  
Me.rejilla.Columns(0).Name = "ID"  
Me.rejilla.Columns(1).HeaderText = "FECHA"  
Me.rejilla.Columns(1).Name = "FECHA"  
Me.rejilla.Columns("FECHA").AutoSizeMode =  
DataGridViewAutoSizeColumnMode.Fill  
Me.rejilla.Columns(2).HeaderText = "DOCUMENTO"  
Me.rejilla.Columns(2).Name = "DOCUMENTO"  
Me.rejilla.Columns("DOCUMENTO").AutoSizeMode =  
DataGridViewAutoSizeColumnMode.Fill  
Me.rejilla.Columns(3).Name = "CANCELADA"  
Me.rejilla.Columns("CANCELADA").Visible = False  
Me.rejilla.Columns(4).Name = "TIPOMOV"  
Me.rejilla.Columns("TIPOMOV").Visible = False  
Me.rejilla.Columns.Add(clmnchk.Clone)  
Me.rejilla.Columns("revisaCheck").DataPropertyName =  
tabla.Columns(3).ToString  
Me.rejilla.Columns("revisaCheck").HeaderText = "CANCELADA"  
Me.rejilla.Columns("revisaCheck").AutoSizeMode =  
DataGridViewAutoSizeColumnMode.ColumnHeader
```

End Select

```

Catch ex As MySqlException
    logError(Me.objModulos, ex)
End Try

```

```

End Sub

```

```

Private Sub tsbddModulos_DropDownItemClicked(ByVal sender As Object, ByVal e As
System.Windows.Forms.ToolStripItemClickedEventArgs) Handles

```

```

tsbddModulos.DropDownItemClicked

```

```

Try

```

```

    Dim numElemento As Integer

```

```

    Dim nombreModulo As String

```

```

    For numElemento = 0 To Me.tsbddModulos.DropDownItems.Count() - 1

```

```

        If Me.tsbddModulos.DropDownItems.Item(numElemento).Selected Then

```

```

            nombreModulo =

```

```

Me.tsbddModulos.DropDownItems.Item(numElemento).ToString()

```

```

            If numElemento + 1 < 8 Then 'si el modulo al que se desea cambiar no es el de
consultas y reportes ni configuraciones

```

```

                Me.frmPadre = Me.Owner

```

```

                If Me.frmPadre.verificarFormulario(numElemento + 1) Then

```

```

                    Me.frmPadre.ponerFocoFomulario(numElemento + 1)

```

```

                Else

```

```

                    If conexionResumen.State = ConnectionState.Open Then

```

```

                        conexionResumen.Close()

```

```

                    End If

```

```

                    Me.objModulos.Modulo = numElemento + 1 'se setea la variable de
accesos y modulos al nuevo modulo

```

```

                    Me.ConfigureConModulo(Me.objModulos) 'se configura el formulario
para el nuevo modulo

```

```

                End If

```

```

            Else

```

```

                'MsgBox(objModulos.moduloToString(numElemento + 1))

```

```

            End If

```

```

        End If

```

```

    Next

```

```

    Me.Refresh()

```

```

Catch ex As Exception

```

```

    logError(Me.objModulos, ex)

```

```

End Try

```

```

End Sub
Private Sub llamarFormsModificar()
    Try
        Select Case Me.objModulos.Modulo
            Case 1 'Expedientes
                If Me.rejilla.SelectedRows.Count = 0 Then
                    MsgBox("Seleccione un expediente", MsgBoxStyle.OkOnly,
Me.objModulos.moduloToString())
                    Return
                End If

                Dim exp As New
expediente(Me.rejilla.Rows(Me.rejilla.SelectedRows(0).Index).Cells("idexpediente").Value
, Me.objModulos)
                exp.ShowDialog()
                If exp.DialogResult = Windows.Forms.DialogResult.OK Then
                    Me.llenarGrid()
                End If
            Case 2 'consultas
                If Me.rejilla.SelectedRows.Count.Equals(0) Then
                    MsgBox("Seleccione una consulta para editar", MsgBoxStyle.OkOnly,
Me.objModulos.moduloToString())
                    Return
                End If
                If
(Me.rejilla.Rows(Me.rejilla.SelectedRows(0).Index).Cells("REVISADA").Value = 0) Then
                    Dim cons As New consulta(Me.objModulos,
Me.rejilla.Rows(Me.rejilla.SelectedRows(0).Index).Cells("ID").Value,
Me.rejilla.Rows(Me.rejilla.SelectedRows(0).Index).Cells("IDEXP").Value, False)
                    cons.ShowDialog()

                Else
                    MsgBox("Imposible editar una consulta revisada.", MsgBoxStyle.OkOnly,
Me.objModulos.moduloToString())
                End If
            Case 3 'consultas
                If Me.arbol.SelectedNode.Text.Equals("REMESAS") Then
                    MsgBox("Si desea editar una cuenta, seleccione la hoja correspondiente a
las cuentas", MsgBoxStyle.OkOnly, Me.objModulos.moduloToString())
                    Return
                End If
                If Me.rejilla.SelectedRows.Count = 0 Then
                    MsgBox("Debe seleccionar una cuenta", MsgBoxStyle.OkOnly,
Me.objModulos.moduloToString())
                    Return
                End If
        End Select
    End Try
End Sub

```

```

        Dim cuentas As New
cargosyabonos(Me.rejilla.Rows(Me.rejilla.SelectedCells(0).RowIndex).Cells("idexpediente
").Value, Me.objModulos, True)
        cuentas.ShowDialog()
    Case 4 ' citas
        If Me.rejilla.SelectedRows.Count.Equals(0) Then
            MsgBox("Seleccione una consulta", MsgBoxStyle.OkOnly,
Me.objModulos.moduloToString())
            Return
        End If

        If Me.rejilla.Item("EDITABLE", Me.rejilla.SelectedRows(0).Index).Value = 0
Then
            MsgBox("Las consultas programadas de fecha menor que hoy no se pueden
editar", MsgBoxStyle.OkOnly, Me.objModulos.moduloToString())
            Return
        End If
        Dim editCita As New registroCita(Me.rejilla.Item("IDCITA",
Me.rejilla.SelectedRows(0).Index).Value, Me.objModulos, If(Me.rejilla.Item("IDCITA",
Me.rejilla.SelectedRows(0).Index).Value.Equals(0), False, True))
        editCita.ShowDialog()
        Me.crearConsulta()
        Me.llenarGrid()
    Case 5 ' practicantes

    Case 6 ' mantenimientos de equipo
        If Me.rejilla.SelectedRows.Count.Equals(0) Then
            MsgBox("Seleccione un mantenimiento para editar", MsgBoxStyle.OkOnly,
Me.objModulos.moduloToString())
            Return
        End If
        Dim revisiones As New revisionesequipo(Me.rejilla.Item("IDM",
Me.rejilla.SelectedRows(0).Index).Value.ToString, True, Me.objModulos)
        revisiones.ConfigureConModulo(Me.objModulos)
        revisiones.ShowDialog()
        Me.llenarGrid()

    Case 7 ' movimientos
        If Me.rejilla.SelectedRows.Count.Equals(0) Then
            MsgBox("Seleccione un movimiento para ver", MsgBoxStyle.OkOnly,
Me.objModulos.moduloToString())
            Return
        End If
        Dim movi As New movimientomaterial(Me.rejilla.Item("ID",
Me.rejilla.SelectedCells(0).RowIndex).Value, False, Me.objModulos)
        movi.ShowDialog()
        Me.llenarGrid()

```

```

    End Select
    Catch ex As Exception
        logError(Me.objModulos, ex)
    End Try

End Sub
Private Sub llamarFormsAgregar()
    Try
        Select Case Me.miModulo
            Case 1 ' expedientes ...
                Dim exp As New expediente(0, Me.objModulos)
                exp.ShowDialog()
                If exp.DialogResult = Windows.Forms.DialogResult.OK Then
                    Me.llenarGrid()
                End If
            Case 2 ' consultas
                If Me.rejilla.SelectedRows.Count > 0 Then
                    Dim consulta As New trataconsulta(Me.objModulos,
Me.rejilla.Rows(Me.rejilla.SelectedCells(0).RowIndex).Cells("IDEXP").Value)
                    consulta.ShowDialog()
                Else
                    Dim consulta As New trataconsulta(Me.objModulos)
                    consulta.ShowDialog()
                End If
                Me.llenarGrid()
            Case 3 ' cuentas por cobrar ..
                If Me.arbol.SelectedNode.Text.Equals("REMESAS") Then
                    MsgBox("Si desea agregar un cargo o abono una cuenta, seleccione la hoja
correspondiente a las cuentas", MsgBoxStyle.OkOnly, Me.objModulos.moduloToString())
                    Return
                End If
                If Me.rejilla.SelectedRows.Count = 0 Then
                    MsgBox("Debe seleccionar una cuenta", MsgBoxStyle.OkOnly,
Me.objModulos.moduloToString())
                    Return
                End If
                Dim cuentas As New
cargosyabonos(Me.rejilla.Rows(Me.rejilla.SelectedCells(0).RowIndex).Cells("idexpediente
").Value, Me.objModulos, False)
                cuentas.ShowDialog()
            Case 4 ' CITAS ..
                Dim cita As New registroCita(Me.objModulos)
                cita.ShowDialog()
                'Me.llenarGrid()
            Case 6 ' ma
                Dim revisiones As New revisionesequipo()
                revisiones.ConfigureConModulo(Me.objModulos)
        End Select
    End Try
End Sub

```

```

        revisiones.ShowDialog()
        Me.llenarGrid()
    Case 7
        If Not (Me.arbol.SelectedNode.Level.Equals(0)) Then
            MsgBox("Por favor seleccione un tipo de movimiento para agregar",
                MsgBoxStyle.OkOnly, Me.objModulos.moduloToString())
            Return
        End If
        Dim movimientoMateriales As New
        movimientomaterial(Me.arbol.SelectedNode.Tag, Me.objModulos)
        movimientoMateriales.ConfigureConModulo(Me.objModulos)
        movimientoMateriales.ShowDialog()
        Me.llenarGrid()

    End Select
Catch ex As Exception
    logError(Me.objModulos, ex)
End Try
End Sub
Private Sub llenarPaginador()
    Try
        If Me.conexionResumen.State = ConnectionState.Closed Then
            Me.conexionResumen.Open()
        Else
            Me.conexionResumen.Close()
            Me.conexionResumen.Open()
        End If
        Dim i As Integer
        Me.cbbPaginacion.Items.Clear()
        Select Case objModulos.Modulo
            Case 1 'consulta para expedientes
                comandoResumen.CommandText = "SELECT COUNT(*) "
                comandoResumen.CommandText += "FROM tblexpediente b LEFT JOIN
tblpersonas a ON(a.cmpidpersona=b.cmpidpersona) "
                comandoResumen.CommandText += "WHERE b.cmpidpersona > 0 "
                If Me.filtroArbol.Length > 0 Then
                    Dim filtros As String()
                    filtros = Me.filtroArbol.Split("\")
                    If filtros(0).Equals("Pacientes PPC") Then
                        comandoResumen.CommandText += " AND b.cmpppc = 1 "
                    Else
                        comandoResumen.CommandText += " AND b.cmpppc = 0 "
                    End If
                    Select Case filtros.Length
                        Case 2
                            comandoResumen.CommandText += " AND a.cmpidgrupo = '" +
filtros(1) + "'"

```

```

Case 3
comandoResumen.CommandText += " AND a.cmpidgrupo = '" +
filtros(1) + "'"
comandoResumen.CommandText += " AND a.cmpgenero = '" +
filtros(2) + "'"

Case 4
comandoResumen.CommandText += " AND a.cmpidgrupo = '" +
filtros(1) + "'"
comandoResumen.CommandText += " AND a.cmpgenero = '" +
filtros(2) + "'"
comandoResumen.CommandText += " AND (a.cmpnombrespersona
like '" + filtros(3) + "%' OR a.cmpapellidospersona like '" + filtros(3) + "%')
End Select
End If
Case 2 'consulta para consulta..
comandoResumen.CommandText = "SELECT COUNT(*)"
comandoResumen.CommandText += " FROM tblconsulta c LEFT JOIN
tblexpediente e ON(e.CmpIdExpediente=c.CmpIdExpediente) "
comandoResumen.CommandText += " LEFT JOIN tblpersonas p
ON(p.CmpIdPersona=e.CmpIdPersona)"
comandoResumen.CommandText += " WHERE c.cmpidconsulta > 0 "
If Me.filtroArbol.Length > 0 Then
Dim filtros As String()
filtros = Me.filtroArbol.Split("\")
If filtros(0).Equals("Pacientes PPC") Then
comandoResumen.CommandText += " AND e.cmpppc = 1 "
Else
comandoResumen.CommandText += " AND e.cmpppc = 0 "
End If
Select Case filtros.Length
Case 2 'filtro por grupo...
comandoResumen.CommandText += " AND p.cmpidgrupo = '" +
filtros(1) + "'"
Case 3 'filtro por sexo...
comandoResumen.CommandText += " AND p.cmpidgrupo = '" +
filtros(1) + "'"
comandoResumen.CommandText += " AND p.cmpgenero = '" +
filtros(2) + "'"
Case 4 'filtro por ano...
comandoResumen.CommandText += " AND p.cmpidgrupo = '" +
filtros(1) + "'"
comandoResumen.CommandText += " AND p.cmpgenero = '" +
filtros(2) + "'"
comandoResumen.CommandText += " AND
DATE_FORMAT(c.cmpfechaconsulta, '%Y') = '" + filtros(3) + "'"
Case 5 'filtro por mes...

```

