



UPAEP

UNIVERSIDAD POPULAR AUTÓNOMA DEL ESTADO DE PUEBLA

LICENCIATURA EN SISTEMAS COMPUTACIONALES

TRABAJO RECEPTACIONAL

***"Automatización Administrativa, Contable y Clínica
DE LA CLÍNICA DE ODONTOLOGÍA U.P.A.E.P."***

**QUE PARA OBTENER EL TÍTULO DE
LICENCIADO EN SISTEMAS COMPUTACIONALES**

PRESENTAN:

**Oscar Ariel Merino Gómez
Alejandro León Hernández**

ASESORES:

**Lic. Rocío Morales Salgado
Dra. Ma. Trinidad Elvira Rayón**

H. Puebla de Zaragoza 1994



UPAEP – Secretaría General

Dirección General de Apoyos Académicos

Dirección del Centro de Recursos para el Aprendizaje y la Investigación.

Biblioteca Central - **Karol Wojtyla**

Tesis Digitales Restricciones de uso:

DERECHOS RESERVADOS ©

PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

El uso de textos, imágenes, gráficas, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente de donde la obtuvo mencionando el autor o autores involucrados en el documento.

Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

UPAEP
BIBLIOTECA CENTRAL
TESIS
USO ÚNICAMENTE EN SALA



A DIOS

Por darme la vida y guiar mi transcurrir por ella, por acompañarme en los momentos más difíciles y brindarme la oportunidad de enmendar mis errores y saldar todas mis deudas, por la alegría, la salud, la enfermedad, por el trabajo y el cansancio. Por hacerme nacer de una mujer maravillosa, tierna, firme y valiente. Por darme una familia llena de gentes valiosas, quienes nunca escatimaron su cariño, apoyo y comprensión. Por regalarme la felicidad y compañía de mis amigos. Por poner en mi camino a la mujer de mi vida y dejarme conocerla y amarla, y por ayudarme a corregir el camino.

A MI MADRE

Quien supo ser tierna cuando hubo lágrimas en mis ojos, firme cuando los azares de la vida me hicieron equivocar el rumbo, y valiente cuando más necesité de su apoyo y cariño.

Quien siempre antepuso mi bienestar al suyo y me ha regalado los mejores años de su vida, demostrándome que el amor verdadero no es el afecto, sino el dar la vida por aquel a quien se ama.

(Mamá gracias por no darme todo, me has hecho fuerte, constante y maduro)

A MI FAMILIA

PAPA

Gracias por ser mi amigo y compañero, por compartir mis proyectos y enseñarme a ser emprendedor. Por heredarme el amor por el arte, la facilidad de palabra y el gusto por las costumbres y lugares de mi patria.

MIS HERMANOS

Miguel y Max, ustedes a quien tanto quiero y agradezco su cariño y apoyo incondicional. Les deseo el mismo éxito o más que el que este trabajo representa para mí y que sean dichosos.

MIS TÍOS, PRIMOS Y DEMÁS FAMILIARES

No incluyo nombres para no omitir a nadie, pero ustedes saben que el apoyo que me dieron fue importante para lograr todo esto.

A MIS AMIGOS

Especialmente a:

Alejandro León, Abraham Lorencéz, Alfredo Planell, David Hernández, Verónica Jiménez, Lucia Regil.

Gracias por darme su tiempo y dejarme ser parte de su vida, por acompañarme durante toda mi carrera.

A MI NOVIA

Quien ha sabido hacerme feliz y ha estado conmigo en los momentos de triunfo y los de fracaso. Por quien siento tanto amor y sé que me ama. Gracias por apoyarme para ser quien soy.

A MIS PROFESORES

Y todas aquellas personas que contribuyeron con sus conocimientos y sabios consejos a mi formación profesional. Especialmente al Ing. José Bedolla, por apoyarme durante toda mi formación universitaria y darme ejemplo de disciplina en todos los aspectos.

A MIS PADRES:

EFREN LEON RAMOS

SOLEDAD HERNANDEZ DE LEON

A mi padre y a mi madre, en especial a mi madre, le dedico este trabajo ya que sin la ayuda y el apoyo que siempre me brindó, esta meta jamás se habría logrado, porque gracias a su sacrificio un sueño se hace realidad, esto es algo que nunca podré olvidar, es por eso que estaré por siempre agradecido, A TI MADRE mi más humilde y sincero agradecimiento. Gracias.

A MIS HERMANAS:

ELIZABETH LEON HERNANDEZ

JACQUELINE LEON HERNANDEZ

ADRIANA LEON HERNANDEZ

A mis hermanas:Elizabeth, Jacqueline, Adriana. También les estoy muy agradecido por todo el apoyo que siempre me han brindado ya que sin éste, un sueño podría no haberse hecho realidad. Todo lo que han hecho por mí pueden estar seguras que nunca lo olvidaré. Gracias.

A MIS FAMILIARES:

A todos mis familiares, no escribo ningún nombre por temor a omitir alguno, pero ustedes saben perfectamente quienes son, a todos ustedes que de alguna manera me ayudaron y me apoyaron durante todo este tiempo, también les ofresco mis más humildes y sinceros agradecimientos. Gracias

A MIS PROFESORES:

Para todos los profesores, en especial a mi director José Bedolla Cordero y aquellos que ayudaron en mi formación, les doy las gracias por permitirme ser su alumno y aprender algo de ustedes, gracias por todo el apoyo que brindaron durante mi permanencia en la escuela. Gracias.

A MIS AMIGOS:

A todos mis amigos les doy las gracias por el apoyo y la amistad que no merecía, a todos ustedes les ofresco mi humilde amistad y lealtad, sinceramente quiero ofrecerles mis agradecimientos, en especial a:

Arturo Sánchez Barragán, Hugo Teissier, Gabriela Rodríguez Castelán, Oscar A. Merino Gómez, Abhaham Lorencéz Villa, Alfredo Romero Planell, Oscar David Hernández, Pilar Weaver Velasco, Iris Cervantes Acosta, Lucía T. Regil González, Bárbara González Jiménez, y a todos aquellos a los que no pude recordar en estos momentos. Gracias.

INDICE

	TEMAS	PAGINA
A	INTRODUCCION	
A.1	- Planteamiento del Proyecto de Tesis	1
A.1.1	- Justificación	1
A.1.2	- Objetivo General	2
A.1.3	- Objetivos Específicos	2
A.1.4	- Alcances	3
A.1.5	- Planteamiento del Problema	3
A.1.6	- Desarrollo del Problema	3
A.1.7	- Pasos a seguir en la Solución del Problema	4
A.1.8	- Planteamiento de Hipótesis	5
A.1.9	- Comprobación de Hipótesis	5
A.2	- Breve Descripción del Trabajo	6
	CAPITULO 1:	
1.	- LA UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE PUEBLA	7
1.1	- Breve Historia	7
1.2	- Ideario y Misión de la U.P.A.E.P	8
1.3	- Cultura	11
1.4	- Educación	11
	CAPITULO 2:	
2.	- LA CLINICA DE ODONTOLOGIA U.P.A.E.P.	
2.1	- Antecedentes	13
2.2	- Estructura y Administración	14
2.2.1	- Areas que la Comprenden	14
2.2.1.1	- Recepción	14
2.2.1.2	- Central de Equipos	15
2.2.1.3	- Clínicas	15
2.2.1.4	- Laboratorio	16
2.2.1.4.1	- Anatomía Dental	16
2.2.1.4.2	- Ortodoncia	16
2.2.1.4.3	- Oclusión	16
2.2.1.4.4	- Operatoria Dental	16
2.2.1.4.5	- Prostodoncia	16

	TEMAS	PAGINA
2.2.1.4.6	- Prótesis	16
2.3	- Sistema de Información Manual	17
2.3.1	- Secuencia de Actividades	17
2.4	- Modelo de "Asistencia-Docencia"	18

CAPITULO 3:

3. - ADMINISTRACION Y COMPUTACION

3.1	- Administración e Información	19
3.1.1	- Exposición General de Administración	19
3.1.2	- Planeación	20
3.1.3	- Control	20
3.1.3.1	- Elementos de Retroalimentación y Control	21
3.1.3.2	- Tipos de Control	22
3.1.3.2.1	- Sistema de Circuito Abierto	22
3.1.3.2.2	- Sistema de Circuito Cerrado	23
3.1.4	- La Toma de Decisiones	23
3.1.5	- Elementos de la Toma de Decisiones	23
3.1.5.1	- Modelo	23
3.1.5.2	- Criterios	23
3.1.5.3	- Restricciones	24
3.1.5.4	- Optimización	24
3.1.5.5	- Clases de Decisión	24
3.1.5.5.1	- Decisión Programada	24
3.1.5.5.2	- Decisión no Programada	24
3.1.5.6	- Proceso de la Toma de Decisiones	25
3.1.5.6.1	- Información Inadecuada	26
3.1.5.6.2	- Objetivos Incorrectamente Especificados	26
3.1.5.7	- Niveles de la Toma de Decisiones	26
3.1.5.7.1	- Nivel Estratégico	26
3.1.5.7.2	- Nivel Táctico	26
3.1.5.7.3	- Nivel Técnico	26
3.1.5.8	- Información Necesaria para la Toma de Decisiones	26
3.1.6	- El Método Computacional	28
3.1.6.1	- Exposición General de los Sistemas	28
3.1.6.2	- Clasificación de los Sistemas	28
3.1.6.2.1	- Nivel de Predictibilidad	28
3.1.6.2.2	- Nivel de Complejidad	28
3.1.6.2.3	- Sistema Determinista Sencillo	29
3.1.6.2.4	- Sistema Determinista Complejo	29
3.1.6.3.	- Análisis de Causa y Efecto	29
3.1.6.4.	- Análisis Determinista y Probabilista	30

TEMAS

PAGINA

CAPITULO 4: PROBLEMATICA

4.

4.1	- Análisis de la Problemática	31
4.1.1	- Requerimientos de Información	31
4.1.2	- Entidades Relacionadas	31
4.1.2.1	- Rectoría	32
4.1.2.2	- Contraloría y Tesorería	32
4.1.2.3	- Escuela de Odontología	32
4.1.2.4	- Departamento Escolar	33
4.1.2.5	- Entrevistas con Usuarios	33
4.1.2.5.1	- Dirección de la Clínica	33
4.1.2.5.2	- Personal Administrativo	35
4.1.2.5.3	- Catedráticos	35
4.1.2.5.4	- Alumnos	36

CAPITULO 5:

5.

- MODELO PROPUESTO DE SOLUCION

5.1	- Areas Contable	37
5.2	- Area Administrativa	37
5.3	- Area Clínica	38
5.4	- Area Académica	38
5.4.1	- La Formación	38
5.4.2	- La Evaluación	38
5.5	- Funcionamiento en Red	38
5.6	- Planteamiento Teórico de Redes	38
5.6.1	- Para qué Sirven las Redes	38
5.6.2	- Qué es una Red de Computadoras	39
5.6.3	- Ventajas de las Redes	39
5.6.4	- Topologías de Red	39
5.6.5	- Topologías más Comunes	40
5.6.5.1	- Topología Jerárquica	40
5.6.5.2	- Topología Horizontal	40
5.6.5.3	- Topología de Estrella	41
5.6.5.4	- Topología de Anillo	41
5.6.5.5	- Topología de Malla	42
5.7	- Componentes de una Red Local	43
5.8	- Estaciones de Trabajo	43

	TEMAS	PAGINAS
5.9	- Servidor	43
5.10	- Sistema de Cableado	43
5.11	- Tarjetas de Red	44
5.12	- Sistema Operativo de Red	45
5.13	- Sistemas Operativos	46
5.14	- Software de las Lan's	46
5.15	- La Mejor Red Local	46

CAPITULO 6:

6. - PROGRAMACION ORIENTADA A OBJETOS

6.1	- Introducción	47
6.1.1	- Enfoque por Procedimientos	49
6.1.2.	- Enfoque por Objetos	50
6.2	- Análisis y Diseño Orientado a Objetos	50
6.3	- Metodologías para la Ingeniería Orientada a Objetos	52
6.4	- Representando el Mundo con Objetos	52
6.4.1	- Abstracción	54
6.4.2	- Encapsulación	55
6.4.3	- Herencia	56
6.4.4	- Poliformismo	57

CAPITULO 7:

7. - SISTEMA COMPUTACIONAL

7.1	- Paradox	59
7.2	- ObjetPAL	59
7.2.1	- Que es ObjetPAL	59
7.2.2	- Objetos en ObjetPAL	59
7.2.3	- Programación en ObjetPAL	62
7.3	- Documentación del Sistema Computacional	63
7.3.1	- Diagrama de Navegación	64
7.3.2	- DFD's	67
7.3.3	- TABLAS (BASES DE DATOS)	79

	TEMAS	PAGINAS
8.	MANUAL DEL USUARIO	94
9.	CONCLUSIONES	134
10.	APENDICE	
10.1.1	- Indice de métodos	135
11.	BIBLIOGRAFIA	

A.1 INTRODUCCION

A.1.1 JUSTIFICACION

En la educación a nivel superior es necesario brindar a los educandos elementos que apoyen lo aprendido en las aulas con prácticas de campo.

En el área de la salud éstas prácticas sólo pueden apoyarse mediante la creación de instituciones con modelos de funcionamiento "Atención-Docencia", ésto es, instituciones donde los alumnos aplican sus conocimientos en la atención de problemas reales de la población, mientras son supervisados por personal capacitado y con experiencia.

Estas instituciones generan volúmenes de información enormes, equivalentes a la unión de una facultad y una institución de salud, por lo tanto la administración de las mismas requiere el proceso de información relativa a pacientes, alumnos, recursos, ingresos económicos, horarios, etc.

El procesamiento de la información consume recursos excesivos que obstaculizan o limitan la labor docente que debiera ser la actividad principal y a la que debieran destinarse la mayor parte de los recursos.

En la U.P.A.E.P. se ha creado una institución de éste tipo para apoyar la labor formativa de la Escuela de Odontología, tal es el caso de la "Clínica de Odontología".

En un principio los datos generados por la atención al público significaban un volumen pequeño, debido al reducido número de pacientes atendidos y tratamientos brindados; sin embargo, la clínica, con el objetivo de mejorar su eficiencia ha ampliado el número de servicios prestados, al mismo tiempo, la aceptación lograda entre la población ha hecho que se incremente el número de pacientes.

Actualmente la información generada en la clínica constituye un volumen considerablemente amplio, puesto que se atienden alrededor de 80 pacientes diarios, de los cuales aproximadamente el 60% son pacientes de un sólo tratamiento, pero deben ser registrados en un expediente clínico, el 40% restante son pacientes con un plan de tratamiento extenso que involucra citas sucesivas y que los mismos alumnos deben ir programando para la mejor atención de los pacientes, todo ésto bajo la supervisión de los doctores que están a cargo de las clínicas, por lo que debe almacenarse un expediente clínico y un plan de tratamiento.

Todos los tratamientos efectuados en la clínica generan ingresos y egresos, es decir un cargo en efectivo por la atención y el consiguiente gasto en materiales y equipo, el cual debe registrarse con exactitud.

Junto con el problema del volumen de datos generados por la clínica, se encuentra el de procesarlos para producir información suficiente y oportuna para diferentes entidades como son: Dirección de la clínica, Dirección de la escuela, Contraloría, Consejo Universitario, Rectoría, etc.

La solución a estos problemas requiere la implementación de procedimientos y sistemas que mejoren tanto el control interno como el proceso de la información, en ambos casos un auxiliar eficiente a nivel tecnológico es el uso de computadoras.

Por todo esto se hace necesario crear un sistema de control y proceso de información auxiliado por computadora, dicho sistema será creado en base al estudio del funcionamiento de la Clínica de odontología de la U.P.A.E.P. y cubrirá sus necesidades administrativas y contables, adicionalmente, apoyará la labor docente por medio de un sistema de consulta para alumnos de nominado "Revista Digital".

A.1.2. OBJETIVO GENERAL

Que en la clínica de odontología U.P.A.E.P. se controle eficientemente la información, de modo que sea suficiente y oportuna para tomar decisiones con más certidumbre y en menos tiempo, que se fortalezcan otras áreas de la administración y se brinde un mejor servicio tanto a los pacientes como a los alumnos, además de que se controlen los ingresos y egresos de efectivo, así como los recursos materiales e insumos.

A.1.3 OBJETIVOS ESPECIFICOS

- Determinar puntos neurálgicos, problemas y cuellos de botella por medio de un análisis del sistema administrativo y de procesamiento de información de la clínica de odontología U.P.A.E.P.

- Realizar un estudio de las operaciones y procesos de la clínica de odontología que permita diseñar un sistema de información que combine eficientemente un modelo de control interno y el uso de computadoras.

- Crear un sistema de cómputo que automatice la captura, sumarización y recuperación de la información generada en la clínica y que apoye la toma de decisiones.

- Aplicar el sistema diseñado, familiarizar al personal con su uso y mejorarlo de modo tal que cumpla los alcances mencionados en el punto siguiente.

A.1.4 ALCANCES

-El sistema de información odontológico se desarrollará en el manejador de bases de datos PARADOX de la empresa Microsoft.

- El sistema de información odontológico sólo funciona dentro del ambiente operativo Windows de Microsoft incluyendo la versión para trabajo en grupo.

- El sistema de información odontológico será desarrollado y operado en computadoras personales compatibles con el estándar IBM.

- El sistema de información odontológico cubrirá el manejo de redes de área local.

- Todos los reportes impresos se elaborarán en modo gráfico y pueden imprimirse en equipos de matriz de puntos, tinta o láser.

A.1.5 PLANTEAMIENTO DEL PROBLEMA

En la clínica de odontología de la U.P.A.E.P. no se lleva un control eficaz de los ingresos y egresos de efectivo ni del inventario, tampoco se registra la casuística ni el uso de las instalaciones y equipo. El registro de ingresos depende de una sola persona, lo cual podría fomentar malos manejos, pérdidas, errores, etc.

El manejo de los expedientes, datos de pacientes, alumnos, materiales, horarios y calificaciones se realiza de manera manual, lo cual hace lenta y poco eficiente la atención, dificulta el aprendizaje y la evaluación y limita la toma de decisiones a corto y mediano plazo.

La dirección de la clínica no cuenta con información suficiente ni organizada para determinar la eficiencia del servicio ni del uso de recursos.

La escuela de odontología no tiene acceso eficiente a la información generada en la clínica sobre el desempeño de los alumnos dentro de ella.

A.1.6 DESARROLLO DEL PROBLEMA

La administración de una clínica en cualquier área de la medicina implica una serie de procesos que requieren manejar una gran cantidad de datos a la vez con rapidez y precisión. Estos datos, al mismo tiempo de servir como una base para la evaluación del desempeño funcional, también sirven para establecer políticas y objetivos a corto, mediano y largo plazo.

La cantidad de la información requerida implica la dedicación de una cantidad considerable de recursos a las tareas de recopilación, proceso y recuperación de ésta. Los procesos manuales pueden garantizar la seguridad y la calidad de la información, pero sacrificando considerablemente la rapidez, al mismo tiempo que produce un desperdicio de recursos valiosos como son el tiempo, el personal, los canales de comunicación, etc.

La escuela de odontología de la U.P.A.E.P. persigue, en el desarrollo de cada una de las materias y acciones de servicio, formar cirujanos dentistas para práctica general en el aspecto preventivo, con un amplio sentido de servicio como respuesta a las necesidades de salud.

La escuela de odontología de la U.P.A.E.P. posee un modelo de "Asistencia-Docencia", esto es, que en sus instalaciones se realizan actividades académicas que corresponden a las fases preclínicas y clínicas en todas las áreas y al mismo tiempo se realizan los procedimientos para la atención de personas que solicitan servicio de estomatología, esto sobre todo a personas de bajos recursos económicos, participando así en el servicio prestado a la comunidad por la universidad.

Las áreas funcionales que integran la clínica son: **Administrativa, Recepción/Caja, Central de Equipos, Clínicas Especializadas, Laboratorio y Radiología.**

A.1.7 PASOS A SEGUIR EN LA SOLUCION DEL PROBLEMA

- Manejo efectivo de expedientes de modo tal que sea posible saber en cualquier momento su clasificación y datos más importantes.

- Registro, sumariazación y reporte de los ingresos económicos derivados de la atención a pacientes, y de los egresos en general.

- Registro y presentación adecuada del uso de recursos a través del control de casos y uso de instalaciones y equipo.

- Control de horarios y calificaciones.

- Posibilitar la edición y consulta de la revista digital de la escuela de odontología U.P.A.E.P. que incluya textos e imágenes digitalizadas.

- Posibilitar el acceso a la información por medio de computadoras independientes enlazadas en una red de área local.

- Con el sistema a realizar para la clínica de odontología se llevará un control eficaz de los ingresos y egresos de efectivo y de inventario, se registrará la casuística y el control del uso de las instalaciones y equipo, con lo cual se evita que haya malos manejos, pérdidas, errores, etc.

- El manejo de los expedientes, datos de pacientes, alumnos, materiales, horarios y calificaciones al realizarse de manera automática hará que se haga el trabajo más rápido y la atención más eficiente, facilitando el aprendizaje, la evaluación y la toma de decisiones a corto y mediano plazo.

- La dirección de la clínica contará con la información suficiente y organizada para determinar la eficiencia del servicio y el uso de los recursos.

- La escuela de odontología tendrá un acceso eficiente a la información generada en la clínica sobre el desempeño de los alumnos dentro de ella.

A.1.8 PLANTEAMIENTO DE HIPOTESIS

El desarrollo de un sistema de cómputo para la clínica de odontología de la U.P.A.E.P. que automatice el control de los ingresos y egresos de efectivo y del inventario, que facilite el registro de la casística y del uso de las instalaciones y equipo, que dicho control pueda delegarse a diferentes personas sin perder de vista la seguridad e integridad de la información, que controle los datos de los pacientes, alumnos, materiales, horarios, calificaciones, uso de recursos y materiales, así como el desempeño del personal y alumnos de manera automática, que brinde acceso eficiente a dicha información para que colabore con el aprendizaje y la evaluación y que apoye la toma de decisiones a corto y mediano plazo.

A.1.9 COMPROBACION DE HIPOTESIS:

- Las empresas que reducen la incertidumbre por medio del manejo adecuado de información pueden tomar decisiones inteligentes y conocer el resultado de éstas en menor tiempo por medio de una adecuada retroalimentación.

- Las computadoras y la tecnología asociada a ellas utilizados adecuadamente y analizados de acuerdo con la relación Costo/Beneficio, influyen en gran medida a la efectividad de los sistemas de información, sobre todo cuando el volumen, la complejidad y las demandas de oportunidad y cálculo son elevados.

- Los sistemas de información basados en el uso de computadoras agilizan los procesos de captura, recuperación y resumen de la información, por lo tanto facilitan la toma de decisiones.

- Cuando el volumen de información manejado dentro de una empresa o institución es elevado el método computarizado rebasa en efectividad a los métodos manual y electromecánico.

- Las instituciones y empresas que utilizan equipos de cómputo para apoyar las tareas de control y toma de decisiones disminuyen los recursos asignados a estas tareas y se vuelven más productivas.

- Los usuarios de equipos de cómputo obtienen resultados efectivos en menor tiempo, evitando así el tedio y aumentando la productividad.

A.2 BREVE DESCRIPCION DEL TRABAJO

En el capítulo 1 se presentan las generalidades de la institución de enseñanza que sustenta, da origen y rige las actividades académicas de la clínica de odontología U.P.A.E.P.

En el capítulo 2 se presentan las características organizacionales y operativas de la clínica de odontología U.P.A.E.P. De la misma manera se expone el resultado del análisis sobre el sistema de información manual que se aplica en la clínica de odontología con sus aciertos y puntos débiles.

En el capítulo 3 se analiza el valor de la información en la toma de decisiones, así como la teoría general de la administración que fundamenta la afirmación de que las computadoras y los sistemas computacionales son herramientas excelentes para ayudar a eficientar las tareas de planeación, organización y control.

CAPITULO 1:

Dado el objetivo general de este trabajo, que es que en la clínica de odontología U.P.A.E.P. se controle eficientemente la información, de modo que sea suficiente y oportuna para tomar decisiones con más certidumbre y en menos tiempo, que se fortalezcan otras áreas de la administración y se brinde un mejor servicio tanto a los pacientes como a los alumnos, además de que se controlen los ingresos y egresos de efectivo, así como los recursos materiales e insumos. En este capítulo se presentan las generalidades de la institución de enseñanza que sustenta, da origen y rige las actividades académicas de la clínica de odontología U.P.A.E.P.

1. LA UNIVERSIDAD POPULAR AUTONOMA DEL ESTADO DE PUEBLA

La Universidad Popular Autónoma de Estado de Puebla ha nacido en un momento en el que la universidad tecnocrática es sacudida por su propio fracaso, ya que esta institución es asaltada por doctrinas que niegan la libertad de cátedra, la libertad de investigación y con ello aniquilan la autonomía universitaria. Nacida la Universidad Popular Autónoma de Estado de Puebla no en un estado de paz, sino en la lucha contra la escalada violenta del sectarismo, la universidad anhela realizarse como una institución que atiende la finalidad última de la enseñanza, que es la de forjar el espíritu humano, con el ánimo dispuesto a la defensa de la verdad. Sería absurdo el crear una nueva universidad si esta careciera de un ideario y de una muy personal forma de ver la realidad fundamentada en una verdad.

Del contenido del ideario de la Universidad Popular Autónoma de Estado de Puebla surge la decisión de encausarla por un camino duro y difícil que da a todos y cada uno de sus integrantes un estilo diferente de ver la vida y de afrontarla, ya que la Universidad Popular Autónoma de Estado de Puebla busca principalmente formar primero al hombre antes de pensar en la profesión que éste desarrollará durante el resto de su vida.

1.1 BREVE HISTORIA

El hombre, siendo una criatura dotada de inteligencia y voluntad lo vuelve una persona libre y responsable de sus actos con una vocación única. Por ser portador de una dignidad superior a cualquiera de las demás criaturas que poblan la tierra, por lo tanto el hombre está sujeto de deberes y derechos. El hombre siendo social por propia naturaleza, es capaz de transformar el mundo en su provecho, de progresar, de construir su sociedad y vivir conforme al derecho que el mismo instituye como sociedad por lo que tiene derecho a bienes inalienables como lo son la verdad, el bien y la justicia.

Todo hombre tiene derecho a poseer bienes materiales y espirituales que son necesarios para desarrollarse totalmente conforme a su dignidad de persona humana y así alcanzar su fin último sabiéndose responsable de su libertad. En la sociedad cada ser humano tiene su propio lugar y como una persona libre responsable de sus actos y sabedora de sus derechos y obligaciones, aporta sus conocimientos y esfuerzo al servicio de sus semejantes para lograr la construcción de una sociedad justa y ordenada. La vida íntegra y sana de la sociedad depende de la plenitud que gocen los hombres que la conforman, esto es necesario para garantizar la libertad de las personas y el cumplimiento del fin de la sociedad, la familia, la empresa, las escuelas, etc, constituyen el cuerpo de la sociedad ya que sin ella el pueblo se convierte en masa.

1.2 IDEARIO Y MISION DE LA U.P.A.E.P.

La universidad es una institución que existe con el tiempo, pero que posee como cualidad primordial la de mantenerse fiel a la verdad que no está sujeta a la dinámica temporal. Su propia estructura está diseñada por la misión de la enseñanza y la investigación, como actividades que la complementan. En sí misma, la universidad lleva la vocación de conocer la universalidad de ideas, por eso la universidad actual ha de hacer suyo el objetivo y misión que le dió origen hace siglos, que es el de definirse como una comunidad organizada y jerárquica de maestros y alumnos que tienen como fin la transmisión, la investigación, organización y defensa de la verdad al servicio de la sociedad de la cual forma parte.

La Universidad Popular Autónoma de Estado de Puebla, es una comunidad ordenada y organizada de maestros, alumnos y autoridades, esta unión se basa en la verdad, compartida y buscada en el enriquecimiento mutuo, teniendo un mismo fin los alumnos y maestros trabajan investigando; los maestros como guías que convierten a los alumnos en colaboradores; los alumnos redescubriendo y haciendo suya la verdad y la verdad encontrada por la propia investigación.

La Universidad Popular Autónoma de Estado de Puebla sabe de lo valioso del patrimonio cultural de la humanidad y es por ello que se compromete a incrementar, difundir y defender el acervo cultural de nuestra patria que forma parte del ser del pueblo de México, que es la base y el cauce de su futuro progreso y enriquecimiento cultural.

La Universidad Popular Autónoma de Estado de Puebla sostiene como principio y criterio básico el respeto a la libertad, fuente de la dignidad humana, y entendemos con ello la capacidad que tiene el hombre de autoestimarse y guiarse al cumplimiento y realización de su fin último y por lo mismo, responsable de sus actos y del cumplimiento de su destino.

La Universidad Popular Autónoma de Estado de Puebla respeta la conciencia de todos los miembros de la comunidad universitaria, por lo tanto se entiende que la verdad no se impone, sino que se propone y se convence cuando es correctamente expuesta, por lo tanto se afirma que se tiene el derecho a ser educado en la verdad.

La Universidad Popular Autónoma de Estado de Puebla entiende la libertad de cátedra como el derecho de enseñar la verdad, aunado al derecho que tiene todo maestro de no enseñar aquello de lo que no está convencido, y al derecho que tiene el alumno de no aceptar una tesis mediante coacción.

La Universidad Popular Autónoma de Estado de Puebla asume la responsabilidad, que es la otra cara de la moneda ya que esto implica el espíritu propio del ideario de la universidad, en el que han sido invitados a participar libremente a todos los integrantes de la comunidad universitaria como lo son los maestros, alumnos, trabajadores, y demás autoridades universitarias.

La Universidad Popular Autónoma de Estado de Puebla reconoce también que cualquier aportación verdadera de la ciencia y la cultura vale muy independientemente de la raza, nación credo o filosofía de los hombres que la hayan realizado.

La Universidad Popular Autónoma de Estado de Puebla defiende también el derecho de la libre investigación científica al que no se conocen otros límites que el respeto a la verdad en todas sus manifestaciones, a las leyes de la moral y la justicia.

La Universidad Popular Autónoma de Estado de Puebla afirma también la autonomía como un valor esencial de la universidad y como la garantía indispensable para el desarrollo de su vida y el cumplimiento de su misión. Esta autonomía se entiende como la libertad para autodeterminarse conforme a sus principios y leyes en todo aquello que atañe a la realización de sus fines, organización interna, nombramiento de sus autoridades, personal académico y administrativo, admisión de alumnos, elaboración de planes de estudio y elección de su contenido, elección de sus sistemas pedagógicos, adquisición y administración de sus recursos económicos. Aunque el significado propio de autonomía no significa que la universidad deja de pertenecer a la sociedad de la cual forma parte, ni que sea un estado dentro de otro estado, por lo que reconoce la obligación colaborar y estar sujetas a las exigencias del bien común de nuestra sociedad.

La Universidad Popular Autónoma de Estado de Puebla colabora y se enriquece con la experiencia propia de otras universidades e instituciones de enseñanza superior, por lo que busca la relación y el intercambio con ellas.

La Universidad Popular Autónoma de Estado de Puebla se considera una comunidad dentro de la gran comunidad nacional, la actitud de la universidad es positiva, no se quiere destruir sino construir, aportar y compartir con todos la verdad como un fundamento de unidad.

La Universidad Popular Autónoma de Estado de Puebla considera a los maestros como el elemento principal de la comunidad universitaria. El mejor maestro no es aquel que sabe más sino el que es más capaz de comunicar la verdad y de enseñar a amarla, el buen maestro es aquel que enseña a sus alumnos el hábito del estudio y la investigación.

La Universidad Popular Autónoma de Estado de Puebla se propone lograr la excelencia académica, cuidando que ésta se armonice con una equilibrada formación integral de los alumnos, para ello se atiende a las actividades docentes y a los constantes progresos de la ciencia y la pedagogía y se considera a la investigación como la actividad superior de la vida universitaria.

La Universidad Popular Autónoma de Estado de Puebla considera a los alumnos como elemento activo e integrado a la comunidad universitaria, responsables de sí mismos y de los fines de la universidad, dentro de la jerarquía de la universidad los alumnos participan con su estudio e investigación empeñando todas las fuerzas en la formación integral y en el engrandecimiento de la propia universidad. Congruentes con ésta concepción de educación, la universidad concibe al alumno como el agente principal pero no único de su propia afirmación pues es necesario que exista un maestro que le comunique los conocimientos, el acervo cultural y científico logrado con

el esfuerzo común de la universidad y que lo guíe y lo estimule a investigar. Activo en la vida universitaria, guiado por las autoridades y maestros de forma en la comunicación e intercambio con sus compañeros y maestros de todas las escuelas.

El fin primordial de la Universidad Popular Autónoma de Estado de Puebla es introducir a los alumnos en el conocimiento y análisis de los problemas sociales, económicos y políticos de nuestra patria, y el de crearles conciencia de su responsabilidad en la promoción del bien común, es por ello que la universidad considera como una parte esencial de la formación de los alumnos la realización de su servicio social, que ha de colaborar con el desarrollo del pueblo, y se acerque a los sectores sociales más necesitados y les forme un sentido de solidaridad y de servicio que impregne el ejercicio de su profesión.

La Universidad Popular Autónoma de Estado de Puebla siendo responsable de su función, pondrá todo su empeño en la selección, capacitación, actualización, etc. de los maestros y con respecto a sus alumnos, se les apoyará y orientará en sus iniciativas que contribuyan a su afirmación, y al engrandecimiento mismo de la universidad.

Quien es universitario lo es para siempre, y se caracteriza porque en todas las circunstancias de su vida se guía por la verdad conquistada en el estudio y reflexión personal, éste es el universitario que queremos formar. La misión del universitario llega a su plenitud cuando aplica competentemente los conocimientos que ha adquirido y vive los valores de los que se nutrió en la universidad.

Para tal fin, la Universidad Popular Autónoma de Estado de Puebla instrumentará los mecanismos necesarios para mantener el vínculo de sus egresados y colaborará en su actualización profesional.

La Universidad Popular Autónoma de Estado de Puebla se propone crecer y abarcar equilibradamente la diversidad de las disciplinas científicas, pero se darán prioridades en el desarrollo de las ciencias y actividades más urgentes que tenga la comunidad y que contribuyan más eficazmente al progreso de nuestra región. Por lo que la universidad no es indiferente a las necesidades que tiene el pueblo.

Las soluciones a los problemas que tiene el hombre no se pueden improvisar, la Universidad Popular Autónoma de Estado de Puebla contribuye a la solución de los problemas fundamentados con un estudio serio y metódico, trabaja desde hoy en la transformación de los hombres por la verdad y la cultura.

Se quieren universitarios que tengan dignidad y coraje y que luchen por sus ideales. La mejor manera de solucionar los problemas urgentes de la sociedad es el estudio serio de los mismos y una gran preparación, estar capacitados para resolverlos en su momento.

1.3 CULTURA

La cultura en su acepción más noble y elevada es la formación del hombre, es decir, el cultivo y la maduración perfecta de las potencialidades y capacidades humanas y el conjunto de obras, instituciones, virtudes, ideas, principios, etc. Cada generación ha de recibir la cultura como el conjunto de obras y realizaciones valiosas de los hombres en los distintos terrenos de la actividad humana. La cultura es un patrimonio de todos los hombres, y cada uno de ellos tiene derecho a tener la oportunidad de conocer esta cultura. El hombre no debe conformarse con la cultura que le dan sino que debe incrementarla y proyectarla hacia las nuevas realizaciones para el bien de las futuras generaciones.

La cultura es formación y no solo instrucción o información, la cultura es armónica, orgánica y jerarquizada, adecuada al orden de las potencialidades del hombre, en la cultura es más importante la calidad y profundidad que la cantidad y la acumulación.

1.4 EDUCACION

La educación es el proceso permanente de actualización ordenada y jerarquizada de todas las potencialidades y capacidades del hombre, de tal manera que logre una plenitud y madurez plena. En este sentido la educación es tarea fundamental del hombre, el fin de la educación es lograr que el hombre llegue a ser lo más perfecto posible. El agente principal de la educación es el propio educando, ya que es éste el que asimila, conquista y forma a su propio criterio las obras valiosas que su maestro le aproxima y le presenta de forma ordenada y progresiva.

La educación se logra por mediación de la cultura, que forma a quien la acoge y la integra a su persona, es por ello que ha de ponerse empeño educativo, en la calidad académica de la enseñanza con el fin de lograr que los hombres sean cada vez mas grandes y mas generosos con sus semejantes.

Todo hombre tiene derecho a la educación, pero todos aquellos que tienen la oportunidad de acercarse a una educación de nivel superior deben tener conciencia de que su servicio sea humilde para que fructifique en la educación del pueblo. Contra el orgullo intelectual y elitista de que el hombre educado no es el que sabe más, sino el que es mejor.

CAPITULO 2:

Dado el objetivo general de este trabajo, que es que en la clínica de odontología U.P.A.E.P. se controle eficientemente la información, de modo que sea suficiente y oportuna para tomar *decisiones con más certidumbre y en menos tiempo, que se fortalezcan otras áreas de la administración y se brinde un mejor servicio tanto a los pacientes como a los alumnos, además de que se controlen los ingresos y egresos de efectivo, así como los recursos materiales e insumos.* En este capítulo se presentan las características organizacionales y operativas de la clínica de odontología U.P.A.E.P. De la misma manera se expone el resultado del análisis sobre el sistema de información manual que se aplica en la clínica de odontología con sus aciertos y puntos débiles.

2. LA CLINICA DE ODONTOLOGIA U.P.A.E.P.

2.1 ANTECEDENTES

La escuela de odontología, busca poseer un equilibrio entre la teoría impartida en las aulas y la práctica, de tal manera que, *además de brindar a sus alumnos la atención de un plantel de catedráticos bien capacitados, trata de impulsar al alumnado a desarrollarse en la práctica clínica.*

A la preocupación de la escuela por apoyar a sus alumnos, se suma la necesidad de cumplir con un mínimo de un año en servicio social y prácticas profesionales que estipula la reglamentación estatal de educación y de salud.

Es así como se creó la clínica de odontología, para proporcionar el entrenamiento en laboratorios y clínicas necesario para reforzar el conocimiento en estomatología y con el fin de proporcionar a la población que lo necesite, servicios dentales como institución privada.

La escuela persigue en el desarrollo de cada una de las materias y acciones de servicio, formar cirujanos dentistas para la práctica general en el aspecto preventivo en un amplio sentido de servicio como respuesta a las necesidades de salud.

Los estudios de odontología comprenden nueve semestres, durante los cuales el alumno cursa 64 materias, que incluyen conocimientos básicos de las ciencias de la salud, materias preclínicas, laboratorios, y a partir del tercer semestre se inicia propiamente el entrenamiento del alumno en clínicas.

La clínica de odontología posee un modelo "Asistencia-Docencia", esto sobre todo a personas de bajos recursos económicos y participando así en el servicio prestado a la comunidad por la universidad.

Todos los procedimientos clínicos que se dan en los servicios dentales deben estar planeados y organizados para ser cumplidos por pacientes, alumnos, profesores y personal, desde el registro de pacientes hasta su alta.

El número de pacientes atendidos diariamente en la clínica es en promedio de 65, en un horario de 8:00 a 22:00 horas de lunes a viernes. Las operaciones realizadas en su totalidad son supervisadas por los catedráticos.

2.2 ESTRUCTURA Y ADMINISTRACION

2. 2.1 AREAS QUE LA COMPREDEN

2. 2.1.1 RECEPCION

La recepción es un área donde se conjuntan las tareas administrativas y clínicas de la institución, pues el personal que ahí labora debe tener un control total tanto de las cuestiones administrativas (como los ingresos y egresos de efectivo), el control de los expedientes y vales de material, de modo que debe poseer conocimientos y herramientas que le permitan determinar en un momento dado, cuanto hay que cobrar por un tratamiento, si el paciente es de nuevo ingreso o de cita subsecuente, qué alumno lo va a tratar y en qué unidad, clínica y horario, así como el material necesario para hacer el vale que se entregará en central de equipos para obtener los insumos necesarios para el tratamiento.

Los pacientes son recibidos en esta área, ahí se les informa de los tratamientos que ofrece la clínica así como de las políticas de cobros y atención.

El personal del área de recepción recibe el dinero de los pacientes y les asigna o confirma el alumno tratante, actualiza la hoja de control y el expediente, emite el recibo y entrega el vale para el alumno.

Al llegar el final de cada periodo (mes, semestre, año), el personal de recepción ejecuta las tareas administrativas de sumariación y reporte (además de las mismas tareas diarias), esto es, emite un reporte de ingresos y egresos, de manejo de caja chica y de almacén, al departamento de tesorería.

De la misma manera, se encarga de elaborar los reportes de calificaciones, de uso de equipo e instalaciones y de atención para la dirección de la clínica y de la escuela de odontología.

2.2.1.2 CENTRAL DE EQUIPOS

En esta área se encuentran los equipos, materiales y medicamentos necesarios para cada uno de los servicios que presta la clínica.

La enfermera tiene la función de preparar todo el instrumental y los insumos necesarios para el procedimiento operatorio, su actividad empieza antes que todo el equipo dental.

Otras funciones son las de mantener el instrumental esterilizado así como su organización, distribución y almacenamiento en el momento que se requiera, tener el campo o bulto quirúrgico para el personal aséptico, consultando directamente con el cirujano, tener la capacidad de conocer e identificar los instrumentos, materiales, sustancias químicas y fármacos que poseen propiedades antisépticas, germicidas y bactericidas.

También debe tener amplio conocimiento sobre la utilización del equipo, instrumentos y material que se utiliza en todos y cada uno de los procedimientos del laboratorio y clínicas, al igual que tener un control absoluto y conteo exacto de éste.

La central de equipos y esterilización, así como la persona encargada de esto, son indispensables para el funcionamiento de una clínica dental, para lograr que haya la debida asepsia y antisepsia en el lugar en donde se atienden a numerosas personas.

2.2.1.3 CLINICAS

Las clínicas son el lugar donde se practican todos los tratamientos a los pacientes, es considerada un área blanca, por lo tanto es de vital importancia su asepsia y antisepsia.

En las clínicas se encuentran las unidades odontológicas, las cuales constan de: Un sillón que permita la comodidad del paciente y la facilidad de acceso para el médico o alumno tratante, dos piezas de mano para colocar las diferentes fresas quirúrgicas que permitirán preparar las cavidades en las piezas dentales, y un extractor o succionador de material de residuo y saliva o sangre del paciente; todo esto además de una lámpara y una mesa de instrumental.

Los pacientes permanecen en las clínicas durante todo el tiempo que dura su tratamiento, exceptuando algunas visitas esporádicas a la sala de rayos X, donde se toman las impresiones de las piezas dentales para trabajar en base a un plan predeterminado.

Cuando el paciente llega a la clínica de tratamientos, el alumno ha hecho antes una sesión de entrevista para conocer tanto sus datos generales como su historia clínica, estos datos le permitirán al médico o alumno determinar las causas del padecimiento actual, así como las acciones y tratamientos a seguir.

2.2.1.3 LABORATORIO

Es un área específica e imprescindible para el desahogo del conocimiento y un desarrollo integral del estudiante de odontología en las prácticas propias del área preclínica y clínica en gran parte de las especialidades dentales.

El alumno debe conocer el proceso de manufactura y adaptación de los diferentes medios en los que basará la posible rehabilitación del paciente ya sean prótesis o simples aparatos preventivos que le permitan un desarrollo completo e integral de su futura profesión.

De éste modo se hace imprescindible en una institución de Asistencia-Docencia odontológica, la existencia de un laboratorio, pues es el lugar donde el estudiante obtendrá la familiaridad y práctica para conocer adecuadamente el uso particular y específico de cada uno de los diferentes materiales dentales.

Las materias que precisan del uso de laboratorio son:

Anatomía dental.- Por medio de encerados y tallados que desarrollan además la habilidad manual del futuro odontólogo.

Ortodoncia.- Fabricación de aparatos preventivos e interceptores, estudio y fabricación de modelos.

Oclusión.- Tal vez la materia que, unida a la de prótesis, exige más conocimiento del laboratorio por parte del estudiante.

Operatoria dental.- La fabricación y vaciado de algunas restauraciones y el estudio preclínico que el alumno llevará a cabo antes de operar en el paciente.

Prostodoncia.- Fabricación de las prótesis que el paciente llevará en su dentadura y que previamente el alumno practicó repetidas veces por medio del estudio en ceras, acrílicos, yesos y un sinnúmero de materiales.

Prótesis.- El encerado de las curaciones parciales , totales y removibles durante todo el proceso, pasando por el corrido de modelos, recortando el estudio requerido de estos al vaciarlos, pulirlos y adaptarlos.

2.3 SISTEMA DE INFORMACION MANUAL

2.3.1 SECUENCIA DE ACTIVIDADES

Paciente —————>Recepción
(Solicita servicio y paga 1ª consulta: Dx, estudio Rx) (forma 1 y 2)

Recepción —————>Alumno
(Entrega expediente y el alumno llena historia clínica y toma estudio Rx) (Forma 3 y 4)

Alumno —————> Paciente
(Saca Dx y Plan de tratamiento y remite al departamento de Exodoncia, Operatoria, etc.)
(Forma 5)

Paciente —————>Recepción
(Paga derecho de atención y recibe comprobante de pago) (Forma 1)

Recepción —————>Alumno
(Entrega a alumno tarjeta de control de atención clínica endodoncia, Operatoria, etc. y comprobante de pago) (Formas 1, 6, 8)

Alumno —————>Central de Equipo
(Con el comprobante de pago solicita material clínico, medicamentos o instrumental)
(Forma 7)

Central de Equipo————>Alumno
(Entrega material clínico, medicamentos, instrumental y equipo con vale) (Forma 7)

Alumno —————>Paciente
(Da atención clínica al paciente con autorización y supervisión del maestro) (Forma 5, 8)

Paciente —————>Alumno
(Se coordina una nueva cita) (Forma 5)

Alumno —————>Recepción
(Regresa el expediente y tarjetas firmadas a la recepción) (Formas 1, 2, 3, 4 y 8)

2.4 MODELO ASISTENCIA-DOCENCIA

La Clínica de Odontología posee un modelo de "Asistencia-Docencia", ésto es, que en las instalaciones de la clínica se realizan actividades académicas que corresponden a las faces preclínicas y clínicas en todas las áreas de servicio y al mismo tiempo se realizan los *procedimientos para la atención de personas que solicitan el servicio de estomatología (odontología general y de especialidades)*, esto sobre todo a personas de bajos recursos económicos y participando así en el servicio prestado a la comunidad por la universidad.

La clínica al poseer este modelo logra un equilibrio entre la teoría, el entrenamiento en laboratorios y la capacitación dentro de las clínicas por medio de trabajo de campo, ésto hace que los egresados posean una alta capacitación en todo lo que concierne a las necesidades orales del país. La escuela persigue en el desarrollo de cada una de la materias y acciones de servicio el formar cirujanos dentistas para prácticas en general en el aspecto preventivo en un amplio sentido de servicio como respuesta a las necesidades de salud. Tomando en cuenta que todos los trabajos odontológicos que realizan los alumnos son bajo la estrecha supervisión de un doctor, y los alumnos no pueden hacer ningún trabajo si no tiene el visto bueno de los doctores.

CAPITULO 3:

Dado el objetivo general de este trabajo, que es que en la clínica de odontología U.P.A.E.P. se controle eficientemente la información, de modo que sea suficiente y oportuna para tomar decisiones con más certidumbre y en menos tiempo, que se fortalezcan otras áreas de la administración y se brinde un mejor servicio tanto a los pacientes como a los alumnos, además de que se controlen los ingresos y egresos de efectivo, así como los recursos materiales e insumos. *En este capítulo se analiza el valor de la información en la toma de decisiones, así como la teoría general de la administración que fundamenta la afirmación de que las computadoras y los sistemas computacionales son herramientas excelentes para ayudar a eficientar las tareas de planeación, organización y control.*

3 ADMINISTRACION Y COMPUTACION

3.1 ADMINISTRACION E INFORMACION

3.1.1 EXPOSICION GENERAL DE ADMINISTRACION

Cuál es la función de la administración? Algunas autoridades afirman que la tarea fundamental de la administración consiste en hacer frente a las condiciones cambiantes. Otros sostienen que esa tarea consiste en reconocer y asimilar los cambios tecnológicos, de manera que se produzcan valores prácticos que serán entregados a la sociedad en forma ordenada, oportuna y económica. Otros más pretenden que la administración consiste simplemente en "lograr que las cosas se hagan mediante el esfuerzo de los demás".

Un examen de lo que se ha escrito al respecto, revela una multiplicidad de tareas asociadas con la función administrativa. Entre ellas figuran las siguientes: planeación, programación, dirección, organización, contratación, entrenamiento, control, supervisión, etc. Tradicionalmente, la función administrativa se ha identificado con la planeación, la organización y el control. Se van a dividir esa función en tres tareas fundamentales: Planeación, Control y toma de decisiones. Si los analistas de sistemas tienen que diseñar sistemas efectivos de información que sean útiles para el administrador, es obvio que debe entender no sólo la función administrativa, sino también sus necesidades de información.

En los sistemas, la administración puede tomar decisiones , establecer planes y llevar el control señalando las entradas; pero como ya se dijo, no se tiene control sobre todas esas entradas. Por otra parte, puede encontrarse presente una buena dosis de incertidumbre con respecto a aquellos sobre los cuales ejerce control.

3.1.2 LA PLANEACION

Es momento de señalar que todos los sistemas persiguen un objetivo. La planeación, establece con anterioridad a toda actividad, los criterios que los administradores deben satisfacer para alcanzar las metas señaladas. La planeación es básica para el control y la toma de decisiones, aunque podría aducirse que es necesario decidirse primero por un plan. Sin la planeación no haría falta el control, puesto que aquella es el fundamento de la administración y el control, así como de las otras actividades, como la organización y la programación. La administración necesita información para fijar objetivos y cursos de acción. Es imposible proporcionar toda la información necesaria para abarcar todos los aspectos y contingencias de un plan; pero la que se aporta, ayuda a disminuir la incertidumbre, amenaza constante del administrador.

Analizando más a fondo la planeación encontramos cinco tareas básicas que debe llevar a cabo el que planea:

1. Establecer metas y objetivos.
2. Identificar los eventos y actividades que deben realizarse para alcanzar los objetivos.
3. Describir los recursos y la habilidad requeridos para realizar cada actividad.
4. Definir la duración de cada actividad señalada.
5. Determinar la secuencia, si la hay, en que deben llevarse acabo las actividades.

De esta descripción de la planeación se deduce que es una condición permanente de la administración, si no en todas, si en la mayoría de las organizaciones. Además, en casi todas ellas, la tarea de hacer planes la comparten en grados diversos todos los niveles de la administración.

3.1.3 EL CONTROL

Las cosas, en verdad, rara vez marchan de acuerdo con los planes. Las desviaciones se deben a factores que están mas allá del control de los administradores; por ejemplo, errores de estimación y equivocaciones cometidas por los subordinados. En la mayor parte de los sistemas ocurre siempre que algo marcha mal, por lo cual se impone una actividad de control. El control es un proceso que consiste en tres etapas básicas: **Evaluar lo que produce el sistema, Comparar el resultado con lo planeado para encontrar las desviaciones, Corregir las desviaciones desfavorables poniendo el remedio que sea necesario.** Subrayemos aquí que, para que el administrador pueda emprender una acción correctiva, se supone que tiene el poder de modificar o sustituir algún recurso; suposición muy importante.

Como se dijo antes, la planeación establece los objetivos, y el control garantiza el logro de esos objetivos a través de la retroalimentación de informes a la administración.

Elementos de retroalimentación y control. Si una persona permanece sentada por largo tiempo con las piernas cruzadas, al levantarse descubre, que le es difícil sostenerse de acuerdo con lo planeado. Lo que ocurre es que la retroalimentación de los sensores situados en la base de los pies ha desaparecido prácticamente. En consecuencia, no puede ejercer control para adoptar la posición planeada.

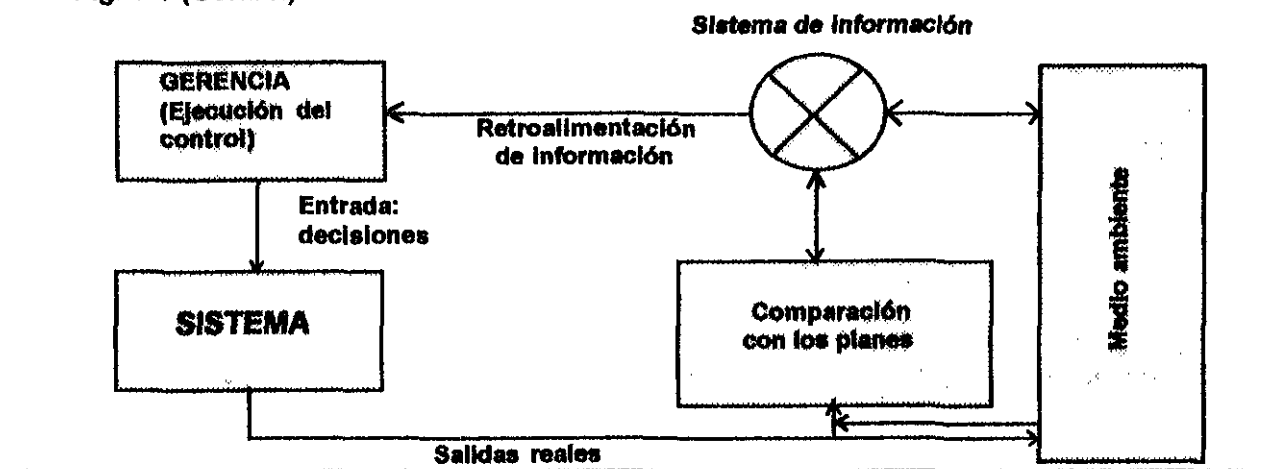
Al lanzar una pelota de fútbol, los zagueros confían en un sistema de retroalimentación que evalúa continuamente sus errores de posición, la rapidez y posición de sus receptores, las posiciones y velocidad de los jugadores de línea y el nivel de protección. Para controlar debidamente el lanzamiento, deben reducir el error a cero, o casi cero.

Los elementos del sistema que han sido enumerados son la entrada, la salida y el proceso. Estos elementos representan los parámetros del sistema, una parte de los cuales se puede medir y evaluar. Para medir la efectividad general que posee el sistema para lograr sus fines, acudimos al parámetro de salida, que representa el punto central de control.

Para lograr este parámetro se disponen de tres elementos de control:

1. Establecimiento de un estándar para el resultado, es decir, el rendimiento deseado u objetivo del sistema.
2. Diseño e implementación de un elemento sensor que recoja información relacionada con el resultado, para pasar los resultados obtenidos a la dirección de la administración.
3. Empleo de uno o más supervisores o de un mecanismo, que tomen acción correctiva si la información obtenida así lo señala. Dicha acción correctiva da lugar a la liberación de una decisión, que actúa como elementos de entrada para el sistema (En la figura 3.1 se da un ejemplo sencillo de control).

Fig. 3.1 (Control)



El sistema de información mide y evalúa los resultados del proceso. Los resultados pueden consistir en las ventas, el volumen de ventas, las utilidades, el control de calidad, etc. Estos resultados emanan del proceso del sistema.

Si se trata de una organización comercial o de cualquier otra clase, el sistema de información debe entregarla a los usuarios externos e internos. Los usuarios externos, como son los accionistas o el gobierno pueden influir en alguna forma en las decisiones tomadas por el grupo administrativo interno. El sistema de información debe igualmente recoger información del exterior, acerca del nivel de la demanda, el precio de los materiales, la disponibilidad de mano de obra, el precio de la misma, etc.

El control de un sistema será efectivo si:

1. Si los resultados reales son debidamente medidos y comparados con los deseados,
2. Si la acción necesaria es puesta en ejecución y,
3. Si tanto la retroalimentación de informes como la toma de decisiones son lo suficientemente rápidos como para efectuar correcciones antes de que los factores que intervienen en el proceso se vuelvan inconsistentes con las mismas.

3.1.3.2 TIPOS DE CONTROL

El elemento de control se relaciona directamente con la característica del sistema sobre si es abierto o cerrado al medio ambiente. Existen dos tipos de control de acuerdo a los sistemas.

3.1.3.2.1 SISTEMA DE CIRCUITO ABIERTO

Este sistema se establece para lograr un determinado propósito y no hay retroalimentación de información. Por ejemplo, una máquina lavadora puede ser gobernada por un control de tiempo, de manera que realice un ciclo de lavado de duración determinada. Podríamos poner la máquina en marcha, dejarla y volver más tarde, sólo para encontrarla conque el ciclo de lavado se llevo a cabo sin agua porque la llave no fue abierta. Si el dispositivo de control hubiera sido informado de la situación mediante un sistema de retroalimentación, se habrían tomado las medidas necesarias para corregir la falla. Podemos pensar igualmente en instituciones que no operan con eficiencia ya sea por causa de un sistema deficiente de retroalimentación, o debido a que el mecanismo de control no toma en cuenta la información disponible.

A una escala mayor y más compleja, los sistemas comerciales que se manejan basándose en un circuito abierto pueden llegar a situaciones críticas cuya solución está más allá de las posibilidades de la administración. Por ejemplo, la producción de artículos alimenticios se debe revisar con todo cuidado a fin de detectar la presencia de contaminantes. La información obtenida mediante ese control se debe comunicar a los administradores a quienes corresponde corregir cualquier desviación. Sin ésta retroalimentación de información, la empresa corre el riesgo de un desastre imposible de corregir.

3.1.3.2.2 SISTEMA DE CIRCUITO CERRADO.

Este tipo de sistema es aquel en que el mecanismo de control recibe y utiliza la información retroalimentada. En éste ejemplo de la máquina de lavar, si el mecanismo de control ha sido ajustado de manera que el ciclo de lavado solamente pueda comenzar después de que una

-determinada cantidad de agua haya entrado en el recipiente, el lavado se efectuará sólo si dicho volumen de agua se encuentra presente. El mecanismo de control es advertido por el circuito de retroalimentación.

Todas nuestras organizaciones e instituciones deberían contar con un flujo de información a través de un sistema de circuito cerrado. La razón por la cual se necesita tal sistema se evidencia cuando consideramos que todo sistema, si ha de alcanzar sus objetivos, debe entregar periódicamente al sistema administrativo la información que le ponga al tanto del nivel de logros. Así podemos afirmar en general que todo sistema que se le proponga alcanzar sus metas debe utilizar el sistema de circuito cerrado para ejercer el debido control.

3.1.4 LA TOMA DE DECISIONES.

Una de las actividades básicas de la administración consiste en compenetrarse de las características del sistema hasta un grado que le permita juzgar el rendimiento con toda propiedad, mejorándolo dentro de las restricciones establecidas. La toma de decisiones, pues, es el proceso de elegir entre varias alternativas, que pueden ser cualitativas o cuantitativas, aquella que sea la mejor para resolver un problema o arreglar un conflicto.

Elementos de la toma de decisiones: La causa por la que un administrador debe tomar una decisión es la confrontación de un problema o la presencia de una situación conflictiva. El hecho de decidir resuelve el problema o arreglar el conflicto. Un proceso ordenado para llegar a una decisión contiene cuatro elementos:

3.1.5.1 MODELO:

Un modelo es una descripción cuantitativa o cualitativa del problema.

3.1.5.2 CRITERIOS:

Los criterios establecidos representan las metas u objetivos del problema de decisión. Cuando varios criterios entran en conflicto, quien toma la decisión debe elegir un término medio.

3.1.5.3. RESTRICCIONES:

Son varios los factores que se deben considerar cuando se trata de resolver el problema de decisión. La falta de recursos es una restricción.

3.1.5.4. OPTIMIZACION:

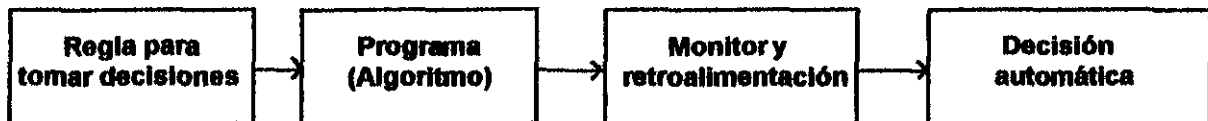
Una vez planteado claramente el problema (el modelo), el administrador determina las necesidades (los criterios) y lo que puede hacerse (las restricciones). En ese punto, quien toma la decisión está listo para elegir la mejor solución, es decir, la óptima.

Clases de decisión: Los problemas de decisión y los conflictos aparecen por todas partes. Algunos son sencillos y deterministas, con pocas ramificaciones; pero otros son bastante complejos y probabilistas y sus efectos pueden ser considerables. La toma de decisiones puede ser rutinaria y bien estructurada. De manera que, en términos generales, hay dos clases de decisiones:

3.1.5.5.1 DECISION PROGRAMADA:

Esta clase de decisión implica una respuesta automática de acuerdo con las políticas previamente establecidas. Todos los problemas repetitivos y rutinarios, con parámetros bien definidos, se prestan a la decisión programada. La identificación de esta clase de decisiones y la aportación de métodos mediante los cuales puedan ser adoptadas siempre que sea posible, son un reto para el analista. Para tomar esta clase de decisiones es preciso haber establecido y definido claramente una regla de decisión. Una vez establecida esta regla, todo se reduce a desarrollar un programa que permita llegar a la decisión de modo rutinario y automático. En muchas empresas se presenta la oportunidad de adoptar la decisión programada, porque gran parte de las decisiones se toman de acuerdo con procedimientos estándar y rutinarios de operación. El beneficio de la decisión programada consiste en que los administradores quedan libres para enfocar sus tareas en otras cosas mas importantes. Una vez implantado, el proceso de toma de decisiones del que se habla es del tipo ilustrado por la fig. 3.2

Fig. 3.2 (Decisión Programada)



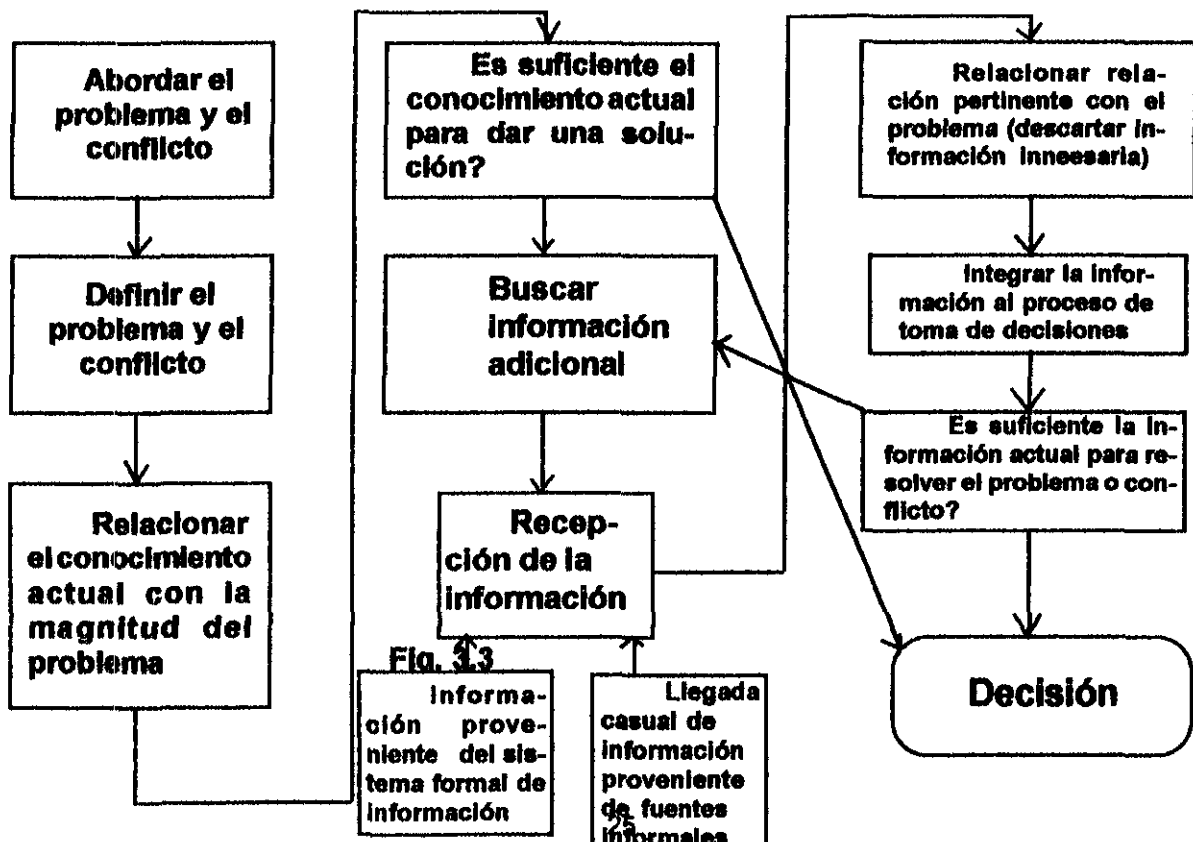
3.1.5.5.2 DECISION NO PROGRAMADA:

Esta clase de decisión representa el proceso de afrontar problemas poco definidos. Generalmente dichos problemas son complejos y sólo se conoce una parte de sus parámetros, y ocurre que muchos de los parámetros conocidos representan un alto grado de probabilidad. La toma

correcta de decisiones no programadas requiere de todo el talento del administrador experimentado, mas el auxilio del sistema de información. La ampliación de la planta, el desarrollo de un nuevo producto, las políticas de procesamiento y publicidad, el manejo de personal, etc. son problemas que exigen la decisión no programada.

Proceso de la toma de decisiones. Las personas que toman decisiones poseen en diversos grados la facultad de asimilar la información, lo cual determina la eficiencia de su procesamiento. Los conocimientos de una persona, unidos a su eficiencia para procesar la información, determina su capacidad de decisión. Frente a varias alternativas, elige un objetivo y trata de alcanzarlo escogiendo, con base a lo que sabe, la mejor alternativa. Si reconoce que sus conocimientos no le permiten comprender el significado de cada alternativa, busca información suficiente, surgirán nuevos problemas y tendrá que redoblar sus esfuerzos. La falta de información proviene de la incapacidad de las fuentes para proporcionar toda la que hace falta, o de la incapacidad de quien toma las decisiones para determinar exactamente sus necesidades. En la figura 3.3 se presenta un diagrama del proceso de la toma de decisiones.

Fig. 3.3 (Proceso Toma de Decisiones)



Las dificultades que se encuentran al tomar las decisiones pueden imputarse a los siguientes factores:

3.1.5.6.1 Información inadecuada. Es decir, información incorrecta o incompleta con respecto a las diferentes alternativas de cursos de acción o con respecto a las diferentes alternativas de curso de acción o con respecto a sus implicaciones en cuanto a los resultados.

3.1.5.6.2. Objetivos incorrectamente especificados; o sea que no se ha indicado claramente cuáles resultados son los más deseables.

3.1.5.7 Niveles de la toma de decisiones. Las decisiones pueden ser desde muy superficiales y rutinarias (programadas) hasta muy complejas, o sea aquellas que producen un efecto significativo en el sistema (no programadas). Para su clasificación se sitúan en tres niveles:

Nivel estratégico. Las decisiones estratégicas se caracterizan por un alto grado de incertidumbre y están orientadas hacia el futuro. Estas decisiones establecen planes a largo plazo que influyen en toda la organización. Se fijan las metas de la empresa y se crea una serie de estrategias que pueden comprender, por ejemplo, la expansión de la planta, la determinación de la gama de productos, etc.

Nivel táctico. Las decisiones tácticas se refieren a las actividades a corto plazo y a la distribución de los recursos para el logro de los objetivos. Esta clase de decisiones comprende las áreas de la formulación de presupuestos, análisis de flujo de recursos, distribución de la planta, problemas de personal etc.

Mientras en la toma de decisiones estratégicas implica que más que nada una actividad de planeación y control y su potencial para decisiones programadas es escaso, si es que existe. Para la toma de decisiones tácticas, la mayoría de las reglas de decisión está mal estructurada y no son adaptables a la rutina y a la autoregulación.

Nivel técnico. A éste nivel de decisión, los estándares son fijos y los resultados son deterministas. La toma de decisiones técnicas es un proceso para asegurarse de que la ejecución de tareas específicas se lleva a cabo de manera efectiva y eficiente. Esta clase de decisiones exige que se impartan órdenes específicas para controlar las operaciones igualmente específicas. La principal función administrativa a la que corresponde este tipo de decisión es la de control; la planeación figura en una escala mas bien limitada.

Información necesaria para la toma de decisiones. Los diferentes niveles de decisión tienen diferentes necesidades de información. Lo anterior se ilustra en la figura 3.4

Fig. 3.4 (Necesidades de Información)



Las líneas de separación no pueden establecerse tan nítidamente como en la figura, ya que, en la práctica, los límites de las categorías son inciertos y tienden a traslaparse. No obstante los analistas deben estar al tanto de esta clase de decisiones y conocer la forma de diseñar el sistema de información con el fin de satisfacer las diversas necesidades. Las características de la información requerida se indican en la figura 3.5

Fig. 3.5 (Información Requerida)

Información estratégica	1. Información externa (a) Acciones de la competencia (b) Acciones de los clientes (c) Disponibilidad de recursos (d) Estudios demográficos (e) Acciones gubernamentales 2. Información predictiva (Tendencias a largo plazo) 3. Información simulada - ¿Que pasa si...?
Información táctica	1. Información descriptiva-histórica 2. Información sobre el rendimiento presente 3. Información predictiva-futuro (corto plazo) 4. Información simulada-¿Que pasa si...?
Información técnica	1. Información descriptiva-histórica 2. Información sobre el rendimiento presente

3.1.6 EL METODO COMPUTACIONAL

3.1.6.1 EXPOSICION GENERAL DE LOS SISTEMAS

La palabra "sistemas" es un término oblicuo que sirve para describir un gran número de cosas, objetos, métodos, o grupos. Un grupo ordenado de hechos, principios doctrinas, creencias, y tradiciones; por ejemplo, el sistema de gobierno democrático, se describe como un sistema. El sistema respiratorio de un ser viviente es un conjunto de órganos o partes que desempeñan una o mas funciones vitales. Este sistema mantiene vivo al ser, el que a su vez puede ser considerado como un sistema. Un equipo de cómputo, que comprende cierto número de impresoras, scanners, unidades de cinta magnéticos, etc. a menudo reciben nombre de sistema, porque representa un grupo de objetos relacionados entre sí que forman una red unificada.

Es sumamente necesario entender los sistemas y trabajar con ellos en forma inteligente. Por esta razón es preciso aplicar el enfoque de sistemas para conocer las características significativas de uno de ellos, las que a su vez sugerirían los cambios que pueden ser introducidos para mejorarlo. Además, la aplicación del enfoque de sistemas implica que los componentes del sistema se distribuirán e integrarán de manera que su efectividad general sea óptima. Es decir, la aplicación del enfoque de sistemas que permite lograr un efecto sinérgico, de manera que la acción unificada de las diferentes partes del sistema produzca un efecto mayor que la suma de dichas partes.

3.1.6.2 CLASIFICACION DE LOS SISTEMAS

Un modo conceptual de clasificar los sistemas se basa en dos criterios distintos, que son los siguientes:

Nivel de predictibilidad. Este criterio está basado en un doble esquema: Determinista y Probabilista.

Nivel de complejidad. Adoptando éste criterio, es posible clasificar los sistemas en tres categorías: sencillos, complejos y sumamente complejos.

Cuando es posible pronosticar con certeza la manera en que responderá un sistema, o cómo funcionará en circunstancias diferentes, decimos que el sistema es **determinista**. No hay ninguna duda sobre los resultados. Dado el estado último del sistema y la cantidad de información actualmente presente y determinada, será posible predecir, sin riesgo de equivocarse el estado subsecuente del sistema. A la inversa, cuando no es posible pronosticar con certeza los resultados de un sistema en particular, dicho sistema es **probabilista**. Podría aducirse que un sistema probabilista es, a fin de cuentas, un sistema determinista no logramos entender del todo. Esta observación encierra un elemento de verdad, como quiera que sea, todos los administradores y analistas tienen que enfrentarse a los sistemas probabilistas.

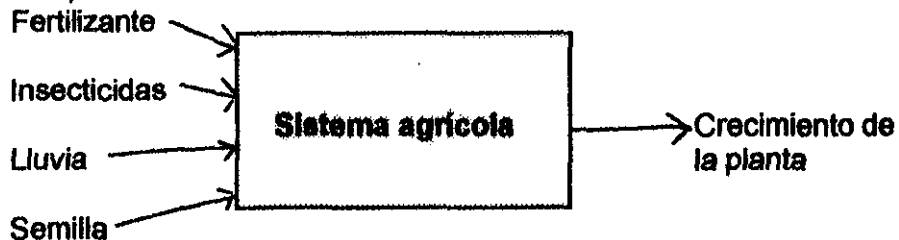
Un sistema **determinista** sencillo es aquel que contiene pocos subsistemas e interrelaciones y revela un comportamiento enteramente predecible. Unos balines de acero colocados sobre una pista (como las máquinas de juego de canicas) dan ejemplo de un sistema **determinista sencillo**. Sin embargo, una vez que los balines salen de la pista, sus caminos se vuelven **probabilistas**. Una hilera de producción se puede estudiar y disponer de manera que se reduzca al mínimo la distancia que recorran los materiales. Cuando se hace necesario estudiar lo que ocurre realmente cuando los materiales comienzan a fluir, el sistema se toma inmediatamente **probabilista**. Las mismas consideraciones son aplicables cuando estudiamos un sistema **determinista complejo**. La computadora es un sistema complejo; pero es determinista en el sentido de que hará aquello para lo cual fue programada. Un almacén automatizado es determinista. Cuando el comportamiento de un sistema determinista complejo, significa simplemente que ha habido alguna descompostura (una falla).

Un sistema sencillo también puede ser probabilista. Por ejemplo, el lanzar al aire una moneda es un sistema sencillo; pero a la vez es notoriamente probabilista. Un sistema de control de calidad que pronostica el número de piezas defectuosas en un conjunto, es un sistema probabilista.

3.1.6.3 ANALISIS DE CAUSA Y EFECTO.

Los sistemas se pueden analizar en términos de entradas y salidas. Los elementos que entran se consideran las causas que interactúan para producir una salida; la salida constituye el efecto. Esta relación de causa y efecto se ilustra en la figura 3.6. El sistema actúa como una caja de funciones que cambian los elementos de entrada en resultados o elementos de salida.

Fig. 3.6 (Causa Efecto)



En un sistema agrícola, la combinación de los elementos de entrada, es decir, fertilizante, insecticida, lluvia y semillas, hace que éstos interactúen de manera complicada para producir un efecto o elemento de salida. En este caso, la salida consiste en el crecimiento de una planta.

3.1.6.3.1 ANALISIS DETERMINISTA Y PROBABILISTA.

Un ejemplo clásico de un sistema **determinista sencillo** lo tenemos en un sistema de calefacción. En la mayoría de estos sistemas, la decisión de poner a funcionar la calefacción, o de cerrarla, está bien definida y es cosa de rutina. Cuando la temperatura del local ha alcanzado el nivel indicado por la posición del termostato, éste cierra la calefacción; cuando la temperatura desciende, la pone nuevamente a funcionar. El termostato actúa a la vez como mediador a la salida y como controlador a la entrada. Sus contactos cierran un circuito que activa a la fuente productora de calor. El calor producido por éste último es retroalimentado al termostato a través del aire, el cual, al calentar el elemento metálico de aquél, abre nuevamente los contactos. El aire calentado enlaza al productor de calor y al termostato en un circuito cerrado de retroalimentación relacionando la acción de uno y otro en una cadena sin fin de causa y efecto.

CAPITULO 4:

Dado el objetivo general de este trabajo, que es que en la clínica de odontología U.P.A.E.P. se controle eficientemente la información, de modo que sea suficiente y oportuna para tomar decisiones con más certidumbre y en menos tiempo, que se fortalezcan otras áreas de la administración y se brinde un mejor servicio tanto a los pacientes como a los alumnos, además de que se controlen los ingresos y egresos de efectivo, así como los recursos materiales e insumos. En este capítulo se presentan los hechos que dieron origen a este trabajo, así como la descripción del análisis efectuado para determinar los requerimientos generales del sistema que sustenta este trabajo.

4.1 ANALISIS DE LA PROBLEMÁTICA

4.1.1 REQUERIMIENTOS DE INFORMACION

En general, la información manejada en la clínica gira en torno a las siguientes entidades o elementos:

- **Los pacientes** (sus datos, tratamientos y estadísticas)
- **Los alumnos** (Sus datos, calificaciones y pacientes)
- **Los catedráticos** (Sus datos, horarios y alumnos)
- **Los tratamientos** (Sus estadísticas)
- **Los ingresos y egresos** (Su registro, control y reporte)
- **Los materiales e insumos** (Su registro, control y reporte)
- **Las instalaciones** (Su uso, control y estadísticas).

4.1.2 ENTIDADES RELACIONADAS

Debido a la estructura administrativa de la universidad y de la clínica misma, se involucra la participación de una u otra forma de las siguientes entidades y/o departamentos:

Administrativas:

**Rectoría
Contraloría
Tesorería**

Escolares:

**Escuela de odontología
Departamento escolar**

4.1.2.1 RECTORIA

Todas las actividades desarrolladas en la universidad se involucran directamente con la rectoría, quien se encarga de controlar y vigilar que cada una de ellas cumpla con el objetivo de la universidad.

La clínica debe entregar un reporte de actividades y desempeño a la rectoría para que ésta pueda tomar decisiones acerca de los recursos asignados a la clínica y evalúe el aprovechamiento de los mismos, este reporte también es útil para que se integre a todos los reportes de servicios prestados dentro de la universidad, para que exista una visión clara de lo que la universidad ofrece a la sociedad y a la comunidad universitaria.

La rectoría define o aclara los objetivos y planes de la clínica respetando su autonomía y administración propias.

4.1.2.2 CONTRALORIA Y TESORERIA

Los reportes proporcionados a estos departamentos están profundamente relacionados e incluso son similares, pues en general, dan una idea de cómo se están administrando los recursos materiales y económicos de la clínica, así como dan base y fundamentan las peticiones presupuestales de la clínica y escuela de odontología.

Dentro de los reportes hechos para estos departamentos se incluyen algunos periódicos (sobre control de efectivo e inventario) y otros esporádicos (manejo de caja chica, facturas pagadas y por pagar, etc.).

4.1.2.3 ESCUELA DE ODONTOLOGIA

La clínica de odontología debe coordinar sus actividades con la escuela de odontología, pues su fin básico es la formación de profesionales en el área de estomatología.

Los reportes a la escuela son periódicos y expresan primordialmente las estadísticas de tratamientos y uso de las instalaciones; otro tipo de reportes sintetizan el desempeño de los alumnos dentro de la clínica por medio de listas de calificaciones y estadísticas de prácticas y tratamientos.

La escuela de odontología, por su parte, determina las materias que se impartirán durante el semestre, autoriza la planta de catedráticos y coordina los horarios de clase con los de prácticas.

4.1.2.4 DEPARTAMENTO ESCOLAR

La clínica reporta periódicamente las calificaciones de los alumnos por medio de la escuela de odontología, de acuerdo a las políticas y estructura de la universidad.

4.1.2.5 ENTREVISTAS CON USUARIOS

Una fase importante dentro del desarrollo de un sistema de control en el área administrativa y de control es la de entrevistas con los futuros usuarios, se realiza esta importancia cuando el sistema a desarrollar implicara el uso de equipos de cómputo, pues los usuarios serán quienes aprovecharán las características del sistema y, por otra parte, nadie conoce mejor las necesidades de información y problemas dentro del medio ambiente del sistema.

Los usuarios del sistema de automatización de la clínica serán los siguientes:

- 1. DIRECCION DE LA CLINICA**
- 2. PERSONAL ADMINISTRATIVO**
- 3. CATEDRATICOS**
- 4. ALUMNOS**

4.1.2.5.1 DIRECCION DE LA CLINICA

Involucra al director de la escuela de odontología y al director de la clínica, el resultado de las entrevistas con estos usuarios arroja los siguientes resultados:

-Servicios necesarios:	Reportes y consultas
-Información necesaria:	Efectivo, calificaciones, uso del equipo e instalaciones, casuística.
-Información extra:	Tipos de población, servicios especiales
-Información generada:	Políticas, materias, acciones a seguir.
-Periodicidad de la información:	Alguna periódica y otra esporádica.
-Importancia de la información:	Crítica
-Presentación deseable:	Resúmenes e información gráfica de alta calidad
-Experiencia en el uso de computadoras:	Suficiente
-Resistencia a la implementación del sistema:	Nula

4.1.2.5.2 PERSONAL ADMINISTRATIVO

Involucra al personal de las áreas de recepción, caja y central de equipos, el resultado de las entrevistas con estos usuarios arroja los siguientes resultados:

-Servicios necesarios:	Captura de datos, resumen de los datos, reportes en general, modificación de información.
-Información necesaria:	Efectivo, calificaciones, horarios.
-Información extra:	Materiales y equipos.
-Información generada:	Datos en general.
-Periodicidad de la información:	Diaria, mensual y semestral.
-Importancia de la información:	Media
-Presentación deseable:	Por lo general información detallada, algunos resúmenes e información gráfica.
-Experiencia en el uso de computadoras:	No suficiente
-Resistencia a la implementación del sistema:	Nula

4.1.2.5.3 CATEDRATICOS

Involucra a los responsables de área y catedráticos en general, el resultado de las entrevistas con estos usuarios arroja los siguientes resultados:

-Servicios necesarios:	Reportes y consultas
-Información necesaria:	Calificaciones, horarios, expedientes.
-Información extra:	Tipos de población
-Información generada:	Calificaciones, artículos de revista
-Periodicidad de la información:	Mensual, esporádica
-Importancia de la información:	Media
-Presentación deseable:	Resúmenes e información gráfica de alta calidad
-Experiencia en el uso de computadoras:	Variable
-Resistencia a la implementación del sistema:	Generalmente nula.