```

        Dim m As Integer
        For Each u As Integer In Me.meses.Keys
            If Me.meses(u).Equals(filtros(4)) Then
                m = u
            End If
        Next
        comandoResumen.CommandText += " AND p.cmpidgrupo = '" +
filtros(1) + "'"
        comandoResumen.CommandText += " AND p.cmpgenero = '" +
filtros(2) + "'"
        comandoResumen.CommandText += " AND
DATE_FORMAT(c.cmpfechaconsulta, '%Y') = '" + filtros(3) + "'"
        comandoResumen.CommandText += " AND
DATE_FORMAT(c.cmpfechaconsulta, '%c') = '" + m.ToString + "'"
    End Select

    End If
    Case 3 'cuentas por cobrar..
        comandoResumen.CommandText = "SELECT COUNT(*) "
        comandoResumen.CommandText += "FROM tblexpediente b LEFT JOIN
tblpersonas a ON(a.cmpidpersona=b.cmpidpersona) "
        comandoResumen.CommandText += "WHERE b.cmpidpersona > 0 "
        If Me.filtroArbol.Length > 0 Then
            Dim filtros As String()
            filtros = Me.filtroArbol.Split("\")
            Select Case filtros(0)
                Case "CUENTAS"
                    Select Case filtros.Length
                        Case 2
                            If filtros(1).Equals("Paciente PPC") Then
                                comandoResumen.CommandText += " AND b.cmpppc = 1 "
                            Else
                                comandoResumen.CommandText += " AND b.cmpppc = 0 "
                            End If
                        Case 3
                            comandoResumen.CommandText += " AND a.cmpidgrupo = '" +
filtros(2) + "'"
                        Case 4
                            comandoResumen.CommandText += " AND a.cmpidgrupo = '" +
filtros(2) + "'"
                            comandoResumen.CommandText += " AND a.cmpgenero = '" +
filtros(3) + "'"
                        Case 5
                            comandoResumen.CommandText += " AND a.cmpidgrupo = '" +
filtros(2) + "'"
                    End Select
                End Select
            End If
        End If
    End Case
End Select

```

```

comandoResumen.CommandText += " AND a.cmpgenero = '" +
filtros(3) + "'"
comandoResumen.CommandText += " AND
(a.cmpnombrespersona like '" + filtros(4) + "%' OR a.cmpapellidospersona like '" +
filtros(4) + "%')"
```

End Select

```

Case "SALDOS AZULES"
comandoResumen.CommandText += " AND (((SELECT
SUM(cmpvalorabono)-SUM(cmpvalorcarga) FROM tblmovimientocuenta WHERE
cmpidexpediente = b.cmpidexpediente AND cmpcancelada = 0) > 0) OR (SELECT
SUM(cmpvalorabono)-SUM(cmpvalorcarga) FROM tblmovimientocuenta WHERE
cmpidexpediente = b.cmpidexpediente AND cmpcancelada = 0)is null )"
```

Select Case filtros.Length

```

Case 2
If filtros(1).Equals("Paciente PPC") Then
comandoResumen.CommandText += " AND b.cmpppc = 1 "
Else
comandoResumen.CommandText += " AND b.cmpppc = 0 "
End If
```

```

Case 3
If filtros(1).Equals("Paciente PPC") Then
comandoResumen.CommandText += " AND b.cmpppc = 1 "
Else
comandoResumen.CommandText += " AND b.cmpppc = 0 "
End If
comandoResumen.CommandText += " AND a.cmpidgrupo = '" +
filtros(2) + "'"
```

```

Case 4
If filtros(1).Equals("Paciente PPC") Then
comandoResumen.CommandText += " AND b.cmpppc = 1 "
Else
comandoResumen.CommandText += " AND b.cmpppc = 0 "
End If
comandoResumen.CommandText += " AND a.cmpidgrupo = '" +
filtros(2) + "'"
comandoResumen.CommandText += " AND a.cmpgenero = '" +
filtros(3) + "'"
```

```

Case 5
If filtros(1).Equals("Paciente PPC") Then
comandoResumen.CommandText += " AND b.cmpppc = 1 "
Else
comandoResumen.CommandText += " AND b.cmpppc = 0 "
End If
comandoResumen.CommandText += " AND a.cmpidgrupo = '" +
filtros(2) + "'"
```

```

comandoResumen.CommandText += " AND a.cmpgenero = '" +
filtros(3) + "'"
comandoResumen.CommandText += " AND
(a.cmpnombrespersona like '" + filtros(4) + "%' OR a.cmpapellidospersona like '" +
filtros(4) + "%')"
End Select
Case "SALDOS ROJOS"
comandoResumen.CommandText += " AND ((SELECT
SUM(cmpvalorabono)-SUM(cmpvalorcarga) FROM tblmovimientocuenta WHERE
cmpidexpediente = b.cmpidexpediente AND cmpcancelada = 0) < 0 )"
Select Case filtros.Length
Case 2
If filtros(1).Equals("Paciente PPC") Then
comandoResumen.CommandText += " AND b.cmpppc = 1 "
Else
comandoResumen.CommandText += " AND b.cmpppc = 0 "
End If
Case 3
If filtros(1).Equals("Paciente PPC") Then
comandoResumen.CommandText += " AND b.cmpppc = 1 "
Else
comandoResumen.CommandText += " AND b.cmpppc = 0 "
End If
comandoResumen.CommandText += " AND a.cmpidgrupo = '" +
filtros(2) + "'"
Case 4
If filtros(1).Equals("Paciente PPC") Then
comandoResumen.CommandText += " AND b.cmpppc = 1 "
Else
comandoResumen.CommandText += " AND b.cmpppc = 0 "
End If
comandoResumen.CommandText += " AND a.cmpidgrupo = '" +
filtros(2) + "'"
comandoResumen.CommandText += " AND a.cmpgenero = '" +
filtros(3) + "'"
Case 5
If filtros(1).Equals("Paciente PPC") Then
comandoResumen.CommandText += " AND b.cmpppc = 1 "
Else
comandoResumen.CommandText += " AND b.cmpppc = 0 "
End If
comandoResumen.CommandText += " AND a.cmpidgrupo = '" +
filtros(2) + "'"
comandoResumen.CommandText += " AND a.cmpgenero = '" +
filtros(3) + "'"

```

```

                comandoResumen.CommandText += " AND
(a.cmpnombrespersona like '"' + filtros(4) + "%' OR a.cmpapellidospersona like '"' +
filtros(4) + "%')
            End Select
        End Select

    End If
    Case 4 'planificacion de citas..
        Me.comandoResumen.CommandText = " SELECT COUNT(*)"
        Me.comandoResumen.CommandText += " FROM tblcitas c LEFT JOIN
tblexpediente e ON(c.cmpidexpediente = e.cmpidexpediente) "
        Me.comandoResumen.CommandText += " LEFT JOIN tblpersonas p
ON(p.cmpidpersona = e.cmpidpersona) "
        Me.comandoResumen.CommandText += " LEFT JOIN tblpersonas k
ON(k.cmpidpersona = c.cmpidpersona) "
        Me.comandoResumen.CommandText += " WHERE c.cmpidcita > 0 "
        If Me.filtroArbol.Length > 0 Then
            If Me.arbol.SelectedNode.Text.Equals("HOY") Then
                Dim hoy As New Date
                Me.comandoResumen.CommandText += " AND
date_format(c.cmphoracita, '%Y/%m/%d') = '" + Format(hoy.Today,
"yyyy/MM/dd").ToString + "' "
            Else
                Dim filtros As String()
                filtros = Me.filtroArbol.Split("\")
                Select Case filtros.Length
                    Case 1
                        comandoResumen.CommandText += " AND
DATE_FORMAT(c.cmphoracita, '%Y') = '" + Me.arbol.SelectedNode.Tag + "' "
                    Case 2
                        comandoResumen.CommandText += " AND
DATE_FORMAT(c.cmphoracita, '%Y') = '" + Me.arbol.SelectedNode.Parent.Text + "' "
                        comandoResumen.CommandText += " AND
DATE_FORMAT(c.cmphoracita, '%c') = '" + Me.arbol.SelectedNode.Tag + "' "
                End Select
            End If
        End If

    End If
    Case 5 ' practicantes
        Me.comandoResumen.CommandText = " SELECT COUNT(*)"
        Me.comandoResumen.CommandText += " FROM tblpersonas a LEFT JOIN
tbloperadores b ON(a.cmpidpersona = b.cmpidpersona) "
        Me.comandoResumen.CommandText += " WHERE b.cmpidpersona > 0 "
        If Me.filtroArbol.Length > 0 Then
            Dim filtros As String()
            filtros = Me.filtroArbol.Split("\")
            Select Case filtros.Length

```

```

        Case 1
            comandoResumen.CommandText += " AND
DATE_FORMAT(b.cmpiniciorotacion, '%Y') = " + Me.arbol.SelectedNode.Tag + " "
        Case 2
            comandoResumen.CommandText += " AND
DATE_FORMAT(b.cmpiniciorotacion, '%Y') = " + Me.arbol.SelectedNode.Parent.Text + "
"
            comandoResumen.CommandText += " AND UPPER(b.cmpciclo) = ""
+ UCase(Me.arbol.SelectedNode.Tag) + "" "
        End Select
    End If
    Case 6 'equipos e instrumentos...
        comandoResumen.CommandText = "SELECT COUNT(*) "
        comandoResumen.CommandText += "FROM tblmantenimientoequipo a
LEFT JOIN tblequipo b ON(a.cmpidequipo=b.cmpidequipo) "
        comandoResumen.CommandText += "WHERE a.cmpidmantenimiento > 0 "
        If Me.filtroArbol.Length > 0 Then
            Dim filtros As String()
            filtros = Me.filtroArbol.Split("\")
            Select Case filtros(0)
                Case "REALIZADOS"
                    comandoResumen.CommandText += "AND a.cmprealizado = 1 "
                    Select Case filtros.Length
                        Case 2
                            comandoResumen.CommandText += "AND a.cmpaprobado = " +
Me.arbol.SelectedNode.Tag
                        Case 3
                            comandoResumen.CommandText += " AND a.cmpaprobado = " +
Me.arbol.SelectedNode.Parent.Tag
                            comandoResumen.CommandText += " AND
DATE_FORMAT(a.cmpfecha, '%Y') = " + filtros(2) + " "
                        Case 4
                            comandoResumen.CommandText += " AND a.cmpaprobado = " +
Me.arbol.SelectedNode.Parent.Parent.Tag
                            comandoResumen.CommandText += " AND
DATE_FORMAT(a.cmpfecha, '%Y') = " + Me.arbol.SelectedNode.Parent.Text + " "
                            comandoResumen.CommandText += " AND
DATE_FORMAT(a.cmpfecha, '%c') = " + Me.arbol.SelectedNode.Tag + " "
                        End Select
                    Case "NO REALIZADOS"
                        comandoResumen.CommandText += "AND a.cmprealizado = 0 "
                        Select Case filtros.Length
                            Case 2
                                comandoResumen.CommandText += " AND
DATE_FORMAT(a.cmpfecha, '%Y') = " + filtros(1) + " "
                            Case 3

```

```

                                comandoResumen.CommandText += " AND
DATE_FORMAT(a.cmpfecha, '%Y') = " + filtros(1) + " "
                                comandoResumen.CommandText += " AND
DATE_FORMAT(a.cmpfecha, '%c') = " + Me.arbol.SelectedNode.Tag + " "
                                End Select
                                End Select
                                End If