4.1.2.5.4 ALUMNOS

El resultado de las entrevistas con estos usuarios arroja los siguientes resultados:

-Servicios necesarios:	Consultas
-Información necesaria:	Calificaciones, horarios, artículos de revista, expedientes.
-Información extra:	Pacientes, casuística.
-Información generada:	Tratamientos efectuados..
-Periodicidad de la información:	Esporádica.
-Importancia de la información:	Media y baja.
-Presentación deseable:	Completa.
-Experiencia en el uso de computadoras:	Variable
-Resistencia a la implementación del sistema:	Nula

CAPITULO 5:

Dado el objetivo general de este trabajo, que es que en la clínica de odontología U.P.A.E.P. se controle eficientemente la información, de modo que sea suficiente y oportuna para tomar decisiones con más certidumbre y en menos tiempo, que se fortalezcan otras áreas de la administración y se brinde un mejor servicio tanto a los pacientes como a los alumnos, además de que se controlen los ingresos y egresos de efectivo, así como los recursos materiales e insumos. En este capítulo se resumen los puntos generales y características del modelo propuesto de solución para la problemática presentada. Este modelo sirvió de guía para elaborar el presente trabajo.

5. MODELO PROPUESTO DE SOLUCION

Para el apoyo en la eficientización del control y administración de la clínica de odontología es necesario implementar un sistema computacional que apoye en las siguientes áreas:

- 5.1 - CONTABLE**
- 5.2 - ADMINISTRATIVA**
- 5.3 - CLINICA**
- 5.4 - ACADEMICA**

El área **CONTABLE** estará a cargo, en sus funciones básicas (cobros, emisión de comprobantes, cortes de caja, control de inventarios, etc.) del personal administrativo de la clínica en el área de caja/recepción, se integra al sistema de cómputo la capacidad de registrar estas operaciones de manera confiable y ser verificadas o corregidas por el personal directivo y el departamento de contraloría y tesorería, también cuenta con la capacidad de elaborar reportes de alta precisión y calidad, así como gráficas comparativas y estadísticas, como apoyo a las decisiones de dirección y control.

El área **ADMINISTRATIVA** comprende el registro y control de los expedientes de los pacientes, de manera precisa y confiable, el control de los horarios de atención, guardias, uso de equipo, insumos e instalaciones, etc. Así como el registro de frecuencia de casos, eficiencia en la atención, etc. Todo lo anterior ilustrado en reportes y gráficas adaptables de frecuencia variable. La función de evaluación se relaciona comúnmente con el área administrativa, pero que debe considerarse en el plano académico, ya que su principal objetivo es dotar al alumno de la capacidad de reflexión y retroalimentación, de manera que fomente su autoestima y deseo de superación, así que se describe dentro de dicha área.

El área **CLINICA** contempla el seguimiento de cada tratamiento por paciente, clínica y alumno, de manera que cada alumno al recibir a un paciente puede saber de manera segura y exacta el avance del tratamiento, la respuesta del paciente y el siguiente paso en el procedimiento normal predefinido, de la misma manera, en caso de emergencias, consultas extemporáneas o cualquier otra contingencia, se puede tener la información del paciente y su tratamiento.

Se ha prestado especial atención al área **ACADEMICA** del sistema, tomando en cuenta que uno de los objetivos básicos de la clínica es colaborar en la formación de odontólogos correctamente capacitados. En esta área se incluyen dos aspectos fundamentales: La formación y la evaluación.

La parte **formativa** incluye un sistema de consulta de procedimientos, en la cual el alumno podrá tener acceso a la información adecuada sobre como realizar algún tratamiento o procedimiento de los incluidos como servicio de la clínica, de la misma manera puede consultar un banco de datos que se presentará de manera similar al de una revista médica, esto es, se incluirá en este cualquier artículo que pudiera ser de interés, así como avisos importantes y casos poco usuales, esta información será proporcionada por catedráticos y alumnos, y contará con ilustraciones gráficas relacionadas y, cuando lo amerite, video digitalizado, que ayuden a comprender mejor la información que se desea brindar.

La **evaluación** contempla el registro de calificaciones de los alumnos, el promedio general por grupos, por clínicas o por individuos, de manera que se pueda consultar esa evaluación en cualquier momento y se pueda elaborar un reporte inmediato y confiable a la dirección de la escuela o al departamento escolar. Los catedráticos podrán incluir comentarios de retroalimentación hacia esas evaluaciones, de manera que se cumpla con el fin de mejoramiento continuo.

5.5 FUNCIONAMIENTO EN RED

5.5.1 PLANTEAMIENTO TEORICO DE REDES.

5.5.1.1 ¿PARA QUE SIRVEN LAS REDES?

Durante la última década, las computadoras y las redes de computadoras han producido un impacto en nuestra sociedad de enormes consecuencias. Nosotros decimos que estamos en la época de la informática. Lo que hay de cierto en esto es que, estas revolucionarias herramientas han multiplicado en mucho la productividad y la eficiencia del trabajo, tanto para las empresas como para los usuarios individuales.

Día a día, infinidad de usuarios acuden a redes de computadoras para atender sus necesidades privadas y comerciales y éstas tendencias aumentan a medida que las empresas y los usuarios van descubriendo la potencia de estos medios. En nuestros días las computadoras se ocupan de registrar las transacciones que tienen lugar día a día en un almacén, en las operaciones bancarias, reservaciones de hoteles y reservaciones de vuelos, es por eso que ahora se incluye una red para lograr un mejor desempeño de la clínica de odontología de la UPAEP. Pero no debemos olvidar que la información depende por igual de las computadoras y de las redes que las comunican entre sí.

5.5.1.2 ¿QUE ES UNA RED DE COMPUTADORAS?

Son varias las definiciones que son aceptadas para explicarlo, pero la que es probablemente la más sencilla de todas para poderlo comprender correctamente es la siguiente: Es un grupo de computadoras (y terminales en general) interconectados entre sí a través de uno o varios caminos o medios de transmisión. Las redes tienen la finalidad concreta de transferir e intercambiar información entre dos o más computadoras, esto es lo que nosotros comúnmente llamamos como LAN's (que por sus siglas en inglés significa Local Area Network).

5.5.1.3 VENTAJAS DE LAS REDES

Las redes de computadoras presentan varias ventajas importantes de cara a los usuarios, ya sean empresas o particulares, a continuación se presentan varias de ellas:

- Posibilidad de compartir con gran facilidad información, los programas y recursos del servidor (o servidores).
- Facilidad en las labores de automatización de las labores de respaldo.
- Mejor integración del personal al utilizar programas especiales para red local, enfocados al trabajo en grupo.
- Posibilidad de trabajar con archivos y bases de datos realmente grandes (Incluso de algunos gigabytes).
- Conectividad mayor entre las micros y las computadoras mayores para lo cual se requiere el uso de los llamados gateways.
- Mayor seguridad en las PC's individuales, equivalente al de una mini o un mainframe (y a veces con mejores resultados).

El empleo de redes confiere una gran flexibilidad a los entornos laborales, los empleados pueden laborar desde su casa, utilizando terminales conectados con la computadora central de su trabajo. Hoy día es frecuente observar personas que viajan con su computadora portátil y la conectan a la computadora de su oficina por medio de cualquier línea telefónica.

El empleo del nombre de "era de la informática" es bastante adecuado. La sociedad de nuestros días emplea la información para reducir los costes de producción de los bienes que se consumen, y en general para mejorar la calidad de vida de las personas.

5.5.2 TOPOLOGIAS DE RED

La configuración física de una red suele conocerse como **TOPOLOGIA** de la misma. El término topología es un concepto geométrico con el que se alude al aspecto de una cosa, cuando se tiene que establecer la topología de una red, la persona que ha de diseñarla se tiene que plantear tres objetivos principales:

- Proporcionar la máxima confiabilidad posible, para garantizar la transmisión correcta de los datos que transitan por la red.
- Encaminar el tráfico entre el ETD (Equipo Terminal de Datos) transmisor y el receptor a través del camino más económico dentro de la red.
- Proporcionar al usuario final un tiempo de respuesta óptimo y un caudal eficaz máximo.

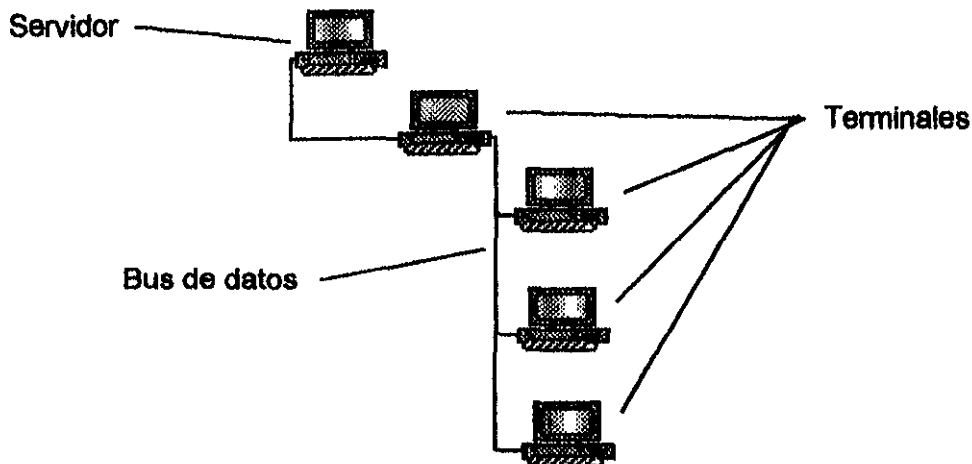
Cuando se habla de la fiabilidad de una red se está refiriendo a la capacidad que tiene la misma para transportar los datos correctamente sin errores de un ETD a otro. Ello también incluye la capacidad de recuperación de errores o datos perdidos en la red, ya sea por fallo del canal, del ETD, del ETDC (Equipo Terminal de Circuitos de Datos), o del ECD (Equipo de Conmutación de Datos).

5.5.2.1 TOPOLOGIAS MAS COMUNES

5.5.2.2 TOPOLOGIA JERARQUICA

La estructura jerárquica es una de las más extendidas en la actualidad, el software que utiliza es relativamente sencillo y su topología proporciona un punto de concentración de las tareas de control, en la mayoría de los casos el ETD situado en el nivel más alto de la jerarquías, osea es el que controla la red.

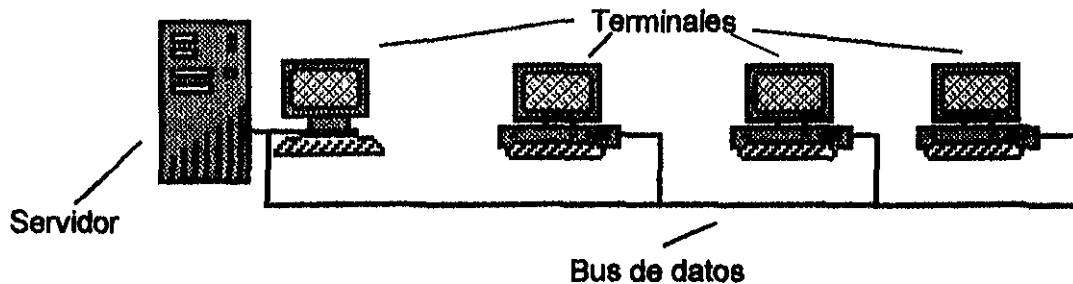
Fig. 5.1 (Jerárquica)



5.5.2.2 TOPOLOGIA HORIZONTAL (BUS LINEAL)

Esta estructura es frecuente en las redes de área local, es relativamente fácil de controlar el flujo de datos entre los distintos ETD, ya que el bus permite que todas las estaciones reciban todas las transmisiones, es decir, una estación puede difundir la información a todas las demás. Una de las limitantes que tiene este tipo de topología es que sólo cuenta con un canal de comunicaciones entre las computadoras, por lo tanto, si éste falla toda la red deja de funcionar.

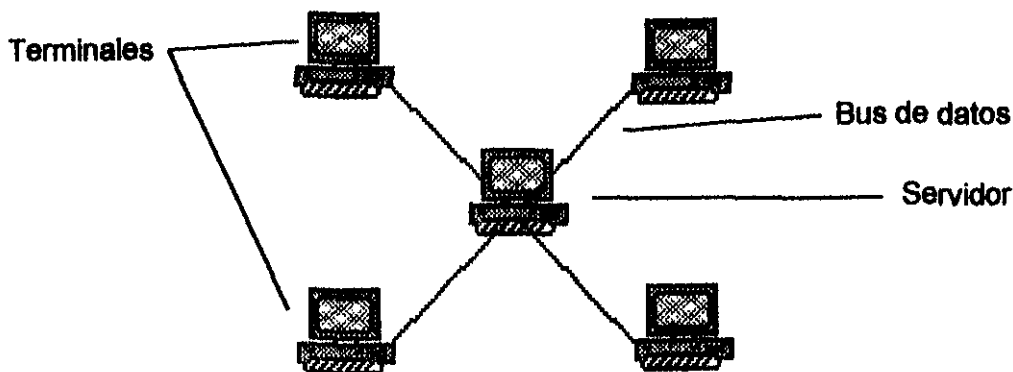
Fig. 5.2 (Bus Lineal)



5.5.2.3 TOPOLOGIA DE ESTRELLA

Esta topología es de las más empleadas dentro de los sistemas de comunicación de datos, su software es sencillo al igual que su flujo de tráfico, todo el tráfico emana del núcleo de la estrella que es el nodo central de la red el cual posee el control total de los ETD. El nodo central es el responsable de encaminar el tráfico hacia el resto de los componentes, en caso de una avería es fácil de localizar ya que es posible aislar las líneas para identificar rápidamente el problema.

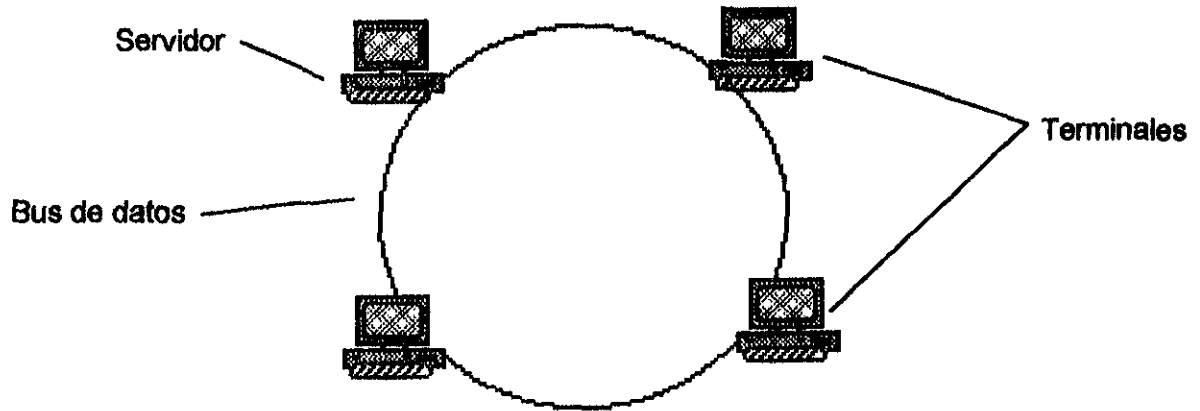
Fig. 5.3 (Estrella)



5.5.2.4 TOPOLOGIA DE ANILLO

Esta topología es fácil de identificar por el flujo de datos en esta forma, en la mayoría de los casos la información fluye en una sola dirección, y cada estación recibe la señal y la retransmite a la siguiente estación, la organización en esta topología resulta atractiva ya que en muy rara ocasión ocurren embotellamientos de datos que ocurren en forma tan frecuente en las topología jerárquica y de estrella.

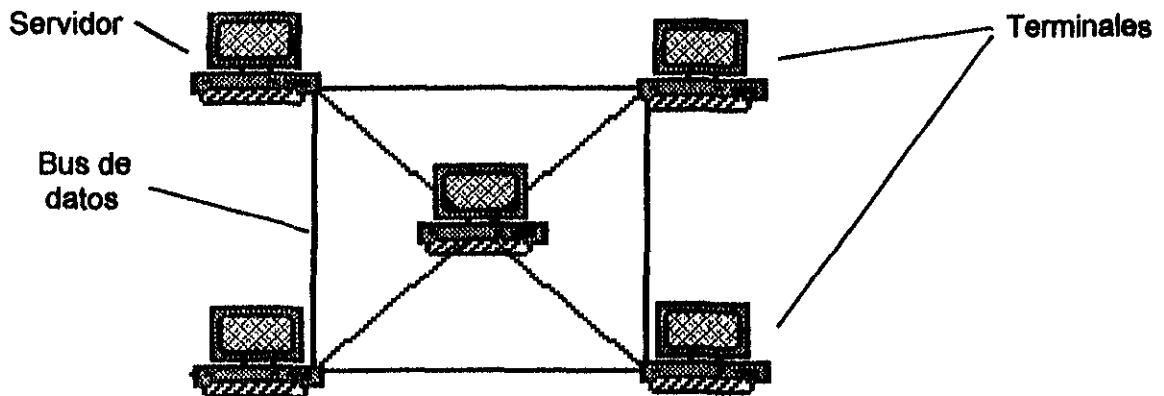
Fig. 5.4 (Anillo)



5.5.2.5 TOPOLOGIA DE MALLA

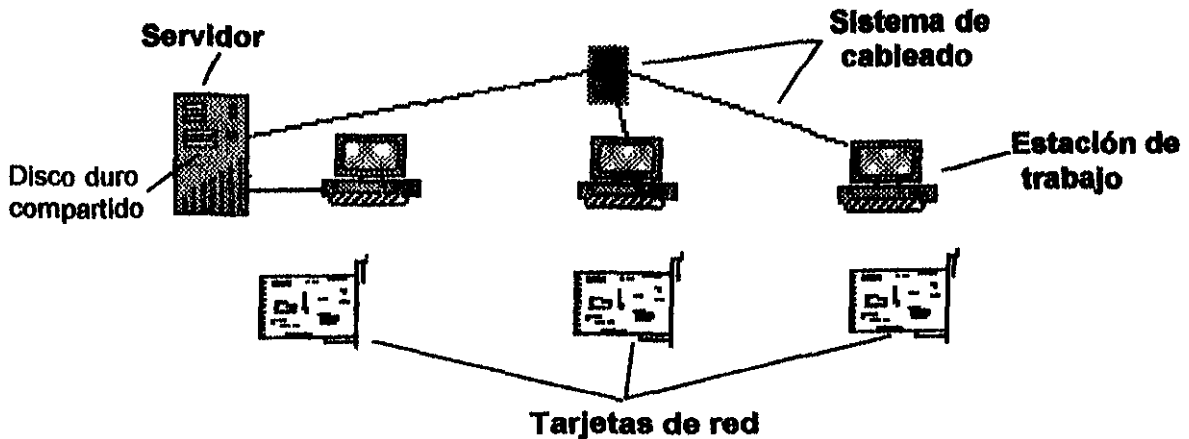
Lo que hace atractiva la utilización de este tipo de topología es su relativa inmunidad a los problemas de embotellamiento y averías, gracias a la multiplicidad de caminos que ofrece a través de los distintos ETD y ECD, es posible orientar el tráfico de datos por trayectorias alternativas en caso de que exista una avería o esté algún nodo ocupado.

Fig. 5.5 (Malla)



5.5.3 COMPONENTES DE UNA RED LOCAL

La figura muestra las partes principales que conforman una red local



5.5.3.1 ESTACIONES DE TRABAJO:

Cada micro que se conecta a la red por medio de su tarjeta, software y todos los componentes de una red, para poder usarla como una PC individual, y también se denomina como una estación de trabajo. Estas estaciones de trabajo pueden ser "tontas" (sin procesador propio) o "inteligentes" (con procesador propio).

5.5.3.2 SERVIDOR:

Cuando una computadora es capaz de compartir alguno(s) de sus recursos, se denomina servidor. Por lo común, el servidor comparte los archivos y programas que se encuentran en su(s) disco(s) duros y se denomina servidor de archivos. También el servidor puede compartir sus demás periféricos. Si el servidor puede ser utilizado como otra estación de trabajo será un servidor no dedicado, en contraposición a lo que se llama servidor dedicado.

5.5.3.3 SISTEMA DE CABLEADO:

Para que todas las computadoras estén unidas, se utiliza algún tipo de cable o todo un sistema de cableado, que puede incluir algunos dispositivos especiales, los cables pueden ser de diversos tipos y componentes, algunos de los más comunes son los cables coaxiales, triaxiales, fibra óptica, algunas veces se utiliza rayo láser que funciona como cable, etc.

5.5.3.4 TARJETAS DE RED:

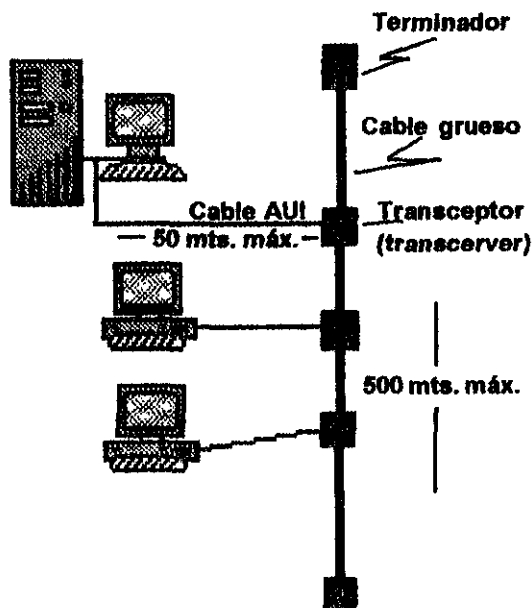
Regularmente las computadoras no traen de fábrica un puerto especial para red local, por lo que es necesario agregarle una tarjeta de red que se conecte al sistema de cableado. Dicha tarjeta se inserta en algunas de las ranuras libres del equipo y dependiendo del tipo de computadora pueden ser para una PC, AT, Microcanal o para EISA.

Tres estándares dominan en las tarjetas de red:

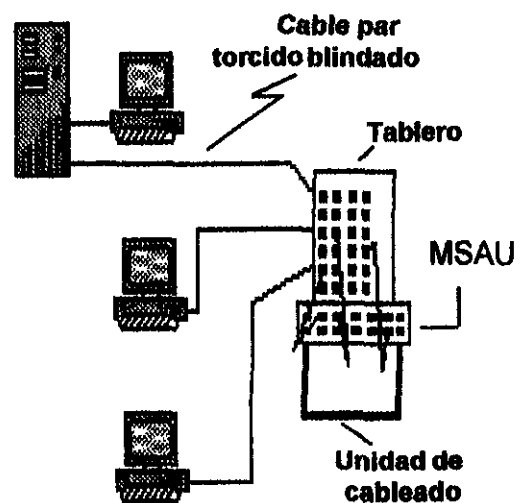
- Ethernet
- Token Ring
- Arcnet

Sin duda alguna Ethernet es la tarjeta de más abolengo, la más utilizada en el mundo (cerca del 50% del mercado) y la que permite mejor integración entre todo tipo de equipos: PC's, minis, mainframes y hasta algunos equipos grandes de IBM.

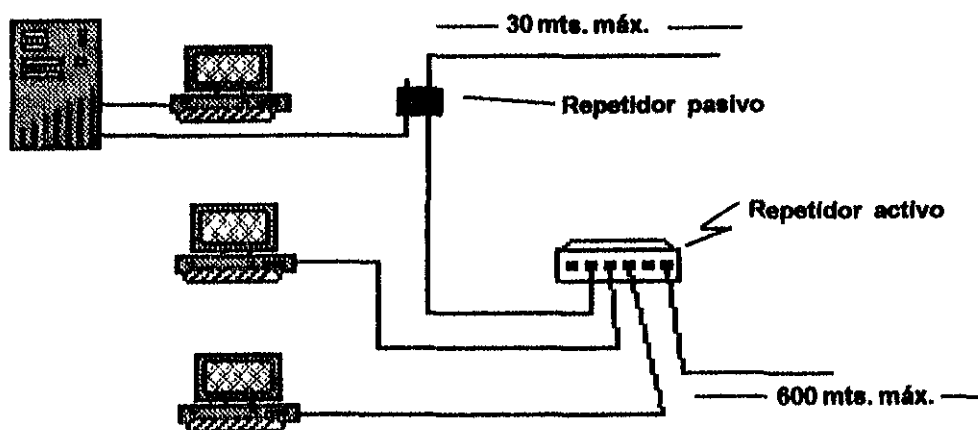
En las siguientes figuras se resumen las principales características de cada una de ellas:



Red tipo Ethernet



Red tipo Token-Ring



5.5.3.5 SISTEMA OPERATIVO DE RED:

Dado que el sistema operativo MS-DOS no cuenta con las características para poder trabajar con una red de área local, se requiere de un sistema operativo que lo complete para trabajar en una red. Por lo regular este sistema operativo tiene dos formas de instalarse, una para estación de trabajo y otra para el servidor.

Hasta aquí, lo más importante que el usuario percibe de la red son dos cosas: Además de sus unidades de discos (flexibles y duro) el usuario se encuentra en posibilidades de utilizar las unidades de discos duros del servidor, como si fueran las propias.

Así como el disco duro es la unidad C, el servidor puede verse como las unidades F y G, de igual forma el usuario puede manejar la impresora del servidor desde los mismos programas comunes, sin importar que además cuente con una impresora. En este caso, cuando el usuario ordena imprimir su trabajo, se envía al disco del servidor para "formarse" en una cola de impresión en caso de que existan mas impresiones por realizarse al mismo tiempo que el usuario sigue usando su terminal como si ya hubiera impreso su trabajo, en caso de que la impresora estuviera desocupada la impresión saldría de inmediato.

5.5.3.6 SISTEMAS OPERATIVOS:

A pesar de existir varios sistemas operativos, son dos los que predominan en el mercado:

- **NETWARE de NOVELL**
- **LAN-Manager de MICROSOFT.**

Cuatro versiones fundamentales tiene Netware, las dos primeras para redes pequeñas y las otras dos para redes mas poderosas que llegan hasta cientos de computadoras dentro de la misma red.

Netware se caracteriza desde sus inicios por ser un producto de excelente calidad y velocidad, compatibilidad con muchos tipos de tarjeta de red, alta seguridad y arquitectura propietaria. Pero en nuestros días se habla de estándares y arquitecturas abiertas, basadas en sistemas operativos no propietarios y esto lleva a Novell a tener que aceptar estos nuevos estándares.

Por otra parte LAN-Manager desarrollado por Microsoft, ha tomado definiciones distintas aunque al final de cuentas ambos productos llevan el mismo rumbo.

5.5.3.7 SOFTWARE DE LA LAN:

Se ha dicho anteriormente que una red permite, en forma simple, compartir los recursos con los que cuente la red, en caso de que los recursos sean bases de datos, es necesario que el sistema operativo y los demás sistemas están preparados para que dichos datos sean siempre consistentes, y puedan trabajar correctamente cuando varios usuarios accedan las bases de datos al mismo tiempo.

Por fortuna, casi todos los paquetes y software del mercado están totalmente diseñados para trabajar en red y de hecho existen versiones especiales para tal efecto.

5.5.4. LA MEJOR RED LOCAL:

No es sencillo diagnosticar sin ver al enfermo, lo mismo sucede al querer instalar una nueva red, se tienen que saber con toda certeza cuáles son las necesidades reales de cada usuario ya que cada usuario tiene sus propias necesidades, para ello se proponen los siguientes consejos para la mayoría de los casos en que se tenga que instalar una nueva red.

- Definir primero la aplicaciones que se desean que operen en la futura red así como sus prioridades (el manejo de las bases de datos o documentos de procesador de palabras, por ejemplo), volumen de información y ubicaciones actuales y futuras de las computadoras.

- Si existe alguna tarea particularmente crítica, o alguno de los valores del punto anterior es extremo (grandes volúmenes de información o grandes distancias) entonces es preferible realizar un estudio a fondo para la instalación de la nueva red.

CAPITULO 6:

Dado el objetivo general de este trabajo, que es que en la clínica de odontología U.P.A.E.P. se controle eficientemente la información, de modo que sea suficiente y oportuna para tomar decisiones con más certidumbre y en menos tiempo, que se fortalezcan otras áreas de la administración y se brinde un mejor servicio tanto a los pacientes como a los alumnos, además de que se controlen los ingresos y egresos de efectivo, así como los recursos materiales e insumos. En este capítulo se expone de manera somera las técnicas y teorías involucradas en la programación orientada a objetos, puesto que el sistema computacional desarrollado durante el presente trabajo está basado en ellas con el fin de facilitar su creación.

6. PROGRAMACION ORIENTADA A OBJETOS

6.1 INTRODUCCION

El mundo está compuesto de objetos. Los objetos nuevos y las clases de objetos son creados por un proceso de conocimiento y diseño. Existiendo objetos y pudiendo estar entendiendo a través de una práctica colateral, de práctica conocimiento y análisis, en el cual los principios científicos son usados para analizar el mundo. Estos dos procesos, "Análisis" y "Diseño", hacen más en el campo de la ingeniería, la cual es generalmente definida para ser la aplicación de principios científicos en problemas de tipo práctico.

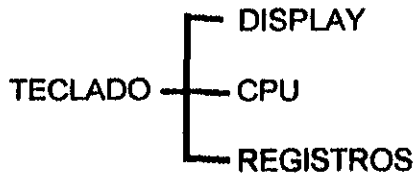
Hay muchas perspectivas en la representación de objetos en el mundo, algunos objetos requieren solamente representación de estructura, mientras que otros requieren de representación de estructura y conducta. Una piedra o pintura puede ser un ejemplo en el cual la conducta de el objeto es incidente a su estructura. Muchos objetos requieren ambas estructuras y especificaciones de conducta.

La programación orientada a objetos es una disciplina prácticamente nueva en la ingeniería de software.

Es una manera diferente de planear, conceptualizar, diseñar, escribir y hacer mantenimiento de software.

Los lenguajes de alto nivel representan una interfase inteligente entre el programador y la computadora, una evolución paralela entre las técnicas de la ingeniería de software y los métodos, esto es motivado por la decisión de crear un software re-utilizable, lo que posibilitaría la reducción de ciclos y de tiempo en el desarrollo de software.

Fig. 6.1 (Calculadora de bolsillo)



La figura muestra un esquema de los principales componentes de una calculadora: El teclado, la pantalla, el CPU y los registros internos. La misma figura también divide los componentes de la computadora dentro de dos campos: el teclado, el cual provee de funcionalidad y las otras partes, las cuales definen el estado de la calculadora. El teclado provee la funcionalidad de correr a la calculadora, de este modo afectando su estado. El estado de la máquina incluye si está encendida o apagada, en suma, mientras la calculadora está encendida, el contenido de la pantalla y los registros internos son afectados por las teclas que son pulsadas, advirtiéndole que el número de teclas es tan grande como el número de estados que la calculadora puede poseer (desde un número hasta el máximo de combinaciones que pueden aparecer en la pantalla). La funcionalidad está representada por la colocación de las teclas. Cuando uno presiona una tecla de la calculadora se envía un impulso eléctrico al CPU lo que causa que realice una tarea que altere el estado de la calculadora. La programación orientada a objetos observa el proceso de presionar la tecla como mandar el mensaje a el CPU el cual es el cerebro de la calculadora.

UN MENSAJE es una solicitud enviada a un objeto para llamar alguno de sus métodos.

UN METODO es un procedimiento o función que está contenido en un objeto, un método específica como un mensaje es ejecutado.

Un lenguaje basado en objetos mejora la programación en el sentido de que proporciona herramientas que reflejan con mayor exactitud la forma en que concebimos el mundo. Con la programación basada en objetos, es posible reproducir la complejidad de los sistemas del mundo real y desarrollar aplicaciones a gran escala de la forma más eficaz posible.

6.1.1 ENFOQUE POR PROCEDIMIENTOS

En la programación por procedimientos, son las funciones las que dictan la organización del programa. Por ejemplo, suponga que trabajamos con dos tipos de animales: caballos y delfines. El proceso que hace que un caballo salte es bastante diferente del que hace que salte un delfín. en un lenguaje por procedimientos la función "saltar" sólo se codificaría una vez, de forma similar a:

```

saltar (elanimal)
  if elanimal = "caballo" then
    correnendireccion= adelante
    patasdelanteras(elevar)
    patastraseras(empujar)
  else if elanimal = "delfin" then
    nadarendireccion= arriba
    velocidadnado= rápida
    cola(aletazo)
  else
    message("la función ha recibido datos erróneos")
  endif

```

En este ejemplo, una misma función debe gestionar varios tipos de datos diferentes. La toma de decisiones se lleva a cabo dentro de la función, y su eficacia depende en gran medida de la capacidad del programador para prever la cantidad de tipos de datos entre los que la función deberá elegir. A medida que el programa evoluciona, la función "saltar" se utilizará para una cantidad cada vez mayor de animales, algunos de los cuales habrá previsto el programador, pero otros no. En un lenguaje por procedimientos, se escribe una sola función para realizar muchas tareas diferentes.

6.1.2 ENFOQUE POR OBJETOS

En un lenguaje basado en objetos, se deja que el procesador de lenguaje decida "el método" que debe utilizarse según el tipo de objetos que llama el método: el código de programación subyacente está escrito para los objetos de forma específica. En un contexto de programación basada en objetos el método general se llama "saltar", pero habría un método para caballos y otro para delfines. De esta forma, habría que especificar el animal que se desea que salte.

Por ejemplo:
 caballo.saltar() o bien delfin.saltar()

Es así de simple, sólo debe recordarse el animal con que se está trabajando y lo que se desea que haga. El lenguaje se encarga del resto.

Es conveniente apuntar que no todos los métodos resultan adecuados para todos los tipos de objetos. No tendría mucho sentido por ejemplo, escribir "silla.saltar()".

6.2 ANALISIS Y DISEÑO ORIENTADO A OBJETOS

Considerando la figura (6.2) en la cual la naturaleza de la ingeniería es esquemáticamente dedicada. Los diferentes tipos de ingeniería son listadas en la esquina superior izquierda de la figura, incluyendo diseñadores, analistas y simulación orientada a ingenieros. Los ingenieros

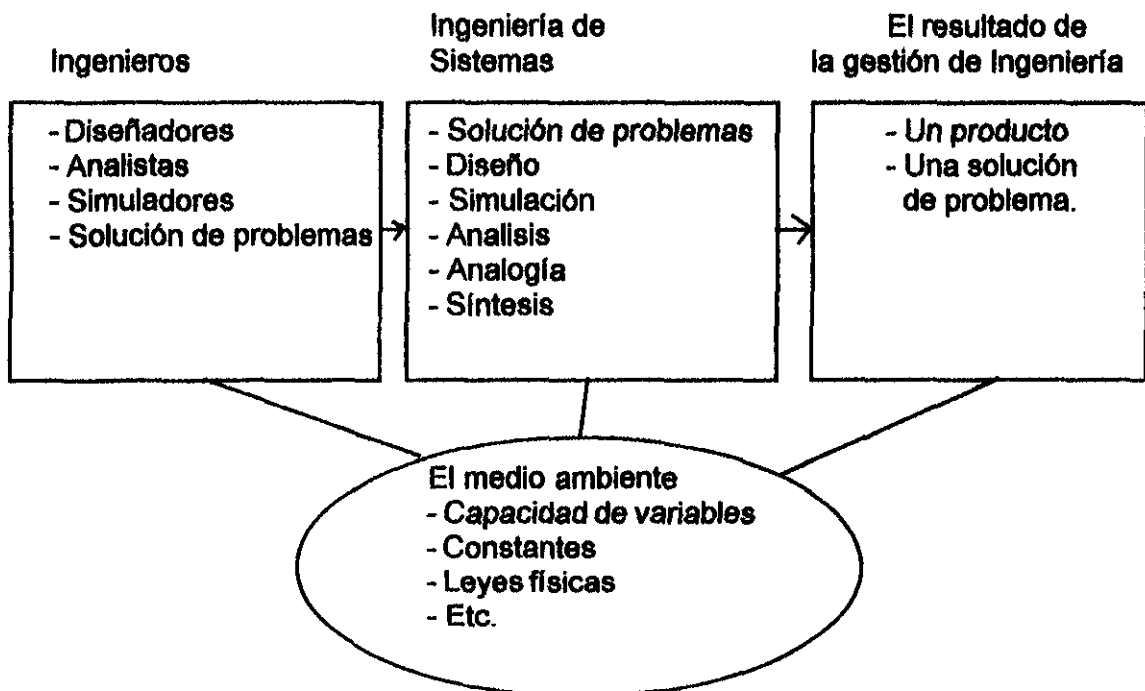
típicamente utilizan diferentes maneras de resolución de problemas; las metodologías usadas son actualmente los más diferentes compuestos de estos estereotipos, incluyendo una buena secuencia de problema-solución que consistirá en:

- 1: Análisis del problema
- 2: Diseño de un nuevo objeto
- 3: Simulación de objetos usando la computadora

La ingeniería de sistemas, como se muestra en el centro de la figura consiste en una mezcla de muchas técnicas de solución de problemas. El resultado de las actividades de la ingeniería como se muestra en el extremo derecho de la figura consistentes en dos principales resultados:

Productos y Soluciones de problemas. El análisis de problemas y la creación de nuevos productos resuelven alrededor de el dominio seleccionado.

Fig. 6.2 (Productos y soluciones)



6.3 METODOLOGIAS PARA LA INGENIERIA ORIENTADA A OBJETOS

La programación orientada a objetos es una metodología que permite la descripción de objetos, incluyendo su función, medio ambiente e información declarativa. La función se refiere a **para qué haga qué**, **fué** diseñado un objeto. El medio ambiente se refiere a **qué está haciendo actualmente**. Y las declarativas de información consistente de información acerca del estado de un objeto.

Una definición típica de qué es un objeto:

- 1: Tiene un estado local
- 2: Características heredadas de más objetos abstractos
- 3: Responde a mensajes, decodificando los mensajes en un contexto local de objetos, tales características hacen objetos muy usados para la descripción de cómo las cosas trabajan en el mundo real y una forma ideal en la solución de problemas que facilitan el trabajo a los ingenieros.

6.4 REPRESENTANDO EL MUNDO CON OBJETOS

El mundo está compuesto por objetos, entre los cuales se encuentran animados e inanimados. Siempre se ha entendido a la naturaleza con objetos, como lo son los árboles, rocas y todo lo que la rodea, así como los objetos que el hombre ha desarrollado, los cuales tienen conocidas características tal como los objetos de medición, automóviles, calentadores de agua, aviones, y otros muchos artefactos de ingeniería. Pocos entienden a los objetos como una representación abstracta, en algún tipo de relación mental, como algunos instrumentos de ingeniería que están compuestos por otros muchos objetos interactuando entre sí.

Para describir un objeto u objetos en lenguajes de programación orientada a objetos, por lo general hay cuatro términos que son frecuentemente utilizados, los cuales son:

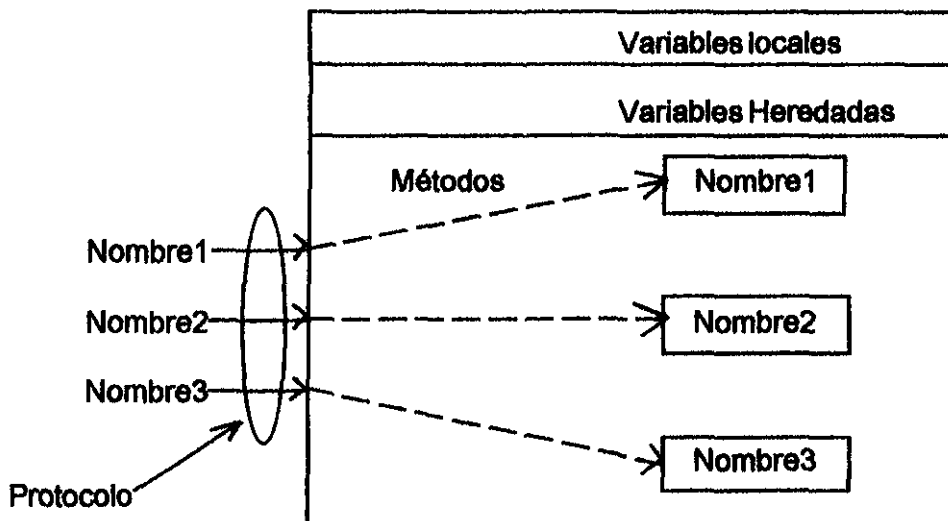
- 1: Encapsulación
- 2: Abstracción
- 3: Herencia
- 4: Poliformismo

La **encapsulación** se refiere a lo que se tiene oculto en la implementación de un objeto desde el mundo exterior. Al tener oculta la estructura interna de un objeto, entendemos que el medio ambiente de un objeto está reducido a la cantidad con la que están separadas las cualidades de herencia, o la propiedades de un objeto desde un objeto físico actual al que ellos pertenecen.

La definición de objetos que viene en diccionario dice: "Cualquier cosa perceptible por uno o más de los sentidos, especialmente algunas cosas que pueden ser vistas o sentidas; un material cualquiera". Esta definición es familiar y es comúnmente usada de la misma manera en el mundo de la programación orientada a objetos, los objetos pueden ser representados en abstracciones

de cosas materiales o algunos objetos abstractos. La figura (6.4) explica la anatomía de un objeto, con variables y métodos, las variables son de dos tipos: Casos y Clases. Las clases de variables son las que están comúnmente en todos los casos de una clase. Los casos de variables pueden ser diferentes para cada caso de una clase.

Fig. 6.4 (Anatomía)



Los mensajes son enviados a los objetos para obtener diferentes conductas. Enviando mensajes a los objetos es la única manera de comunicarse con los otros objetos. Enviando un mensaje en específico a un objeto dispara la ejecución de un código por un método, el cual tiene muchos métodos diferentes que pudieron haber sido creados instantáneamente en una clase específica.

Los métodos pueden residir en cada clase específica o en la descripción de clases que son heredadas por las clases anteriores. Los métodos pueden ser agrupados por entendimientos acorde a la funcionalidad dentro de unidades llamadas **PROTOCOLOS**.

Un protocolo es solamente una etiqueta que identifica a un grupo de métodos que ejecutan la relación de operaciones, por ejemplo: acceso inicialización, actividades, terminación y así sucesivamente. El uso de protocolos es primordialmente para ayudar a los usuarios inexpertos en el entendimiento de la conducta de objetos.

La tabla siguiente (6.5) muestra los términos básicos asociados con las características de la programación orientada a objetos. Cada término será descrito a detalle posteriormente.

Fig. 6.5 (Términos básicos)

- **ABSTRACCION:** Separa las cualidades heredadas o las propiedades de un objeto desde un objeto físico actual al cual ellos pertenecen.
- **ENCAPSULACION:** La implementación oculta de un objeto desde el mundo externo.
- **HERENCIA:** La habilidad de una clase para usar el estado local (variables) y conductas (métodos) desde las clases mas abstractas.
- **POLIFORMISMO:** La capacidad de muchos objetos diferentes a responder al mismo mensaje.

6.4.1 ABSTRACCION

Está definido como: "El acto o proceso de separación de cualidades heredadas o propiedades de algunas cosas desde el actual estado físico del objeto o al concepto al cual ellos pertenecen". La concreta manifestación de esta definición de la metodología en la programación orientada a objetos es la colección de tantas cualidades como sea posible heredar dentro de clases abstractas las cuales pueden estar especializadas por clases abstractas bajas que hereda las características de las clases abstractas altas.

Hay algunas confusiones acerca de las diferencias entre la abstracción y la generalización. La generalización esta definida como: "La reducción a una forma general, clase o ley". Por ende la generalización es una ley pariente, viendo los principios que describen las clases de cosas. En contraste, la abstracción ve por una manera concreta de características amalgamamiento de clases que son comunes a las clases más especializadas. La metodología de la programación orientada a objetos usan la abstracción como una técnica estándar, no generalizada.

En la creación de clases para la representación de cosas, un programador podría considerar la siguiente metodología. En primer lugar observa las características mas abstractas de un objeto el cual va ha ser representado, por ejemplo se considera la siguiente representación:

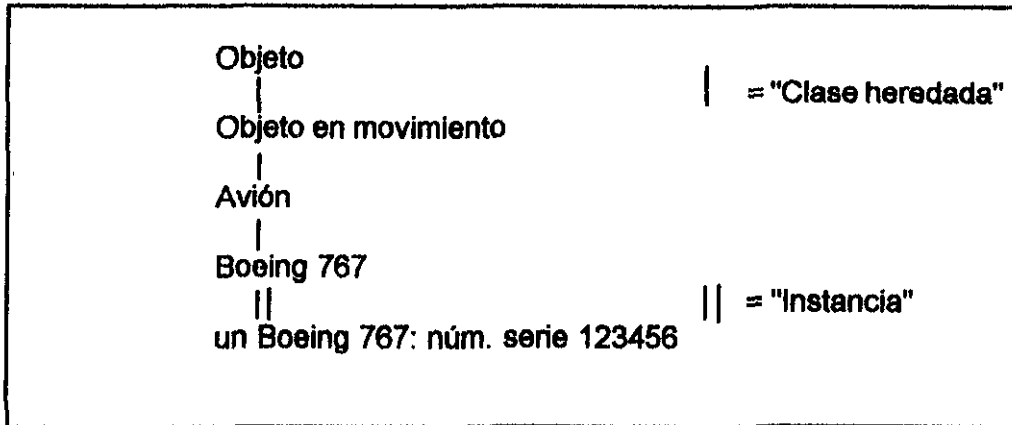


Fig. 6.6 (Abstracción)

Empezando desde el principio, la tarea representa en específico a un avión Boeing 767. El objeto en específico es llamado **INSTANCIA**. En específico el Boeing 767 es una instancia de una clase llamada Boeing 767 desde las cuales muchas instancias de los aviones que pueden ser fabricadas. Noten que la clase Boeing 767 hereda la información de un Aeroplano, el cual, en su turno hereda información acerca del Movimiento de Objetos. La utilidad de esta metodología se basa principalmente en la capacidad de reducción de información que pueda ser descrita en nuevos objetos. Las características de todos los objetos que se mueven pueden ser heredadas no solamente por los aeroplanos, sino también por automóviles, animales, objetos astronómicos, barcos, y así sucesivamente. De la misma manera, las características de todos los aeroplanos que pueden heredar por los diferentes tipos de aviones están representadas.

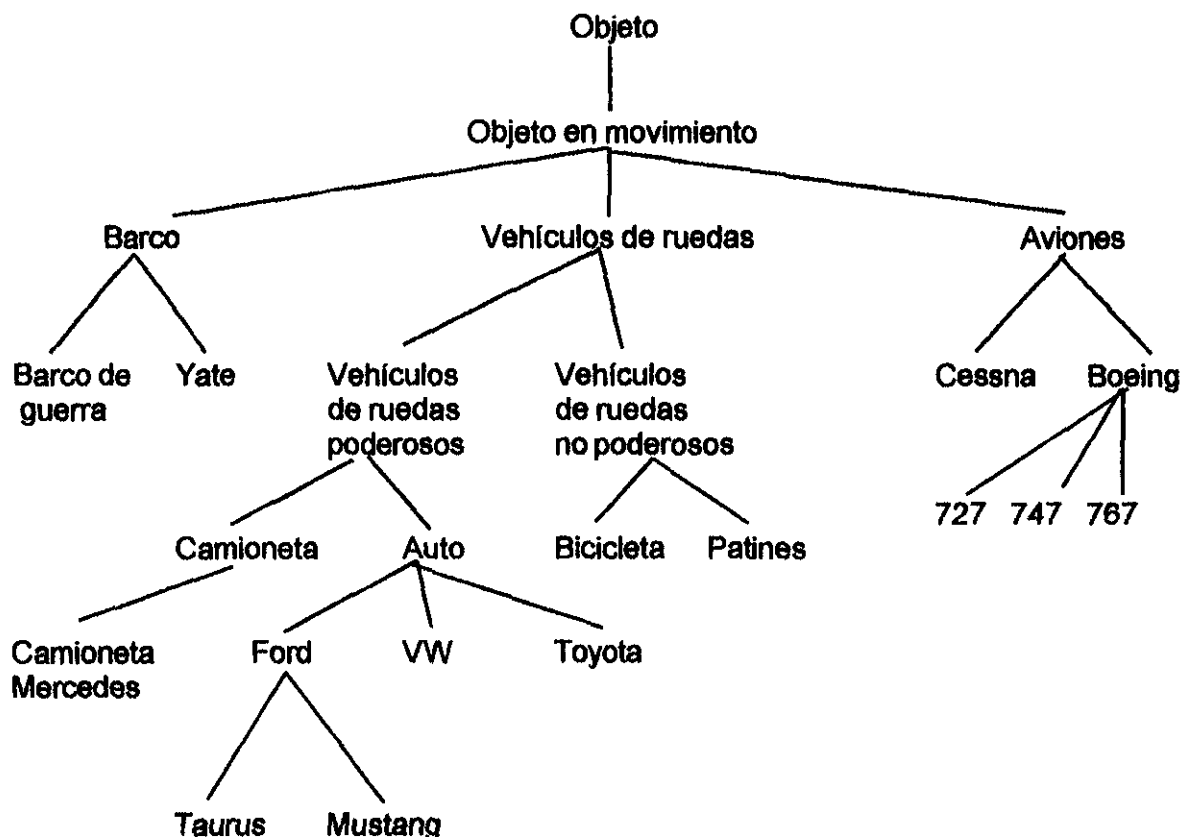
6.4.2 ENCAPSULACION

La encapsulación se refiere a la captura del estado y conducta de un objeto dentro de otro objeto. Como se indico previamente, un objeto puede contener diferentes tipos de variables: clases de variables, variables de instancia y variables locales, las cuales son conocidas solamente en los métodos el las cuales ellas están definidas. Todas esas variables son encapsuladas dentro del objeto y definen el estado del objeto. En suma la conducta de un objeto es encapsulado dentro de otro objeto como una colección de métodos que definen como un objeto se conducirá cuando el mensaje o los mensajes le sean enviados.

6.4.3 HERENCIA

La herencia es la capacidad de los objetos para obtener información acerca de los estados internos y métodos desde los antecesores mas abstractos. Considerando la figura (6.7)

Fig. 6.7 (Herencia)



en la cual se ejemplifican objetos como barcos, vehículos de ruedas y aeroplanos la información heredada desde la clase abstracta "Objeto en movimiento". Como se muestra, Objeto en Movimiento es la superclase de la clase abstracta de Barcos, Vehículos de ruedas y aeroplanos, y en suma éstos pueden contener todos los métodos y variables que podrían ser asociados con barcos, vehículos de ruedas y aeroplanos. La superclase es un término que se refiere a los más abstractos objetos en su jerarquía ancestral. Controversialmente, las sub-clases son unos términos que se refieren a las clases que están contenidas dentro de otras clases. El objeto en movimiento puede contener variables concernientes en dirección, rapidez, peso y métodos que contienen código por computadora de rapidez, dirección, y otras cosas. Esos métodos y variables podrían ser igualmente aplicables a cualquier sub-clase usando Objeto en movimiento como una

superclase. Los vehículos de rueda son también una clase abstracta en estos servicios, como una superclase para todos los tipos de vehículos que tiene ruedas. Coche podría heredar toda la información de estas superclases y en tal caso, agregar variables como "ruedas" o "girar hacia".

El principal punto a ser discutido es que la herencia permite especificar objetos para ser creados por el uso de una herencia de clases. Esta técnica capacita a los diseñadores a seleccionar clases que son construidos al mismo tiempo y se especializan esas clases para inicializar objetos que son necesitados para un diseño.

De aquí, existen aparentemente dos ventajas:

1: Reducción de la cantidad de código por la repartición de código en la abstracción heredada.

2: Simplificación en la creación de nuevos objetos por la especialización de clases.

La herencia puede ser estrictamente ancestral o de forma contraria, como se muestra en la figura anterior (6.7). Puede ser también conocida como múltiple herencia heredada, permite tener herencia de información desde múltiples antecesores, la figura anterior (6.7) es primordialmente una representación hereditaria.

6.4.4 POLIFORMISMO

El poliformismo se refiere en general a que hay muchas partes y formas, en la programación orientada a objetos, el poliformismo es la capacidad de diferentes objetos a responder de la misma manera a los mensajes. La figura (6.4.4) representa un ejemplo de poliformismo, como lo son los cuatro objetos que son mostrados en este ejemplo:

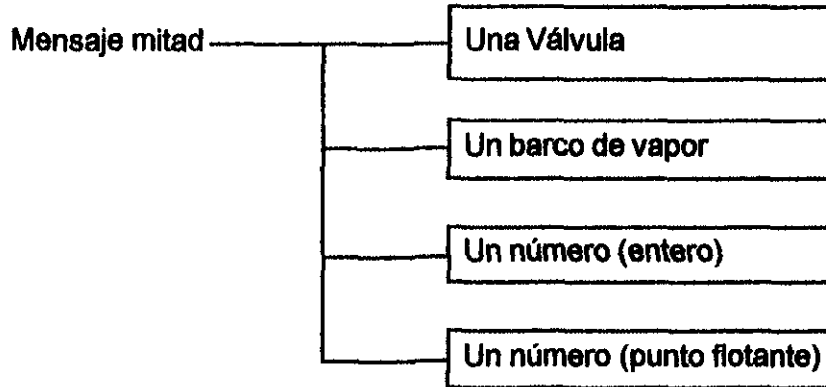
- 1: Una válvula
- 2: Un Bote de vapor
- 3: Un número entero
- 4: Un número de punto flotante

El mismo mensaje será enviado a cada objeto:

Ajustar un Objeto: Mitad

[Un objeto] ajustar: Mitad

Fig. 6.8 (Poliformismo)



Cuando un **objeto** es el objeto, **ajustar** es el mensaje a enviar al objeto que activa el método **ajustar**; y **mitad** es el argumento pasan con el mensaje a los métodos. El mensaje **ajusta** tiene una diferente implementación para cada uno de los cuatro diferentes objetos. La válvula se abriría a la mitad de su capacidad, el barco de vapor aumentaría o disminuiría su velocidad a la mitad de su capacidad, y para cada uno de los diferentes tipos de números sería divididos en dos, cada uno acorde al tipo de número que es.

El punto central de esta discusión es que la definición de conducta de poliformismo que permite a los objetos una mejor manera de entender la conducta de los objetos. En el ejemplo anterior, el mensaje **mitad** da como resultado una conducta diferente en cada uno de los ejemplos, siempre el mismo mensaje será enviado a los diferentes objetos.

CAPITULO 7:

Dado el objetivo general de este trabajo, que es que en la clínica de odontología U.P.A.E.P. se controle eficientemente la información, de modo que sea suficiente y oportuna para tomar decisiones con más certidumbre y en menos tiempo, que se fortalezcan otras áreas de la administración y se brinde un mejor servicio tanto a los pacientes como a los alumnos, además de que se controlen los ingresos y egresos de efectivo, así como los recursos materiales e insumos. En este capítulo se describe el funcionamiento del sistema computacional desarrollado para solventar los problemas que originaron la necesidad de implementar el presente trabajo.

7. SISTEMA COMPUTACIONAL

7.1 PARADOX

Paradox para Windows es un paquete de bases de datos relacionales que proporcionan una gran rapidez al usuario, para su completa utilización es fácil aprender su manejo, proporciona una máxima funcionalidad, ya que se puede adaptar a las necesidades de los múltiples usuarios que este paquete posee. Paradox es un paquete que se diseñó para que pueda ser manejado por personas que tengan mucha o poca experiencia en el manejo de bases de datos, ya que siendo una base de datos relacional ofrece la opción de poder ser funcional sin necesidad de tener programación.

La mayoría de los paquetes de bases de datos convencionales han sido difíciles de aprender y han tenido problemas con su funcionamiento, porque entre más grande es el sistema en cuestión, su funcionamiento es más lento, cosa que no sucede con Paradox, porque es un programa de máxima eficiencia con el que se pueden realizar tareas de un alto grado de complejidad, además que ofrece la facilidad de realizar tareas que funcionan en monousuario y multiusuario.

7.2 ObjetPAL

ObjetPAL es el lenguaje de programación que trae integrado PARADOX para Windows. Puede utilizarse para programar aplicaciones que paradox no trae integrada.

Los programadores que apenas empiezan a realizar programas se darán cuenta que en ObjetPAL pueden realizar aplicaciones con relativa facilidad.

A continuación se muestran algunos ejemplos de lo que pueden realizar:

- Añadir botones para realizar acciones que se repiten con frecuencia
- Crear cuadros de diálogo que, si la base de datos contiene un volumen grande de datos, ayuden a los usuarios a obtener los registros que deseen cuando para ello se precise algo más complejo que una búsqueda normal.
- Puede crearse una rutina de edición que vigile lo que los usuarios introducen y responda ante entradas incorrectas.
- Con ObjetPAL se pueden crear efectos de animación

Como Paradox y ObjetPAL están entre sí estrechamente ligados, entre más se sepa acerca de la programación dentro del paquete Paradox, más se podrá aprovechar en la programación dentro de ObjetPAL.

7.2.1 QUE ES ObjetPAL

En un sentido mas formal, ObjetPAL es un lenguaje de programación visual de alto nivel, guiado por sucesos y basados en objetos, ya que en este lenguaje se pueden desarrollar aplicaciones personalizadas con botones, cuadros de diálogo, menús, mensajes, advertencias, etc, se pueden realizar interfaces de usuario para aplicaciones de base de datos o aplicaciones que no tengan nada que ver con base de datos.

ObjetPAL es considerado como una herramienta que aumenta la potencia de paradox interactivo, de esta forma se podrán realizar tareas que sin éste serían difíciles o prácticamente imposibles de realizar.

Como ObjetPAL trabaja con objetos y lo primero que debe recordar sobre los objetos es que los objetos tienen propiedades. Cuando se crea un objeto, tiene definidas las propiedades que le han de regir en cuanto su aspecto y comportamiento. Por ejemplo, las propiedades de un cuadro son el tamaño, la posición, el color y el marco. Dentro de ObjetPAL pueden cambiarse o crearse todas las propiedades que se utilizan con el funcionamiento interactivo dentro de paradox.

Lo siguiente que debe recordar es que los objetos existen en un contexto. El contexto de un objeto concreto viene definido por los objetos que lo contienen, esta característica de ObjetPAL le proporciona a los programadores experimentados una gran potencia y flexibilidad. Y a los programadores que no tienen mucha experiencia les bastará recordar que la ficha contiene a todos los demás objetos. Cuando se le coloca un objeto a una ficha se le está dando un contexto.

Esta forma de colocar objetos se le denomina programación visual, ya que la ficha permite ver la interfaz de usuario de la aplicación al mismo tiempo que se está programando. Para crear aplicaciones nuevas con ObjetPAL se colocan objetos como campos, listas de selección, listas

editables, botones, etc, y la mismo tiempo se definen sus propiedades. Una vez satisfecho con la ficha, se puede utilizar ObjetPAL para cambiar el comportamiento sólo de aquellos objetos que no se adapte satisfactoriamente a las necesidades del usuario.

Este trabajo difiere en mucho al que supondría la creación de una aplicación dentro de un entorno no visual, ya que en otros lenguajes, se crea el interfaz del usuario escribiendo el código en algún editor de texto, posteriormente ha de compilarse el código, depurarlo, volverlo a ejecutar, etc., hasta que funciona, por lo tanto cada vez que se cambia a ciegas la posición de un elemento ha de accederse el código, con lo que se pueden crear más errores.

Como en el ambiente Windows ninguna aplicación tiene el control absoluto y permanente, por lo que todas las aplicaciones son totalmente dependientes de Windows, que les proporciona tiempo de proceso, espacio en el monitor y otros recursos, en otras palabras Windows es quien controla todas las aplicaciones Windows. La naturaleza de Windows invita a todas las aplicaciones que sean guiadas por sucesos. Un interfaz guiado por sucesos es aquel que sólo responde a acciones concretas del sistema o del usuario, la aplicación toma el control del sistema durante el tiempo que sea necesario para responder a una petición del usuario y del sistema, y para cuando termina de realizar esta función, retoma el control a Windows hasta que la aplicación vuelva a ser requerida por un suceso. Teniendo en cuenta que estas aplicaciones son guiadas por sucesos, una descripción más elaborada diría que los objetos tienen un contexto que determina su relación con los demás objetos.

Una de las tareas principales de los programadores de ObjetPAL consiste en volver a definir la respuesta de los objetos, ya que los objetos tienen un conjunto de respuestas por default, y si se desea modificar la respuesta por default de algún objeto, habrá de modificar uno o más de sus métodos estándar. La mayor parte de la programación en ObjetPAL consistirá en modificar la respuesta por default de los métodos estándar.

Como se mencionaba que el ObjetPAL se puede realizar tanta o tan poca programación como se desee ya que ObjetPAL ofrece esta flexibilidad gracias a que los objetos son modulares, es decir los objetos son contenedores de sí mismos, por lo que es posible cambiar el comportamiento de un objeto de una ficha, sin cambiar al mismo tiempo todos los demás objetos. Esta característica implica que entre más grande sea un sistema, con una programación basada en objetos puede realizarse de una forma relativamente más sencilla, ya que en los lenguajes tradicionales, entre más grande sea el sistema, más probabilidades hay de que sea más inestable, y no sólo porque ésto implica mas líneas de código sino que puede acarrear más errores.

Dentro de una programación basada en objetos se intenta hacer una imitación del mundo real, con un enfoque mas orgánico, osea que dentro de ellos hay piezas más pequeñas (objetos), por otro lado la programación basada en objetos permiten un proceso de desarrollo gradual por lo que se puede volver a programar cuantas veces sea necesario hasta terminar de definirlo. Esto quiere decir que se puede programar y reprogramar un objeto hasta que quede perfeccionado sin que corra peligro todo el sistema.

EL lenguaje de programación ObjetPAL permite las siguientes funciones:

- Eficaz introducción de datos
- Tipos de datos definidos por el usuario
- *Matrices redimensionables, matrices dinámicas y registros*
- Control del programa estructurado
- Biblioteca de métodos y procedimientos
- Métodos y procedimientos definidos por el usuario
- Llamadas a funciones y procedimientos escritos en otros lenguajes de programación como Pascal, C, C++, etc.

En el lenguaje cotidiano, los objetos no son más que cosas, en términos de programación, los objetos son datos y código estrechamente ligados. Los objetos se crean de forma interactiva o mediante ObjetPAL.

7.2.2 OBJETOS EN ObjetPAL

Al igual que los animales en los ejemplos anteriores, los objetos de ObjectPaL se organizan por tipos. Los objetos de un tipo determinado tienen las mismas propiedades y los mismos métodos. *Por ejemplo, todos los archivos de texto tienen propiedades en común y todas las tablas también, pero las propiedades de las tablas son diferentes a las de los archivos de texto.* Por lo tanto, los archivos de texto y las tablas son objetos de tipos diferentes y deberá utilizarse un conjunto de métodos para los primeros y otro conjunto para actuar sobre las segundas.

Los métodos para los distintos tipos de objetos pueden tener los mismos nombres. De la misma forma que en el ejemplo anterior se utilizaba “saltar” para caballos y para delfines, ObjectPaL utiliza un método “abrir” para abrir una tabla y otro para abrir un archivo de texto. La sintaxis de ObjectPaL recuerda al lenguaje humano. Por ejemplo, uno pueda pensar “quiero abrir una tabla” o “quiero abrir un archivo de texto”. El código subyacente que abre una tabla es distinto del que abre un archivo de texto, pero el verbo “abrir” es el mismo y es lo único que hace falta recordar. ObjectPaL ejecuta el método open adecuado para el tipo de objeto, sin que sea necesario recordar dos comandos diferentes como “abretabla” o “abrearchivotexto”, como se haría en un lenguaje por procedimientos.

Por otro lado, el código que actúa sobre un objeto no debe estar necesariamente anexo al objeto. Por ejemplo, el código que abre una tabla puede estar anexo a un botón, un objeto de campo o cualquier otro objeto que pueda colocarse en una ficha (pantalla o ventana).

En ObjectPaL los objetos tienen propiedades (color, posición, tipo y grosor de línea) y métodos (código que define la forma en que se comporta el objeto), las propiedades son datos y los métodos, código.

El entorno ObjectPaL se basa totalmente en objetos. Los objetos de paradox tienen código estándar incorporado, por lo que la mayor parte del trabajo de programación ObjectPaL puede considerarse como "personalización" de los métodos y no como programación desde cero.

La creación de aplicaciones en Paradox consiste en gran medida en colocar "objetos" en "fichas" y escribir o modificar "métodos" con ObjectPaL que definan el comportamiento de dichos "objetos". Estas aplicaciones podrían definirse como aplicaciones "haz esto" (objetos y verbos).

En un programa tradicional no interactivo basado en procedimientos, el código del programa controla lo que ocurre y el momento en que ocurre. La ejecución suele ser lineal. Un programa de éste tipo podría, por ejemplo mostrar una tabla, realizar un cálculo, mostrar el resultado, etc., todo bajo el control del programa.

ObjectPaL es diferente. Las aplicaciones elaboradas con ObjectPaL otorgan el control al usuario, que actúa sobre los objetos (botones, tablas, menús, etc.) en el orden que desea. Aquí no puede aplicarse el enfoque basado en procedimientos; es preciso adoptar una estrategia basada en objetos.

Antes de escribir la primera línea de código es necesario tener una idea clara del objetivo de la aplicación, de los objetos que van a utilizarse y de la manera en que el usuario va a manipularlos. Cuanto más complicada vaya a ser la aplicación más importante va a ser ésta fase. Para desarrollar un aplicación de Paradox, han de seguirse los pasos siguientes:

- 1.- Definir el objetivo
- 2.- Generar las tablas
- 3.- Diseñar la(s) ficha(s)
- 4.- Anexar y/o modificar el código

(Para mayor información sobre estos tópicos diríjase a la "Guía del programador ObjectPaL", Borland, 1992).

7.2.3 PROGRAMACION EN ObjetPAL

La programación dentro de ObjetPAL es similar en muchos aspectos a los demás lenguajes y también cuenta con algunas diferencias. ObjetPAL es similar a otros lenguajes porque también utiliza variables, ofrece estructuras de control como el IF..THEN..ELSE, bucles como FOR y el WHILE, realiza cálculos y proporciona un sistema para la creación de funciones, dentro de ObjetPAL se denominan Procedimientos y Métodos.

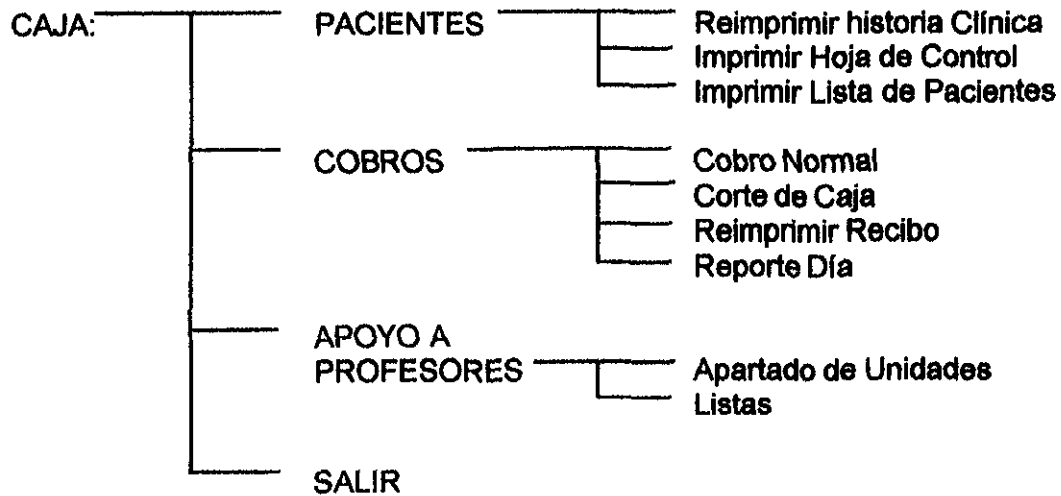
Una de las diferencias principales entre ObjetPAL y los lenguajes tradicionales de programación es que se trata de un lenguaje basado en objetos, en los otros lenguajes la programación se propone como un todo o nada, o bien se controla toda la aplicación de principio a fin o no programa nada. Como ObjetPAL se centra en objetos la programación a realizarse se convierte en una tarea relativamente más sencilla, dependiendo de lo que se desee realizar, y éstos objetos pueden ser tantos o tan pocos como se deseen.

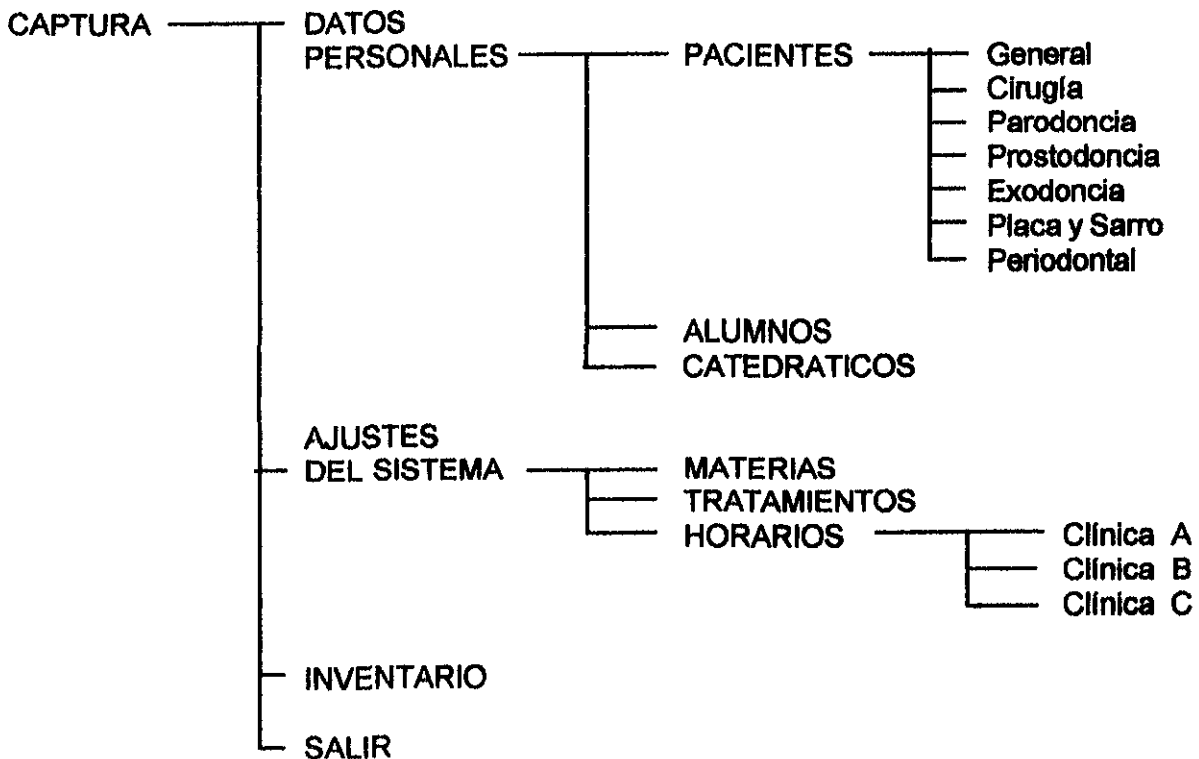
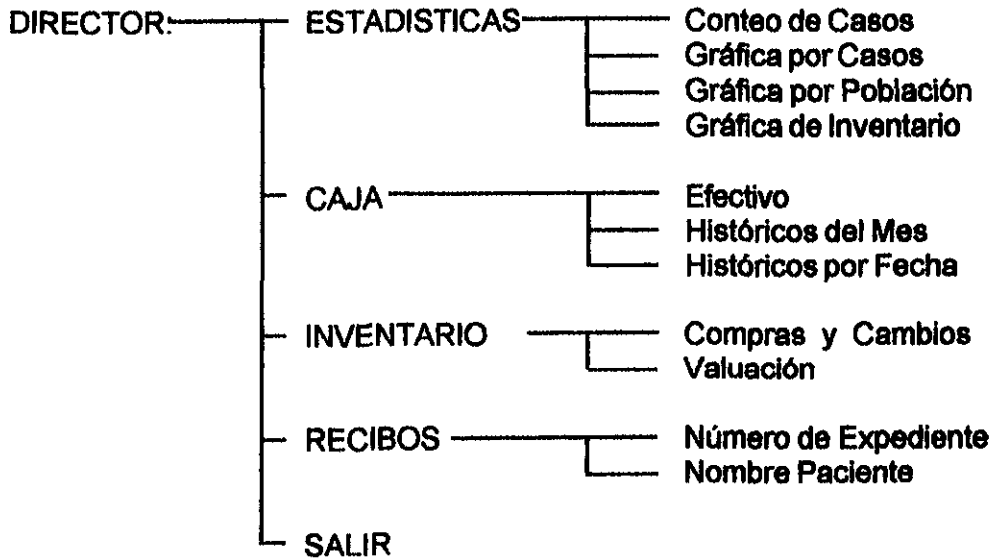


7.3 DOCUMENTACION DEL SISTEMA COMPUTACIONAL

7.3.1 DIAGRAMA DE NAVEGACION

SISTEMA DE CONTROL ADMINISTRATIVO, CONTABLE, CLINICO Y ACADEMICO





CONSULTA:

- Por Título
- Por Autor
- Por Fecha
- Calificación
- Horario Clínica A
- Horario Clínica B
- Horario Clínica C
- Pacientes

AJUSTES:

- Modificar una Tabla
- Recuperar Registros de Hoy
- Reimprimir un Informe
- Cierre del Mes
- Cierre de Semestre
- Respalda Datos
- Recuperar archivos

DFD SISTEMA PROPUESTO (NIVEL 0)

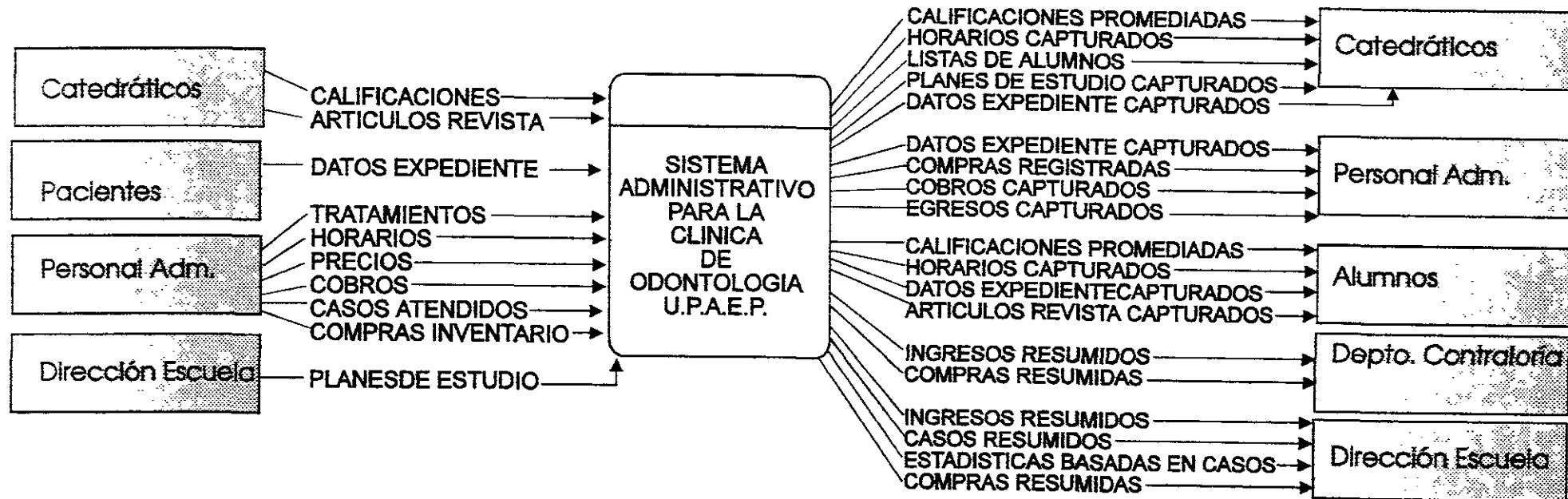


Diagrama de Flujo de Datos SISTEMA PROPUESTO (NIVEL 1)

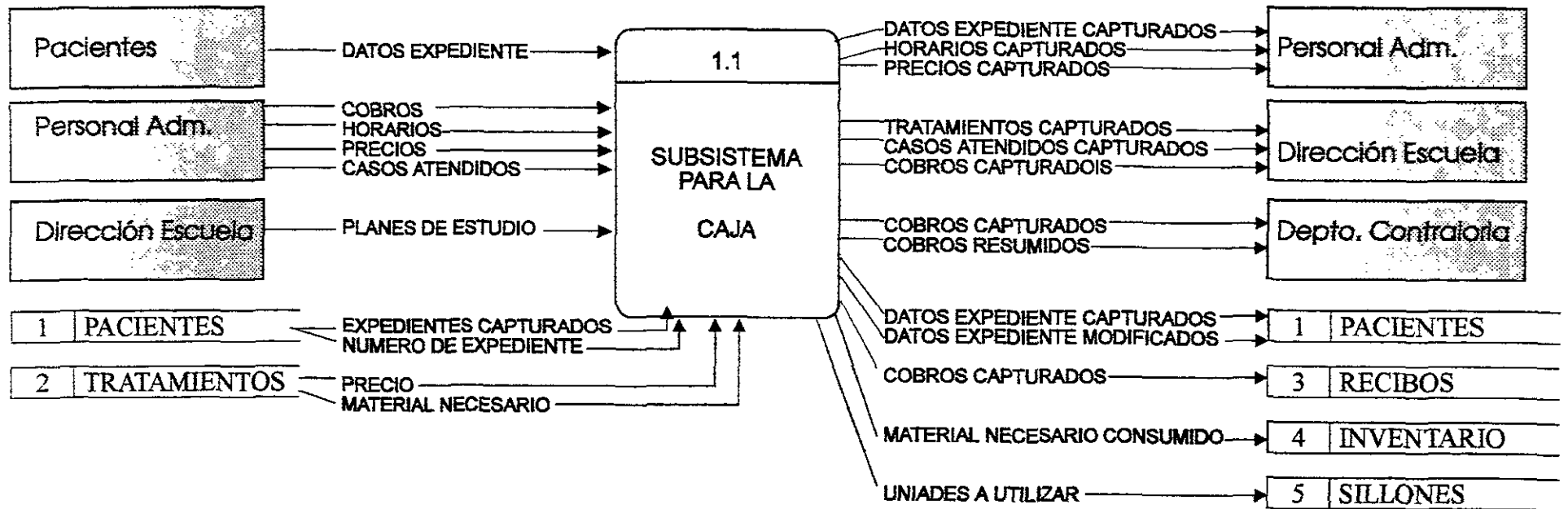


Diagrama de Flujo de Datos SISTEMA PROPUESTO (NIVEL 1)

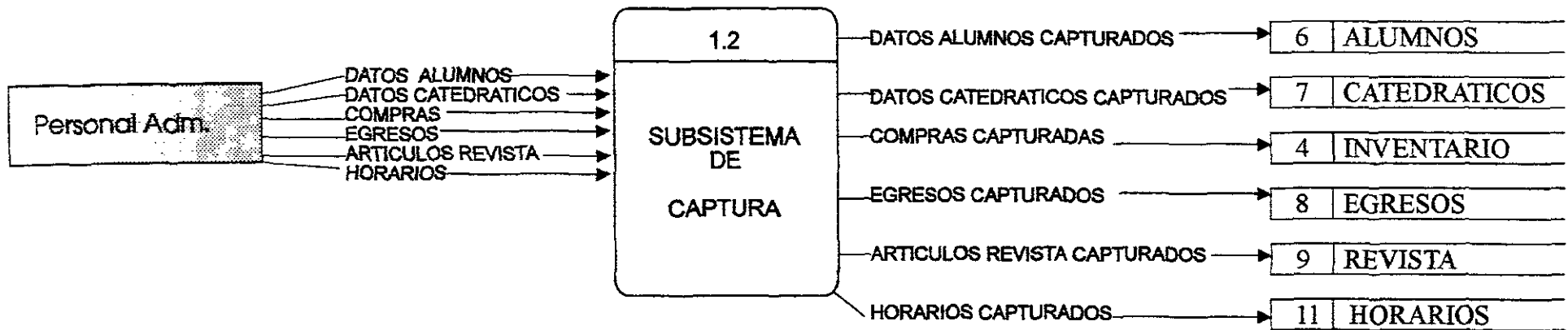


Diagrama de Flujo de Datos SISTEMA PROPUESTO (NIVEL 1)

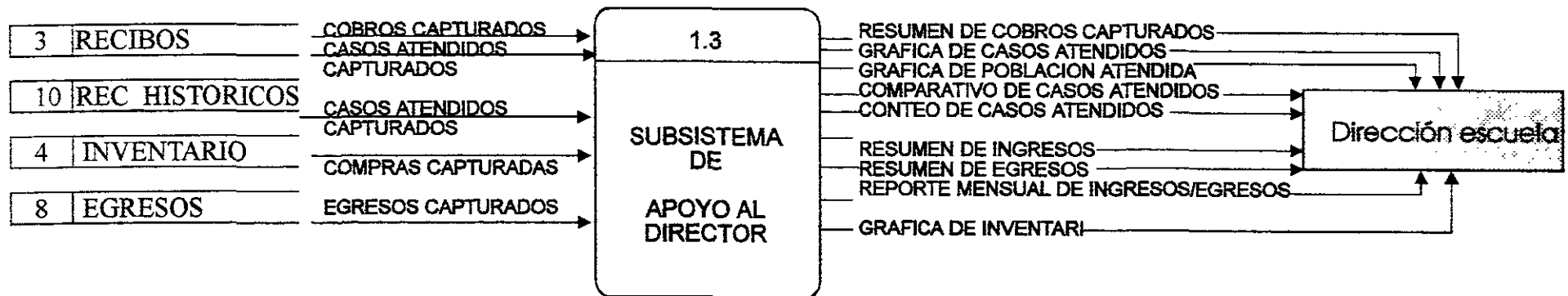


Diagrama de Flujo de Datos SISTEMA PROPUESTO (NIVEL 1)