Case 7 ' medicamentos
If Me.filtroArbol.Length > 0 Then
    Me.comandoResumen.CommandText = "SELECT COUNT(*) "
    comandoResumen.CommandText += " FROM tblmovimientomateriales a "
    comandoResumen.CommandText += " "
    comandoResumen.CommandText += " WHERE a.cmpidmovimiento > 0 "
    Dim filtros As String()
    filtros = Me.filtroArbol.Split("\")
    Select Case filtros.Length
        Case 1 'filtro por tipo de movimiento...
            comandoResumen.CommandText += " AND a.cmpidtipomov = "" +
Me.arbol.SelectedNode.Tag + ""
            Case 2 'filtro por año..
                comandoResumen.CommandText += " AND a.cmpidtipomov = "" +
Me.arbol.SelectedNode.Parent.Tag + ""
                comandoResumen.CommandText += " AND
DATE_FORMAT(a.cmpfechamovimiento, '%Y') = " + Me.arbol.SelectedNode.Text + " "
                Case 3 'filtro por mes
                    comandoResumen.CommandText += " AND a.cmpidtipomov = "" +
Me.arbol.SelectedNode.Parent.Parent.Tag + ""
                    comandoResumen.CommandText += " AND
DATE_FORMAT(a.cmpfechamovimiento, '%Y') = " + Me.arbol.SelectedNode.Parent.Text
+ " "
                    comandoResumen.CommandText += " AND
DATE_FORMAT(a.cmpfechamovimiento, '%c') = " + Me.arbol.SelectedNode.Tag + " "
                    Case 4 'filtro por tipo de materiales..
                        comandoResumen.CommandText += " AND a.cmpidtipomov = "" +
Me.arbol.SelectedNode.Parent.Parent.Parent.Tag + ""
                        comandoResumen.CommandText += " AND
DATE_FORMAT(a.cmpfechamovimiento, '%Y') = " +
Me.arbol.SelectedNode.Parent.Parent.Text + " "
                        comandoResumen.CommandText += " AND
DATE_FORMAT(a.cmpfechamovimiento, '%c') = " + Me.arbol.SelectedNode.Parent.Tag
+ " AND ("
                            comandoResumen.CommandText += " (SELECT COUNT(*) FROM
tbldetallesalidas WHERE cmpidmovimiento=a.cmpidmovimiento AND "
                                comandoResumen.CommandText += " cmpidmaterial IN (SELECT
tmp.cmpidmaterial FROM tblmateriales tmp WHERE tmp.cmpidtipomaterial=" +
Me.arbol.SelectedNode.Tag + ")) > 0 OR "

```

```

                comandoResumen.CommandText += " (SELECT COUNT(*) FROM
tbldetalleentradas WHERE cmpidmovimiento=a.cmpidmovimiento AND "
                comandoResumen.CommandText += " cmpidmaterial IN (SELECT
tmp.cmpidmaterial FROM tblmateriales tmp WHERE tmp.cmpidtipomaterial=" +
Me.arbol.SelectedNode.Tag + ") > 0 )"
            End Select
        End If
        Case 8 'Datos de inventarios...
    End Select
    Me.totalRegistros = Me.comandoResumen.ExecuteScalar
    For i = 1 To Math.Ceiling(Me.totalRegistros / paginadorGeneral)
        Me.cbbPaginacion.Items.Add(New cbbItem("pagina " + i.ToString + " de " +
Math.Ceiling(Me.totalRegistros / paginadorGeneral).ToString, i.ToString))
    Next
    If cbbPaginacion.Items.Count > 0 Then
        Me.cbbPaginacion.SelectedIndex = 0
    End If
    Me.conexionResumen.Close()
    Catch ex As Exception
        logError(Me.objModulos, ex)
    End Try
End Sub
Private Sub configurarArbol()
    Try
        Dim abecedario As String = "ABCDEFGHIJKLMÑNOPQRSTUVWXYZ"
        Dim i As Integer
        Me.filtroArbol = ""
        Me.arbol.Nodes.Clear()
        Dim ppc As TreeNode
        Dim ppctemp As TreeNode
        Dim temp As TreeNode
        Select Case Me.actualModulo
            Case 1 'Expedientes...
                ppc = Me.arbol.Nodes.Add("Paciente PPC", "Pacientes PPC", 7)
                ppctemp = ppc.Nodes.Add("Paciente Adulto", "Paciente Adulto",
0).Nodes.Add("Hombre", "Hombre")
                For i = 0 To abecedario.Length - 1
                    ppctemp.Nodes.Add(abecedario.Chars(i), abecedario.Chars(i))
                Next
                ppctemp = Me.arbol.Nodes("Paciente PPC").Nodes("Paciente
Adulto").Nodes.Add("Mujer", "Mujer")
                For i = 0 To abecedario.Length - 1
                    ppctemp.Nodes.Add(abecedario.Chars(i), abecedario.Chars(i))
                Next
                ppctemp = ppc.Nodes.Add("Paciente Infante", "Paciente
Infante").Nodes.Add("Hombre", "Hombre")
                For i = 0 To abecedario.Length - 1

```

```

        ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
    Next
    ppctemp = Me.arbol.Nodes("Paciente PPC").Nodes("Paciente
Infante").Nodes.Add("Mujer", "Mujer")
    For i = 0 To abcedario.Length - 1
        ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
    Next

    ppc = Me.arbol.Nodes.Add("Paciente No PPC", "Pacientes No PPC")
    ppctemp = ppc.Nodes.Add("Paciente Adulto", "Paciente Adulto",
0).Nodes.Add("Hombre", "Hombre")
    For i = 0 To abcedario.Length - 1
        ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
    Next
    ppctemp = Me.arbol.Nodes("Paciente No PPC").Nodes("Paciente
Adulto").Nodes.Add("Mujer", "Mujer")
    For i = 0 To abcedario.Length - 1
        ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
    Next
    ppctemp = ppc.Nodes.Add("Paciente Infante", "Paciente
Infante").Nodes.Add("Hombre", "Hombre")
    For i = 0 To abcedario.Length - 1
        ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
    Next
    ppctemp = Me.arbol.Nodes("Paciente No PPC").Nodes("Paciente
Infante").Nodes.Add("Mujer", "Mujer")
    For i = 0 To abcedario.Length - 1
        ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
    Next
Case 2 'Consultas...
    Dim mes As Integer
    If Me.conexionResumen.State = ConnectionState.Closed Then
        Me.conexionResumen.Open()
    Else
        Me.conexionResumen.Close()
        Me.conexionResumen.Open()
    End If
    Me.comandoResumen.CommandText = "SELECT
@mas:=if(max(DATE_FORMAT(cmpfechaconsulta,'%Y')),max(DATE_FORMAT(cmpfe
chaconsulta,'%Y')),0) anomax FROM tblconsulta"
    Dim anoMax As Integer = Me.comandoResumen.ExecuteScalar
    Me.comandoResumen.CommandText = "SELECT
@min:=if(min(DATE_FORMAT(cmpfechaconsulta,'%Y')),min(DATE_FORMAT(cmpfec
haconsulta,'%Y')),0) anomin FROM tblconsulta"
    Dim anoMin As Integer = Me.comandoResumen.ExecuteScalar
    ppc = Me.arbol.Nodes.Add("Paciente PPC", "Pacientes PPC")

```

```

        ppctemp = ppc.Nodes.Add("Paciente Adulto", "Paciente
Adulto").Nodes.Add("Hombre", "Hombre")
        For i = anoMax To anoMin Step -1
            ppctemp.Nodes.Add(i.ToString, i.ToString)
            For Each mes In Me.meses.Keys
                ppctemp.Nodes(i.ToString).Nodes.Add(mes.ToString, Me.meses(mes))
            Next
        Next
        ppctemp = Me.arbol.Nodes("Paciente PPC").Nodes("Paciente
Adulto").Nodes.Add("Mujer", "Mujer")
        For i = anoMax To anoMin Step -1
            ppctemp.Nodes.Add(i.ToString, i.ToString)
            For Each mes In Me.meses.Keys
                ppctemp.Nodes(i.ToString).Nodes.Add(mes.ToString, Me.meses(mes))
            Next
        Next
        ppctemp = ppc.Nodes.Add("Paciente Infante", "Paciente
Infante").Nodes.Add("Hombre", "Hombre")
        For i = anoMax To anoMin Step -1
            ppctemp.Nodes.Add(i.ToString, i.ToString)
            For Each mes In Me.meses.Keys
                ppctemp.Nodes(i.ToString).Nodes.Add(mes.ToString, Me.meses(mes))
            Next
        Next
        ppctemp = Me.arbol.Nodes("Paciente PPC").Nodes("Paciente
Infante").Nodes.Add("Mujer", "Mujer")
        For i = anoMax To anoMin Step -1
            ppctemp.Nodes.Add(i.ToString, i.ToString)
            For Each mes In Me.meses.Keys
                ppctemp.Nodes(i.ToString).Nodes.Add(mes.ToString, Me.meses(mes))
            Next
        Next
        ppc = Me.arbol.Nodes.Add("Paciente No PPC", "Pacientes No PPC")
        ppctemp = ppc.Nodes.Add("Paciente Adulto", "Paciente
Adulto").Nodes.Add("Hombre", "Hombre")
        For i = anoMax To anoMin Step -1
            ppctemp.Nodes.Add(i.ToString, i.ToString)
            For Each mes In Me.meses.Keys
                ppctemp.Nodes(i.ToString).Nodes.Add(mes.ToString, Me.meses(mes))
            Next
        Next
        ppctemp = Me.arbol.Nodes("Paciente No PPC").Nodes("Paciente
Adulto").Nodes.Add("Mujer", "Mujer")
        For i = anoMax To anoMin Step -1
            ppctemp.Nodes.Add(i.ToString, i.ToString)
            For Each mes In Me.meses.Keys
                ppctemp.Nodes(i.ToString).Nodes.Add(mes.ToString, Me.meses(mes))
            Next
        Next

```

```

    Next
    Next
    ppctemp = ppc.Nodes.Add("Paciente Infante", "Paciente
Infante").Nodes.Add("Hombre", "Hombre")
    For i = anoMax To anoMin Step -1
        ppctemp.Nodes.Add(i.ToString, i.ToString)
        For Each mes In Me.meses.Keys
            ppctemp.Nodes(i.ToString).Nodes.Add(mes.ToString, Me.meses(mes))
        Next
    Next
    Next
    ppctemp = Me.arbol.Nodes("Paciente No PPC").Nodes("Paciente
Infante").Nodes.Add("Mujer", "Mujer")
    For i = anoMax To anoMin Step -1
        ppctemp.Nodes.Add(i.ToString, i.ToString)
        For Each mes In Me.meses.Keys
            ppctemp.Nodes(i.ToString).Nodes.Add(mes.ToString, Me.meses(mes))
        Next
    Next
    Next
    Case 3 ' CUENTAS POR COBRAR Y REMESAS
    ppc = Me.arbol.Nodes.Add("CUENTAS", "CUENTAS",
7).Nodes.Add("Paciente PPC", "Paciente PPC")
    ppctemp = ppc.Nodes.Add("Paciente Adulto", "Paciente Adulto",
0).Nodes.Add("Hombre", "Hombre")
    For i = 0 To abcedario.Length - 1
        ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
    Next
    ppctemp = ppc.Nodes("Paciente Adulto").Nodes.Add("Mujer", "Mujer")
    For i = 0 To abcedario.Length - 1
        ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
    Next
    Next
    ppctemp = ppc.Nodes.Add("Paciente Infante", "Paciente
Infante").Nodes.Add("Hombre", "Hombre")
    For i = 0 To abcedario.Length - 1
        ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
    Next
    Next
    ppctemp = ppc.Nodes("Paciente Infante").Nodes.Add("Mujer", "Mujer")
    For i = 0 To abcedario.Length - 1
        ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
    Next
    Next

    ppc = Me.arbol.Nodes("CUENTAS").Nodes.Add("Paciente No PPC",
"Pacientes No PPC")
    ppctemp = ppc.Nodes.Add("Paciente Adulto", "Paciente Adulto",
0).Nodes.Add("Hombre", "Hombre")
    For i = 0 To abcedario.Length - 1
        ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
    Next
    Next

```

```

ppctemp = ppc.Nodes("Paciente Adulto").Nodes.Add("Mujer", "Mujer")
For i = 0 To abcedario.Length - 1
    ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
Next
ppctemp = ppc.Nodes.Add("Paciente Infante", "Paciente
Infante").Nodes.Add("Hombre", "Hombre")
For i = 0 To abcedario.Length - 1
    ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
Next
ppctemp = ppc.Nodes("Paciente Infante").Nodes.Add("Mujer", "Mujer")
For i = 0 To abcedario.Length - 1
    ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
Next

ppc = Me.arbol.Nodes.Add("SALDOS ROJOS", "SALDOS ROJOS",
7).Nodes.Add("Paciente PPC", "Paciente PPC")
ppctemp = ppc.Nodes.Add("Paciente Adulto", "Paciente Adulto",
0).Nodes.Add("Hombre", "Hombre")
For i = 0 To abcedario.Length - 1
    ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
Next
ppctemp = ppc.Nodes("Paciente Adulto").Nodes.Add("Mujer", "Mujer")
For i = 0 To abcedario.Length - 1
    ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
Next
ppctemp = ppc.Nodes.Add("Paciente Infante", "Paciente
Infante").Nodes.Add("Hombre", "Hombre")
For i = 0 To abcedario.Length - 1
    ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
Next
ppctemp = ppc.Nodes("Paciente Infante").Nodes.Add("Mujer", "Mujer")
For i = 0 To abcedario.Length - 1
    ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
Next

ppc = Me.arbol.Nodes("SALDOS ROJOS").Nodes.Add("Paciente No PPC",
"Paciente No PPC")
ppctemp = ppc.Nodes.Add("Paciente Adulto", "Paciente Adulto",
0).Nodes.Add("Hombre", "Hombre")
For i = 0 To abcedario.Length - 1
    ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
Next
ppctemp = ppc.Nodes("Paciente Adulto").Nodes.Add("Mujer", "Mujer")
For i = 0 To abcedario.Length - 1
    ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
Next

```

```

        ppctemp = ppc.Nodes.Add("Paciente Infante", "Paciente
Infante").Nodes.Add("Hombre", "Hombre")
        For i = 0 To abcedario.Length - 1
            ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
        Next
        ppctemp = ppc.Nodes("Paciente Infante").Nodes.Add("Mujer", "Mujer")
        For i = 0 To abcedario.Length - 1
            ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
        Next

        ppc = Me.arbol.Nodes.Add("SALDOS AZULES", "SALDOS AZULES",
7).Nodes.Add("Paciente PPC", "Paciente PPC")
        ppctemp = ppc.Nodes.Add("Paciente Adulto", "Paciente Adulto",
0).Nodes.Add("Hombre", "Hombre")
        For i = 0 To abcedario.Length - 1
            ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
        Next
        ppctemp = ppc.Nodes("Paciente Adulto").Nodes.Add("Mujer", "Mujer")
        For i = 0 To abcedario.Length - 1
            ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
        Next
        ppctemp = ppc.Nodes.Add("Paciente Infante", "Paciente
Infante").Nodes.Add("Hombre", "Hombre")
        For i = 0 To abcedario.Length - 1
            ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
        Next
        ppctemp = ppc.Nodes("Paciente Infante").Nodes.Add("Mujer", "Mujer")
        For i = 0 To abcedario.Length - 1
            ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
        Next

        ppc = Me.arbol.Nodes("SALDOS AZULES").Nodes.Add("Paciente No PPC",
"Paciente No PPC")
        ppctemp = ppc.Nodes.Add("Paciente Adulto", "Paciente Adulto",
0).Nodes.Add("Hombre", "Hombre")
        For i = 0 To abcedario.Length - 1
            ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
        Next
        ppctemp = ppc.Nodes("Paciente Adulto").Nodes.Add("Mujer", "Mujer")
        For i = 0 To abcedario.Length - 1
            ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
        Next
        ppctemp = ppc.Nodes.Add("Paciente Infante", "Paciente
Infante").Nodes.Add("Hombre", "Hombre")
        For i = 0 To abcedario.Length - 1
            ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
        Next

```

```

ppctemp = ppc.Nodes("Paciente Infante").Nodes.Add("Mujer", "Mujer")
For i = 0 To abcedario.Length - 1
    ppctemp.Nodes.Add(abcedario.Chars(i), abcedario.Chars(i))
Next
ppc = Me.arbol.Nodes.Add("REMESAS", "REMESAS", 7)
Case 4 ' PLANIFICACIONES Y CITAS
If Me.conexionResumen.State = ConnectionState.Closed Then
    Me.conexionResumen.Open()
Else
    Me.conexionResumen.Close()
    Me.conexionResumen.Open()
End If
Me.comandoResumen.CommandText = "SELECT
@mas:=if(max(DATE_FORMAT(cmpchoracita,'%Y')),max(DATE_FORMAT(cmpchoracita,
,'%Y')),0) anomax FROM tblcitas"
Dim anoMax As Integer = Me.comandoResumen.ExecuteScalar
Me.comandoResumen.CommandText = "SELECT
@min:=if(min(DATE_FORMAT(cmpchoracita,'%Y')),min(DATE_FORMAT(cmpchoracita,'
%Y')),0) anomin FROM tblcitas"
Dim anoMin As Integer = Me.comandoResumen.ExecuteScalar
If anoMax > 0 Then
    For i = anoMax To anoMin Step -1
        If i = anoMax Then
            Me.arbol.Nodes.Add("HOY", "HOY")
        End If
        temp = Me.arbol.Nodes.Add(i.ToString, i.ToString)
        Me.arbol.Nodes(i.ToString).Tag = i.ToString
        For Each m As Integer In Me.meses.Keys
            temp.Nodes.Add(m.ToString, Me.meses(m))
            temp.Nodes(m.ToString).Tag = m.ToString
        Next
    Next
End If
Case 5 ' PRACTICANTES
If Me.conexionResumen.State = ConnectionState.Closed Then
    Me.conexionResumen.Open()
Else
    Me.conexionResumen.Close()
    Me.conexionResumen.Open()
End If
Me.comandoResumen.CommandText = "SELECT
@mas:=if(max(DATE_FORMAT(cmpiniciorotacion,'%Y')),max(DATE_FORMAT(cmpini
ciorotacion,'%Y')),0) anomax FROM tbloperadores"
Dim anoMax As Integer = Me.comandoResumen.ExecuteScalar
Me.comandoResumen.CommandText = "SELECT
@min:=if(min(DATE_FORMAT(cmpiniciorotacion,'%Y')),min(DATE_FORMAT(cmpini
ciorotacion,'%Y')),0) anomin FROM tbloperadores"

```

```

Dim anoMin As Integer = Me.comandoResumen.ExecuteScalar
If anoMax > 0 Then
    For i = anoMax To anoMin Step -1
        temp = Me.arbol.Nodes.Add(i.ToString, i.ToString)
        Me.arbol.Nodes(i.ToString).Tag = i.ToString
        For Each m As String In Me.ciclos.Keys
            temp.Nodes.Add(m, Me.ciclos(m))
            temp.Nodes(m).Tag = m
        Next
    Next
End If
Case 6 ' MANTENIMIENTO DE EQUIPOS
If Me.conexionResumen.State = ConnectionState.Closed Then
    Me.conexionResumen.Open()
Else
    Me.conexionResumen.Close()
    Me.conexionResumen.Open()
End If
Me.comandoResumen.CommandText = "SELECT * FROM tbltipoequipo t;"
Dim lector As MySqlDataReader = Me.comandoResumen.ExecuteReader
Dim tipos As New Hashtable
While lector.Read
    tipos.Add((lector!cmpidtipo).ToString, (lector!cmpdescripcion).ToString)
End While
lector.Close()
Me.comandoResumen.CommandText = "SELECT
@mas:=if(max(DATE_FORMAT(cmpfecha,'%Y')),max(DATE_FORMAT(cmpfecha,'%Y')
)),0) anomax from tblmantenimientoequipo"
Dim anoMax As Integer = Me.comandoResumen.ExecuteScalar
Me.comandoResumen.CommandText = "SELECT
@min:=if(min(DATE_FORMAT(cmpfecha,'%Y')),min(DATE_FORMAT(cmpfecha,'%Y')
),0) anomin from tblmantenimientoequipo"
Dim anoMin As Integer = Me.comandoResumen.ExecuteScalar
ppc = Me.arbol.Nodes.Add("REALIZADOS", "REALIZADOS")
ppc.Tag = "1"
ppc.Nodes.Add("APROBADOS", "APROBADOS")
ppc.Nodes("APROBADOS").Tag = "1"
ppc.Nodes.Add("NO APROBADOS", "NO APROBADOS")
ppc.Nodes("NO APROBADOS").Tag = "0"
ppc = ppc.Nodes("APROBADOS")
For i = anoMax To anoMin Step -1
    temp = ppc.Nodes.Add(i.ToString, i.ToString)
    For Each m As Integer In Me.meses.Keys
        temp.Nodes.Add(m.ToString, Me.meses(m))
        temp.Nodes(m.ToString).Tag = m.ToString
    For Each s As String In tipos.Keys
        temp.Nodes(m.ToString).Nodes.Add(s, tipos(s))
    Next
    Next

```

```

        temp.Nodes(m.ToString).Nodes(s).Tag = s
    Next
Next
Next
ppc = Me.arbol.Nodes("REALIZADOS").Nodes("NO APROBADOS")
For i = anoMax To anoMin Step -1
    temp = ppc.Nodes.Add(i.ToString, i.ToString)
    For Each m As Integer In Me.meses.Keys
        temp.Nodes.Add(m.ToString, Me.meses(m))
        temp.Nodes(m.ToString).Tag = m.ToString
        For Each s As String In tipos.Keys
            temp.Nodes(m.ToString).Nodes.Add(s, tipos(s))
            temp.Nodes(m.ToString).Nodes(s).Tag = s
        Next
    Next
Next
ppc = Me.arbol.Nodes.Add("NO REALIZADOS", "NO REALIZADOS")
For i = anoMax To anoMin Step -1
    temp = ppc.Nodes.Add(i.ToString, i.ToString)
    For Each m As Integer In Me.meses.Keys
        temp.Nodes.Add(m.ToString, Me.meses(m))
        temp.Nodes(m.ToString).Tag = m.ToString
        For Each s As String In tipos.Keys
            temp.Nodes(m.ToString).Nodes.Add(s, tipos(s))
            temp.Nodes(m.ToString).Nodes(s).Tag = s
        Next
    Next
Next
Case 7
If Me.conexionResumen.State = ConnectionState.Closed Then
    Me.conexionResumen.Open()
Else
    Me.conexionResumen.Close()
    Me.conexionResumen.Open()
End If
Me.comandoResumen.CommandText = "SELECT
@mas:=if(max(DATE_FORMAT(cmpfechamovimiento,'%Y')),max(DATE_FORMAT(cm
pfechamovimiento,'%Y'),0) anomax from tblmovimientomateriales"
Dim anoMax As Integer = Me.comandoResumen.ExecuteScalar
Me.comandoResumen.CommandText = "SELECT
@min:=if(min(DATE_FORMAT(cmpfechamovimiento,'%Y')),min(DATE_FORMAT(cm
pfechamovimiento,'%Y'),0) anomin from tblmovimientomateriales"
Dim anoMin As Integer = Me.comandoResumen.ExecuteScalar
Me.comandoResumen.CommandText = "SELECT cmpdescripcion,
IF(cmpsigno=0,'-', '+') signo, cmpidtipomovimiento FROM tbltiposmovimiento t WHERE
cmpactivo=1"
Dim nodo As TreeNode

```

```

        Dim lector As MySqlDataReader = Me.comandoResumen.ExecuteReader
        While lector.Read
            nodo = Me.arbol.Nodes.Add((lector!cmpidtipomovimiento).ToString,
lector!cmpdescripcion)
            nodo.Tag = (lector!cmpidtipomovimiento).ToString
        End While
        lector.Close()
        Me.comandoResumen.CommandText = "SELECT * FROM tbltipomaterial"
        lector = Me.comandoResumen.ExecuteReader
        Dim tipos As New Hashtable
        While lector.Read
            tipos.Add((lector!cmpidtipomaterial).ToString,
(lector!cmpnombretipo).ToString)
        End While
        lector.Close()
        For Each nodo In Me.arbol.Nodes
            For i = anoMax To anoMin Step -1
                temp = nodo.Nodes.Add(i.ToString, i.ToString)
                For Each m As Integer In Me.meses.Keys
                    temp.Nodes.Add(m.ToString, Me.meses(m))
                    temp.Nodes(m.ToString).Tag = m.ToString
                    For Each s As String In tipos.Keys
                        temp.Nodes(m.ToString).Nodes.Add(s, tipos(s))
                        temp.Nodes(m.ToString).Nodes(s).Tag = s
                    Next
                Next
            Next
        End Select
        Me.llenarPaginador()
        Catch ex As Exception
            logError(Me.objModulos, ex)
        End Try

    End Sub
    Private Sub InitializeComponent()
        Me.components = New System.ComponentModel.Container
        Dim resources As System.ComponentModel.ComponentResourceManager = New
System.ComponentModel.ComponentResourceManager(GetType(summary))
        Dim DataGridViewCellStyle5 As System.Windows.Forms.DataGridViewCellStyle =
New System.Windows.Forms.DataGridViewCellStyle
        Dim DataGridViewCellStyle6 As System.Windows.Forms.DataGridViewCellStyle =
New System.Windows.Forms.DataGridViewCellStyle
        Dim DataGridViewCellStyle7 As System.Windows.Forms.DataGridViewCellStyle =
New System.Windows.Forms.DataGridViewCellStyle
        Dim DataGridViewCellStyle8 As System.Windows.Forms.DataGridViewCellStyle =
New System.Windows.Forms.DataGridViewCellStyle
    