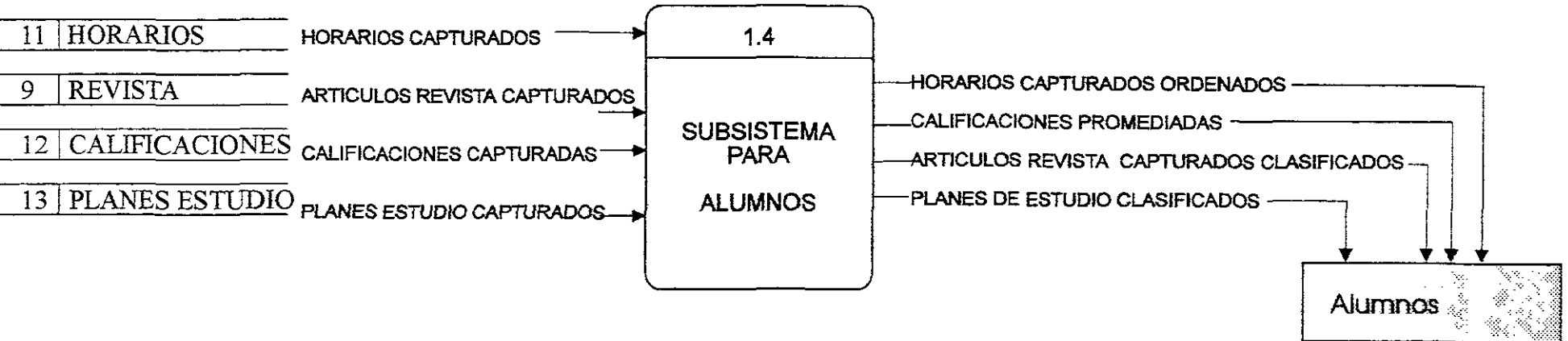


Diagrama de Flujo de Datos SISTEMA PROPUESTO (NIVEL 1)

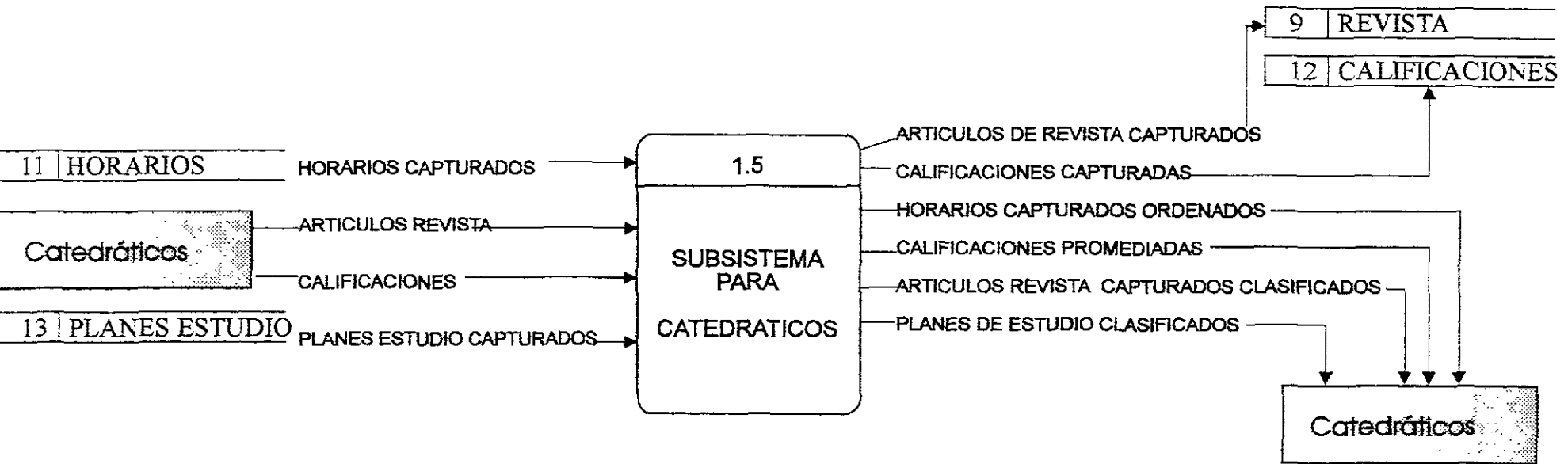


Diagrama de Flujo de Datos SISTEMA PROPUESTO (NIVEL 1)

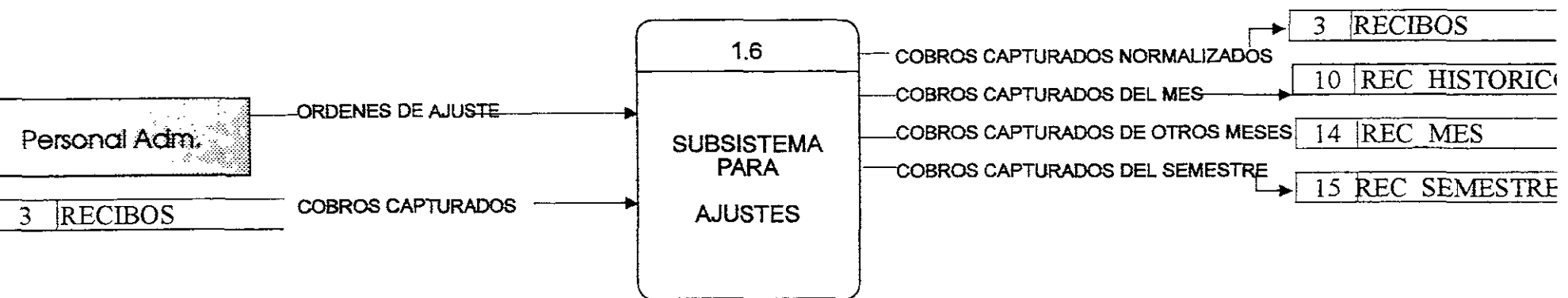


Diagrama de Flujo de Datos

SISTEMA PROPUESTO

(Subsistema de CAJA NIVEL 2)

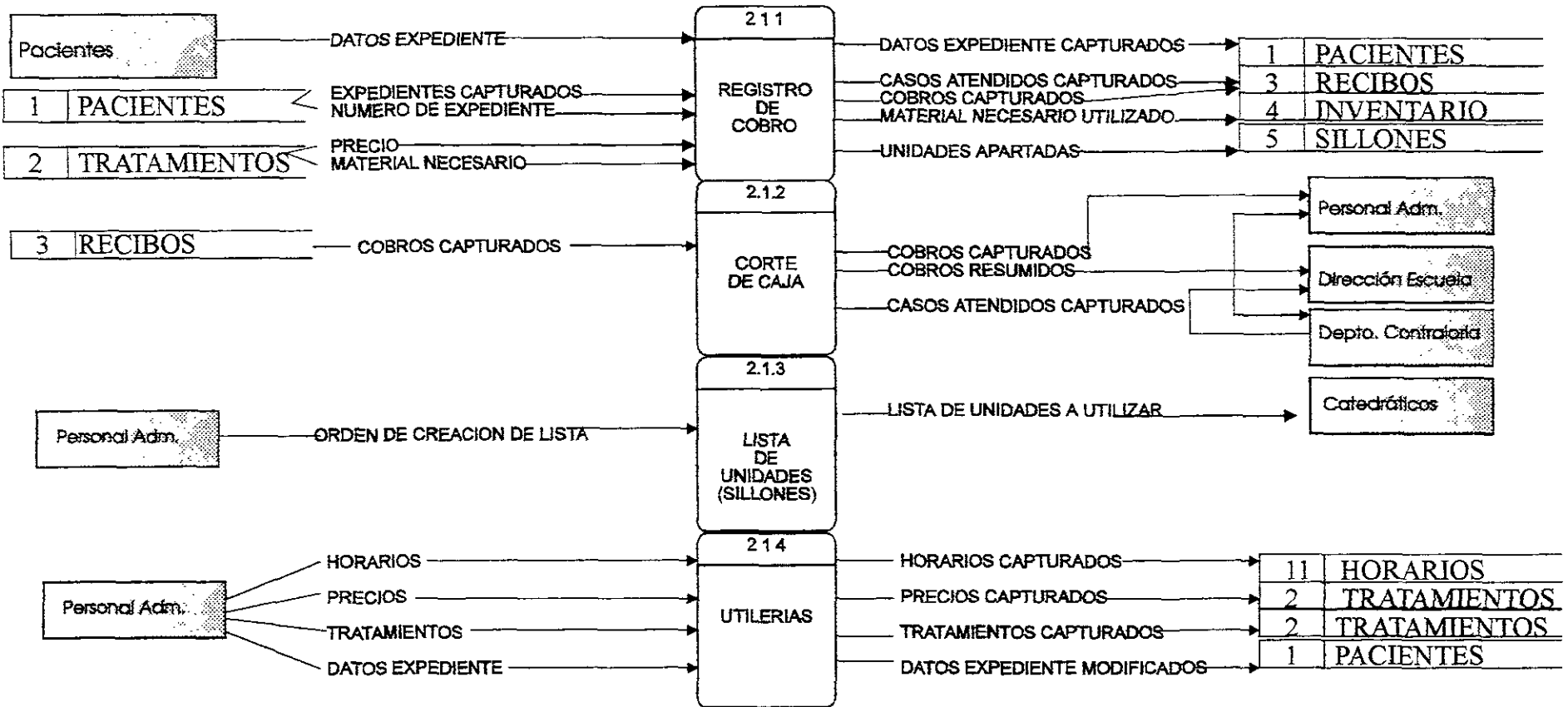


Diagrama de Flujo de Datos

SISTEMA PROPUESTO

(Subsistema de apoyo al DIRECTOR NIVEL 1)

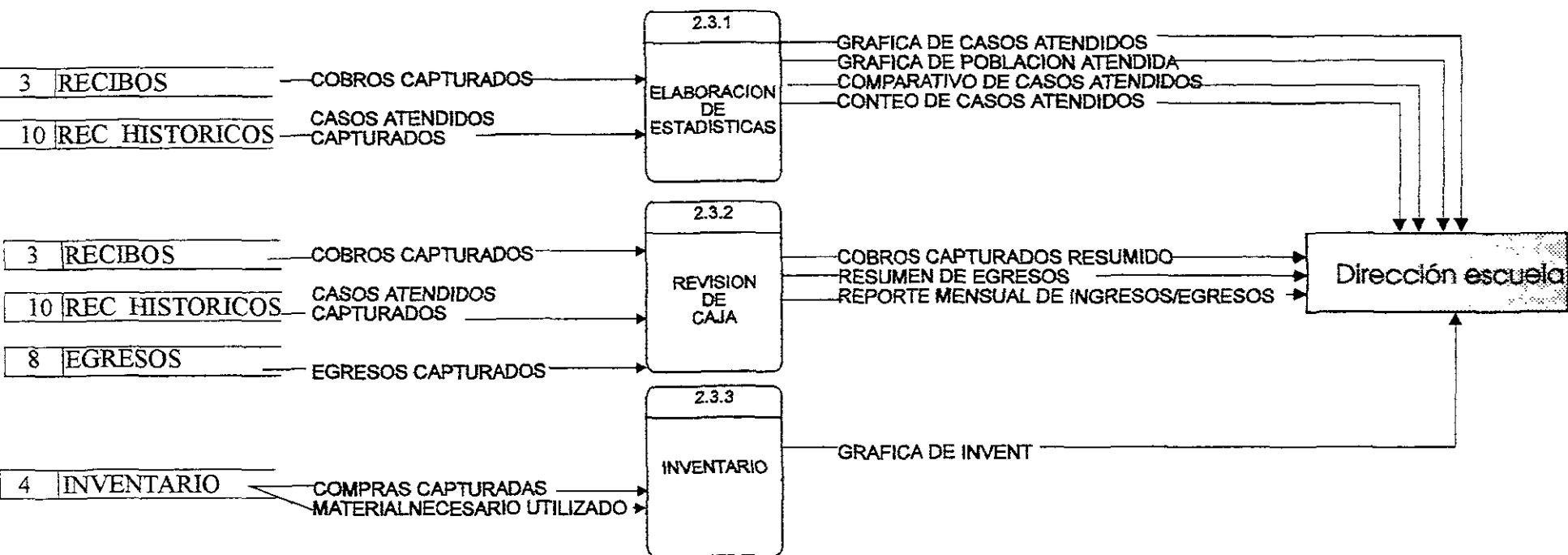


Diagrama de Flujo de Datos SISTEMA PROPUESTO (Subsistema para ALUMNOS NIVEL 2)

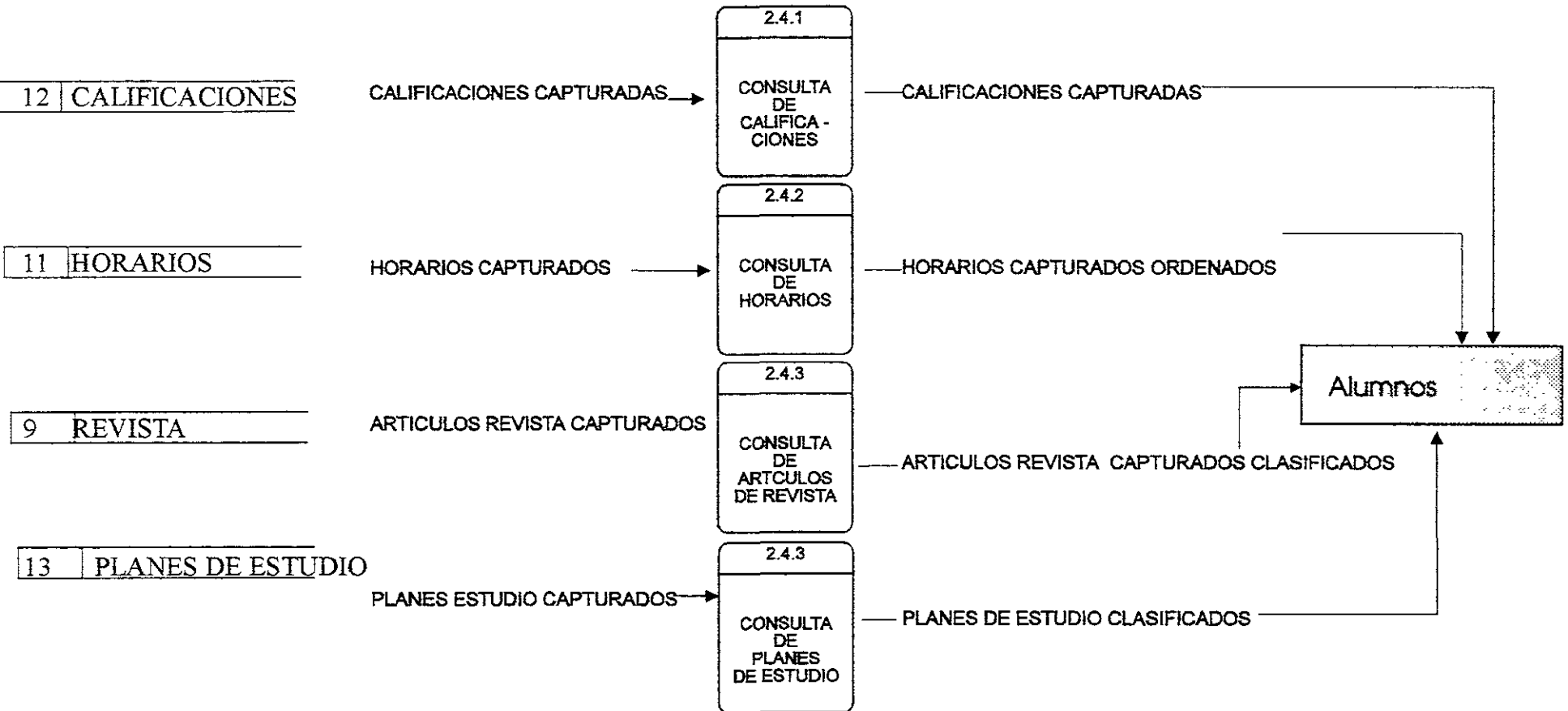


Diagrama de Flujo de Datos

SISTEMA PROPUESTO

(Subsistema para CATEDRATICOS NIVEL 2)

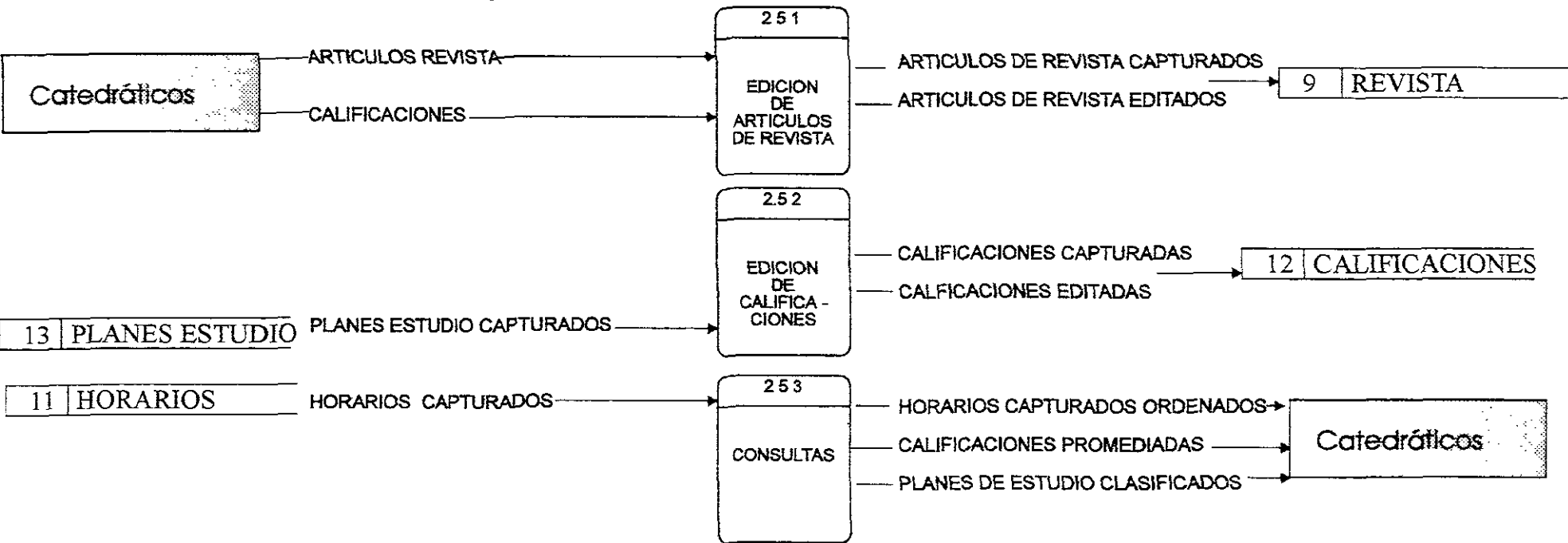
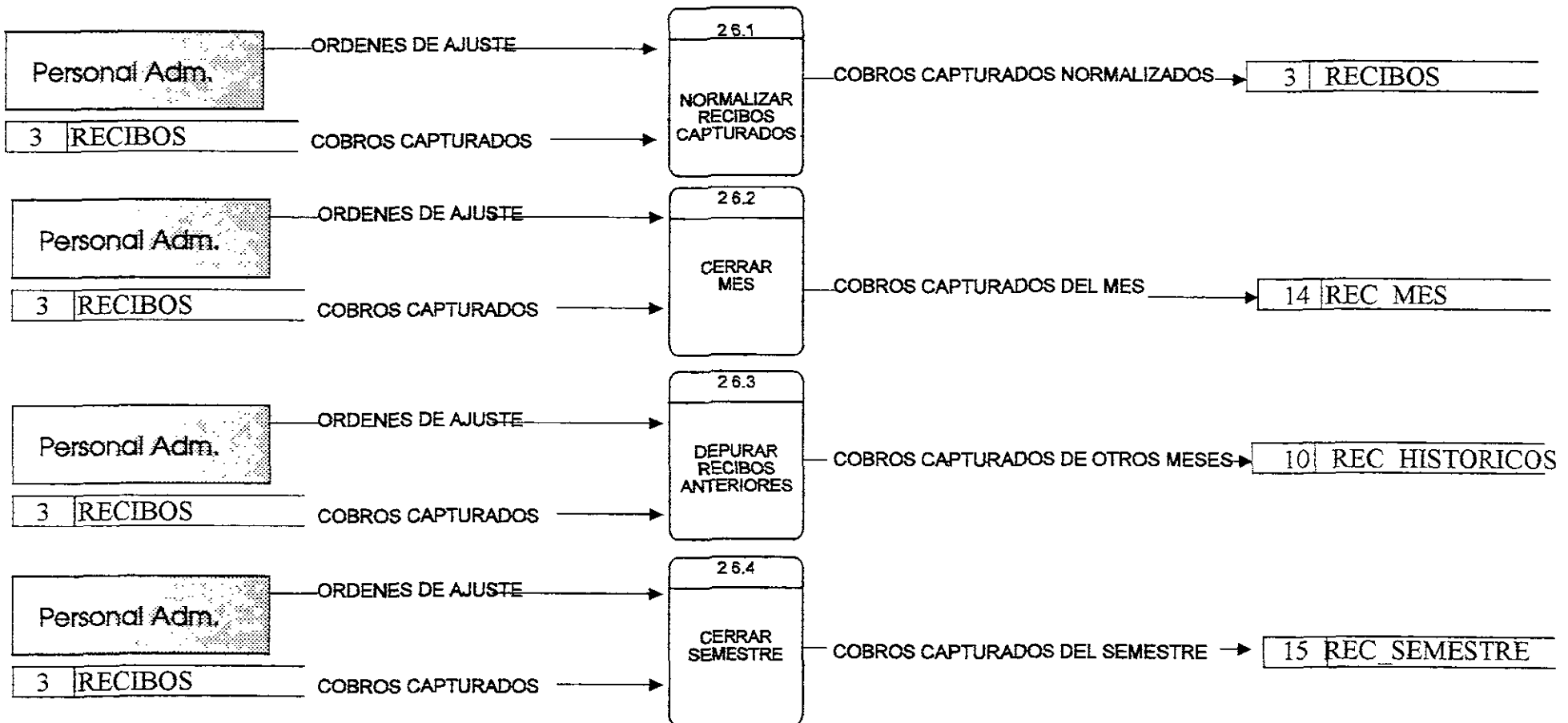


Diagrama de Flujo de Datos

SISTEMA PROPUESTO

(Subsistema para AJUSTES NIVEL 2)



7.3.3 - TABLAS (BASES DE DATOS)

Información de estructura Paradox para Windows Tabla: ALUMNO1.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Clave
1	IDENTIFICACION	N		*
2	Nombre	A	30	
3	Grupo	A	2	
4	Dirección	A	20	
5	Teléfono	A	10	

Propiedades de tabla:

Controles de validación

Campos necesarios

Aceptar Ayuda

ALUMNO1

Información de estructura dBASE III Tabla: ALUMNOS.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Dec
1	IDENTIFICACION	N	6	0
2	NOMBRE	C	40	
3	SEMESTRE	N	1	0

Propiedades de tabla:

Indices

Información detallada

Bloqueo de registros:

Ampliar

Aceptar Ayuda

ALUMNOS

Información de estructura Parados para Windows Tabla: CLI_CIR.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Clave
1	FECHA	N		
2	NOMBRE	A	40	
3	PRO_PRIN	A	40	
4	TIE_MO	A	40	
5	BAJO_TRA	A	40	
6	OPERA	A	40	
7	ALERGI	A	40	
8	EXTRA_CO	A	10	
9	TRA_RA	A	10	
10	SAN_CO	A	10	
11	MARCA	A	10	
12	EMBA	A	10	

Propiedades de tabla:

Controles de validación

1 Campo necesario

Aceptar Ayuda

CLI_CIR

Información de estructura QUASE III Tabla: CLINICAS.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Dec
1	CLINICA	C	1	
2	MATERIA	C	15	
3	SEMESTRE	N	1	0
4	DOCTOR	C	30	
5	FECHA	D		
6	HORA_INI	N	2	0
7	HORA_FIN	N	2	0
8	S1	N	6	0
9	S2	N	6	0
10	S3	N	6	0
11	S4	N	6	0
12	S5	N	6	0
13	S6	N	6	0

Propiedades de tabla:

Indice

Información detallada

Bloqueo de registros:

1 año

Aceptar Ayuda

CLINICAS

Información de estructura Parados para Windows Tabla: CONSULTA.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Clave
1	Mat_op	N		*
2	Fecha	D		
3	Nombre	A	30	
4	Tratamiento	A	20	
5	Paso	N		
6	Mat_op	N		

Propiedades de tabla:

Controles de validación

☐ 1 Campo necesario

☒ Aceptar ☐ Ayuda

CONSULTA

Información de estructura Parados para Windows Tabla: CONTRARECIBO.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Clave
1	Contrarecibo	N		*
2	Factura	N		
3	Proveedor	A	30	
4	Concepto	A	40	
5	Fecha factura	D		
6	Importe	N		
7	Fecha pago	D		

Propiedades de tabla:

Controles de validación

☐ 1 Campo necesario

☒ Aceptar ☐ Ayuda

CONTRARE

Información de estructura de BASE III Tabla: CONTROL.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Dec
1	FECHA	N	6	0
2	FECHA	D		
3	TRAT	C	25	
4	PIEZA	N	2	0
5	OPERADOR	N	6	0

Propiedades de tabla:

Indices

Información detallada

Bloqueo de registros:

☐ Variedad

☒ Aceptar

CONTROL

Información de estructura Paradox para Windows Tabla: EGRE193.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Clave
1	Concepto	N		*
2	Concepto	A	30	
3	Proveedor	A	30	
4	Importe	N		
5	Fecha	D		
6	Reposicion	N		

Propiedades de tabla:

Controles de validación

... 1 Campo necesario

☒ Aceptar

EGRE193

Información de estructura Paradox para Windows Tabla: EGRESOS.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Clave
1	Reposicion	N		*
2	Reposicion	N		*
3	Concepto	A	30	
4	Proveedor	A	30	
5	Importe	N		
6	Fecha	D		

Propiedades de tabla:

Controles de validación

☐ 1 Campo necesario

☒ Aceptar ☐ Ayuda

EGRESOS

Información de estructura Paradox para Windows Tabla: FACTURAS.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Clave
1	Num factura	N		*
2	Num factura	N		
3	Importe	N		
4	Fecha exp	D		
5	Fecha pago	D		
6	Contrarecibo...	N		

Propiedades de tabla:

Controles de validación

☐ 1 Campo necesario

☒ Aceptar ☐ Ayuda

FACTURAS

Información de estructura de BASE III Tabla: HIS_PARO.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Dec
1	PARO	N	6	0
2	ULTIMO_EX	D		
3	ENFER_GRAV	C	30	
4	HOSPITALI	C	30	
5	DOCTOR	C	30	
6	NOM_OPERA	C	30	
7	INFLAMA	N	8	0
8	MOV_DENT	C	40	
9	PROF_BOL	C	40	
10	FURCAS_INV	C	40	
11	DES_FACETA	C	40	
12	OTRAS	C	30	
13	PER_HUE_AN	C	40	

Propiedades de tabla:

Indices

Información detallada

Bloqueo de registros:

Tamaño

Aceptar Ayuda

HIS_PARO

Información de estructura de BASE III Tabla: HOR_CLIN.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Dec
1	DIAS	C	6	
2	LUNES	N	4	0
3	MARTES	N	4	0
4	MIERCOLES	N	4	0

Propiedades de tabla:

Indices

Información detallada

Bloqueo de registros:

Tamaño

Aceptar Ayuda

HOR_CLIN

Información de estructura Parados para Windows Tabla: HORA1.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Clave
1	Fecha	D		
2	Hora	N		
3	Lunes	A	10	
4	Martes	A	10	
5	Miercoles	A	10	
6	Jueves	A	10	
7	Viernes	A	10	
8	Sabado	A	10	

Propiedades de tabla:

Controles de validación

1 Campo necesario

Aceptar Ayuda

HORA1

Información de estructura dBASE III+ Tabla: IDENT.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Dec
1	IDENTIFICACION	N	6	0
2	FECHA	D		
3	NOMBRE	C	25	
4	EDAD	N	2	0
5	SEXO	C	1	
6	CIVIL	C	1	
7	HIJOS	N	2	0
8	DOMICILIO	C	25	
9	TEL	C	15	
10	OCUPACION	C	15	
11	RESP	C	20	
12	MEDICO	C	20	
13	TEL_MED	C	15	

Propiedades de tabla:

Indices

Información detallada

Bloqueo de registros:

1 Tamaño

Aceptar Ayuda

IDENT

Información de estructura Paradox para Windows Tabla: INVENTARIO

Definición de campo:

	Nombre campo	Tipo	Tamaño	Clave
1	Idioma	N		
2	Descripcion	A	30	
3	Precio compra	N		
4	Fecha	D		
5	Precio_promedio	N		
6	Existencia	N		
7	Dias surtido	N		

Propiedades de tabla:

Controles de validación ☐

☐ 1. Campo necesario

☒ Aceptar ☐ Ayuda

INVENTAR

Información de estructura Paradox para Windows Tabla: MAT_RE

Definición de campo:

	Nombre campo	Tipo	Tamaño	Clave
1	Idioma	N		*
2	SEMESTRE	A	10	
3	MATERIA	A	30	

Propiedades de tabla:

Controles de validación ☐

☐ 1. Campo necesario

☒ Aceptar ☐ Ayuda

MAT_RE

Información de estructura dBASE III Tabla: MATERIAS.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Dec
1	CLAVE	C	6	
2	MATERIA	C	15	
3	SEMESTRE	N	2	0
4	MAESTRO	C	34	
5	CLINICAS	C	4	

Propiedades de tabla:

Indices

Información detallada

☒ Aceptar ☐ Ayuda

Bloqueo de registros:

1 tamaño

MATERIAS

Información de estructura dBASE IV Tabla: PLAN_TR.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Dec
1	SEMENTE	N	6	0
2	TRAT	M		
3	REQUER	C	12	
4	LAB	C	40	
5	MED	C	40	
6	INTER	C	15	
7	RECU	C	15	
8	COMUNI	C	15	
9	RESTAUR	C	15	
10	MALPO	C	15	
11	INCLU	C	15	
12	RAIRE	C	15	
13	RESRA	C	15	

Propiedades de tabla:

Indices

Información detallada

☒ Aceptar ☐ Ayuda

Bloqueo de registros:

1 tamaño

PLAN_TR

Información de estructura DBASE III Tabla: REC_CONS.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Dec
1	NUMERO	N	6	0
2	NUM_EXP	N	6	0
3	FECHA	D		
4	CONCEPTO	C	25	
5	CANTIDAD	N	2	0
6	PRECIO	N	12	2
7	IMPORTE	N	12	2
8	MAT_OP	N	6	0
9	NOM_OP	C	40	
10	MES	N	1	0

Propiedades de tabla:

Indices

Información detallada

Bloqueo de registros:

☐ Tamaño

Aceptar Ayuda

REC_CONS

Información de estructura DBASE III Tabla: REC_GRAT.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Dec
1	NUMERO	N	6	0
2	NUM_EXP	N	6	0
3	NOM_PAC	C	30	
4	FECHA	D		
5	CONCEPTO	C	25	
6	CANTIDAD	N	2	0
7	PRECIO	N	12	2
8	IMPORTE	N	12	2
9	MAT_OP	N	6	0
10	NOM_OP	C	40	
11	MES	N	1	0

Propiedades de tabla:

Indices

Información detallada

Bloqueo de registros:

☐ Tamaño

Aceptar Ayuda

REC_GRAT

Información de estructura dBASE IV Tabla: REC_H1.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Dec
1	NUM_EXP	N	6	0
2	NUM_EXP	N	6	0
3	NOM_PAC	C	30	
4	FECHA	D		
5	CONCEPTO	C	25	
6	CANTIDAD	N	2	0
7	PRECIO	N	12	2
8	IMPORTE	N	12	2
9	MAT_OP	N	6	0
10	MES	N	1	0

Propiedades de tabla:

Indice

RII_FECHA
RII_NUM

Información detallada

Bloqueo de registros:

Tamaño

✓ Aceptar ? Ayuda

REC_H1

Información de estructura dBASE IV Tabla: REC_HIST.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Dec
1	NUM_EXP	N	6	0
2	NUM_EXP	N	6	0
3	NOM_PAC	C	30	
4	FECHA	D		
5	CONCEPTO	C	25	
6	CANTIDAD	N	2	0
7	PRECIO	N	12	2
8	IMPORTE	N	12	2
9	MAT_OP	N	6	0
10	MES	N	1	0

Propiedades de tabla:

Indice

RII_FECHA
RII_NUM

Información detallada

Bloqueo de registros:

Tamaño

✓ Aceptar ? Ayuda

REC_HIST

Información de estructura dBASE IV Tabla: REC1.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Dec
1	NUMERO	N	6	0
2	NUM_EXP	N	6	0
3	NOM_PAC	C	30	
4	FECHA	D		
5	CONCEPTO	C	25	
6	CANTIDAD	N	2	0
7	PRECIO	N	12	2
8	IMPORTE	N	12	2
9	MAT_OP	N	6	0
10	MES	N	1	0

Propiedades de tabla:

Indice

RH_FECHA
RH_NUM

Información detallada

Bloqueo de registros:

Tamaño

✓ Aceptar ? Ayuda

REC1

Información de estructura dBASE III Tabla: RECIBOS.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Dec
1	NUMERO	N	6	0
2	NUM_EXP	N	6	0
3	NOM_PAC	C	30	
4	FECHA	D		
5	CONCEPTO	C	25	
6	CANTIDAD	N	2	0
7	PRECIO	N	12	2
8	IMPORTE	N	12	2
9	MAT_OP	N	6	0
10	NOM_OP	C	40	
11	MES	N	1	0

Propiedades de tabla:

Indice

Información detallada

Bloqueo de registros:

Tamaño

✓ Aceptar ? Ayuda

RECIBOS

Información de estructura Paradox para Windows Tabla: REPO193.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Clave
1	Identificación	N		*
2	Cobro	D		
3	Importe	N		

Propiedades de tabla:

Controles de validación

☐ 1 Campo necesario

☒ Aceptar ☐ Ayuda

REPO193

Información de estructura Paradox para Windows Tabla: REPUESTO.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Clave
1	Identificación	N		*
2	Cobro	D		
3	Importe	N		

Propiedades de tabla:

Controles de validación

☐ 1 Campo necesario

☒ Aceptar ☐ Ayuda

REPUESTO

Información de estructura Paradox para Windows Tabla: REVISTA.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Clave
1	NUMERO	N		*
2	TITULO	A	30	
3	AUTOR	A	30	
4	RESUMEN	A	40	
5	FECHA	D		
6	CONTENIDO	F		
7	Imagen	G		

Propiedades de tabla:

Controles de validación

☒ 1 Campo necesario

Aceptar Ayuda

REVISTA

Información de estructura dBASE III Tabla: SILLONES.DBF

Definición de campo:

	Nombre campo	Tipo	Tamaño	Dec
1	FECHA	D		
2	HORA	N	5	2
3	HORA_FIN	N	5	2
4	CLINICA	C	1	
5	MATERIA	C	15	
6	SEMESTRE	N	1	0
7	DOCTOR	C	30	
8	SILLON	N	2	0
9	MATRICULA	N	6	0
10	NOMBRE	C	40	

Propiedades de tabla:

Indices

Información detallada

Bloqueo de registros:

Tamaño

Aceptar Ayuda

SILLONES

Información de estructura Paradox para Windows Tabla: TRATS.DIB

Definición de campo:

	Nombre campo	Tipo	Tamaño	Clave
1	Identificación	A	25	
2	Costo	N		
3	Lista material	A	30	
4	Consultas	N		

Propiedades de tabla:

Controles de validación 

| | Campo necesario

 Aceptar  Ayuda

TRATS

CONTENIDO DE ESTA GUIA

INTRODUCCIÓN

Qué hace el sistema?
Qué equipo necesito?
Cómo se instala?
Cómo usar esta guía?
Qué significan los tipos de letra?
Qué significan los iconos?

INDICACIONES GENERALES

Partes del sistema
Módulos que lo integran
Elementos del sistema
Ventanas
Menús
Botones
Campos
Informes
Textos
Mensajes
Uso de los menús
Funcionamiento
Menús y submenús del sistema

DESCRIPCIÓN FUNCIONAL DE LOS MÓDULOS

CAJA
CAPTURA
DIRECTOR
ALUMNOS
REVISTA DIGITAL

QUE HACE EL SISTEMA?

Este sistema fue desarrollado para auxiliar al personal de la Clínica de Odontología de la U.P.A.E.P. en las áreas: Contable, Administrativa, Clínica y Académica.

Con este sistema se optimiza el manejo de la información generada y utilizada en la clínica, de la siguiente manera:

1: REDUCE EL TIEMPO DE REGISTRO DE DATOS

Los datos generales de los pacientes, alumnos, catedráticos, etc. se registran una sola vez de acuerdo a la frecuencia de cambio, esto es:

- * Los datos de los pacientes solamente se registran cuando llegan por primera vez a la clínica, después solamente se modifican en caso de ser necesario.

- * Los datos de los catedráticos se registran solamente cuando se contratan, cada semestre se modifican las materias impartidas y los grupos en los que se imparten las mismas.

- * Los datos de los alumnos se registran cuando el plan de estudios establece que deben empezar sus prácticas clínicas, se modifican las materias en las que se inscribe cada semestre y sus calificaciones mensuales y semestrales.

- * Los datos de las tablas auxiliares como: tratamientos, precios, horarios, materias por grupo, etc. se registran una sola vez y se modifican en caso de ser necesario.

2: AGILIZA PROCESOS

Algunos procesos toman mucho tiempo, trámites y papeles, ésto los hace tardados y poco eficaces, ejemplos de ésto son:

- * **Manejo de expedientes:** Normalmente resulta difícil controlar los expedientes tanto en su espacio físico como la seguridad, exactitud oportunidad, etc. de la información, lo mas difícil es saber si un paciente ya ha sido registrado y que expediente le corresponde, esto se soluciona mediante el uso de tarjetas de consulta, credenciales, etc. pero aun así no se puede garantizar que el paciente conserve éstos y de cualquier manera tomará tiempo localizarlos. En un sistema como éste, los expedientes se encuentran archivados en un dispositivo magnético que garantiza su localización rápida y al mismo tiempo se archivan de manera física por seguridad y para hacer posible el préstamo para consulta.

*** Manejo de calificaciones:** Las calificaciones se van registrando periódicamente con el fin de controlar el aprovechamiento del alumno y facilitar la tarea de resumen y reporte del mismo para la Escuela de Odontología, esto evita el proceso de promedio manual y los errores derivados del mismo.

*** Reportes contables:** Tanto el registro como la sumarización del efectivo manejado por la caja se realizan de forma automática, de éste modo se reduce el trabajo y tiempo y aumenta la exactitud de la información entregada al departamento de contraloría.

*** Manejo de inventario:** Resulta difícil y en ocasiones imposible determinar cuanto se gasta en material, cuando se ocupa y cuando se debe volver a comprar. El registro de materiales y su uso se hace de manera automática también, lo cual ayudará a la dirección a establecer políticas de compra, uso y/o rotación de inventario.

3: INFORMA SUFICIENTE Y OPORTUNAMENTE

Cada departamento e individuo de personal tiene acceso a información suficiente sobre aquellos tópicos que sean de su interés e incumbencia, de manera sencilla y bien organizada, ya sea en forma de reporte, tabla o gráfica.

4 :BRINDA ELEMENTOS NUEVOS DE ENSEÑANZA

Por medio de la revista digital, -única en la universidad- es posible informar sobre asuntos generales, poner a disposición de alumnos y personal, información de interés o asesorar de manera general sobre algún tema en particular por medio de artículos que incluyen texto -el cual se puede tomar de cualquier archivo de texto, incluyen los artículos de discos compactos como los que cuenta la biblioteca- y gráficas ya sea hechas a mano, copiadas de libros, revistas o fotos, o digitalizadas desde una cámara.

QUE EQUIPO NECESITO

Existen diferentes configuraciones con las que puede utilizar el sistema, dependiendo del uso que le desee dar, necesita:

Para utilizar **UN MODULO FUNCIONAL** (Caja, Director, Catedráticos, Alumnos) y el **MODULO DE CAPTURA**:

- Una computadora personal con procesador 386 o superior
- 8 MB en RAM mínimo
- Disco duro de 102 MB mínimo
- Monitor VGA color o B/W
- Dispositivo señalador (Mouse, trackball, etc)
- Impresora
- Windows 3.1 o posterior
- Paradox para windows

Para utilizar **TODOS LOS MÓDULOS FUNCIONALES** (Caja, Director, Catedráticos, Alumnos) y el **MODULO DE CAPTURA**:

- Dos computadoras personales con procesador 386 o superior
- 8 MB en RAM mínimo
- Disco duro de 80 MB mínimo
- Monitor VGA color o B/W
- Dispositivo señalador (Mouse, trackball, etc.).
- Impresora.
- Windows 3.1 o posterior.
- Paradox para windows.
- Sistemas operativos para red (NOVELL Netware 2.0 o superior).
- Tarjetas para red.
- Cables para red.

Para utilizar el **MODULO DE REVISTA DIGITAL** en cualquiera de los casos anteriores necesita:

- Disco duro de 200 MB mínimo.
- Monitor VGA color mínimo.
- Scanner y/o tarjeta digitalizadora de video.

COMO SE INSTALA

Para instalar el equipo consulte con su proveedor o con nuestro equipo de soporte técnico.

Para instalar Windows y Paradox siga sus propios instructivos o consulte a nuestro equipo de soporte técnico.

Puede realizar la instalación del sistema siguiendo nuestras instrucciones o bien pedir la asistencia de nuestro equipo de soporte técnico.

INSTRUCCIONES DE INSTALACION

- Introduzca el disco **INSTALAR** en el drive A o B
- Cambie el directorio actual a ese drive tecleando a: o b: y [Enter]
- Teclee "Instalar"<disco destino> y oprima [Enter], ejemplo
A:> Instalar C:
- A continuación verá aparecer diferentes nombres de archivos, espere hasta que aparezca la frase "Oprima [Enter] para continuar...", Pulse la tecla señalada.
- Una vez hecho esto, usted puede empezar a trabajar con el sistema, para ello:
- Cambie el directorio actual al disco donde instaló el programa.
- Teclee "Win pdoxwin" y oprima [Enter].

COMO USAR ESTA GUÍA

Usted puede leer esta guía como un libro, esto es, de principio a fin, o bien, consultar aquellos tópicos que sean de su interés, sin embargo, le recomendamos que antes de empezar a saltar entre los capítulos lea la descripción y las indicaciones generales del sistema.

En la sección de **INDICACIONES GENERALES**, se describen los conceptos necesarios para entender el funcionamiento del sistema, estos incluyen manejo del entorno Windows y del sistema en sí, si Ud. está familiarizado con Windows, puede evitar esta sección, aunque no estaría mal repasarla y al mismo tiempo familiarizarse con el manejo del sistema. Si empieza a tener errores al intentar manejar el sistema, lea cuidadosamente ésta sección o la del módulo que está empleando, también puede consultar ésta información en las opciones de ayuda del sistema.

En el capítulo de **DESCRIPCIÓN FUNCIONAL DE LOS MÓDULOS**, encontrará para cada módulo:

- Descripción general del módulo
- Opciones del menú
- Instrucciones de uso.
- Ejemplo de operación

QUE SIGNIFICAN LOS TIPOS DE LETRAS?

MAYÚSCULAS: Se emplean para hacer referencia a secciones del manual o módulos del programa.

**** ÍNDICE:** Se refiere al índice del manual.

**** "COMILLAS":** Texto tal como aparece en la pantalla.

"Oprima una tecla para continuar"

- Texto en pantalla

[Corchetes] : Nombres de teclas.

"Oprima [Enter] para seguir..."

- Oprima la tecla con la etiqueta "Enter".

(Paréntesis): Texto que Ud. puede sustituir por lo necesario, por lo general es el nombre de lo que tiene que escribir.

(Nombre del paciente)

- Escriba un nombre.

QUE SIGNIFICAN LOS ICONOS?

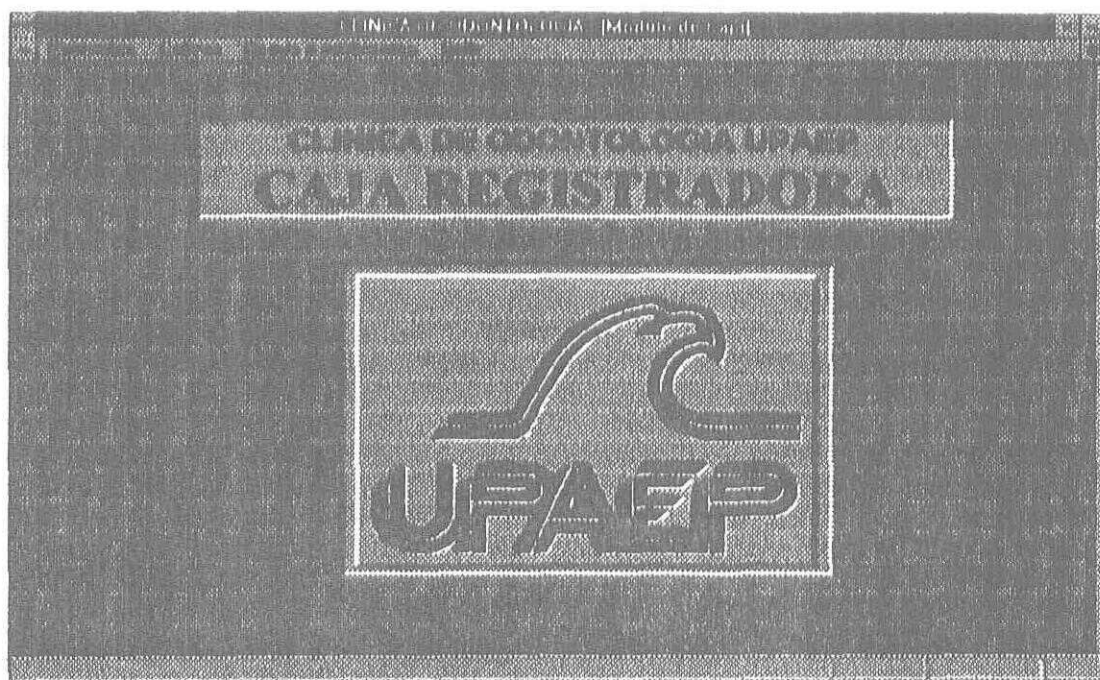
El icono [****Impresora] indica que lo que aparece en la pantalla se puede imprimir y da las instrucciones para ello.

El icono [****Rayo] Indica la manera mas corta de hacer algo.

El icono [****Admiración] indica una nota especial.

El icono [****ManoStop] indica una nota sobre la cual debe poner especial atención, ya que si hace algo puede causar un desperfecto irreparable en las bases de datos o el sistema mismo.

El icono [****Ratón] indica que ésta opción del sistema sólo se puede ejecutar con el uso del ratón (mouse).



PARTES DEL SISTEMA (MÓDULOS QUE LO INTEGRAN)

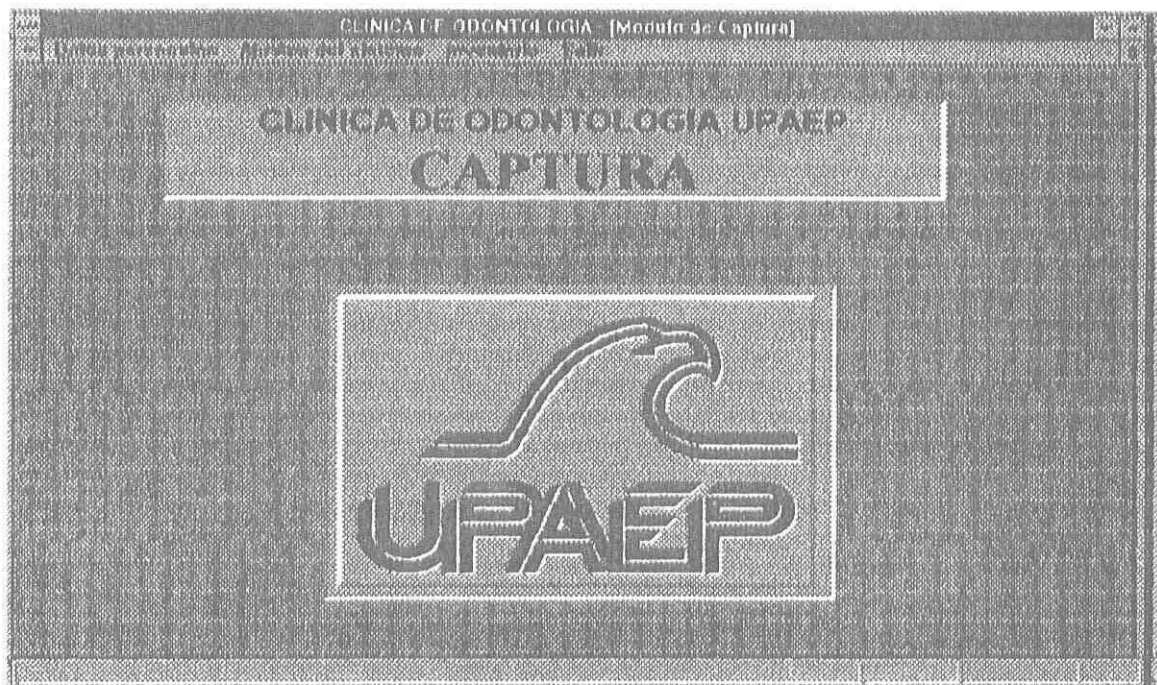
1: MODULO DE CAJA

Esta parte imita el funcionamiento de una caja registradora, y fue creada para facilitar el manejo de efectivo dentro de la clínica, pensando en las necesidades del personal del área de Caja-recepción, contempla las siguientes opciones:

PACIENTES COBROS APOYO A PROFESORES SALIR



2: MODULO DE CAPTURA:

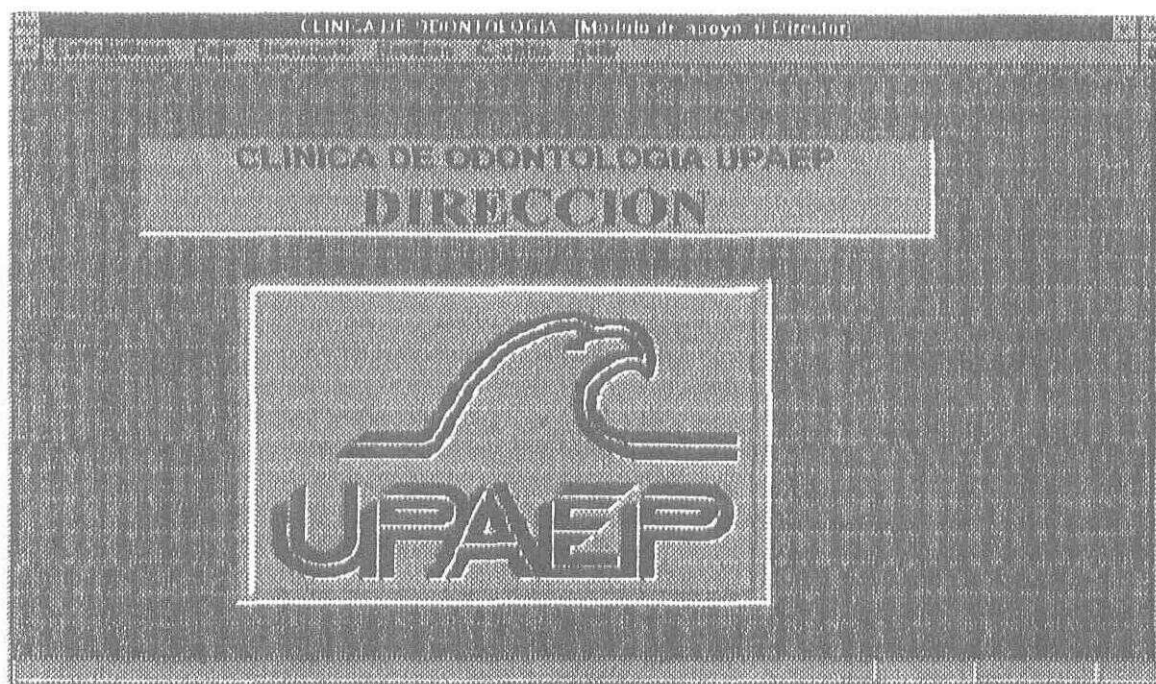


El sistema requiere de muchos datos que no es necesario proporcionar en línea, y que además se pueden agrupar para reducir el tiempo necesario para su introducción. Estos datos incluyen:

DATOS GENERALES DE LOS PACIENTES, ANTECEDENTES MÉDICOS, DATOS DE LOS ALUMNOS, DATOS DE LOS CATEDRÁTICOS, REVISTA DIGITAL, HORARIO DE LA CLÍNICA, CALIFICACIONES, AJUSTES DEL SISTEMA

Para mas información sobre este módulo consulte el capítulo DESCRIPCIÓN FUNCIONAL DE LOS MÓDULOS.

3: MODULO DE APOYO AL DIRECTOR

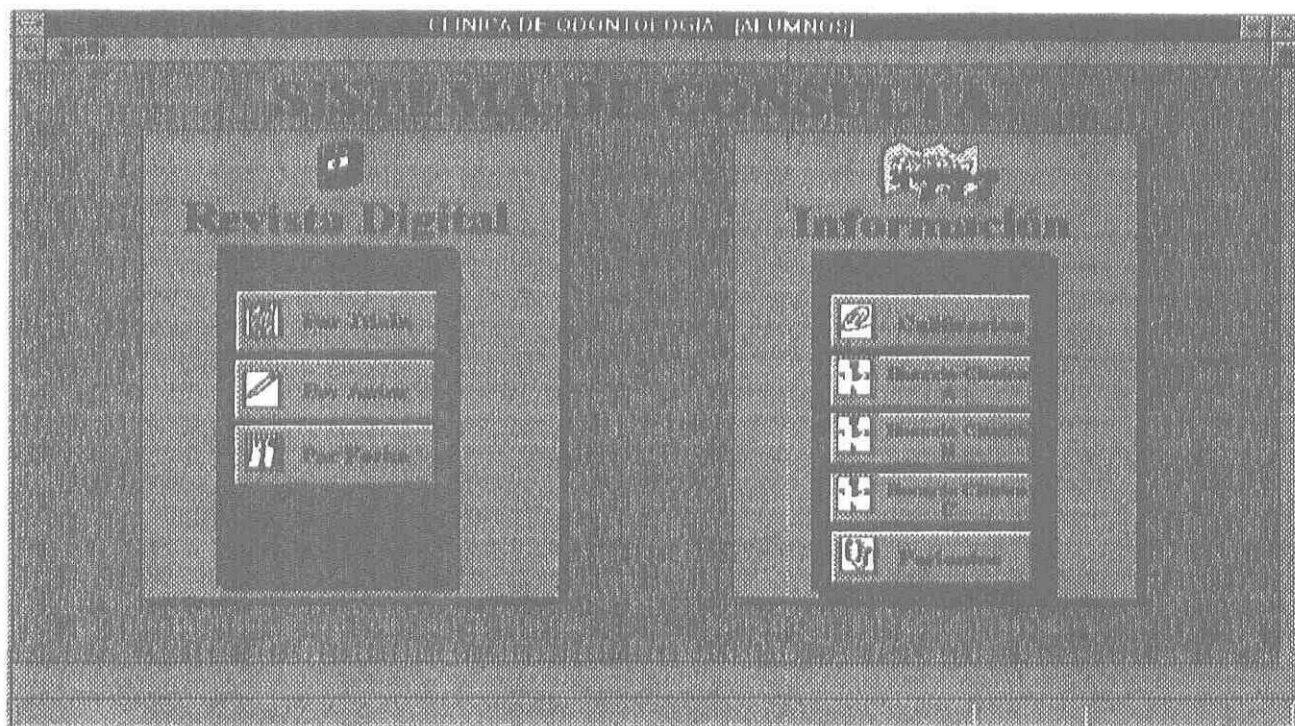


Esta parte se creó con el fin de apoyar al personal directivo en la toma de decisiones, brindándole información relevante y resumida de manera gráfica o de reporte. En cualquier momento el director o personal autorizado puede consultar:

GRÁFICA DE CASOS, GRÁFICA DE POBLACIÓN, GRÁFICA DE INVENTARIO, EFECTIVO EN CAJA, INGRESOS HISTÓRICOS, REVISTA DIGITAL, COMPRAS, VALUACIÓN Y CAMBIO AL ARCHIVO DE INVENTARIO.

Para más información sobre este módulo consulte el capítulo DESCRIPCIÓN FUNCIONAL DE LOS MÓDULOS.

5: MODULO DE CONSULTA



Es un sistema por sí mismo, en el cual, los alumnos tienen acceso a cierta información que es de su interés, con un diseño de pantalla bien estudiado que incluye manejo de gráficas digitalizadas y su manejo es sencillo e intuitivo por medio de botones virtuales, para su implementación se recomienda un sistema de autoconsulta con un mueble especial que impida el mal uso. Entre la información que puede consultar está:

CONSULTA DE LA REVISTA DIGITAL POR TITULO, AUTOR O FECHA

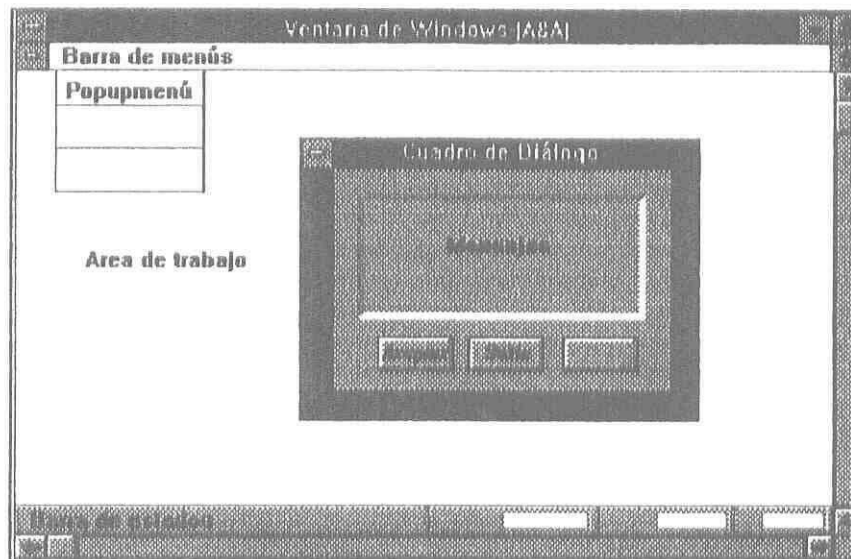
CONSULTA DE CALIFICACIONES, HORARIOS Y PACIENTES

Para más información sobre este módulo consulte el capítulo
DESCRIPCIÓN FUNCIONAL DE LOS MÓDULOS.

(****NOTA) La opción de modificación a la revista a la que tienen acceso el director, catedáticos y capturistas se describe como un módulo aparte en la sección de DESCRIPCIÓN FUNCIONAL DE LOS MÓDULOS.

ELEMENTOS DEL SISTEMA:

VENTANAS:



Las ventanas se pueden definir como secciones de la pantalla en las cuáles se muestra información, cuando utilizamos una ventana se limita la escritura de caracteres en el monitor a esa área, debido a lo cual es posible ver sólo los caracteres que quepan en ella.

LAS VENTANAS POSEEN CIERTAS CARACTERISTICAS:

- **Marco:** Es una línea o líneas de forma rectangular que limitan el área utilizable de la pantalla, ese marco se puede estirar o encoger en cualquier dirección con el fin de cambiar el tamaño del área de la ventana, esto ocurre cuando el cursor del mouse se posiciona sobre este marco y su forma de flecha se cambia por una flecha de doble punta, entonces sin soltar el botón del mouse se estira la ventana hasta el tamaño que se desee, soltando hasta ese momento el botón del mouse.

- **Título:** En un renglón en la parte superior de la ventana aparece el título o nombre de la misma, ésta área no se puede utilizar para escribir caracteres en ella.

- **Botón de control:** Es un cuadro en la esquina superior izquierda del marco, dentro del renglón de título. Sirve como un botón virtual (V. Botones), al accionarlo muestra un menú de opciones que incluyen

- **Maximizar:** Ampliar el tamaño de la ventana al máximo posible.

- **Minimizar:** Cambiar el tamaño al mínimo posible o mostrar sólo un icono con el título de la ventana.

- **Restaurar:** Cambiar el tamaño de la ventana al anterior.

- **Cerrar:** Al cerrar una ventana desaparece de la pantalla y se detiene la acción que se estaba realizando en ella.

- **Tamaño:** Cuando no se cuenta con un ratón, se utiliza ésta opción y las teclas de flechas para modificar el tamaño.

- **Cambiar a:** Podemos dejar de utilizar una ventana por un momento y mientras utilizar otra, podemos cambiar de ventana cuantas veces sea necesario.

- **Botón maximizar:** Está situado en el renglón de título en la esquina superior derecha, muestra una punta de flecha hacia arriba. Funciona igual que la opción "Maximizar" del botón de control, cuando la ventana está maximizada, se convierte en la opción restaurar y muestra dos puntas de flecha, una hacia arriba y otra hacia abajo, cuando la ventana está minimizada, desaparece y sólo se puede usar el botón control.

- **Botón minimizar:** Está situado en el renglón de título en la esquina superior derecha, a la izquierda del botón maximizar, muestra una punta de flecha hacia abajo. Funciona igual que la opción "Minimizar" del botón de control, cuando la ventana está minimizada, desaparece y sólo se puede usar el botón control.

- **Barras de desplazamiento:** Cuando el contenido de una ventana excede su tamaño, se muestran un renglón en la parte inferior del marco y una columna en la parte derecha, las cuales muestran que área del contenido se está visualizando por medio de un cuadro pequeño, esto es, si el cuadro se encuentra en la orilla derecha de la barra inferior, se muestra el final de renglón de el contenido, si está a la izquierda se muestra el inicio del renglón, de la misma manera, si el cuadro está en la parte superior de la barra derecha se muestra el inicio del contenido y si está en el extremo inferior se muestra el final del contenido, las barras permiten, por medio de botones en las orillas, "mover la ventana de modo que se visualice otra área del contenido, para entender esto:

Corte en una hoja, cartón o similar un rectángulo pequeño en el centro y luego colóquelo(a) sobre una página escrita, sólo podrá ver un área de lo escrito, si quiere ver lo demás tiene que mover el rectángulo descubierto en esa dirección.

- **Barra de estado:** Esta barra en realidad es una pequeña ventana que abarca un renglón en la parte inferior de la ventana, sobre la barra de desplazamiento horizontal. En esta ventana se muestran mensajes del sistema para el usuario que le indican que está sucediendo, o cualquier indicación pertinente.

(****Nota) En la barra de estado se muestran mensajes cuando se necesita que haga algo o cuando ocurre un error. No la pierda de vista.

MENÚS

Los menús en general son grupos de palabras que le indican las acciones que es posible realizar desde un sistema, una ventana, un módulo etc.

Existen básicamente dos tipos de menús:

A: MENÚS DE LINEA O DE DESPLIEGUE HORIZONTAL:

Que aparecen en una sola línea o renglón en la parte superior de la ventana, bajo el título de la ventana.

B: MENÚS SUBYACENTES O DE DESPLIEGUE VERTICAL:

Que aparecen en tantas líneas como opciones contengan.

Para activar una de esas acciones, se debe "seleccionar" la palabra que exprese lo que se desea hacer, esto se hace de la siguiente manera:

SELECCIÓN DE OPCIONES DE UN MENÚ

Existe una "barra de selección", que es un renglón o sección del menú en un color diferente y contrastante al resto del menú, éste(a) enmarca una de las palabras, dicha palabra se convierte en la "opción seleccionada" es decir que al oprimir la tecla [Enter], se inicia la acción que describe esa palabra.

USO DE LOS MENÚS

Para iniciar una acción incluida en un menú usted debe:

1- Oprima la tecla [F10] para llegar al menú.

2- Mueva la "barra de selección" a la palabra que desee por medio de las teclas de flechas.

3- Presione la tecla [Enter]

4- En algunos casos, aparecer otro menú conteniendo mas opciones, siga los pasos 2 y 3 para seleccionar la acción deseada.

(****Paso rápido) También puede seleccionar una opción oprimiendo la tecla [Alt] y al mismo tiempo, oprimir también la tecla que contenga la **LETRA SUBRAYADA** de la palabra que describe la acción deseada.

(****Nota) La letra subrayada de una opción no siempre es la primera, si desea seleccionar esa opción debe utilizar [Alt]+[<letra subrayada>], si oprime [Alt]+[<primera letra>] puede que seleccione otra opción cuya letra subrayada coincide con la primera letra, Ejemplos:

Si aparece un menú con las palabras "Escribir" y "Leer" y Ud. desea escribir, debe: Oprimir la tecla [F10] y, puesto que la "Barra de selección" está sobre la palabra "Escribir", solo debe oprimir la tecla [Enter]. O bien: Oprimir la tecla [Alt] junto con la tecla [E].

Si aparece un menú con las palabras "Escribir" y "Estudiar", y Ud. desea estudiar, debe: Oprimir la tecla [F10], mover la "Barra de selección" hasta que cubra la palabra "Estudiar" y, oprimir la tecla [Enter]. O bien: Oprimir la tecla [Alt] y la tecla [S]. (Si oprime [Alt]+[E], en lugar de [Alt]+[S] se accionar la opción "Estudiar")

BOTONES VIRTUALES

Los botones virtuales son objetos gráficos que simulan la acción de un botón real, esto es: Cuando Ud. presiona un botón real este se mueve hacia abajo y con esto se inicia una acción, por ejemplo, si oprime un botón de un interruptor se enciende o apaga una luz o un aparato.

Al igual que los botones reales, algunos se mantienen oprimidos y otros vuelven a su posición original (suben), dependiendo de la acción que inicie un botón se dividen en:

- **Botones de comando:** Que son los que se oprimen y vuelven a su posición y que inician una acción, esto se asemeja a la selección de opciones de un menú.

- **Botones de radio:** Son áreas de la pantalla que no imitan el movimiento de botones reales, pero inician una acción.

- **Botones de verificación:** No inician ninguna acción, sólo aparece una marca o cambian de color, lo cual indica que se accionó el botón.

Independientemente de su tipo, todos los botones cuentan con una "Etiqueta", la cual es una palabra o frase que se muestra sobre ellos o a un lado y que indica de manera breve la acción que se iniciará al "oprimir" ese botón.

(****Nota) Las teclas de una computadora son los botones reales más parecidos a los virtuales, pues tienen una etiqueta y se oprimen y vuelven a su posición original.

USO DE LOS BOTONES:

Para iniciar la acción relacionada con un botón, y que se describe en su etiqueta, se debe:

1: Enfocar el botón, esto es: posicionar el ratón sobre el dibujo del botón, o bien oprimir la tecla [Tab] hasta que el botón aparezca enmarcado o con un rectángulo de distinto color rodeando la etiqueta.

2: "Oprimirlo", esto es: accionar el botón izquierdo del ratón o bien oprimir la tecla [Enter] o [Space].

CAMPOS

Los campos son, por lo general, espacios de almacenamiento donde se archivan datos, que forman parte de los "registros", sin embargo, como elementos gráficos, son áreas de las ventanas donde se muestra la información contenida en los campos del archivo.

Los campos pueden contar o no con una etiqueta que describe el nombre o tipo de la información que muestran.

TABLAS

Las tablas son un conjunto de líneas y renglones en los cuales se muestra la información contenida en un archivo cuando se requiere mostrar más de un dato, por lo general cuentan con un renglón de etiquetas que muestran los nombres de los campos que se incluyen, en cada renglón se muestran datos de un individuo en particular.

Si se incluyen más campos o líneas de las que caben en una ventana, la tabla también muestra barras de desplazamiento. (V. VENTANAS)

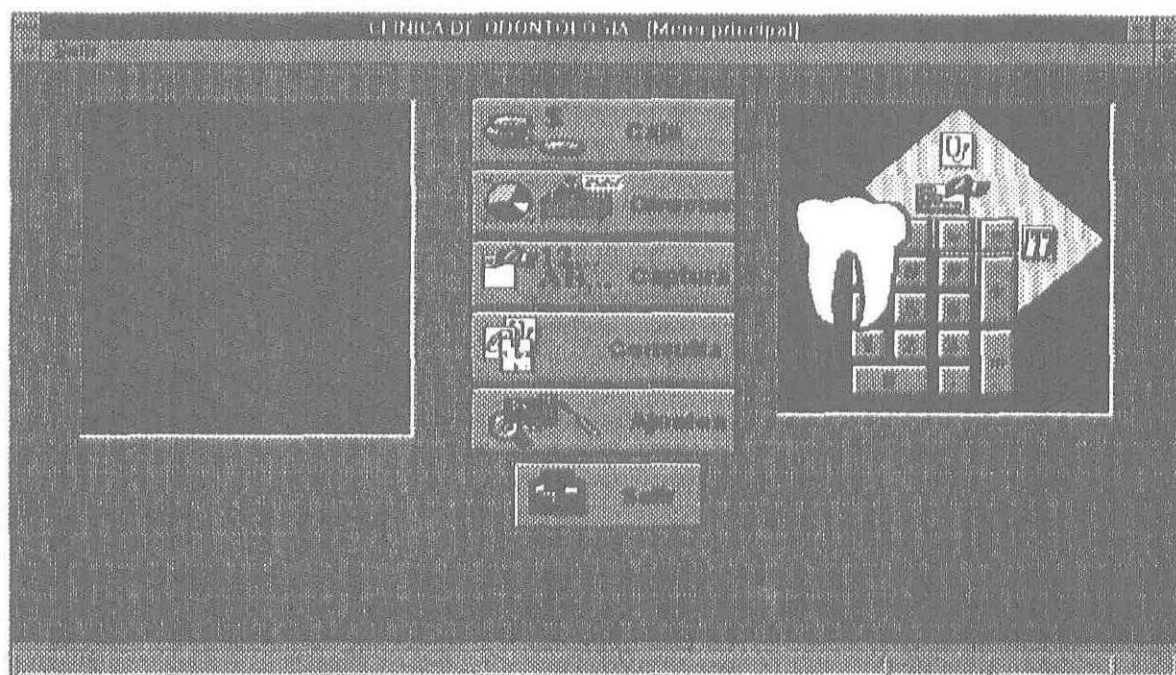
INFORMES

Son un tipo especial de tabla, en el cual se muestra información resumida en forma de líneas y columnas, los informes se pueden imprimir y por lo general se muestran como resultado de consultas específicas que no incluyen a todos los individuos de un archivo, sino sólo los que cumplen con una característica dada.

DESCRIPCIÓN FUNCIONAL DE LOS MÓDULOS

1: MENÚ PRINCIPAL

CARATULA:



La carátula del Sistema de Control de la Clínica de Odontología U.P.A.E.P. presenta seis botones virtuales, los cuales tienen una función específica dentro del Sistema, al pulsarlos ya sea con el mouse o con las teclas abre una ventana que muestra las operaciones que se pueden realizar en esa parte del sistema.

A continuación se explicarán cada una de las funciones que realizan estos botones, para ayudar al usuario a tener una mejor interrelación con el Sistema y así sacarle el máximo provecho al mismo (en un apartado anterior se explica la manera de accesar botones y menús con mouse y con teclas).

En la ventana inicial aparece el menú principal del sistema:

Seleccione el módulo que desea utilizar oprimiendo botón izquierdo del ratón sobre la palabra que describe el módulo deseado.

Esta pantalla aparece cada vez que termine de usar un módulo, desde aquí puede abandonar el sistema y cerrar la ventana por medio del botón de control o seleccionando "Salir" del menú principal.

- Oprima el botón "Salir" o...
- Cierre la ventana por medio del botón de control.

CAJA:



Fig. No. 1

Si pulsamos el botón virtual de esta opción se llega a una pantalla que se detalla en la fig. (1), el menú que presenta esta pantalla es el siguiente:

Paciente Cobros Apoyo a profesores Salir

A continuación se explicará la función de cada uno de los componentes del menú:

PACIENTES:

Al seleccionar esta operación aparece un submenú que contiene las siguientes opciones; si selecciona:

REIMPIMIR HISTORIA CLINICA : Se realiza la siguiente función:

Aparece un cuadro de diálogo que solicita el número de expediente que se desee reimprimir; una vez que se haya introducido el número de expediente se pulsa el botón de [aceptar], la computadora solicita la confirmación de que la impresora se encuentre lista y posteriormente la computadora automáticamente reimprime el expediente.

IMPRIMIR LISTA DE DE PACIENTES: Se realiza la siguiente función:

Al seleccionar esta opción se procesa un informe, el cual manda a impresión (se tiene que recordar que la impresora se encuentre lista para imprimir).

PARA IMPRIMIR:

- Seleccione imprimir del menú.
- Aparece un cuadro de diálogo con la configuración de la impresora.
- Oprima "Aceptar", "Cancelar" o "Configurar" según desee.
- Espere sin pulsar teclas o botones mientras esté en pantalla el cuadro de diálogo "Imprimiendo".

PARA SALIR:

- Seleccione "Salir" del menú o..
- Cierre la ventana por medio del botón de control.

COBROS:

Si pulsamos el botón virtual de esta opción se llega a una pantalla del cual el menú que presenta es el siguiente:

Cobro Normal Corte de Caja Reimprimir Recibo Reporte del Día

A continuación se explicará la función de cada uno de los componentes del menú:

COBRO NORMAL: Se realiza la siguiente función:

The screenshot shows a software interface titled "COBRO NORMAL". It contains several input fields for patient information: "Paciente", "Fecha", "Hr de Cobro", "Distribuir Rec. Expediente", and "Ver lista de pacientes". Below these are fields for "Cobro", "Monto", "Monto", and "Monto". At the bottom, there is a menu with four options: "Cobro", "Corte de Caja", "Reimprimir Recibo", and "Reporte del Día".

Al seleccionar esta opción, aparece una pantalla en la que solicita los datos del paciente y del operador así como el concepto, cantidad e importe, al a vez tiene un par de botones virtuales con los que se pueden realizar dos operaciones distintas [Paciente por número] y [Ver lista de pacientes] los cuales, facilitan la búsqueda de algún paciente en caso de no saber su número de expediente o su nombre, cuando no se sabe el nombre del paciente se ocupa [Ver lista de pacientes], entonces el sistema despliega la lista de los pacientes existentes desde el primero que se haya introducido en la base de datos del sistema hasta donde alcance la página o pantalla del monitor, para ver los siguientes pacientes se puede hacer por página pulsando la tecla Página Abajo [Av. Pag] o Página Arriba [Re. Pag] para ir a la página siguiente o anterior según sea el caso. Otra manera de hacer un desplazamiento hacia arriba o hacia abajo es con el mouse pulsando la barra de desplazamiento rápido que se encuentra a la derecha de la lista de pacientes.

En caso de saber el nombre correcto del paciente y cómo está escrito dentro del sistema se puede buscar de una manera aún más sencilla: A continuación se detalla:

1: Se posiciona el cursor (barra sombreada de color gris oscuro) en el campo con el cual se desea realizar la búsqueda (éste método sirve para hacer la búsqueda por el número de expediente o nombre del pacientes) en este caso lo posicionaremos en el campo **NOMBRE** ya sea con las teclas o con el mouse.

2: Con el mouse se va al menú REGISTRO.

3: Cuando aparece el popuymenú se selecciona la opción BUSCAR.

4: Se selecciona a su vez la opción VALOR e inmediatamente después un cuadro de diálogo solicita el valor con el que se va a buscar el paciente y se pulsa [Aceptar] si los datos son correctos el cursor se posiciona en el registro que se haya elegido, si los datos que se le dieron a la computadora son incorrectos o no se encuentra el valor la computadora simplemente no hace nada y regresa al menú anterior.

Esta ventana cuenta con otros tres botones los cuales tienen las siguientes funciones:

GUARDAR Y SEGUIR:

Este botón tiene la función de almacenar en un archivo los datos del paciente para posteriormente hacer consultas en el expediente del paciente, posteriormente si se desea imprimir el recibo y la historia clínica del paciente la computadora pregunta si ese es el caso, solicitando la confirmación de que la impresora se encuentre lista para imprimir, si no se desea imprimir simplemente se cancela la operación de impresión.

SERVICIO GRATUITO:

Esta opción guarda en un archivo los datos del paciente sin cargar en Caja los costos por los servicios prestados, ya que la Clínica de Odontología tiene la facultad de prestar servicios gratuitos a diferentes instituciones de beneficencia, por lo que es necesario la utilización de esta opción.

CANCELAR Y SALIR:

Con esta opción se tiene la facilidad de no guardar los datos del paciente en el archivo de pacientes y salir de la aplicación.

CORTE DE CAJA:

Cuando se elige esta opción aparece la pantalla de cobros del día la cual desgloza de manera detallada todos los movimientos que se hallan realizado en el día indicando fecha, número de recibo, matrícula del operador, número que se le asignó al paciente para su expediente, el concepto del pago, la cantidad de trabajos que pagó el paciente, el importe que pagó por cada uno de los servicios y el importe total que pagó el paciente.

A su vez esta ventana cuenta con tres botones que realizan las siguientes funciones:

[Imprimir corte] Al seleccionar esta opción la computadora solicita la confirmación de que esté lista la impresora e inmediatamente después imprime el reporte de los movimientos del día.

[Salir y Cerrar Históricos] Al seleccionar esta opción automáticamente la computadora cierra el corte del día y sale de la ventana de cobros del día, dejando vacío el archivo de cobros del día.

[Salir sin Históricos] Esta opción sale de la ventana sin hacer el corte de caja dejando el archivo de cobros sin hacerle ninguna modificación.

REIMPRIMIR RECIBO:

Al seleccionar esta opción la computadora solicita el número del recibo que se desea volver a imprimir y al proporcionarlo se pulsa el botón de [Aceptar], la computadora solicita la confirmación de que la impresora esté lista e imprime el recibo seleccionado.

REPORTE DEL DIA:

Al seleccionar esta opción la computadora realiza un informe de los movimientos que se hayan realizado durante el día, solicitando la confirmación de que se encuentre lista la impresora.

APOYO A PROFESORES:

Al seleccionar esta operación aparece un submenú que contiene las siguientes opciones; si selecciona:

APARTADO DE UNIDADES:

The screenshot shows a window titled "Unidades para Clase" with a grid of input fields. The grid is organized into four columns and five rows. The columns are labeled at the top: "FECHA", "HORA INICIAL", "HORA FINAL", and "MATERIA". Each column contains five input fields. Below the grid, there are two buttons: "Aceptar" and "Cancelar".

Al seleccionar esta opción la computadora abre una ventana que se llama "Unidades para Clase" la cual solicita los siguientes datos:

- Fecha:** Se le da a la computadora la fecha en que se desea apartar la unidad. Por default la computadora da la fecha del día siguiente a la fecha actual, si se desea cambiar la fecha, con el mouse se posiciona el cursor en este campo para poder hacer los cambios pertinentes.
- Hora inicial:** En este campo se introduce la hora a la que inicia la materia dentro de las clínicas.
- Hora final:** En este campo se introduce la hora en la que finaliza la materia dentro de las clínicas.
- Semestre:** En este campo se introduce el semestre a la que pertenece la materia.
- Materia:** En este campo se introduce el nombre de la materia en la que se está haciendo el apartado.
- Doctor:** En este campo aparece el nombre del doctor que está a cargo de la materia en cuestión, pero la computadora automáticamente despliega el nombre del doctor.

Inmediatamente después el cursor se posiciona en el primer sillón de la clínica a la que esté asignada la materia en cuestión, ya que cuando una materia no ocupa las tres clínicas disponibles al mismo tiempo con la misma materia, la computadora despliega una letra mayúscula en la parte izquierda de la pantalla (si la materia está asignada sólo a la clínica A la computadora desplegará una "A" al lado izquierdo de la pantalla, y así sucesivamente con las demás materias dentro de las otras clínicas). El cursor se puede mover dentro de ésta área ya sea con las flechas o con el mouse.

El botón de **[Seguir]** sirve para guardar los cambios realizados hasta el momento y limpiar la pantalla y poder apartar otro sillón en otra hora, fecha o materia.

El Botón de **[Terminar]** sirve para guardar los cambios efectuados hasta el momento y salir de ésta aplicación regresando a la ventana anterior.

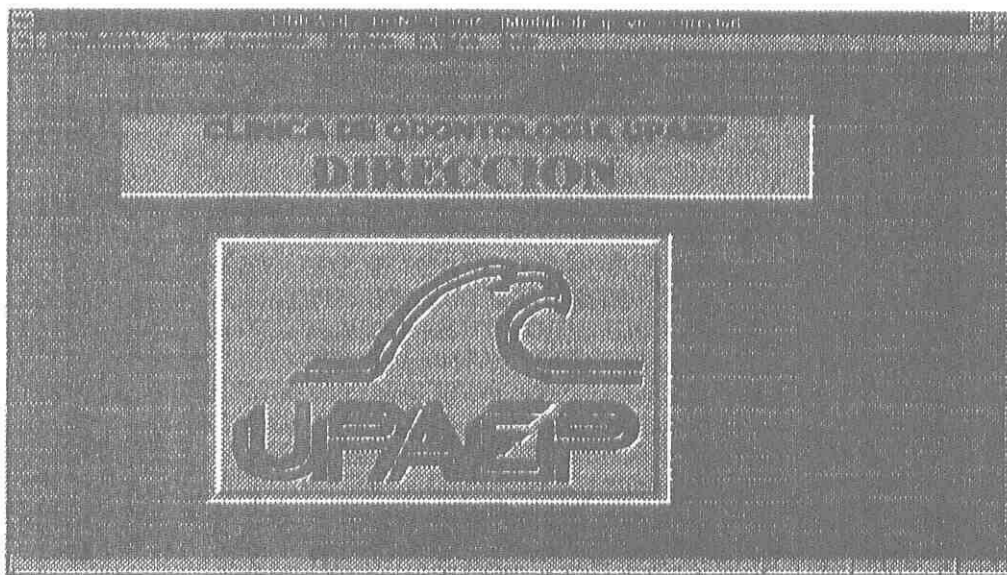
NOTA: Cabe aclarar que para dejar esta ventana sin los datos desplegados antes de "Salir" o "Terminar" pulsemos el botón de "Terminar" para que la próxima vez que se utilice ésta aplicación se abra la ventana sin ningún dato desplegado.