```

```

Me.panel = New System.Windows.Forms.SplitContainer
Me.arbol = New System.Windows.Forms.TreeView
Me.treeImages = New System.Windows.Forms.ImageList(Me.components)
Me.rejilla = New System.Windows.Forms.DataGridView
Me.barra = New System.Windows.Forms.ToolStrip
Me.tsbVer = New System.Windows.Forms.ToolStripButton
Me.tsbAgregar = New System.Windows.Forms.ToolStripButton
Me.tsbModificar = New System.Windows.Forms.ToolStripButton
Me.tsbEliminar = New System.Windows.Forms.ToolStripButton
Me.tsbSalir = New System.Windows.Forms.ToolStripButton
Me.tsbddModulos = New System.Windows.Forms.ToolStripDropDownButton
Me.ToolStripSeparator1 = New System.Windows.Forms.ToolStripSeparator
Me.tsbevaluaciones = New System.Windows.Forms.ToolStripButton
Me.cbbPaginacion = New System.Windows.Forms.ToolStripComboBox
Me.panel.Panel1.SuspendLayout()
Me.panel.Panel2.SuspendLayout()
Me.panel.SuspendLayout()
CType(Me.rejilla, System.ComponentModel.ISupportInitialize).BeginInit()
Me.barra.SuspendLayout()
Me.SuspendLayout()
'
'panel
'
Me.panel.Dock = System.Windows.Forms.DockStyle.Bottom
Me.panel.Location = New System.Drawing.Point(0, 38)
Me.panel.Name = "panel"
'
'panel.Panel1
'
Me.panel.Panel1.Controls.Add(Me.arbol)
'
'panel.Panel2
'
Me.panel.Panel2.Controls.Add(Me.rejilla)
Me.panel.Size = New System.Drawing.Size(885, 435)
Me.panel.SplitterDistance = 228
Me.panel.TabIndex = 1
'
'arbol
'
Me.arbol.Dock = System.Windows.Forms.DockStyle.Fill
Me.arbol.ImageIndex = 0
Me.arbol.ImageList = Me.treeImages
Me.arbol.Location = New System.Drawing.Point(0, 0)
Me.arbol.Name = "arbol"
Me.arbol.SelectedIndex = 0
Me.arbol.Size = New System.Drawing.Size(228, 435)

```

```

Me.arbol.TabIndex = 0
'
'treeImages
'
Me.treeImages.ImageStream =
CType(resources.GetObject("treeImages.ImageStream"),
System.Windows.Forms.ImageListStreamer)
Me.treeImages.TransparentColor = System.Drawing.Color.Transparent
Me.treeImages.Images.SetKeyName(0, "")
Me.treeImages.Images.SetKeyName(1, "")
Me.treeImages.Images.SetKeyName(2, "")
Me.treeImages.Images.SetKeyName(3, "")
Me.treeImages.Images.SetKeyName(4, "")
Me.treeImages.Images.SetKeyName(5, "")
Me.treeImages.Images.SetKeyName(6, "")
Me.treeImages.Images.SetKeyName(7, "")
Me.treeImages.Images.SetKeyName(8, "")
'
'rejilla
'
Me.rejilla.AllowUserToAddRows = False
Me.rejilla.AllowUserToDeleteRows = False
Me.rejilla.AllowUserToResizeRows = False
Me.rejilla.AutoSizeColumnsMode =
System.Windows.Forms.DataGridViewAutoSizeColumnsMode.AllCells
Me.rejilla.ColumnHeadersBorderStyle =
System.Windows.Forms.DataGridViewHeaderBorderStyle.[Single]
DataGridViewCellStyle5.Alignment =
System.Windows.Forms.DataGridViewContentAlignment.MiddleLeft
DataGridViewCellStyle5.BackColor = System.Drawing.SystemColors.Control
DataGridViewCellStyle5.Font = New System.Drawing.Font("Microsoft Sans Serif",
8.25!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
DataGridViewCellStyle5.ForeColor = System.Drawing.SystemColors.WindowText
DataGridViewCellStyle5.SelectionBackColor =
System.Drawing.SystemColors.Highlight
DataGridViewCellStyle5.SelectionForeColor =
System.Drawing.SystemColors.HighlightText
DataGridViewCellStyle5.WrapMode =
System.Windows.Forms.DataGridViewTriState.[True]
Me.rejilla.ColumnHeadersDefaultCellStyle = DataGridViewCellStyle5
Me.rejilla.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize
DataGridViewCellStyle6.Alignment =
System.Windows.Forms.DataGridViewContentAlignment.MiddleLeft
DataGridViewCellStyle6.BackColor = System.Drawing.SystemColors.Window

```

```

        DataGridViewCellStyle6.Font = New System.Drawing.Font("Microsoft Sans Serif",
8.25!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
        DataGridViewCellStyle6.ForeColor = System.Drawing.SystemColors.ControlText
        DataGridViewCellStyle6.SelectionBackColor =
System.Drawing.SystemColors.Highlight
        DataGridViewCellStyle6.SelectionForeColor =
System.Drawing.SystemColors.HighlightText
        DataGridViewCellStyle6.WrapMode =
System.Windows.Forms.DataGridViewTriState.[False]
        Me.rejilla.DefaultCellStyle = DataGridViewCellStyle6
        Me.rejilla.Dock = System.Windows.Forms.DockStyle.Fill
        Me.rejilla.Location = New System.Drawing.Point(0, 0)
        Me.rejilla.MultiSelect = False
        Me.rejilla.Name = "rejilla"
        Me.rejilla.ReadOnly = True
        Me.rejilla.RowHeadersBorderStyle =
System.Windows.Forms.DataGridViewHeaderBorderStyle.None
        DataGridViewCellStyle7.Alignment =
System.Windows.Forms.DataGridViewContentAlignment.MiddleLeft
        DataGridViewCellStyle7.BackColor = System.Drawing.SystemColors.Control
        DataGridViewCellStyle7.Font = New System.Drawing.Font("Microsoft Sans Serif",
8.25!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0,
Byte))
        DataGridViewCellStyle7.ForeColor = System.Drawing.SystemColors.WindowText
        DataGridViewCellStyle7.SelectionBackColor =
System.Drawing.SystemColors.Highlight
        DataGridViewCellStyle7.SelectionForeColor =
System.Drawing.SystemColors.HighlightText
        DataGridViewCellStyle7.WrapMode =
System.Windows.Forms.DataGridViewTriState.[True]
        Me.rejilla.RowHeadersDefaultCellStyle = DataGridViewCellStyle7
        Me.rejilla.RowHeadersVisible = False
        DataGridViewCellStyle8.WrapMode =
System.Windows.Forms.DataGridViewTriState.[True]
        Me.rejilla.RowsDefaultCellStyle = DataGridViewCellStyle8
        Me.rejilla.Size = New System.Drawing.Size(653, 435)
        Me.rejilla.TabIndex = 0
    '
    'barra
    '
        Me.barra.BackColor = System.Drawing.Color.White
        Me.barra.Font = New System.Drawing.Font("Arial", 8.25!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.barra.GripStyle = System.Windows.Forms.ToolStripGripStyle.Hidden

```

```

    Me.barra.Items.AddRange(New System.Windows.Forms.ToolStripItem() {Me.tsbVer,
Me.tsbAgregar, Me.tsbModificar, Me.tsbEliminar, Me.tsbSalir, Me.tsbddModulos,
Me.ToolStripSeparator1, Me.tsbevaluaciones, Me.cbbPaginacion})
    Me.barra.Location = New System.Drawing.Point(0, 0)
    Me.barra.Name = "barra"
    Me.barra.RenderMode = System.Windows.Forms.ToolStripRenderMode.Professional
    Me.barra.Size = New System.Drawing.Size(885, 25)
    Me.barra.Stretch = True
    Me.barra.TabIndex = 2
    Me.barra.Text = "ToolStrip1"
    '
    'tsbVer
    '
    Me.tsbVer.Image = Global.resumenForm.My.Resources.Resources.eyes
    Me.tsbVer.ImageTransparentColor = System.Drawing.Color.Magenta
    Me.tsbVer.Name = "tsbVer"
    Me.tsbVer.Size = New System.Drawing.Size(45, 22)
    Me.tsbVer.Text = "Ver"
    '
    'tsbAgregar
    '
    Me.tsbAgregar.Image = Global.resumenForm.My.Resources.Resources.page
    Me.tsbAgregar.ImageTransparentColor = System.Drawing.Color.Magenta
    Me.tsbAgregar.Name = "tsbAgregar"
    Me.tsbAgregar.Size = New System.Drawing.Size(67, 22)
    Me.tsbAgregar.Text = "Agregar"
    '
    'tsbModificar
    '
    Me.tsbModificar.Image = Global.resumenForm.My.Resources.Resources.page_edit
    Me.tsbModificar.ImageTransparentColor = System.Drawing.Color.Magenta
    Me.tsbModificar.Name = "tsbModificar"
    Me.tsbModificar.Size = New System.Drawing.Size(54, 22)
    Me.tsbModificar.Text = "Editar"
    '
    'tsbEliminar
    '
    Me.tsbEliminar.Image = CType(resources.GetObject("tsbEliminar.Image"),
System.Drawing.Image)
    Me.tsbEliminar.ImageTransparentColor = System.Drawing.Color.Magenta
    Me.tsbEliminar.Name = "tsbEliminar"
    Me.tsbEliminar.Size = New System.Drawing.Size(63, 22)
    Me.tsbEliminar.Text = "Eliminar"
    '
    'tsbSalir
    '
    Me.tsbSalir.Alignment = System.Windows.Forms.ToolStripItemAlignment.Right

```

```

Me.tsbSalir.Image = Global.resumenForm.My.Resources.Resources.cancel
Me.tsbSalir.ImageTransparentColor = System.Drawing.Color.Magenta
Me.tsbSalir.Name = "tsbSalir"
Me.tsbSalir.Size = New System.Drawing.Size(48, 22)
Me.tsbSalir.Text = "Salir"
'
'tsbddModulos
'
Me.tsbddModulos.Alignment =
System.Windows.Forms.ToolStripItemAlignment.Right
Me.tsbddModulos.Image = Global.resumenForm.My.Resources.Resources.bricks
Me.tsbddModulos.ImageTransparentColor = System.Drawing.Color.Magenta
Me.tsbddModulos.Name = "tsbddModulos"
Me.tsbddModulos.Size = New System.Drawing.Size(76, 22)
Me.tsbddModulos.Text = "Modulos"
'
'ToolStripSeparator1
'
Me.ToolStripSeparator1.Name = "ToolStripSeparator1"
Me.ToolStripSeparator1.Size = New System.Drawing.Size(6, 25)
'
'tsbevaluaciones
'
Me.tsbevaluaciones.Image = Global.resumenForm.My.Resources.Resources.chart_bar
Me.tsbevaluaciones.ImageTransparentColor = System.Drawing.Color.Magenta
Me.tsbevaluaciones.Name = "tsbevaluaciones"
Me.tsbevaluaciones.Size = New System.Drawing.Size(92, 22)
Me.tsbevaluaciones.Text = "Consolidados"
'
'cbbPaginacion
'
Me.cbbPaginacion.Name = "cbbPaginacion"
Me.cbbPaginacion.Size = New System.Drawing.Size(121, 25)
Me.cbbPaginacion.Text = "Páginas"
'
'summary
'
Me.AutoValidate =
System.Windows.Forms.AutoValidate.EnablePreventFocusChange
Me.BackColor = System.Drawing.SystemColors.ControlLightLight
Me.ClientSize = New System.Drawing.Size(885, 473)
Me.Controls.Add(Me.barra)
Me.Controls.Add(Me.panel)
Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.Fixed3D
Me.Name = "summary"
Me.ShowIcon = False
Me.WindowState = System.Windows.Forms.FormWindowState.Maximized

```

```

Me.panel.Panel1.ResumeLayout(False)
Me.panel.Panel2.ResumeLayout(False)
Me.panel.ResumeLayout(False)
CType(Me.rejilla, System.ComponentModel.ISupportInitialize).EndInit()
Me.barra.ResumeLayout(False)
Me.barra.PerformLayout()
Me.ResumeLayout(False)
Me.PerformLayout()

End Sub
Private Sub Eliminacion()
    Try
        If Me.rejilla.SelectedRows.Count > 0 Then
            Dim respuesta As MsgBoxResult
            respuesta = MsgBox("¿Esta seguro que desea eliminar este registro? " & Chr(13)
-
            , MsgBoxStyle.OkCancel, Me.objModulos.moduloToString())
            If (respuesta = MsgBoxResult.Ok) Then
                Select Case Me.objModulos.Modulo
                    Case 1
                        ' Eliminacion desactivada por convenio -- expedientes
                    Case 2
                        ' Eliminacion desactivada por convenio -- consultas
                    Case 3
                        ' Eliminacion desactivada por convenio -- cuentas por cobrar
                    Case 4
                        ' Eliminacion desactivada por convenio -- planificaciones y citas
                    Case 5
                        ' Eliminacion desactivada por convenio -- practicantes
                    Case 6
                        ' Eliminacion desactivada por convenio -- administracion de materiales
                    Case 7
                        ' Eliminacion desactivada por convenio -- revision de equipos
                End Select
            End If
        Else
            MsgBox("Debe Seleccionar un registro", MsgBoxStyle.Information,
Me.objModulos.moduloToString())
        End If
    Catch ex As Exception
        logError(Me.objModulos, ex)
    End Try

End Sub
Private Sub summary_Resize(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Resize

```

```

        Me.barra.Width = Me.Width
        Me.panel.Height = Me.Height - Me.barra.Height * 2.5
    End Sub
    Private Sub rejilla_RowHeaderDoubleClick1(ByVal sender As Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs) Handles
    rejilla.RowHeaderMouseDoubleClick
        MsgBox(Me.rejilla.SelectedCells().Item(0).Value.ToString())
    End Sub
    Private Sub tsbAgregar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tsbAgregar.Click
        Me.llamarFormsAgregar()
    End Sub
    Private Sub tsbModificar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tsbModificar.Click
        Me.llamarFormsModificar()
    End Sub
    Private Sub tsbSalir_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tsbSalir.Click
        Me.Dispose()
    End Sub
    Private Sub tsbEliminar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tsbEliminar.Click
        Me.Eliminacion()
    End Sub
#End Region
#Region "metodos publicos"
    Public Sub New()
        MyBase.New()
        Me.InitializeComponent()
        formatGrid(Me.rejilla)
    End Sub
    Public Sub ConfigureConModulo(ByVal confAcceso As AccesoModulos)
    Try
        Me.meses(1) = "ENE"
        Me.meses(2) = "FEB"
        Me.meses(3) = "MAR"
        Me.meses(4) = "ABR"
        Me.meses(5) = "MAY"
        Me.meses(6) = "JUN"
        Me.meses(7) = "JUL"
        Me.meses(8) = "AGO"
        Me.meses(9) = "SEP"
        Me.meses(10) = "OCT"
        Me.meses(11) = "NOV"
        Me.meses(12) = "DIC"
        Me.ciclos("CICLO-I") = "CICLO I"
    End Try
    End Sub

```

```

Me.ciclos("CICLO-II") = "CICLO II"
Me.objModulos = confAcceso
Me.actualModulo = Me.objModulos.Modulo
Me.StartPosition = FormStartPosition.CenterScreen
If Me.objModulos.AccesoDeModulo() < 2 Then
    Me.Text = objModulos.moduloToString() & ": ACCESO RESTRINGIDO!"
    MsgBox(Me.Text)
Else
    Me.Text = objModulos.moduloToString() & " - usuario:" &
Me.objModulos.usuario & " " & Me.objModulos.idgrupo
End If
conexionResumen = New MySqlConnection(Me.cadenaConexion.stringConexion)
comandoResumen.Connection = conexionResumen
Me.configurarBarra() ' se inicializa la barra
Me.configurarArbol()
Me.Refresh()
Catch ex As Exception
    logError(Me.objModulos, ex)
End Try

```

```

End Sub
#End Region

```

```

Private Sub summary_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
End Sub

```

```

Private Sub llamarFormsVer()
Try
    If Me.rejilla.SelectedRows.Count > 0 Then
        Select Case Me.objModulos.Modulo
        Case 1
            If Me.rejilla.SelectedRows.Count = 0 Then
                MsgBox("Seleccione un expediente", MsgBoxStyle.OkOnly,
Me.objModulos.moduloToString())
                Return
            End If
            Dim exp As New
expediente(Me.rejilla.Rows(Me.rejilla.SelectedRows(0).Index).Cells("idexpediente").Value
, Me.objModulos, False)
            exp.ShowDialog()
        Case 2
            If Me.rejilla.SelectedRows.Count.Equals(0) Then
                MsgBox("Seleccione una consulta para ver", MsgBoxStyle.OkOnly,
Me.objModulos.moduloToString())
                Return
            End If

```

```

        Dim cons As New consulta(Me.objModulos,
Me.rejilla.Rows(Me.rejilla.SelectedRows(0).Index).Cells("ID").Value,
Me.rejilla.Rows(Me.rejilla.SelectedRows(0).Index).Cells("IDEXP").Value, True)
        cons.ShowDialog()
    Case 3
        If Me.arbol.SelectedNode Is Nothing Then
            MsgBox("Seleccione una entrada en el árbol", MsgBoxStyle.Information,
Me.objModulos.moduloToString())
            Return
        End If
        If Me.arbol.SelectedNode.Text.Equals("REMESAS") Then
            Dim remesa As New Remesas()
            remesa.ShowDialog()
            Return
        End If
        If Me.rejilla.SelectedRows.Count = 0 Then
            MsgBox("Debe Seleccionar una cuenta", MsgBoxStyle.OkOnly,
Me.objModulos.moduloToString())
            Return
        End If
        If Me.rejilla.SelectedRows(0).Cells("idexpediente").Value = 0 Then
            MsgBox("Seleccione un registro de consulta")
            Return
        End If
        Dim verCx C As New VerCuentas(Me.rejilla.Item("idexpediente",
Me.rejilla.SelectedCells(0).RowIndex).Value, Me.objModulos)
        verCx C.ShowDialog()
    Case 4 'planificación de citas ....
        If Me.rejilla.SelectedRows.Count = 0 Then
            MsgBox("Seleccione una consulta", MsgBoxStyle.OkOnly,
Me.objModulos.moduloToString())
            Return
        End If
        Dim verCita As New registroCita(Me.rejilla.Item("IDCITA",
Me.rejilla.SelectedRows(0).Index).Value, Me.objModulos, False)
        verCita.ShowDialog()
        Me.crearConsulta()
        Me.llenarGrid()
    Case 5 'Practicantes...
        If Me.rejilla.SelectedRows.Count = 0 Then
            MsgBox("Seleccione un practicante", MsgBoxStyle.OkOnly,
Me.objModulos.moduloToString())
            Return
        End If
        Dim evals As New
evaluacionesPracticante(Me.rejilla.Item("IDOPERADOR",
Me.rejilla.SelectedRows(0).Index).Value, Me.objModulos)

```

```

        evals.ShowDialog()

    Case 6
        If Me.rejilla.SelectedRows.Count.Equals(0) Then
            MsgBox("Seleccione un mantenimiento para ver", MsgBoxStyle.OkOnly,
Me.objModulos.moduloToString())
            Return
        End If
        Dim revisiones As New revisionesequipo(Me.rejilla.Item("IDM",
Me.rejilla.SelectedRows(0).Index).Value.ToString, False, Me.objModulos)
        revisiones.ConfigureConModulo(Me.objModulos)
        revisiones.ShowDialog()
    Case 7
        If Me.rejilla.SelectedRows.Count.Equals(0) Then
            MsgBox("Seleccione un movimiento para ver", MsgBoxStyle.OkOnly,
Me.objModulos.moduloToString())
            Return
        End If
        'MsgBox(Me.rejilla.Item("ID",
Me.rejilla.SelectedCells(0).RowIndex).Value.ToString)
        Dim movi As New movimientomaterial(Me.rejilla.Item("ID",
Me.rejilla.SelectedCells(0).RowIndex).Value.ToString, False, Me.objModulos)
        movi.ConfigureConModulo(Me.objModulos)
        movi.ShowDialog()
    End Select
Else
    MsgBox("Seleccione un registro", MsgBoxStyle.OkOnly,
Me.objModulos.moduloToString())
End If
Catch ex As Exception
    logError(Me.objModulos, ex)
End Try
End Sub
Private Sub tsbVer_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tsbVer.Click
    Me.llamarFormsVer()
End Sub
Private Sub tsbfecha_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    Try
        Dim fecha As New fechas(Me.comandoResumen.CommandText,
Me.objModulos.Modulo)
        Dim respuesta As DialogResult
        respuesta = fecha.ShowDialog()
    Catch ex As Exception
        logError(Me.objModulos, ex)
    End Try

```

```

End Sub
Private Sub arbol_AfterSelect(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.TreeViewEventArgs) Handles arbol.AfterSelect
    Me.filtroArbol = Me.arbol.SelectedNode.FullPath
    Me.llenarPaginador()
    Me.crearConsulta()
    Me.llenarGrid()
End Sub
Private Sub cbbPaginacion_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles cbbPaginacion.SelectedIndexChanged
    Me.crearConsulta()
    Me.llenarGrid()
End Sub

End Class

```

#### Clases sencillas de lectura de datos de archivos XML

```

Imports System.IO
Imports System.Xml
Imports MySql.Data.MySqlClient
Public Class stringDeConexion
    Private cXML As XmlTextReader
    Private cadena As String = ""
    Private Sub cargarDatosDeConexion()
        Try
            cXML = New XmlTextReader(raiz & "\conexion.xml")
            cXML.WhitespaceHandling = WhitespaceHandling.None
            cXML.Read()
            cXML.Read()
            While cXML.Read()
                Select Case cXML.NodeType
                    Case XmlNodeType.Element
                        cadena = cadena & cXML.Name & "="
                    Case XmlNodeType.Text
                        cadena = cadena & cXML.Value & "; "
                End Select
            End While
            cadena = cadena.Substring(0, cadena.Length - 1)

```

```

        cXML.Close()
    Catch ex As Exception
        MsgBox("Carga de XML de Conexion", MsgBoxStyle.Exclamation, "XML
Conexion")
    End Try

End Sub
Public ReadOnly Property stringConexion()
    Get
        Return cadena
    End Get
End Property

#Region "metodos Publicos"
Public Sub New()
    cargarDatosDeConexion()
End Sub
#End Region
End Class

```

Los utilizados en los PMA que son clases simples basadas en checkbox y labels para la utilización de respectivo chequeo durante la consulta.

Objetos sin interface gráfica para la utilización con controles tales como combobox.

Se desarrollo una clase especial para administración de los permisos y accesos de usuarios al los diferentes módulos llamada AccesoModulos que cuenta con diferentes métodos y funciones, cuya codificación se muestra de la siguiente forma:

```

Imports MySql.Data.MySqlClient

Public Class AccesoModulos
    Private conexion As MySqlConnection
    Private cadenaConexion As New stringDeConexion()
    Private comando As MySqlCommand
    Private comandoNiveles As MySqlCommand
    Private lector As MySqlDataReader
    Private lectorNiveles As MySqlDataReader
    Private maModulos() As String = {"indefinido", "noModulo"}
    Private maIdNiveles() As Integer
    Private maNiveles() As String
    Private maConfAccesoModulos(1, 1) As Byte

```

```

Private miNivel As Integer = 0
Private miModulo As Integer = 0
Private miUsuario As String = "-INDEFINIDO-"
Private miIdUsuario As Integer = 0
Private UsuarioLee As Boolean = 0
Private UsuarioAgrega As Boolean = 0
Private UsuarioModifica As Boolean = 0
Private UsuarioElimina As Boolean = 0
Private grupo As String
Private nombrePersona As String
Private idpersona As Integer
Private seteado As Boolean = False

Private Sub resetear()
    conexion = New MySqlConnection(Me.cadenaConexion.stringConexion)
    comando = New MySqlCommand()
    comando.Connection = conexion
    comando.CommandText = "select count(*) from tblmodulos"
    Try
        conexion.Open()
        Dim n As Integer = 1
        Dim limodulos As Integer = comando.ExecuteScalar()
        ReDim Me.maConfAccesoModulos(1, limodulos)
        Me.maConfAccesoModulos(0, 0) = 0
        For n = 1 To limodulos
            Me.maConfAccesoModulos(0, n) = n
            Me.maConfAccesoModulos(1, n) = 1
        Next
    Catch ex As Exception
        MsgBox(ex.Message)
    End Try
End Sub

#End Region
#Region "propiedades privadas"
Private Property Nivel()
    Get
        Return miNivel
    End Get
    Set(ByVal value)
        Me.miNivel = value
    End Set
End Property

#End Region
#Region "propiedades publicas"
#Region "propiedades de lectura, agregacion, modificacion y eliminacion"
Public ReadOnly Property leeModulo() As Boolean

```

```

    Get
        Return Me.UsuarioLee
    End Get
End Property
Public ReadOnly Property agregaModulo() As Boolean
    Get
        Return Me.UsuarioAgrega
    End Get
End Property
Public ReadOnly Property modifcaModulo() As Boolean
    Get
        Return Me.UsuarioModifica
    End Get
End Property
Public ReadOnly Property eliminamodulo() As Boolean
    Get
        Return Me.UsuarioElimina
    End Get
End Property
#End Region
#Region "propiedades de retorno y seteo de usuario, idusuario, nivel, modulo "
Public ReadOnly Property idUsuario() As Integer
    Get
        Return miIdUsuario
    End Get
End Property
Public ReadOnly Property Persona() As Integer
    Get
        Return idpersona
    End Get
End Property
Public ReadOnly Property usuario() As String
    Get
        Return miUsuario
    End Get
End Property

Public Property Modulo() As Integer
    Get
        Return miModulo
    End Get
    Set(ByVal value As Integer)
        Try
            If value <= UBound(maModulos) And value > 0 Then
                miModulo = value
                conexion = New MySqlConnection(Me.cadenaConexion.stringConexion)
                comando = New MySqlCommand
            End If
        End Try
    End Set
End Property

```