LISTAS:

Esta opción sirve para imprimir la lista de los alumnos que hayan apartado unidad con anterioridad por lo que la computadora abre un cuadro de diálogo el cual solicita la fecha que se desea imprimir una vez que se le haya dado se le da el botón de **[Aceptar]** y posteriormente la computadora solicita la hora que se desea imprimir al pulsar el botón de **[Aceptar]** la computadora solicita la confirmación de que la impresora se encuentre lista para realizar esta operación y posteriormente se realiza la operación de impresión.

SALIR:

Esta opción simplemente sale de la ventana del módulo de caja, regresando a la ventada principal del Sistema.

DIRECTOR:

Este es el segundo botón virtual de la ventana principal del sistema. Al pulsar este botón la computadora despliega la ventana del módulo de apoyo al director la cual tiene las siguientes opciones:

Estadísticas Caja Inventarios Recibos C. Chica Salir

Seleccione la acción que desea utilizar pulsando el botón izquierdo del ratón sobre la palabra que describe esa acción. (Existen otras formas de activar una operación en el menú, consúltelas en la sección de ELEMENTOS DEL SISTEMA, en la parte de MENÚS).

ESTADISTICAS:

Al ocupar esta opción la computadora despliega un submenú el cual contiene otras opciones las cuales son:

CONTEO DE CASOS:

Con esta opción la computadora abre una ventana llamada "conteo de casos" la cual despliega la cantidad de cada uno de los casos que se hayan atendido durante el mes en curso, a la vez que se tiene la opción de imprimir el reporte.

PARA IMPRIMIR:

- Seleccione imprimir del menú.
- Aparece un cuadro de diálogo con la configuración de la impresora.
- Oprima "Aceptar", "Cancelar" o "Configurar" según desee.

- Espere sin pulsar teclas o botones mientras esté en pantalla el cuadro de diálogo "Imprimiendo".

PARA SALIR:

- Seleccione "Salir" del menú o..
- Cierre la ventana `por medio del botón de control.

GRAFICA POR CASOS:

Esta opción abre la ventana de "Gráfica de casos", la cual despliega una gráfica de barras en tercera dimensión que presenta información resumida sobre los casos que se han atendido en la clínica desde su inicio.

PARA IMPRIMIR:

- Seleccione imprimir del menú.
- Aparece un cuadro de diálogo con la configuración de la impresora.
- Oprima "Aceptar", "Cancelar" o "Configurar" según desee.
- Espere sin pulsar teclas o botones mientras esté en pantalla el cuadro de diálogo "Imprimiendo".

PARA SALIR:

- Seleccione "Salir" del menú o..
- Cierre la ventana `por medio del botón de control.

GRAFICA POR POBLACION:

Esta opción despliega una gráfica de barras la cual indica la cantidad de pacientes que se han atendido agrupada por tipos de población que se han atendido en la clínica desde su inicio.

PARA IMPRIMIR:

- Seleccione imprimir del menú.
- Aparece un cuadro de diálogo con la configuración de la impresora.
- Oprima "Aceptar", "Cancelar" o "Configurar" según desee.
- Espere sin pulsar teclas o botones mientras esté en pantalla el cuadro de diálogo "Imprimiendo".

PARA SALIR:

- Seleccione "Salir" del menú o..
- Cierre la ventana `por medio del botón de control.

GRAFICA POR INVENTARIO:

Esta opción despliega una gráfica la cual indica la cantidad de artículos que se encuentran en existencia de una manera gráfica y fácil de entender.

PARA IMPRIMIR:

- Seleccione imprimir del menú.
- Aparece un cuadro de diálogo con la configuración de la impresora.
- Oprima "Aceptar", "Cancelar" o "Configurar" según desee.
- Espere sin pulsar teclas o botones mientras esté en pantalla el cuadro de diálogo "Imprimiendo".

PARA SALIR:

- Seleccione "Salir" del menú o..
- Cierre la ventana por medio del botón de control.

CAJA:

Esta utilidad es para el director de la carrera, la cual tiene las siguientes opciones.

EFFECTIVO:

Esta opción sirve para cuando desee saber cuáles han sido los movimientos que han tenido en la clínica durante el día, ya que despliega los siguientes datos: Número de recibo, Concepto de pago, Cantidad de trabajos realizados, Precio por cada uno de los trabajos e Importe total que paga cada uno de los pacientes, así como un total de efectivo que debe coincidir con lo que se encuentra en caja en ese momento.

PARA IMPRIMIR:

- Seleccione imprimir del menú.
- Aparece un cuadro de diálogo con la configuración de la impresora.
- Oprima "Aceptar", "Cancelar" o "Configurar" según desee.
- Espere sin pulsar teclas o botones mientras esté en pantalla el cuadro de diálogo "Imprimiendo".

PARA SALIR:

- Seleccione "Salir" del menú o..
- Cierre la ventana por medio del botón de control.

HISTORICOS DEL MES:

Esta opción despliega una ventana en la que indica las operaciones que se han realizado durante el mes en curso, es de suma utilidad porque ayuda al Director saber que es lo que ha pasado en la caja y a cuánto asciende el dinero que hay en caja.

PARA IMPRIMIR:

- Seleccione imprimir del menú.
- Aparece un cuadro de diálogo con la configuración de la impresora.
- Oprima "Aceptar", "Cancelar" o "Configurar" según desee.
- Espere sin pulsar teclas o botones mientras esté en pantalla el cuadro de diálogo "Imprimiendo".

PARA SALIR:

- Seleccione "Salir" del menú o..
- Cierre la ventana por medio del botón de control.

HISTORICO POR FECHA:

Esta opción realiza lo mismo que la anterior sólo con la salvedad que pide una fecha específica y despliega las operaciones que se hallan realizado en ese día.

PARA IMPRIMIR:

- Seleccione imprimir del menú.
- Aparece un cuadro de diálogo con la configuración de la impresora.
- Oprima "Aceptar", "Cancelar" o "Configurar" según desee.
- Espere sin pulsar teclas o botones mientras esté en pantalla el cuadro de diálogo "Imprimiendo".

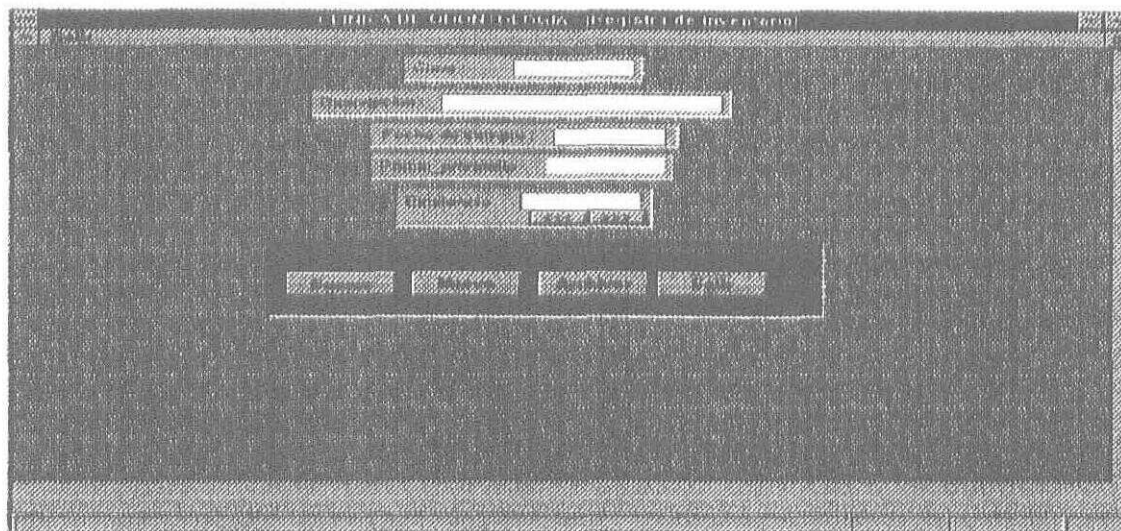
PARA SALIR:

- Seleccione "Salir" del menú o..
- Cierre la ventana por medio del botón de control.

INVENTARIO:

Esta opción tiene el siguiente submenú:

COMPRAS Y CAMBIOS:



En esta ventana se introducen o modifican los datos de los artículos que se compran, utilizan o cambian del inventario, el sistema calcula automáticamente el precio promedio con fines de valuación.

PARA INTRODUCIR UN ARTICULO:

- Pulse el botón "Nuevo".
- El sistema le asigna la clave del artículo.

PARA LOCALIZAR UN ARTÍCULO:

- Oprima el botón "Buscar".
- Aparece un cuadro de diálogo que le solicita la clave.
- Escriba la clave del artículo.
- Oprima el botón "Aceptar" o presione [Enter]

PARA INTRODUCIR O MODIFICAR DATOS:

- Escriba los datos solicitados en todos los campos, puede moverse a cualquier campo por medio de la tecla [TAB] para pasar al siguiente campo, [Shift] + [TAB] para pasar al campo anterior, o las teclas de flechas.

PARA ARCHIVAR LOS DATOS:

- Oprima el botón "Archivar".
- Realice una búsqueda de artículo.
- Salga de la ventana.

PARA SALIR:

- Seleccione "Salir" del menú o ...
- Oprima el botón "Salir" o ...
- Cierre la ventana por medio del botón de control.

VALUACION:

Esta opción realiza un informe resumido de los artículos en existencia en el inventario, así como la valuación de los mismos por el sistema de precios promedio.

PARA IMPRIMIR:

- Seleccione imprimir del menú.
- Aparece un cuadro de diálogo con la configuración de la impresora.
- Oprima "Aceptar", "Cancelar" o "Configurar" según desee.
- Espere sin pulsar teclas o botones mientras esté en pantalla el cuadro de diálogo "Imprimiendo".

PARA SALIR:

- Seleccione "Salir" del menú o..
- Cierre la ventana por medio del botón de control.

RECIBOS:

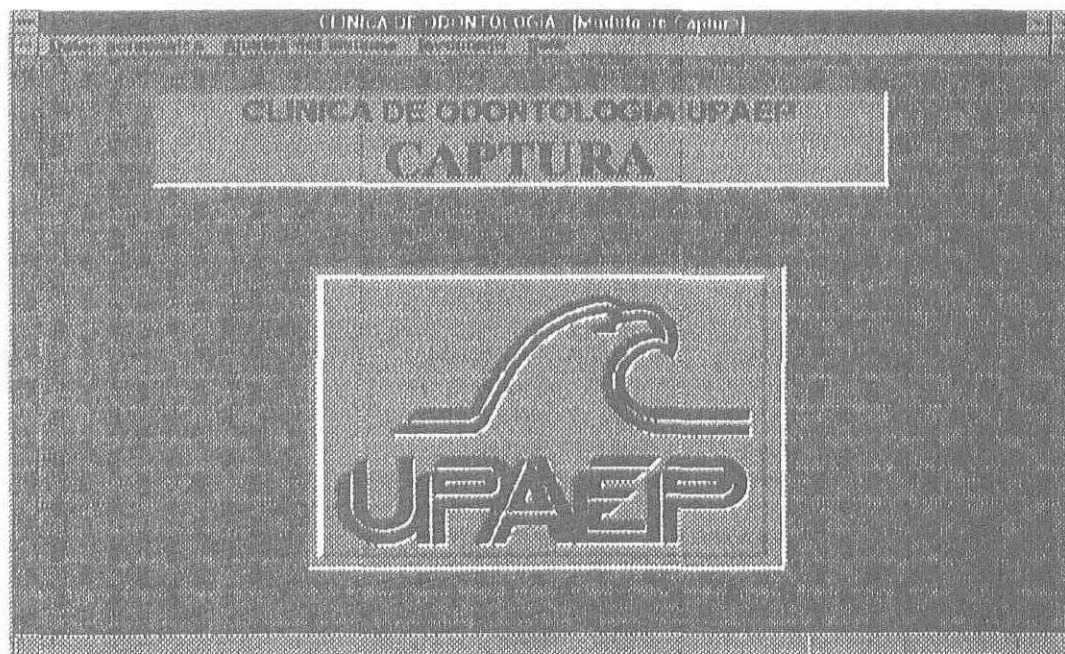
Esta opción despliega un cuadro de diálogo que solicita el número de expediente que desea revisar el director o algún catedrático, si no se sabe el número de expediente también se puede buscar el expediente por medio del nombre del paciente.

CAJA CHICA:

EGRESOS:

Esta pantalla sirve para almacenar los datos de las compras que se efectúan, El **número** lo da la computadora por default, la **fecha** se toma de la fecha de la computadora, en **Ultima Reposición** se guarda el último folio en que se hizo una compra del mismo artículo, en **Proveedor** se guarda el nombre del proveedor al que se le hizo la compra, **Importe** se guarda la cantidad que costó el artículo. El **botón de Guardar seguir** guarda la información que se tenga en ese momento en la ficha, y se continúa trabajando con esta ventana. **Cancelar/Salir** no guarda ninguna información y sale de la aplicación. **Buscar/Modificar** localiza alguna nota que se haya hecho por medio del folio que dió la computadora y también se puede modificar.

CAPTURA:



Este módulo sirve para almacenar los datos de los pacientes que se encuentran dados de alta en la Clínica como de los alumnos que están inscritos en la escuela de Odontología y a continuación se explican las características del módulo.

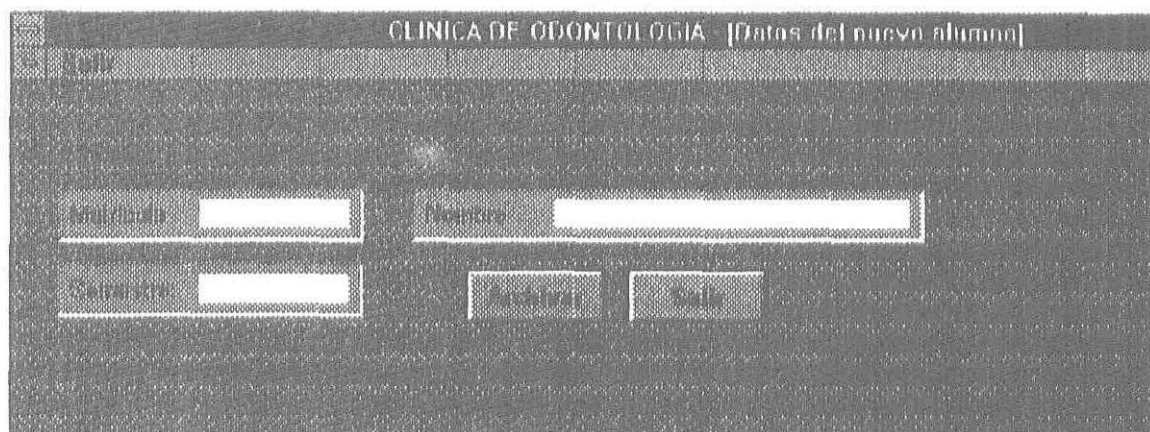
PACIENTES:

Sirve para introducir los datos de los pacientes en las diferentes historias clínicas que se manejan dentro de la clínica de odontología, los cuales son:

GENERAL, CIRUGIA, PARODONCIA, PROSTODONCIA, EXODONCIA, PLACA Y SARRO, PERIODONTAL.

Cada una de estas historias clínicas guarda distintos datos de los pacientes dependiendo de las necesidades que tenga el paciente.

ALUMNOS:



The screenshot shows a window titled "CLINICA DE ODONTOLOGIA [Datos del nuevo alumno]". Inside the window, there are three text input fields arranged vertically on the left side, labeled "Matricula", "Nombre", and "Semestre". To the right of these fields, there are two buttons: "Archivar" and "Salir". The interface has a dark, textured background.

Este módulo sirve para dar de alta en el sistema a nuevos alumnos, y los datos que solicita el sistema son:

Matricula, Nombre y Semestre

Al mismo tiempo que tiene los botones para **Archivar** los datos del alumno así como el de **salir** de esta aplicación.

AJUSTES DEL SISTEMA:

Este módulo tiene la función de hacer modificaciones al sistema cada que sea necesario, ya que estos cambios son necesarios hacerlos cada semestre, se ha implementado este módulo, los cambios que se pueden hacer son los siguientes:

Materias:

Como se cambia de materias cada semestre, se pueden modificar las claves, los nombres, profesores que imparten las materias y el semestre al que pertenecen las materias.

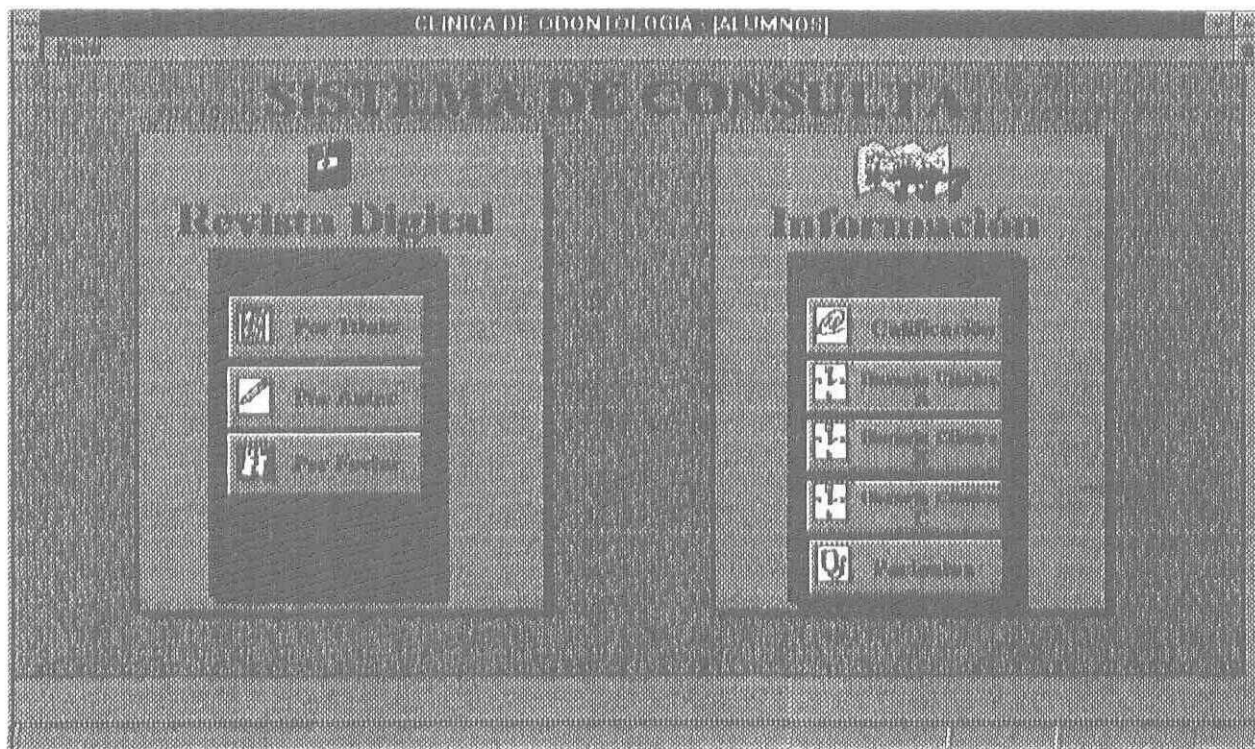
Tratamiento:

En caso de que cambiara el nombre de los tratamientos o se ampliara la gama de servicios que presta la clínica desde esta aplicación se puede realizar dicha operación.

Horarios:

Los horarios de las clínicas A, B y C se cambian continuamente por lo que es necesario que se encuentre una aplicación de cambio de horario dentro del sistema, se puede escoger cuál de las clínicas es la que se desea modificar.

CAPTURA:



Este módulo fué diseñado con el objetivo de dotar a los alumnos con una herramienta de información y parendizaje, en el que se integra un sistema de consulta de datos donde pueden tener acceso a cierta cantidad de información que les incumbe e interesa y al moderno sistema único en la Universidad de REVISTA DIGITAL.

PARA CONSULTAR INFORMACION PERSONAL SOBRE:

CALIFICACIONES:

- Oprima el botón "Calificaciones".
- Aparece el cuadro de diálogo que solicita la matrícula.
- Escribir la matrícula.
- Oprimir el botób "Aceptar" o presionar [Enter].
- Aparece una ventana con las calificaciones del alumno.
- En esa ventana hay un solo botón, el de "Salir".
- Cuando el alumno acaba de leer la información sale por medio de ese botón y regresa al módulo de alumnos.

HORARIO:

- Oprimir el botón "Horario", (Dependiendo de la clínica que se trate A, B o C).
- Aparece una ventana con la tabla del horario.
- En esa ventana hay un solo botón, el de "Salir".
- Regresa al módulo de alumnos.

PACIENTES:

- Oprimir el botón "Pacientes".
- Aparece un cuadro de diálogo que solicita la matrícula.
- Escribir la matrícula.
- Oprimir el botón "Aceptar" o presionar [Enter]
- Aparece una ventana con los datos del (los) pacientes(s) que el alumno está atendiendo actualmente y su(s) próxima(s) cita(s).
- En esa ventana hay un solo botón, el de "Salir".
- Cuando el alumno acaba de leer la información sale por medio de ese botón y regresa al módulo de alumnos.

PARA CONSULTAR LA REVISTA:

BUSCANDO EN EL INDICE POR TITULO:

- Oprimir el botón "Por Titulo".
- Aparece una ventana con una lista de artículos de acuerdo al título.
- escoger el título por medio de las teclas de arriba o abajo y presionando la tecla [Enter] u oprimiendo el botón "Aceptar".
- Aparece una ventana con los datos del artículo, el texto del mismo y un recuadro donde aparece una imagen a la vez que han sido archivadas junto con el texto del artículo.
- En esa ventana hay un solo botón, el de "Salir".
- Cuando el alumno acaba de leer la información sale por medio de ese botón y regresa al módulo de alumnos.

BUSCANDO EN EL INDICE POR AUTOR:

- Oprimir el botón "Por Autor".
- Aparece una ventana con una lista de autores disponibles.
- Escoger el autor por medio de las teclas de arriba o abajo y presionando la tecla [Enter] u oprimiendo el botón "Aceptar".
- Aparece una ventana con los datos del artículo del autor indicado, el texto del mismo y un recuadro donde aparece una imagen a la vez que han sido archivadas junto con el texto del artículo.
- En esa ventana hay un solo botón, el de "Salir".
- Cuando el alumno acaba de leer la información sale por medio de ese botón y regresa al módulo de alumnos.

BUSCANDO EN EL INDICE POR FECHA:

- Oprimir el botón "Por Fecha".
- Aparece una ventana con una lista de fechas disponibles.
- Escoger el título por medio de las teclas de arriba o abajo y presionando la tecla [Enter] u oprimiendo el botón "Aceptar".
- Aparece una ventana con los datos del artículo, el texto del mismo y un recuadro donde aparece una imagen a la vez que han sido archivadas junto con el texto del artículo.
- En esa ventana hay un solo botón, el de "Salir".
- Cuando el alumno acaba de leer la información sale por medio de ese botón y regresa al módulo de alumnos.

SALIR:

Esta opción está protegida con una clave de seguridad que no permite a los alumnos "Salir" del módulo para entrar en otros, en los cuales podría ocasionar daños o mal uso de la información, de modo que el personal autorizado pueda activar o desactivar el módulo.

- Seleccione "Salir" del menu.
- Aparece un cuadro de dialogo que solicita la clave de seguridad.
- Escriba la clave.
- Oprima el botón "Aceptar" o presione [Enter].
- Si la clave es correcta regresa al menú principal.

REVISTA (Sección Especial):

CLINICA DE ODONTOLOGIA - [Modificación de un artículo]

NOMBRE	NOMBRE
NOMBRE	
NOMBRE	
NOMBRE	
NOMBRE	

Guardar Cancelar

Volver a inicio Salir

Esta opción a la que tiene acceso: El capturista, el director y los catedráticos constituye un módulo en sí mismo y sirve para escribir y/o modificar artículos en la Revista Digital.

PARA CONSULTAR LA REVISTA:

BUSCANDO EN EL INDICE POR TITULO:

- Oprima el botón "Por titulo".
- Aparece una ventana con una lista de artículos de acuerdo al titulo.
- Escoja el titulo por medio de las teclas arriba y abajo y presionando la tecla [Enter] u oprimiendo el botón "Aceptar".
- Aparece una ventana con los datos del artículo, el texto del mismo y un recuadro donde aparece una imagen a la vez que han sido archivadas junto con el texto del artículo.
- Cuando acabe de leer el artículo oprima el botón "Salir" para regresar a la ventana de revista.

BUSCANDO EN EL INDICE POR FECHA:

- Oprima el botón "Por fecha".
- Aparece una ventana con una lista de fechas disponibles.
- Aparece una ventana con la lista de artículos de la fecha indicada.
- Escoja el título por medio de las teclas arriba y abajo y presionando la tecla [Enter] u oprimiendo el botón "Aceptar".
- Aparece una ventana con los datos del artículo, el texto del mismo y un recuadro donde aparece una imagen a la vez que han sido archivadas junto con el texto del artículo.
- Cuando acabe de leer el artículo oprima el botón "Salir" para regresar a la ventana de revista.

BUSCANDO EN EL INDICE POR AUTOR:

- Oprima el botón "Por autor".
- Aparece una ventana con una lista de autores disponibles.
- Seleccione el autor deseado por medio de las teclas arriba y abajo o con el ratón.
- Oprima el botón "Aceptar".
- Aparece una ventana con la lista de artículos del autor indicada.
- Escoja el título por medio de las teclas arriba y abajo y presionando la tecla [Enter] u oprimiendo el botón "Aceptar".
- Aparece una ventana con los datos del artículo, el texto del mismo y un recuadro donde aparece una imagen a la vez que han sido archivadas junto con el texto del artículo.
- Cuando acabe de leer el artículo oprima el botón "Salir" para regresar a la ventana de revista.

PARA ESCRIBIR ARTICULOS NUEVOS:

- Oprima el botón "Nuevo".
- Aparece una ventana para la introducción de datos.
- Escriba su nombre en el campo "Autor".
- Escriba el Título del "Artículo".

PARA INTRODUCIR EL TEXTO DEL ARTICULO:

- Muévase al campo "Contenido" por medio del ratón, la tecla [TAB], [Shift]+[TAB] o las teclas de flechas.
- Escriba el texto que quiera u oprima el botón "Texto" o seleccione "Editar" del menú y luego seleccione "Pegar texto".
- Aparecerá el manejador de archivos Paradox similar al de Windows (Consulte su manual de Paradox o Windows para aprender a usarlo), seleccione el archivo donde está guardada la imagen deseada.

- También puede cortar la imagen desde la aplicación de Windows donde la creó o importó y luego seleccionar "Editar" del menú y luego seleccione "Pegar".

- Después de introducir el artículo salga de la ventana oprimiendo el botón "Salir" del menú o cerrándola por medio del botón de control.

PARA MODIFICAR UN ARTICULO:

- Oprima el botón "Modificar".

- Aparece una ventana para introducción de datos.

PARA ENCONTRAR EL ARCHIVO A MODIFICAR:

- Oprima el botón "Siguiete" para ver el siguiente artículo o ...

- Accione el botón "Anterior" para ver el artículo anterior o ...

- Accione el botón "Buscar número", aparecerá un cuadro de diálogo donde se solicita el número y oprima [Enter] o el botón "Aceptar" o ...

- Accione el botón "Buscar título", aparecerá un cuadro de dialogo donde se solicita el Título, escriba el título y oprima [Enter] o el botón "Aceptar".

PARA MODIFICAR EL TEXTO:

- Muévase al campo "Contenido" por medio del ratón, la tecla [TAB], [Shift]+[TAB] o las teclas de flechas.

- Reescriba o modifique el texto que quiera, u oprima el botón "Texto" o seleccione "Editar" del menú y luego seleccione "Pegar Texto".

- Aparecerá el manejador de archivos Paradox similar al de Windows (Consulte su manual de Paradox o Windows para aprender a usarlo), seleccione el archivo donde está guardando el texto deseado.

PARA MODIFICAR LA IMAGEN DEL ARTICULO:

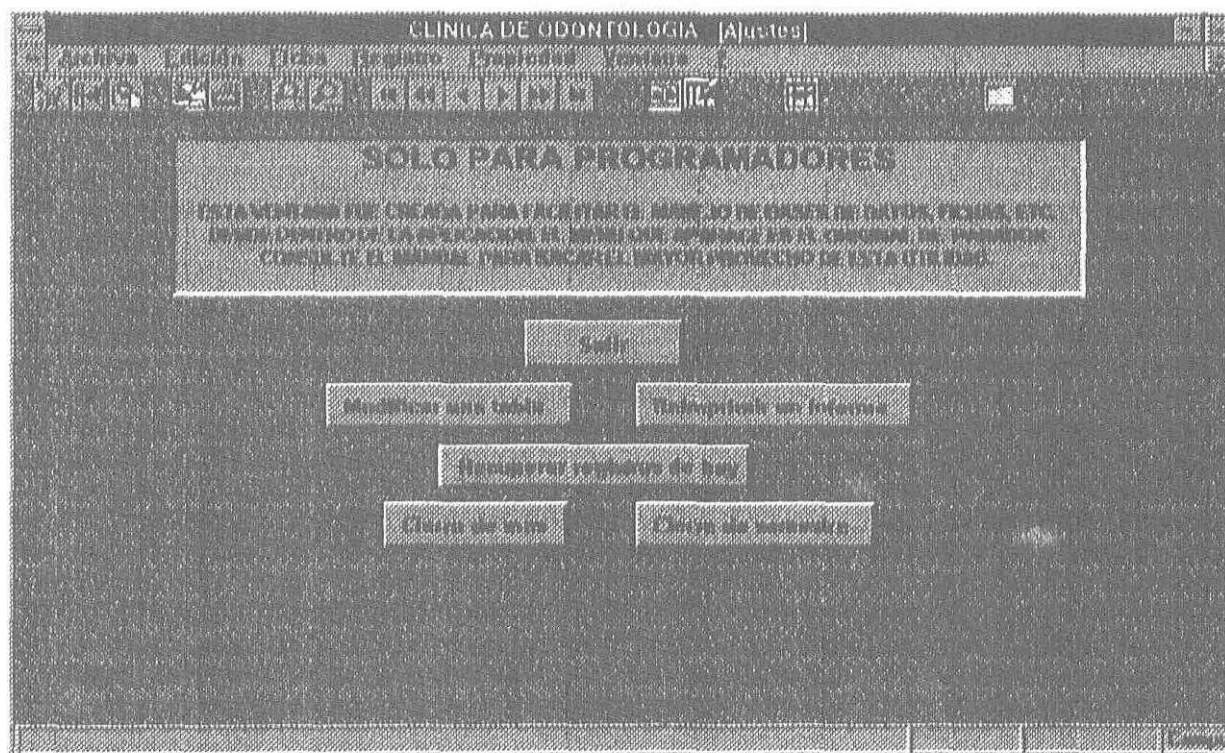
- Oprima el botón "Imagen" o seleccione "Editar" del menú y luego seleccione "Pegar imagen".

- Aparecerá el manejador de archivos Paradox similar al de Windows (Consulte su manual de Paradox o Windows para aprender a usarlo), seleccione el archivo donde está guardanda la imagen deseada.

- También puede cortar la imagen desde la aplicación de Windows donde la creó o importó y luego seleccionar "Editar" del menú y luego seleccione "Pegar".

- Después de terminar de modioficar el artículo salga de la ventana oprimiendo el botón "Salir" o seleccionando "Salir" del menú o cerrándola por medio del botón de control.

AJUSTES:



Esta aplicación fué creada para facilitar el manejo de bases de datos, fichas, etc. Está diseñada especialmente para programadores, es por ello que presenta la ventana principal de Paradox, si no sabe el manejo de este paquete es mejor que salga de la aplicación sin hacer modificaciones, deje ese trabajo a programadores especializados o a los diseñadores del sistema computacional.

9. CONCLUSIONES

Dado que en la clínica de odontología U.P.A.E.P. ahora se controla más eficientemente la información, de modo que es suficiente y oportuna para tomar decisiones con más certidumbre y en menos tiempo, se fortalecieron otras áreas de la administración y ahora se brinda un mejor servicio tanto a los pacientes como a los alumnos, además de que se controlan los ingresos y egresos de efectivo, así como los recursos materiales e insumos se concluye lo siguiente:

- Se determinaron puntos neurálgicos, problemas y cuellos de botella por medio de un análisis del sistema administrativo y del procesamiento de información de la clínica de odontología U.P.A.E.P. lo que ayudó a mejorar notablemente la administración y el proceso de información de la misma clínica, logrando un balance entre el modelo de control interno y el uso de las computadoras.

- El crear un sistema de cómputo que automatiza la captura, sumarización y recuperación de la información generada en la clínica ahora apoya la toma de decisiones.

Las empresas que ahora reducen la incertidumbre por medio del manejo adecuado de información pueden tomar decisiones inteligentes y conocer el resultado de éstas en menor tiempo por medio de una adecuada retroalimentación, son más eficientes como es ahora el caso de la clínica de odontología de la U.P.A.E.P.

- Las computadoras y la tecnología cuando es son asociadas, utilizadas adecuadamente y analizados de acuerdo con la relación Costo/Beneficio, influyen en gran medida a la efectividad de los sistemas de información, sobre todo cuando el volumen, la complejidad y las demandas de oportunidad y cálculo son elevados.

- Cuando el volumen de información manejado dentro de una empresa o institución es elevado, el método computarizado rebasa en efectividad a los métodos manual y electromecánico, afectando en gran medida a la efficientización de los recursos con los que se cuenta en las empresas, apoyando las tareas de control y la toma de decisiones como lo es en este caso la clínica de odontología mejorando y aumentando la productividad.

9.1.1 INDICE DE METODOS.

NOMBRE DEL METODO	No. PAGINA
Ajustes	1
Alumnos	4
Apartuni	8
Caja	27
Camb_art	30
Captura	32
Compras	35
Conteo	37
Control	38
Corregir	39
Corte	40
Ctrl_cli	42
Dat_pac	44
Datos	51
Director	53
Efectivo	57
Graf_cas	58
Graf_inv	59
Graf_pob	60
Historic	61
Ho_cli_a	62
Ho_cli_b	63
Iden2	64
Impr_rec	72
Normal	73
Plan2	75
Rec_cons	82
R_hist	82
Reg_cont	83
Reg_egre	85
Reg_ing	87
Reg_mate	89
Reg_pac	90
Reporte	98
Saldocc	99
Sel_indi	100
Tratams	104
Valua_in	105
Ver_pac	106

AJUSTES.FSL

Página: 1

Objeto	F_Ajustes
Método	open
Fuente	<pre>method open(var eventInfo Event) var opc string endvar if not(eventInfo Isprefilter())then maximize() „Habilitar el uso de la barra rápida para „ que aprovechar sus capacidades showspeedbar() endif endmethod</pre>
Objeto	F_Ajustes
Método	close
Fuente	<pre>method close(var eventInfo Event) „Al salir se esconde la barra rápida para evitar „ errores al seleccionar del menú hidespeedbar() endmethod</pre>
Objeto	F_Ajustes B_CierreSem
Método	pushButton
Fuente	<pre>„Este boton anexa todos los archivos de mes „ a uno de semetre y opcionalmente lo guarda „ en un disco flexible para eviar sobrecargar el disco duro method pushButton(var eventInfo Event) endmethod</pre>
Objeto	F_Ajustes B_Cierre
Método	pushButton
Fuente	<pre>„Este boton se activa cada fin de mes para „ enviar los recibos al archivo de mes method pushButton(var eventInfo Event) var numes number t1 tcursor nomtabla string endvar numes=month(date()-1) „Obtener el número de mes numes view("Número del mes que desea cerrar") „Copiar al archivo de mes los recibos adecuados „ y borrarlos del archivo total if numes>0 and numes<(month(date()+1) then if t1 open("rec_hist.dbf") then if t1 locate("mes",numes) then nomtabla=string("rec_h",int(numes)," dbf") if t1 copy(nomtabla) then t1 edit() while not(t1 eof()) reg=t1 recno() if t1 mes=numes then t1 deleterecord() else t1 skip(1) endif endwhile t1 compact() t1 close() if t1 open(nomtabla) then t1 edit() while not(t1 eof()) reg=t1 recno() if t1 mes<>numes then t1 deleterecord() else t1 skip(1) endif endwhile t1 compact() t1.close() endif endif endif endmethod</pre>
Objeto	F_Ajustes B_RelmpInfo
Método	pushButton

```

nom string
tab report
endvar
nom=""
nom view("Teclee el nombre del Informe")
„Obtener el nombre del Informe a Imprimir
if nom<>""then
„Abrir el Informe y posibilitar su Impresión
if tab open(nom) then
else
beep()
message("No existe ese reporte o no se puede abrir")
endif
endif
endmethod

```

Objeto F_Ajustes B_Recupera

Método *pushButton*

Fuente

```

„Este boton nos permite recuperar los registros del
„ dia actual cuando por error se envían a recibos
„ históricos, es útil para cerrar un corte al día siguiente
method pushButton(var eventinfo Event)
var
tab,t1 tcursor
valor string
num_ant number
endvar
num_ant=0
„ Abrir las tablas de recibos e históricos
if tab open("rec_hist.dbf") and t1 open("Recibos.dbf")then
tab home()
tab edit()
t1 edit()
„Buscar y recuperar cada registro de hoy
while not(tab eof())
reg=tab.reco()
if tab.fecha=date() then
t1 insertrecord()
t1 numero=tab numero
t1 fecha=tab fecha
t1 num_exp=tab num_exp
t1 nom_pac=tab nom_pac
t1 concepto=tab concepto
t1 precio=tab precio
t1 importe=tab importe
t1 mat_op=tab mat_op
t1 mes=tab mes
t1 postrecord()
tab deleterecord()
else
tab skip(1)
endif
endwhile
tab compact()
tab close()
t1 close()
msginfo("FIN","Proceso terminado")
else
message("No se puede abrir el archivo")
endif
endmethod

```

Objeto F_Ajustes B_ModTabla

Método *pushButton*

Fuente

```

method pushButton(var eventinfo Event)
var
nom string
tab tableview
endvar
nom=""
nom view("Teclee el nombre de la tabla")
„Obtener el nombre de la tabla a modificar
if nom<>""then
„Abrir la tabla y posibilitar la modificación
if tab open(nom) then
tab wait()
tab close()
else
beep()
message("No existe esa tabla o no se puede abrir")
endif
endif
endmethod

```

Objeto F_Ajustes B_Gallr

AJUSTES.FSL

Página: 3

Método

pushButton

Fuente

```
method pushButton(var eventInfo Event)
    „Cerrar la ficha al preionar este boton
    formreturn("SALIR")
endmethod
```

```

Objecto      #Página2

Método      Var
Fuente      Var
            linea string
            endVar

Objecto      #Página2

Método      open
Fuente      method open(var eventInfo Event)
            var
            prin menu
            endvar
            maximize()
            hidedesktopbar()
            prin addtext("&Salir")
            prin show()
            linea=""
            endmethod

Objecto      #Página2

Método      keyChar
Fuente      method keyChar(var eventInfo KeyEvent)

            endmethod

Objecto      #Página2

Método      menuAction
Fuente      method menuAction(var eventInfo MenuEvent)
            var
            clave,opc string
            endvar
            opc=eventInfo menuchoice()
            clave="xxxxx"
            if opc="&Salir" then
            formreturn("Salir")
            endif
            endmethod

Objecto      #Página2 #Cuadro34 #Cuadro7 #Botón18

Método      pushButton
Fuente      method pushButton(var eventInfo Event)
            var
            vent form
            endvar

            if vent.open("Ho_cli_c") then
            vent.wait()
            vent.close()
            else
            beep()
            message("Problemas al abrir un archivo")
            endif

            endmethod

Objecto      #Página2 #Cuadro34 #Cuadro7 #Botón6

Método      pushButton
Fuente      method pushButton(var eventInfo Event)
            var
            vent form
            endvar

            if vent.open("Ho_cli_b") then
            vent.wait()
            vent.close()
            else
            beep()
            message("Problemas al abrir un archivo")
            endif

            endmethod

Objecto      #Página2 #Cuadro34 #Cuadro7 #Botón12

Método      pushButton
Fuente      method pushButton(var eventInfo Event)
            var
            vent form
            endvar

```

```

if vent1.open("Ho_cll_a") then
    vent1.wait()
    vent1.close()
else
    beep()
    message("Problemas al abrir un archivo")
endif

endmethod