```

        comando.Connection = Me.conexion
        comando.CommandText = "select cmplee, cmpagrega, cmpmodifica,
        cmpelimina from tblaccesos where cmpidacceso=" & Me.maConfAccesoModulos(1,
        Me.miModulo)
        conexion.Open()
        lector = comando.ExecuteReader()
        If lector.Read Then
            Me.UsuarioLee = lector!cmplee
            Me.UsuarioAgrega = lector!cmpagrega
            Me.UsuarioModifica = lector!cmpmodifica
            Me.UsuarioElimina = lector!cmpelimina
        End If
        lector.Close()
        conexion.Close()

    End If
    Catch ex As Exception
        MsgBox(ex.Message)

    End Try

    End Set
End Property
Public ReadOnly Property name() As String
    Get
        Return Me.nombrePersona
    End Get
End Property
Public ReadOnly Property idgrupo() As String
    Get
        Return Me.grupo
    End Get
End Property
Public ReadOnly Property contarModulos()
    Get
        Return UBound(maModulos)
    End Get
End Property
Public ReadOnly Property contarNiveles()
    Get
        Return UBound(maNiveles)
    End Get
End Property
#End Region
#End Region
#Region "metodos privados"
Private Sub llenarMe()

```

```

Me.maConfAccesoModulos(0, 0) = 0
Me.maConfAccesoModulos(0, 1) = 1
Me.maConfAccesoModulos(1, 0) = 1
Me.maConfAccesoModulos(1, 1) = 0
conexion = New MySqlConnection(Me.cadenaConexion.stringConexion)
comando = New MySqlCommand()
comando.Connection = conexion
comando.CommandText = "select count(*) from tblmodulos"
Try
    conexion.Open()
    Dim n As Integer = 0
    Dim limodulos As Integer = comando.ExecuteScalar() ' SE CONSULTA CUANTO
MODULOS HAY..
    If limodulos >= 1 Then    ' SI HAY POR LO MENOS UNO...
        ReDim Me.maConfAccesoModulos(1, limodulos)    'SE REDIMENSIONA LA
MATRIZ..A DOS FILAS Y TANTAS COLUMNAS COMO MODULOS
        Me.maConfAccesoModulos(0, 0) = 0    'SE SETEA LA ESQUINA (0,0) A 0
INDICANDO MODULO CERO -QUE NO EXISTE-
        Me.maConfAccesoModulos(1, 0) = 0    'EL (1,0) QUE ES DONDE ESTA EL
NUMERO DE NIVEL QUE SIEMRPE SE INICIALIZA CON 0
        For n = 1 To limodulos    'SE PONEN TODOS LOS ACCESOS A 1
QUE CORRESPONDE A NO LEER NINGUN MODULO..
            Me.maConfAccesoModulos(0, n) = n
            Me.maConfAccesoModulos(1, n) = 1
        Next
        ReDim Me.maModulos(limodulos)    ' SE REDIMENSIONA EL ARREGLO
DE LOS NOMBRE DE MODULOS
    End If
    Me.maModulos(0) = "indefinido"    'EL MODULO 0 ES INDIFINIDO
SIEMPRE ...
    comando.CommandText = "select cmpnombre from tblmodulos order by
cmpidmodulo" 'SE HACE LA CONSULTA DE TODOS LOS NOMBRES DE
MODULOS...
    lector = comando.ExecuteReader()
    n = 0
    While lector.Read    'SE LLENAN TODOS LOS NOMBRES DE
MODULOS...
        n += 1
        Me.maModulos(n) = lector!cmpnombre
    End While
    lector.Close()
    conexion.Close()
    conexion.Open()
    comando.CommandText = "select count(*) from tblniveles" ' SE CONSULTA
CUANTOS NIVELES HAY...
    n = comando.ExecuteScalar()
    If (n >= 1) Then

```

```

        ReDim Me.maIdNiveles(n)
        ReDim Me.maNiveles(n)
        comando.CommandText = "select cmpidnivel, cmpdescripcion from tblniveles
order by cmpidnivel"
        lector = comando.ExecuteReader
        Me.maNiveles(0) = "indefinido"
        Me.maIdNiveles(0) = 0
        n = 1
        While lector.Read
            Me.maIdNiveles(n) = lector!cmpidnivel
            Me.maNiveles(n) = lector!cmpdescripcion
            n += 1
        End While
        lector.Close()
        conexion.Close()
    End If

    Catch ex As MySqlException
        MsgBox(ex.Message)
    End Try
End Sub
Private Sub configurarMiNivel()
    conexion = New MySqlConnection(Me.cadenaConexion.stringConexion)
    comando = New MySqlCommand()
    comando.Connection = conexion
    comando.CommandText = "select count(*) from tblmodulos"
    Try
        conexion.Open()
        Dim n As Integer = 1
        Dim limodulos As Integer = comando.ExecuteScalar()
        ReDim Me.maConfAccesoModulos(1, limodulos)
        Me.maConfAccesoModulos(0, 0) = 0
        For n = 1 To limodulos
            Me.maConfAccesoModulos(0, n) = n
            Me.maConfAccesoModulos(1, n) = 1
        Next
        Me.maConfAccesoModulos(1, 0) = Me.Nivel
        comando.CommandText = "select cmpidmodulo, cmpidacceso from
tblnivelesyaccesos where cmpidnivel=" & Me.Nivel & " order by cmpidmodulo"
        lector = comando.ExecuteReader
        n = 1
        While lector.Read
            Me.maConfAccesoModulos(1, n) = lector!cmpidacceso
            n += 1
        End While
        Me.Modulo = Me.Modulo
        lector.Close()
    
```

```

        conexion.Close()
        conexion.Open()
        Me.comando.CommandText = "select
cmpidpersona,cmpnombrespersona,cmpidgrupo from tblpersonas where
cmpusuarioasignado = " + Me.miIdUsuario.ToString
        lector = Me.comando.ExecuteReader
        If lector.Read Then
            Me.grupo = lector!cmpidgrupo
            Me.nombrePersona = lector!cmpnombrespersona
            Me.idpersona = lector!cmpidpersona
        End If
        lector.Close()
        conexion.Close()
    Catch ex As Exception
        MsgBox(ex.Message() + "configurar mi nivel")
    End Try
End Sub
#End Region
#Region "metodos publicos"
Public Sub refrescate()
    Try
        Me.conexion = New MySqlConnection(Me.cadenaConexion.stringConexion)
        Me.comando = New MySqlCommand("select cmpidnivel from tblusuarios where
cmpidusuario=" & Me.idUsuario, Me.conexion)
        Me.conexion.Open()
        Dim lector As MySqlDataReader = Me.comando.ExecuteReader
        If lector.Read Then
            Me.Nivel = lector!cmpidnivel
            Me.configurarMiNivel()
        End If
    Catch ex As MySqlException
        MsgBox("error en refrescar")
    End Try
End Sub
#End Region
#Region "Funciones Publicas"
Public Overloads Function configureConUsuario(ByVal nombreUsuario As String,
ByVal claveUsuario As String) As Boolean
    comando.CommandText = "select cmpidusuario, cmpidnivel from tblusuarios where
cmplogin=" & nombreUsuario & " and cmpclave=" & claveUsuario & " and
cmpactivo=1"
    comando.Connection = conexion
    Try
        conexion.Open()
        lector = comando.ExecuteReader
        If lector.Read Then
            Me.Nivel = lector!cmpidnivel

```

```

    Me.miUsuario = nombreUsuario
    Me.miIdUsuario = lector!cmpidusuario
    conexion.Close()
    lector.Close()
    Me.configurarMiNivel()
    Return True
Else
    Me.miUsuario = "-INDEFINIDO-"
    Me.miIdUsuario = 0
    conexion.Close()
    lector.Close()
    Return False
End If
Catch ex As MySqlException
    Return False
End Try

End Function
Public Overloads Function configureConUsuario(ByVal id As Integer) As Boolean
    comando.CommandText = "SELECT cmpidnivel, cmplogin from tblusuarios where
cmpidusuario=" + id.ToString + " and cmpactivo=1"
    comando.Connection = conexion
    Try
        conexion.Open()
        lector = comando.ExecuteReader
        If lector.Read Then
            Me.Nivel = lector!cmpidnivel
            Me.miUsuario = lector!cmplogin
            Me.miIdUsuario = id
            conexion.Close()
            lector.Close()
            Me.configurarMiNivel()
            Return True
        Else
            Me.miUsuario = "nouser"
            Me.miIdUsuario = 0
            conexion.Close()
            lector.Close()
            Return False
        End If
    Catch ex As MySqlException
        MsgBox("select cmpidnivel, cmplogin from tblusuarios where cmpidusuario=" +
id.ToString + " and cmpactivo=1" + ex.ToString)
        Return False
    End Try

End Function

```

```

Public Overloads Function moduloToString(ByVal numModulo As Integer) As String
    Dim nombreModulo = Me.maModulos(0)
    Dim i As Integer
    For i = 1 To UBound(maModulos)
        If (i = numModulo) Then
            nombreModulo = maModulos(i)
            Exit For
        End If
    Next
    Return nombreModulo
End Function
Public Overloads Function moduloToString() As String
    Dim nombreModulo = Me.maModulos(Me.Modulo)
    Return nombreModulo
End Function
Public Overloads Function indiceDeModulo(ByVal modulo As String) As Integer
    Dim index As Integer
    For index = 1 To UBound(Me.maModulos)
        If (maModulos(index).ToUpper() = modulo.ToUpper()) Then
            Exit For
        End If
    Next
    If index > UBound(Me.maModulos) Then
        Return -1
    Else
        Return index
    End If
End Function
Public Overloads Function indiceDeModulo() As Integer
    Dim index As Integer
    For index = 1 To UBound(Me.maModulos)
        If (maModulos(index).ToUpper() = maModulos(Me.Modulo).ToUpper()) Then
            Exit For
        End If
    Next
    If index > UBound(Me.maModulos) Then
        Return -1
    Else
        Return index
    End If
End Function
Public Function descripcionDeNivel(ByVal nivel As Integer) As String
    Dim descripcion As String = Me.maNiveles(0)
    Dim i As Integer
    For i = 1 To UBound(Me.maNiveles)
        If (nivel = Me.maIdNiveles(i)) Then

```

```

        descripcion = Me.maNiveles(i)
    Exit For
End If
Next
Return descripcion
End Function
Public Overloads Function AccesoDeModulo(ByVal modulo As Integer) As Integer
    Dim acceso As Integer = -1
    If modulo <= UBound(Me.maModulos) Then
        acceso = Me.maConfAccesoModulos(1, modulo)
    End If
    Return acceso

End Function
Public Overloads Function AccesoDeModulo() As Integer
    Dim acceso As Integer = Me.maConfAccesoModulos(1, Modulo)
    Return acceso
End Function
#End Region
#Region "constructor de la clase"
Public Sub New()
    Me.llenarMe()
End Sub
#End Region
End Class

```

Dichas propiedades estan destinadas a proporcionar información sobre los privilegios y accesos del usuario logueado a los diferentes módulos del sistema.

Dicho objeto, en conjunto con otra clase programada basada en formulario llamada summ se utilizan para configurar según los permisos de los usuarios, los controles de agregación, modificación y visualización en cada módulo.

Asimismo, se crearon clases especiales y sencillas como la clase stringDeConexion cuya unica finalidad es la lectura de un XML de conexión al servidor de base datos MySQL.

En la ejecución de diversos procesos se utiliza la lectura de pequeños archivos XML de datos concernientes a la estructura de diversos esquemas como el plan de calificaciones de los estudiantes, algunos datos que se consideran “constantes” en el sistema, como nombres de enfermedades, tratamientos, etc.

Para el desarrollo del componente de navegación se incurrió a la programación con Actionscript 2.0 y un ocx para la comunicación con funciones vb.net de tal manera que sea posible que el componente de navegación se comunique directamente con el programa para la autenticación de usuarios al entrar en cada módulo.

## **4.2 INTEGRACIÓN DE LOS ELEMENTOS Y DEPURACIÓN**

Luego de construir todos los elementos de programa a nivel de código y haber diseñado las interfaces o pantallas a utilizar, se procedió a integrar los componentes y a darle funcionalidad a los subsistemas del sistema en general, depurando los módulos para la detección de inconsistencias o señalamiento de mejoras.

Cabe mencionar que esta aplicación tiene elementos que la hacen única debido a la utilización de tecnologías totalmente distintas que fueron integradas para proporcionar un entorno mas amigable al usuario, estas tecnologías son la plataforma .Net de Microsoft y la plataforma Flash de Adobe.

Con esto se da un paso adelante en la innovación debido a que normalmente ningún programador intenta integrarlas, sin embargo se ha hecho posible gracias a unas librerías de Flash que permiten a las dos plataformas hablar un mismo lenguaje y con eso interactuar como si el código de cada una fuera de la misma plataforma.

Con cada depuración realizada se establecieron las correcciones a realizar y las pruebas requeridas para asegurar el correcto funcionamiento de cada formulario. Existen diversas librerías o recursos que han sido incorporados al proyecto ya sea como referencia o como carpetas de recursos, con eso se pretende facilitar la estructura de la aplicación, como ejemplo se menciona la librería .ocx de flash que permite la interacción entre ambas plataformas y fue incluida dentro de una carpeta de recursos.

### **4.3 ELABORACIÓN DEL PLAN DE IMPLEMENTACIÓN Y DOCUMENTACIÓN**

En esta etapa, se ha desarrollado el plan de implementación del sistema de la clínica de odontología de la Facultad, desarrollando las mejores propuestas para su aplicación, además se ha redactado la documentación necesaria para el sistema, tal como: el manual de usuario.

En este punto se delimitaron los siguientes aspectos:

- La implementación real que tiene el sistema en la clínica ha tenido que realizarse en el equipo con el que actualmente se cuenta en la entidad, con el fin de que el sistema sea utilizado con datos reales, valiéndose de recursos necesarios para optimizar el hardware disponible para que trabaje el sistema.
- Ha sido documentada la implementación real con el equipo de la clínica y los recursos utilizados dentro de las instalaciones.
- Se ha elaborado un plan de implementación ideal o recomendado dentro de la clínica, para que en al adquirir los recursos de hardware e infraestructura adecuada, se pueda implementar el sistema en su manera más idónea. En este plan se mencionan los recursos adecuados tanto de hardware como de software, así como las configuraciones adecuadas en cada equipo.
- Se ha desarrollado una pequeña capacitación para los representantes de la clínica con el fin de exponerles y capacitarlos en la utilización del sistema.

### **4.4 PLAN DE IMPLEMENTACION**

Dentro del presente proyecto se ha decidido ir un paso mas allá de simplemente hacer una investigación, crear un marco teórico y hacer el diseño de un programa más y realizar su presentación, el grupo de trabajo quiere que este proyecto tenga un impacto y un beneficio real para la clínica, de lo contrario no se puede encontrar el valor real de un trabajo de grado.

Es por eso que como uno de los puntos se ha incluido la implementación del sistema dentro de las instalaciones de la clínica odontológica, para que todo lo documentado como beneficio de tener este sistema sea aprovechado realmente para facilitar el trabajo dentro de esta entidad.

A continuación se presentan los requerimientos y pasos propuestos a seguir para la implementación de este programa en la clínica odontológica.

### **IMPLEMENTACIÓN CON EQUIPO ACTUAL**

Debido a que la clínica es una entidad dentro de la universidad que depende totalmente de los recursos que la misma universidad pueda asignarle, tanto en mobiliario, equipo clínico, medicamentos y equipo de cómputo. Es necesario que la implementación y funcionamiento de este programa sea posible con los recursos actuales y disponibles.

En ese sentido el equipo de trabajo se ha dado a la tarea de realizar las tareas necesarias para configurar y habilitar la utilización del sistema. Los recursos con los que se cuenta actualmente son:

- 1 Computadora Dell con las siguientes características
  - Procesador de 1.6 Ghz
  - Disco duro de 40 GB
  - Memoria de 256 MB
  - Tarjeta de Red Fast Ethernet 10/100
- 1 Impresor HP Deskjet 3920

Con estos recursos como se puede notar es imposible realizar el trabajo en red que se ha propuesto, sin embargo es totalmente factible la instalación y utilización del programa haciendo uso de la misma computadora, la cual realizará el rol de servidor de la base de datos y de computadora cliente con acceso a la interfaz del programa.

Desde esta perspectiva de uso es claramente visible que no se estaría aprovechando todo el potencial del programa, pero con todo y estas limitantes siempre es posible tener el beneficio de la automatización de las tareas cotidianas.

Dentro de los pasos realizados para la implementación del programa se pueden mencionar:

1. Migración de datos
2. Instalación y configuración del programa
3. Capacitación del personal

#### **4.4.1 MIGRACIÓN DE DATOS**

La migración de datos siempre tiene un papel importante dentro de todo proceso de automatización en cualquier institución, no siempre es posible realizarla debido a los formatos existentes o a la carencia de éstos durante el período anterior a la implementación. Esto se refiere a que en el caso de que toda la información haya sido registrada en hojas de papel y archivada de esa forma, impide que exista un medio digital desde el cual intentar realizar la migración.

Por otro lado muchas veces la información sí ha sido almacenada en algún medio digital sin embargo el formato en que se ha hecho es tan complicado de ajustar a un patrón que permita su migración, que finalmente se hace mas tedioso y tardado que si se digita manualmente dentro del programa.

En este caso la migración de datos se realizará de forma parcial, debido a que existe una cantidad de información que si se encuentra registrada en hojas de trabajo de Microsoft Excel y otra información que se encuentra ya sea en Microsoft Word o simplemente escrita en los archivos de papel que posee la clínica.

El proceso para migrar estos datos hacia la base de datos es la siguiente:

1. Formatear los archivos de tal manera que tengan el formato de columnas idéntico a como se encuentran en las tablas de la base de datos
2. Una vez formateado el archivo se guarda con una extensión “csv”,
3. Se carga el archivo .csv dentro de la consola de mysql con el comando load file

Esta tarea se ha realizado con las tablas de pacientes y de materiales las cuales representan un avance para el personal de la clínica al no tener que digitar manualmente toda esta información que representa aproximadamente 500 registros entre expedientes y materiales.

#### **4.4.2 INSTALACIÓN Y CONFIGURACIÓN DEL PROGRAMA**

Dentro de esta etapa de la implementación se consideran dos áreas importantes que son las siguientes:

##### **INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR**

Esta parte se enfoca en la instalación y configuración del servidor de base de datos de la aplicación, es en esta computadora donde se almacenará toda la información generada por cualquier número de computadoras cliente que ejecuten el programa de la clínica.

En este caso se supone la instalación en una computadora con sistema operativo Windows, para ello debemos tener el instalador de MySQL 5.0 para plataforma Windows.

Se procede a instalar siguiendo los pasos del asistente para instalación, se coloca la contraseña de acceso y las únicas configuraciones que se deben hacer al instalarse es que MySQL quede funcionando como un servicio del sistema, y que arranque al iniciar Windows.

Como siguiente paso se instalan las herramientas de administración de mysql con las cuales se hace más sencilla la administración de usuarios, bases de datos y respaldos de las mismas.

Luego de esto se procede a crear una base de datos nueva llamada “clínica”, al tener esta base de datos ya creada se procede a restaurar una base de datos desde el administrador de MySQL, haciendo uso de la opción de restaurar se elige el archivo base que ya se tiene y se inicia la restauración.

##### **INSTALACIÓN Y CONFIGURACIÓN DE COMPUTADORAS CLIENTES**

Para esta parte se pretende instalar el software necesario y configurar las computadoras para que sea posible ejecutar el programa creado.

Se inicia con la instalación del .Net Framework 2.0 que es la que se necesita para ejecutar aplicaciones creadas en Visual Studio 2005.

Luego se instala lo que es el conector de MySQL con .Net, este programa es el que permite la comunicación entre una aplicación .Net con la base de datos MySQL.

Se continúa con la instalación de los archivos ejecutables y carpetas que utiliza el sistema. Dentro de las carpetas del sistema se procede a configurar el archivo XML en el que se coloca la dirección de red de la computadora en la que se encuentra trabajando el servidor MySQL con la base de datos.

Finalmente se realiza la primera prueba ejecutando el programa y si no aparece ningún mensaje de no conexión con la base de datos la instalación ha sido exitosa.

#### **4.4.3 CAPACITACIÓN AL PERSONAL DE LA CLÍNICA**

Dentro de la fase de implementación el grupo de trabajo de este proyecto brindará una pequeña capacitación al personal permanente de la clínica, esto con el objetivo que el personal aprenda a utilizar completamente la nueva herramienta que estará a su disposición. Sin una capacitación adecuada el uso que se le pueda dar al programa puede no ser tan beneficioso y no ayudar a facilitarles el trabajo cotidiano dentro de la clínica, por eso se utilizará un tiempo aproximado de 4 horas en las cuales se cubrirán los siguientes temas:

- Administración de la base de datos desde las herramientas externas, esto con el propósito de que el personal pueda restaurar la base de datos en caso de necesitarse.
- Utilización del programa en todos sus módulos, con lo cual se tendrá el conocimiento sobre como realizar la manipulación de registros de cualquier módulo que sea necesario, Ej. Se mostrará el proceso para crear un nuevo expediente, una consulta, una cita, etc.
- Resolución de dificultades a través de la ayuda disponible dentro del programa, con lo cual se dará al usuario la posibilidad de buscar dentro de la ayuda la forma correcta de utilizar cualquier módulo, esto por si se olvidara lo aprendido en la capacitación.

Se desea realizar una capacitación que abarque las 4 horas en un solo día con lo cual no se interrumpirá la asimilación de los conocimientos y se evita tener que volver a recapitular si en caso fuera distribuido en varios días.



**5 RESULTADOS ESPERADOS, CONCLUSIONES Y RECOMENDACIONES**

**CAPÍTULO V**

## **5.1 RESULTADOS ESPERADOS**

Que el personal de la clínica cuente ahora con un sistema completo de administración de la información, para facilitarles las labores cotidianas de la clínica. Esto pensando en la eficiencia en labores como búsqueda de expedientes, generación de expedientes, programación de consultas, etc.

Que los diferentes módulos del sistema cubran todas las necesidades existentes en el manejo de la información de los pacientes y de todas las actividades realizadas tanto por los estudiantes, como por los doctores. Con esto se ha querido dar soporte a todos los ámbitos de la clínica, desde las consultas a pacientes, las actividades de campo, hasta las calificaciones de los estudiantes.

Que mediante la utilización de los reportes del sistema se obtenga toda la información requerida por cualquier persona involucrada en el manejo de la clínica. Con esto se beneficia al sector administrativo a la hora de presentar los reportes que van desde un simple expediente hasta resúmenes consolidados de las operaciones de la clínica.

El sistema permitirá el trabajo en red, con accesos múltiples a los datos y con un manejo eficiente de las operaciones con éstos. Dada su estructura se podrá aumentar de manera fácil el número de personas que pueden utilizar el sistema sin interferir uno con otro.

Este sistema se ha diseñado para ser utilizado sin ninguna dificultad por cualquier estudiante con conocimientos mínimos del uso de un computador; siendo intuitivo y fácil de dar mantenimiento.

Haber desarrollado un sistema con la capacidad de ser escalable y adaptable a futuras necesidades dentro del manejo de los datos o generación de reportes. La escalabilidad proporcionada esta dada tanto en crecimiento de la base de datos como en el número de usuarios en red que pueden acceder simultáneamente al sistema.

Que la documentación de este proyecto sirva de material bibliográfico a los estudiantes que busquen integrar tecnologías de bases de datos Libres con Visual Basic .Net

## **5.2 CONCLUSIONES**

Mediante la elaboración de este proyecto informático se ha hecho más evidente la importancia que tiene en todos los aspectos de la vida cotidiana el uso de un sistema automatizado, que permita agilizar procesos así como acceder a datos de personas y sucesos, en este caso, de pacientes, consultas, practicantes, etc.

Contar con el apoyo de un sistema de información permite dedicar mayores esfuerzos en mejorar diversas áreas del campo práctico de las labores, pues el sistema de información minimiza los tiempos perdidos en la búsqueda de información, presentación y organización de datos críticos.

El sistema desarrollado permitirá a los Administradores y docentes encargados de la evaluación de los estudiantes brindar apreciaciones con mayor grado de objetividad en relación al grado que brindan los procesos actuales.

Que el sistema gestor de base de datos MySQL 5, reúne todas las características importantes de un sistema de bases de datos para misión crítica, y que a pesar de no ser muy utilizado en el entorno empresarial por el momento, tiene el potencial de ganar una cuota importante de mercado frente a los principales competidores en este campo.

## **5.3 RECOMENDACIONES**

Que los directores de las clínicas odontológicas realicen esfuerzos en cuanto a la búsqueda de financiamiento para equipar con computadoras los módulos de atención de pacientes, y de esa manera aprovechar el potencial de agilización que se tendría usando el nuevo sistema en cada computadora.

Que se capacite a los estudiantes que vayan llegando a sus prácticas para que pueda utilizar el sistema y no provocar la situación en que la secretaria o la Dra. Directora tenga que introducir la información al sistema.

Buscar con las autoridades de la Facultad Multidisciplinaria de Occidente la posibilidad de que la base de datos del sistema resida en los servidores de la facultad, con lo cual se lograría tener un mayor nivel de seguridad en la base de datos, y se liberaría también recursos en la computadora principal de la clínica.

## **5.4 ANEXOS**

VER ARCHIVO MANUAL DE USUARIO