```

Objeto #Página2 #Cuadro34 #Cuadro7 #Botón26

Método pushButton

```

method pushButton(var eventInfo Event)
var
    matric number
    cons query
    tab table
    ver form
endvar
matric=0
matric.view("Teclea tu matrícula")
if matric>0 then
    message("Buscando tus pacientes")
    cons=query

    Paciente db | Clave | Nombre | Tratamiento | Mat_op | Fecha |
                | _clave| _nombre| _tratamiento| ~matric| _fecha|

    Consulta db | Clave | Nombre | Tratamiento | Fecha |
                | _clave| _nombre| _tratamiento| _fecha|

endquery
tab.attach("Consulta db")
tab.empty()
if executeQBE(cons, "Consulta db") then
    tab.attach("Consulta db")
    if tab.records()>0 then
        message("")
        ver.open("ver_pac")
        ver.wait()
        ver.close()
    else
        beep()
        message("No tienes pacientes por el momento")
    endif
else
    beep()
    message("")
endif
endif
endmethod

```

Objeto #Página2 #Cuadro34 #Cuadro7 #Botón8

Método pushButton

```

method pushButton(var eventInfo Event)
var
    vista table
    vent form
    cons query
endvar
cons=query

califica db|matricula|materia|grupo|mes1 |mes2 |mes3 |mes4 |ex final|final|
        |check |check |check|check|check|check|check|check |check|

cons_cal db|matricula|materia|grupo|mes1 |mes2 |mes3 |mes4 |ex
final|final|
        |insert|check |check |check|check|check|check|check |check|

endquery
if vista.attach("cons_cal db")then
    vista.empty()
    executeqbe(cons,"cons_cal db")
    if vent.open("cons_cal") then
        vent.wait()
        vent.close()
    endif
else
    beep()
    message("No se puede llegar al archivo")
endif
endmethod

```

Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) var vent form inter tcursor regreso string endvar if inter open("interno") then inter edit() inter.campo="Titulo" inter endedit() inter close() if vent open("sel_indi") then regreso=vent wait() vent close() if regreso<>"Cancelar" and regreso<>"" then inter open("interno") inter edit() inter cadena=regreso inter endedit() inter close() if vent open("hoja_rev") then vent wait() vent close() else beep() message("Problemas al abrir un archivo") endif endif else beep() message("Se cancela la operacion") endif else beep() message("No se puede abrir un archivo") endif endmethod </pre>
Objeto	#Página2 #Cuadro46 #Cuadro27 #Botón39

Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) var vent form inter tcursor regreso string endvar if inter open("Interno") then inter edit() inter.campo="autor" inter endedit() inter close() if vent open("revista") then regreso=vent wait() vent close() if regreso<>"Cancelar" and vent open("Indice") then regreso=vent wait() vent close() if regreso<>"Cancelar" and regreso<>"" then inter open("Interno") inter edit() inter cadena=regreso inter endedit() inter close() if vent open("hoja_rev") then vent wait() vent close() else beep() message("Problemas al abrir un archivo") endif endif else beep() message("Se cancela la operacion") endif else beep() message("No se puede abrir un archivo") endif endmethod </pre>
Objeto	#Página2 #Cuadro46 #Cuadro27 #Botón36
Método	pushButton

```
method pushButton(var eventInfo Event)
var
  vent form
  Inter tcursor
  regreso string
endvar
if Inter open("interno") then
  Inter edit()
  Inter campo="fecha"
  Inter endedit()
  Inter close()
  if vent open("sel_indi") then
    regreso=vent wait()
    vent close()
    if regreso<>"Cancelar" and vent open("indice") then
      regreso=vent wait()
      vent close()
      if regreso<>"Cancelar" and regreso<>" " then
        Inter open("interno")
        Inter edit()
        Inter cadena=regreso
        Inter endedit()
        Inter close()
        if vent open("hoja_rev") then
          vent wait()
          vent close()
        else
          beep()
          message("Problemas al abrir un archivo")
        endif
      endif
    else
      beep()
      message("Se cancela la operacion")
    endif
  else
    beep()
    message("No se puede abrir un archivo")
  endif
endif
endmethod
```

```

Objecto      #Página2

Método       open
Fuente       method open(var eventInfo Event)
              if not(eventInfo isprefilter()) then

                HIDESPEEDBAR()
                self edit()
              endif
            endmethod

Objecto      #Página2

Método       arrive
Fuente       method arrive(var eventInfo MoveEvent)
              var
                prin menu
              endvar
              IF NOT(EVENTINFO ISPREFILTER())THEN
                maximize()
                prin addtext("S&eguir")
                prin addtext("&Terminar")
                prin addtext("&Salir")
                prin show()
              ENDIF
            endmethod

Objecto      #Página2

Método       menuAction
Fuente       method menuAction(var eventInfo MenuEvent)
              var
                opc string
              endvar
              opc=eventInfo menuchoice()
              switch
                case opc="S&eguir".
                  bots pushbutton()
                case opc="&Terminar"
                  bott pushbutton()
                case opc="&Salir"
                  formreturn("Salir")
              endswitch
            endmethod

Objecto      #Página2 bott

Método       pushButton
Fuente       method pushButton(var eventInfo Event)
              formreturn("final")
            endmethod

Objecto      #Página2 bots

Método       pushButton
Fuente       method pushButton(var eventInfo Event)
              message("guardando los datos")
              beep()
              movetorecno(1)
              nc1=""
              nc2=""
              nc3=""
              hora_INI blank()
              hora_FIN blank()
              materia blank()
              semestre blank()
              doctor=""
            endmethod

Objecto      #Página2 CL2

Método       canArrive
Fuente       method canArrive(var eventInfo MoveEvent)
              if nc1="" then
                BEEP()
                eventInfo seterrorcode(cannotarrive)
              endif
            endmethod

Objecto      #Página2 CL2 S11

Método       canDepart
Fuente       method canDepart(var eventInfo MoveEvent)
              var
                tab cursor
                nom string
              endvar

```

```

if tab open("alumnos dbf") and tab locate("matricula",self value) then
message(tab nombre)
tab close()
else
beep()
message("No existe ese alumno registrado")
nom=""
nom view("Teclee el nombre del alumno")
if nom="" then
eventinfo seterrorcode(cannotdepart)
else
tab edit()
tab insertrecord()
tab matricula=self value
tab nombre=nom
tab semestre=semestre
tab postrecord()
tab endedit()
endif
tab close()
endif
endif
endmethod

```

Objecto #Página2 CL2 S11

Método changeValue

Fuente method changeValue(var eventinfo ValueEvent)

endmethod

Objecto #Página2 CL2 S12

Método canDepart

Fuente method canDepart(var eventinfo MoveEvent)

```

var
tab tcursor
nom string
endvar

if self value<>0 then
if tab open("alumnos dbf") and tab locate("matricula",self value) then
message(tab nombre)
tab close()
else
beep()
message("No existe ese alumno registrado")
nom=""
nom view("Teclee el nombre del alumno")
if nom="" then
eventinfo seterrorcode(cannotdepart)
else
tab edit()
tab insertrecord()
tab matricula=self value
tab nombre=nom
tab semestre=semestre
tab postrecord()
tab endedit()
endif
tab close()
endif
endif
endmethod

```

Objecto #Página2 CL2 S12

Método changeValue

Fuente method changeValue(var eventinfo ValueEvent)

endmethod

Objecto #Página2 CL2 S13

Método canDepart

Fuente method canDepart(var eventinfo MoveEvent)

```

var
tab tcursor
nom string
endvar

if self value<>0 then

```

```

tab close()
else
beep()
message("No existe ese alumno registrado")
nom=""
nom view("Teclee el nombre del alumno")
if nom="" then
eventInfo setErrorcode(cannotdepart)
else
tab edit()
tab insertrecord()
tab matricula=self value
tab nombre=nom
tab semestre=semestre
tab postrecord()
tab endedit()
endif
tab close()
endif
endif
endmethod

```

Objeto #Página2 CL2 S13

Método
Fuente
changeValue
method changeValue(var eventInfo ValueEvent)

```

endmethod
#Página2 CL2 S14

```

Objeto #Página2 CL2 S14

Método
Fuente
canDepart
method canDepart(var eventInfo MoveEvent)
var
tab tcursor
nom string
endvar

```

if self value<>0 then
if tab open("alumnos dbf") and tab locate("matricula",self value) then
message(tab nombre)
tab close()
else
beep()
message("No existe ese alumno registrado")
nom=""
nom view("Teclee el nombre del alumno")
if nom="" then
eventInfo setErrorcode(cannotdepart)
else
tab edit()
tab insertrecord()
tab matricula=self value
tab nombre=nom
tab semestre=semestre
tab postrecord()
tab endedit()
endif
tab close()
endif
endif
endmethod

```

Objeto #Página2 CL2 S14

Método
Fuente
changeValue
method changeValue(var eventInfo ValueEvent)

```

endmethod
#Página2 CL2 S15

```

Objeto #Página2 CL2 S15

Método
Fuente
canDepart
method canDepart(var eventInfo MoveEvent)
var
tab tcursor
nom string
endvar

```

if self value<>0 then
if tab open("alumnos dbf") and tab locate("matricula",self value) then

```

```

tab close()
else
  beep()
  message("No existe ese alumno registrado")
  nom=""
  nom view("Teclee el nombre del alumno")
  if nom="" then
    eventinfo seterrorcode(cannotdepart)
  else
    tab edit()
    tab insertrecord()
    tab matricula=self.value
    tab nombre=nom
    tab semestre=semestre
    tab postrecord()
    tab endedit()
  endif
  tab close()
endif
endif
endmethod

```

Objecto	#Página2 CL2 S15
Método	changeValue
Fuente	method changeValue(var eventinfo ValueEvent)
	endmethod
Objecto	#Página2 CL2 S16
Método	canDepart
Fuente	method canDepart(var eventinfo MoveEvent)
	var
	tab tcursor
	nom string
	endvar
	if self.value<>0 then
	if tab.open("alumnos db") and tab locate("matricula",self.value) then
	message(tab.nombre)
	tab close()
	else
	beep()
	message("No existe ese alumno registrado")
	nom=""
	nom view("Teclee el nombre del alumno")
	if nom="" then
	eventinfo seterrorcode(cannotdepart)
	else
	tab edit()
	tab insertrecord()
	tab matricula=self.value
	tab nombre=nom
	tab semestre=semestre
	tab postrecord()
	tab endedit()
	endif
	tab close()
	endif
	endif
	endmethod
Objecto	#Página2 CL2 S16
Método	changeValue
Fuente	method changeValue(var eventinfo ValueEvent)
	endmethod
Objecto	#Página2 CL2 S17
Método	canDepart
Fuente	method canDepart(var eventinfo MoveEvent)
	var
	tab tcursor
	nom string
	endvar
	if self.value<>0 then

```

tab close()
else
beep()
message("No existe ese alumno registrado")
nom=""
nom view("Teclee el nombre del alumno")
if nom="" then
eventInfo setErrorcode(cannotdepart)
else
tab edit()
tab insertrecord()
tab matricula=self value
tab nombre=nom
tab semestre=semestre
tab postrecord()
tab endedit()
endif
tab close()
endif
endif
endmethod

```

Objeto #Página2 CL2 S17

Método
Fuente *changeValue*
method *changeValue*(var eventInfo ValueEvent)

Objeto #Página2 CL2 S18

Método
Fuente *canDepart*
method *canDepart*(var eventInfo MoveEvent)
var
tab tcursor
nom string
endvar

if self value<>0 then
if tab open("alumnos db") and tab locate("matricula",self value) then
message(tab nombre)
tab close()
else
beep()
message("No existe ese alumno registrado")
nom=""
nom view("Teclee el nombre del alumno")
if nom="" then
eventInfo setErrorcode(cannotdepart)
else
tab edit()
tab insertrecord()
tab matricula=self value
tab nombre=nom
tab semestre=semestre
tab postrecord()
tab endedit()
endif
tab close()
endif
endif
endmethod

Objeto #Página2 CL2 S18

Método .
Fuente *changeValue*
method *changeValue*(var eventInfo ValueEvent)

Objeto #Página2 CL2 S19

Método
Fuente *canDepart*
method *canDepart*(var eventInfo MoveEvent)
var
tab tcursor
nom string
endvar

if self value<>0 then

```

tab close()
else
beep()
message("No existe ese alumno registrado")
nom=""
nom view("Teclee el nombre del alumno")
if nom="" then
eventInfo setErrorcode(cannotdepart)
else
tab edit()
tab insertrecord()
tab matricula=self value
tab nombre=nom
tab semestre=semestre
tab postrecord()
tab endedit()
endif
tab close()
endif
endif
endmethod

```

Objecto #Página2 CL2 S19

Método changeValue

Fuente method changeValue(var eventInfo ValueEvent)

endmethod

Objecto #Página2 CL2 S20

Método canDepart

Fuente method canDepart(var eventInfo MoveEvent)

```

var
tab cursor
nom string
endvar

(if self value<>0 then
if tab open("alumnos.db") and tab locate("matricula",self value) then
message(tab nombre)
tab close()
else
beep()
message("No existe ese alumno registrado")
nom=""
nom view("Teclee el nombre del alumno")
if nom="" then
eventInfo setErrorcode(cannotdepart)
else
tab edit()
tab insertrecord()
tab matricula=self value
tab nombre=nom
tab semestre=semestre
tab postrecord()
tab endedit()
endif
tab close()
endif
endif
endmethod

```

Objecto #Página2 CL2 S20

Método changeValue

Fuente method changeValue(var eventInfo ValueEvent)

endmethod

Objecto #Página2 CL3

Método canArrive

Fuente method canArrive(var eventInfo MoveEvent)

```

if nc(3)="" then
BEEP()
eventInfo setErrorcode(cannotarrive)
endif
endmethod

```

Método	canDepart
Fuente	<pre> method canDepart(var eventInfo MoveEvent) var tab tcursor nom string endvar if self.value <> 0 then if tab.open("alumnos dbf") and tab.locate("matricula",self.value) then message(tab.nombre) tab.close() else beep() message("No existe ese alumno registrado") nom="" nom.view("Teclee el nombre del alumno") if nom="" then eventInfo.seterrorCode(cannotdepart) else tab.edit() tab.insertrecord() tab.matricula=self.value tab.nombre=nom tab.semestre=semestre tab.postrecord() tab.endedit() endit tab.close() endif endif endif endmethod </pre>
Objeto	#Página2 CL3 S21

Método	changeValue
Fuente	<pre> method changeValue(var eventInfo ValueEvent) endmethod </pre>
Objeto	#Página2 CL3 S22

Método	canDepart
Fuente	<pre> method canDepart(var eventInfo MoveEvent) var tab tcursor nom string endvar if self.value <> 0 then if tab.open("alumnos dbf") and tab.locate("matricula",self.value) then message(tab.nombre) tab.close() else beep() message("No existe ese alumno registrado") nom="" nom.view("Teclee el nombre del alumno") if nom="" then eventInfo.seterrorCode(cannotdepart) else tab.edit() tab.insertrecord() tab.matricula=self.value tab.nombre=nom tab.semestre=semestre tab.postrecord() tab.endedit() endit tab.close() endif endif endif endmethod </pre>
Objeto	#Página2 CL3 S22

Método	changeValue
Fuente	<pre> method changeValue(var eventInfo ValueEvent) endmethod </pre>
Objeto	#Página2 CL3 S23

Método	canDepart
Fuente	<pre> method canDepart(var eventInfo MoveEvent) var tab tcursor nom string endvar </pre>

```

if tab open("alumnos db") and tab locate("matricula",self.value) then
message(tab nombre)
tab close()
else
beep()
message("No existe ese alumno registrado")
nom=""
nom view("Teclee el nombre del alumno")
if nom="" then
eventInfo setErrorcode(cannotdepart)
else
tab edit()
tab insertrecord()
tab matricula=self.value
tab nombre=nom
tab semestre=semestre
tab postrecord()
tab endedit()
endif
tab close()
endif
endif
endmethod

```

Objeto #Página2 CL3 S23

Método changeValue

Fuente method changeValue(var eventInfo ValueEvent)

endmethod

Objeto #Página2 CL3 S24

Método canDepart

Fuente method canDepart(var eventInfo MoveEvent)

```

var
tab tcursor
nom string
endvar

if self.value<>0 then
if tab open("alumnos db") and tab locate("matricula",self.value) then
message(tab nombre)
tab close()
else
beep()
message("No existe ese alumno registrado")
nom=""
nom view("Teclee el nombre del alumno")
if nom="" then
eventInfo setErrorcode(cannotdepart)
else
tab edit()
tab insertrecord()
tab matricula=self.value
tab nombre=nom
tab semestre=semestre
tab postrecord()
tab endedit()
endif
tab close()
endif
endif
endmethod

```

Objeto #Página2 CL3 S24

Método changeValue

Fuente method changeValue(var eventInfo ValueEvent)

endmethod

Objeto #Página2 CL3 S25

Método canDepart

Fuente method canDepart(var eventInfo MoveEvent)

```

var
tab tcursor
nom string
endvar

if self.value<>0 then
if tab open("alumnos db") and tab locate("matricula",self.value) then
message(tab nombre)

```

```

tab close()
else
  beep()
  message("No existe ese alumno registrado")
  nom=""
  nom view("Teclee el nombre del alumno")
  if nom="" then
    eventInfo setErrorcode(cannotdepart)
  else
    tab edit()
    tab insertrecord()
    tab matricula=self value
    tab nombre=nom
    tab semestre=semestre
    tab postrecord()
    tab endedit()
  endif
  tab close()
endif
endif
endmethod

```

Objeto #Página2 CL3 S25

Método changeValue

Fuente method changeValue(var eventInfo ValueEvent)

endmethod

Objeto #Página2 CL3 S26

Método canDepart

Fuente method canDepart(var eventInfo MoveEvent)

```

var
  tab tcursor
  nom string
endvar

if self value <> 0 then
  if tab open("alumnos.dbf") and tab locate("matricula",self value) then
    message(tab nombre)
    tab close()
  else
    beep()
    message("No existe ese alumno registrado")
    nom=""
    nom view("Teclee el nombre del alumno")
    if nom="" then
      eventInfo setErrorcode(cannotdepart)
    else
      tab edit()
      tab insertrecord()
      tab matricula=self value
      tab nombre=nom
      tab semestre=semestre
      tab postrecord()
      tab endedit()
    endif
    tab close()
  endif
endif
endmethod

```

Objeto #Página2 CL3 S26

Método changeValue

Fuente method changeValue(var eventInfo ValueEvent)

endmethod

Objeto #Página2 CL3 S27

Método canDepart

Fuente method canDepart(var eventInfo MoveEvent)

```

var
  tab tcursor
  nom string
endvar

if self value <> 0 then
  if tab open("alumnos.dbf") and tab locate("matricula",self value) then

```

```

tab close()
else
beep()
message("No existe ese alumno registrado")
nom=""
nom view("Teclee el nombre del alumno")
if nom="" then
eventinfo seterrorcode(cannotdepart)
else
tab edit()
tab insertrecord()
tab matricula=self value
tab nombre=nom
tab semestre=semestre
tab postrecord()
tab endedit()
endif
tab close()
endif
endif
endmethod

```

Objeto #Página2 CL3 S27

Método changeValue

Fuente method changeValue(var eventinfo ValueEvent)

endmethod

Objeto #Página2 CL3 S28

Método canDepart

Fuente method canDepart(var eventinfo MoveEvent)

```

var
tab tcursor
nom string
endvar

if self value<>0 then
if tab open("alumnos dbf") and tab locate("matricula",self value) then
message(tab nombre)
tab close()
else
beep()
message("No existe ese alumno registrado")
nom=""
nom view("Teclee el nombre del alumno")
if nom="" then
eventinfo seterrorcode(cannotdepart)
else
tab edit()
tab insertrecord()
tab matricula=self value
tab nombre=nom
tab semestre=semestre
tab postrecord()
tab endedit()
endif
tab close()
endif
endif
endmethod

```

Objeto #Página2 CL3 S28

Método changeValue

Fuente method changeValue(var eventinfo ValueEvent)

endmethod

Objeto #Página2 CL3 S29

Método canDepart

Fuente method canDepart(var eventinfo MoveEvent)

```

var
tab tcursor
nom string
endvar

if self value<>0 then
if tab open("alumnos dbf") and tab locate("matricula",self value) then

```

```

tab close()
else
beep()
message("No existe ese alumno registrado")
nom=""
nom view("Teclea el nombre del alumno")
if nom="" then
eventinfo seterrorcode(cannotdepart)
else
tab edit()
tab insertrecord()
tab matricula=self value
tab nombre=nom
tab semestre=semestre
tab postrecord()
tab endedit()
endif
tab close()
endif
endif
endmethod

```

Objeto #Página2 CL3 S29

Método
Fuente changeValue
method changeValue(var eventinfo ValueEvent)

```

endmethod
#Página2 CL3 S30

```

Método
Fuente canDepart
method canDepart(var eventinfo MoveEvent)
var
tab tcursor
nom string
endvar

```

if self.value<>0 then
if tab open("alumnos.db") and tab locate("matricula",self value) then
message(tab nombre)
tab close()
else
beep()
message("No existe ese alumno registrado")
nom=""
nom view("Teclea el nombre del alumno")
if nom="" then
eventinfo seterrorcode(cannotdepart)
else
tab edit()
tab insertrecord()
tab matricula=self value
tab nombre=nom
tab semestre=semestre
tab postrecord()
tab endedit()
endif
tab close()
endif
endif
endmethod

```

Objeto #Página2 CL3 S30

Método
Fuente changeValue
method changeValue(var eventinfo ValueEvent)

```

endmethod
#Página2 cl1

```

Método
Fuente canArrive
method canArrive(var eventinfo MoveEvent)
message("llegando")
if ncl1="" then
eventinfo seterrorcode(cannotarrive)
else
errorclear()

```

Objeto      #Página2 cl1 S10
Método .    canDepart

Fuente      method canDepart(var eventInfo MoveEvent)
var
  tab tcursor
  nom string
endvar

If self.value<>0 then
  if tab.open("alumnos.db") and tab.locate("matricula",self.value) then
    message(tab.nombre)
    tab.close()
  else
    beep()
    message("No existe ese alumno registrado")
    nom=""
    nom.view("Teclee el nombre del alumno")
    if nom="" then
      eventInfo.setErrorcode(cannotdepart)
    else
      tab.edit()
      tab.insertRecord()
      tab.matricula=self.value
      tab.nombre=nom
      tab.semestre=semestre
      tab.postRecord()
      tab.endEdit()
    endif
    tab.close()
  endif
endif
endmethod

```

Objeto #Página2 cl1 S10

```

Método      changeValue
Fuente      method changeValue(var eventInfo ValueEvent)

```

```

endmethod
Objeto      #Página2 cl1 S9

```

```

Método      canDepart
Fuente      method canDepart(var eventInfo MoveEvent)
var
  tab tcursor
  nom string
endvar

If self.value<>0 then
  if tab.open("alumnos.db") and tab.locate("matricula",self.value) then
    message(tab.nombre)
    tab.close()
  else
    beep()
    message("No existe ese alumno registrado")
    nom=""
    nom.view("Teclee el nombre del alumno")
    if nom="" then
      eventInfo.setErrorcode(cannotdepart)
    else
      tab.edit()
      tab.insertRecord()
      tab.matricula=self.value
      tab.nombre=nom
      tab.semestre=semestre
      tab.postRecord()
      tab.endEdit()
    endif
    tab.close()
  endif
endif
endmethod

```

Objeto #Página2 cl1 S9

```

Método      changeValue
Fuente      method changeValue(var eventInfo ValueEvent)

```

```

endmethod
Objeto      #Página2 cl1 S8

```

```

Método      canDepart
Fuente      method canDepart(var eventInfo MoveEvent)
var
  tab tcursor
  nom string

```

```

if self value<>0 then
if tab open("alumnos.db") and tab locate("matricula",self value) then
message(tab nombre)
tab close()
else
beep()
message("No existe ese alumno registrado")
nom=""
nom.view("Teclee el nombre del alumno")
if nom="" then
eventInfo seterrorcode(cannotdepart)
else
tab edit()
tab insertrecord()
tab matricula=self value
tab nombre=nom
tab semestre=semestre
tab postrecord()
tab endedit()
endif
tab close()
endif
endif
endmethod

```

Objeto. #Página2.c11 S6

Método
Fuente changeValue
method changeValue(var eventInfo ValueEvent)

```

endmethod
#Página2 c11 S7

```

Objeto

Método
Fuente canDepart
method canDepart(var eventInfo MoveEvent)
var
tab tcursor
nom string
endvar

```

if self value<>0 then
if tab open("alumnos.db") and tab locate("matricula",self value) then
message(tab nombre)
tab close()
else
beep()
message("No existe ese alumno registrado")
nom=""
nom.view("Teclee el nombre del alumno")
if nom="" then
eventInfo seterrorcode(cannotdepart)
else
tab edit()
tab insertrecord()
tab matricula=self value
tab.nombre=nom
tab semestre=semestre
tab postrecord()
tab endedit()
endif
tab close()
endif
endif
endmethod
endmethod

```

Objeto #Página2 c11.S7

Método
Fuente changeValue
method changeValue(var eventInfo ValueEvent)

```

endmethod
#Página2 c11 S6

```

Objeto

Método
Fuente canDepart
method canDepart(var eventInfo MoveEvent)
var
tab tcursor
nom string
endvar

```

if self value<>0 then
if tab open("alumnos.db") and tab locate("matricula",self value) then

```

```

tab close()
else
beep()
message("No existe ese alumno registrado")
nom=""
nom view("Taclea el nombre del alumno")
if nom="" then
eventInfo setErrorcode(cannotdepart)
else
tab edit()
tab insertrecord()
tab matricula=self.value
tab nombre=nom
tab semestre=semestre
tab postrecord()
tab endedit()
endif
tab close()
endif
endif
endmethod

```

Objecto #Página2 c11 S6

Método changeValue

Fuente method changeValue(var eventInfo ValueEvent)

endmethod

Objecto #Página2 c11 S6

Método canDepart

Fuente method canDepart(var eventInfo MoveEvent)

```

var
tab tcursor
nom string
endvar

if self.value<>0 then
if tab open("alumnos db") and tab locate("matricula",self.value) then
message(tab nombre)
tab close()
else
beep()
message("No existe ese alumno registrado")
nom=""
nom view("Taclea el nombre del alumno")
if nom="" then
eventInfo setErrorcode(cannotdepart)
else
tab edit()
tab insertrecord()
tab matricula=self.value
tab nombre=nom
tab semestre=semestre
tab postrecord()
tab endedit()
endif
tab close()
endif
endif
endmethod

```

Objecto #Página2 c11 S6

Método changeValue

Fuente method changeValue(var eventInfo ValueEvent)

endmethod

Objecto #Página2 c11 S4

Método canDepart

Fuente method canDepart(var eventInfo MoveEvent)

```

var
tab tcursor
nom string
endvar

if self.value<>0 then

```

```

tab close()
else
beep()
message("No existe ese alumno registrado")
nom=""
nom view("Teclea el nombre del alumno")
if nom="" then
eventInfo setErrorcode(cannotdepart)
else
tab edit()
tab insertrecord()
tab matricula=self value
tab nombre=nom
tab semestre=semestre
tab postrecord()
tab endedit()
endif
tab close()
endif
endif
endmethod

```

Objeto. #Página2 cl1 S4

Método . changeValue

Fuente method changeValue(var eventInfo ValueEvent)

endmethod

Objeto #Página2 cl1 S3

Método canDepart

Fuente method canDepart(var eventInfo MoveEvent)

```

var
tab tcursor
nom string
endvar

if self value<>0 then
if tab open("alumnos dbf") and tab locate("matricula",self value) then
message(tab nombre)
tab close()
else
beep()
message("No existe ese alumno registrado")
nom=""
nom view("Teclea el nombre del alumno")
if nom="" then
eventInfo setErrorcode(cannotdepart)
else
tab edit()
tab insertrecord()
tab matricula=self value
tab nombre=nom
tab semestre=semestre
tab postrecord()
tab endedit()
endif
tab close()
endif
endif
endmethod

```

Objeto #Página2 cl1 S3

Método changeValue

Fuente method changeValue(var eventInfo ValueEvent)

endmethod

Objeto #Página2 cl1 S2

Método canDepart

Fuente method canDepart(var eventInfo MoveEvent)

```

var
tab tcursor
nom string
endvar

```

```

message(tab nombre)
tab close()
else
beep()
message("No existe ese alumno registrado")
nom=""
nom view("Teclee el nombre del alumno")
if nom="" then
eventInfo setErrorcode(cannotdepart)
else
tab edit()
tab insertrecord()
tab matricula=self value
tab nombre=nom
tab semestre=semestre
tab postrecord()
tab endedit()
endif
tab close()
endif
endif
endmethod

```

Objecto #Página2 cl1 S2

Método changeValue

Fuente method changeValue(var eventInfo ValueEvent)

endmethod

Objecto #Página2 cl1 S1

Método canDepart

Fuente method canDepart(var eventInfo MoveEvent)

```

var
tab tcursor
nom string
endvar

if self value<>0 then
if tab open("alumnos dbf") and tab locate("matricula",self value) then
message(tab nombre)
tab close()
else
beep()
message("No existe ese alumno registrado")
nom=""
nom view("Teclee el nombre del alumno")
if nom="" then
eventInfo setErrorcode(cannotdepart)
else
tab edit()
tab insertrecord()
tab matricula=self value
tab nombre=nom
tab semestre=semestre
tab postrecord()
tab endedit()
endif
tab close()
endif
endif
endmethod

```

Objecto #Página2 cl1 S1

Método changeValue

Fuente method changeValue(var eventInfo ValueEvent)

endmethod

Objecto #Página2 #Cuadro49 MATERIA

Método canDepart

Fuente method canDepart(var eventInfo MoveEvent)

```

var
mats,clin tcursor
mate,OP string
ff logical
fn let number

```

```

if mats open("materias db")then
if not (mats locate("materia",self value,"semestre",semestre value)) then
eventInfo setErrorCode(cannotdepart)
beep()
message("No existe esa materia o no en ese semestre")
else
if mats clinicas search("A")>0 then
if clin open("hora1 DB")then
if clin locate("hora",hora_ini)then
mate=""
clin fieldValue(doword(fecha value)+1,mate)
if mate="" then
beep()
message("No hay clase a esa hora")
else
if materia search(mate substr(1,4))>0 then
fin=hora_fin-1
nci1="A"
doctor=mats maestro
while clin locatenext(doword(fecha value)+1,mate)
fin=clin hora
endwhile
hora_fin=fin+1
else
beep()
message("No hay esa clase ese día a esa hora")
endif
endif
endif
else
op=msgquestion("Cuidado","No se puede checar en el horario, ¿Esta
seguro que las horas de inicio y final están bien?")
if op="Yes"then nci1="A" endif
endif
endif
if mats clinicas search("B")>0 then
if clin open("hora2 DB")then
if clin locate("hora",hora_ini)then
mate=""
clin fieldValue(doword(fecha value)+1,mate)
if mate="" then
beep()
message("No hay clase a esa hora")
else
if materia search(mate substr(1,4))>0 then
fin=hora_fin-1
nci2="B"
doctor=mats maestro
while clin locatenext(doword(fecha value)+1,mate)
fin=clin hora
endwhile
hora_fin=fin+1
else
beep()
message("No hay esa clase ese día a esa hora")
endif
endif
endif
else
op=msgquestion("Cuidado","No se puede checar en el horario, ¿Esta
seguro que las horas de inicio y final están bien?")
if op="Yes"then nci2="B" endif
endif
endif
if mats clinicas search("C")>0 then
if clin open("hora3 DB")then
if clin locate("hora",hora_ini)then
mate=""
clin fieldValue(doword(fecha value)+1,mate)
if mate="" then
beep()
message("No hay clase a esa hora")
else
if materia search(mate substr(1,4))>0 then
fin=hora_fin-1
nci3="C"
doctor=mats maestro
while clin locatenext(doword(fecha value)+1,mate)
fin=clin hora
endwhile
hora_fin=fin+1
else
beep()
message("No hay esa clase ese día a esa hora")
endif
endif
endif
endif
endif

```

```
seguro que las horas de inicio y final están bien?")
  if op=="Yes" then nc13="C" endif
endif
endif
endif
mate close()
else
  message("No se puede abrir el archivo de las materias")
endif
endif
endmethod
```

```

Objecto      #Página2 #Cuadro49 DOCTOR

Método      canDepart
Fuente      method canDepart(var eventInfo MoveEvent)

endmethod

Objecto      #Página2 #Cuadro49 DOCTOR

Método      depart
Fuente      method depart(var eventInfo MoveEvent)
var
ff date
hh,hhh number
mat string
sem smallint
doc string
endvar
if
not(localenext("fecha",fecha,"hora_ini",hora_fin,"materia",materia,"semestr
e",semestre)) then
ff=fecha value
hh=hora_ini value
hhh=hora_fin value
mat=materia value
sem=semestre value
doc=doctor
message("Nuevo registro")
action(databeginedit)
action(datainsertrecord)
fecha=ff
hora_ini=hh
hora_fin=hhh
materia=mat
semestre=sem
doctor=doc
else
beep()
message("Escribiendo en un registro existente")
endif
endmethod

Objecto      #Página2 #Cuadro49 DOCTOR

Método      changeValue
Fuente      method changeValue(var eventInfo ValueEvent)

endmethod

Objecto      #Página2 #Cuadro49 FECHA

Método      open
Fuente      method open(var eventInfo Event)
self value=date()+1
endmethod

```

```

Objeto      F_caja

Método      open
Fuente      method open(var eventInfo Event)
            .. Maximizar la ventana al entrar a la ficha (al activarla)
            .. Se verifica que no ocurra el evento "prefilter", ya que esto
            .. significaría heredar este metodo a todos los objetos
            if not(eventInfo isprefilter()) then
                hidespeedbar()
                maximize()
            endif
            endmethod

Objeto      F_caja

Método      arrive
Fuente      method arrive(var eventInfo MoveEvent)
            var
                menucaja menu      ,menu principal

                menupaciente,menucobros,menuprof      popupmenu
            endvar
            maximize()
            .. Crear y Mostrar los menus
            menupaciente addtext("&Reimprimir Hist clinica")
            menupaciente addtext("&Imprimir &Hoja de control")
            menupaciente addtext("&Imprimir &Lista de pacientes")

            menucobros addtext("&Cobro normal")
            menucobros addtext("&Corte de caja")
            menucobros addtext("&Re&imprimir recibo")
            menucobros addtext("&Reporte dia")

            menuprof addtext("&Apartado de unidades")
            menuprof addtext("&Listas")

            menucaja addpopup("&Pacientes",menupaciente)
            menucaja addpopup("&Cobros",menucobros)
            menucaja addpopup("&Apoyo a profesores",menuprof)
            menucaja addtext("&Salir")
            menucaja show()
            endmethod

Objeto      F_caja

Método      menuAction
Fuente      .. Gestión del menu
            method menuAction(var eventInfo MenuEvent)
            var
                tab_pac,t11,tab_dat tcursor
                opc,opt1 string
                ficha_datos,vent form
                t1 table
                rep report
                num,num1 number
                fech date
            endvar

            opc=eventInfo menuchoice()
            switch
            case opc="&Reimprimir Hist clinica"
                num=0
                num view("&Teclee el número de expediente")
                .. Una vez obtenido el numero se escribe en la tabla
                .. "Interno" para que lo tome de ahí la ficha de impresion
                if num>0 AND t11 open("Interno") then
                    t11 skip(1)
                    t11 edit()
                    t11 numero=num
                    t11 endedit()
                    t11 postrecord()
                    t11 close()
                    if ficha_datos open("Iden2") then
                        ficha_datos wait()
                        ficha_datos close()
                    endif
                    t11 open("Interno")
                    t11 skip(1)
                    t11 edit()
                    t11 numero=0
                    t11 postrecord()
                    t11 close()
                endif
            case opc="&Imprimir &Hoja de control"
                num=0
                num view("&Teclee el número de expediente")
            endswitch

```

```

    .. "Interno" para que lo tome de ahi la ficha de impresion
    if num>0 then
        t1 t open("Interno")
        t1 t edit()
        t1 t numero=num
        t1 t close()
        if ficha_datos open("control") then
            ficha_datos wait()
            ficha_datos close()
        else
            message("Problemas")
        endif
        t1 t open("Interno")

        t1 t edit()
        t1 t numero=0
        t1 t close()
    endif
case opc="Imprimir &Lista de pacientes"
    .. Imprimir el reporte de pacientes
    if rep print("pacient") then
        else
            message("No se pudo imprimir")
        endif
case opc="&Cobro normal"
    .. Activar la ficha de cobros
    if ficha_datos open("dat_pac") then
        opc1=ficha_datos wait()
        ficha_datos close()
    else
        message("Problemas")
    endif
case opc="C&orte de caja"
    .. Activar la ficha de corte
    if ficha_datos open("corte") then
        opc1=ficha_datos wait()
        ficha_datos close()
    else
        message("Problemas")
    endif
case opc="Re&imprimir recibo"
    num=0
    num view("Teclee el numero del recibo?")
    .. Una vez obtenido el numero se escribe en la tabla
    .. "Interno" para que lo tome de ahi la ficha de impresion
    if num<>0 then
        if tab_dat open("Interno") then
            tab_dat skip(1)
            tab_dat edit()
            tab_dat numero=num
            tab_dat postrecord()
            tab_dat close()
            if vent open("Impr_rec") then
                vent wait()
                vent close()
            endif
            tab_dat open("Interno")
            tab_dat skip(1)
            tab_dat edit()
            tab_dat numero=0
            tab_dat postrecord()
            tab_dat close()
        endif
    endif
case opc="&Reporte dia"
    .. Imprimir el reporte del dia
    if rep print("reporte") then
        else
            message("No se pudo imprimir")
        endif
case opc="&Efectivo"
    .. Activar la ficha de efectivo
    if vent open("efectivo") then
        vent wait()
        vent close()
    else
        beep()
        message("Problemas al abrir un archivo")
    endif
case opc="&Horarios"
    .. Activar la ficha de horario
    if vent open("horario") then
        vent wait()
        vent close()
    else
        beep()
        message("Problemas al abrir un archivo")
    endif
case opc="&Apartado de unidades"

```

```

vent wait()
vent close()
else
beep()
message(">Problemas con el archivo")
endif
case opc="&Listas"
setmouseshape(mousewait)
,, Abrir las tablas adecuadas
if t1 open("sillones.dbf") and tab_dat open("clinicas.dbf") and
   tab_pac open("alumnos.dbf") then
   fech=date()
   num=8
   ,, Obtener fecha y hora deseadas
   fech view("Que fecha desea imprimir")
   num view("Que hora desea imprimir")
   setmouseshape(mousewait)
   t1 empty()
   t1 compact()
   t1 edit()
   tab_dat.home()
   ,, Para cada sillón del registro de "apartado" crear un registro
   ,, en la tabla de "sillones", la cual es la base del informe
   while tab_dat locatenext("fecha",fech,"hora_ini",num)
   for num from 8 to 37
   num1=0
   tab_dat fieldvalue(num,num1)
   if num1 <> 0 then
   t1 insertrecord()
   beep()
   message(num1)
   t1 fecha=tab_dat "fecha"
   t1 hora=tab_dat "hora_ini"
   t1 hora_fin=tab_dat "hora_fin"
   t1 materia=tab_dat "materia"
   t1 doctor=tab_dat "doctor"
   t1 semestre=tab_dat "semestre"
   t1 matricula=num1
   num=num-7
   t1 sillón=num mod(10)
   num=num+7
   if t1 sillón<=0 then
   t1 sillón=10
   endif
   ,, Agregar a cada registro el nombre del alumno
   if tab_pac locata("matricula",num1) then
   t1 nombre=tab_pac nombre
   endif
   if (num-7)<11 then
   t1 clinica="A"
   else
   if (num-7)>20 then
   t1 clinica="C"
   else
   t1 clinica="B"
   endif
   endif
   t1 postrecord()
   else
   beep()
   message("No se puede determinar un alumno")
   endif
   endfor
   tab_dat.edit()
   ,, Una vez creados los registros de la tabla "sillones" se
   ,, elimina el registro de la tabla "apartado"
   if tab_dat delete record() then
   tab_dat compact()
   message("Borrado de registro")
   endif
   endwhile
   t1 .endedit()
   t1 .close()
   tab_dat.endedit()
   tab_dat.close()
   tab_pac.close()
   ,, Una vez lista la tabla "sillones" es posible imprimir la lista
   opc=msgquestion("¿desea imprimir")
   if opc="Yes" then
   if rep.print("listas") then
   else
   message("No se pudo imprimir")
   endif
   endif
   setmouseshape(mousearrow)
   ,, Si se escoge salir, cerrar la ficha
   case opc="&Salir"
   form.close()

```

Objeto	#Página2
Método	open
Fuente	<pre> method open(var eventInfo Event) var prin menu endvar maximize() hidespeedbar() prin addtext("&Salir") prin show() self edit() endmethod </pre>
Objeto	#Página2
Método	menuAction
Fuente	<pre> method menuAction(var eventInfo MenuEvent) var opc string endvar opc=eventInfo menuchoice() if opc="&Salir" then formreturn("Salir") endif endmethod </pre>
Objeto	#Página2 #Botón43
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) var lva FileBrowserInfo , variable registro FileBrowserInfo ArchivoElegido String foto graphic endvar lva Alias = "TRABAJO" , busca el directorio de trabajo actual lva selectedtype=fggraphic if fileBrowser(ArchivoElegido, lva) then if foto readfromfile(archivoelegido) then imagen=foto endif else message("Se eligió cancelar") endif endmethod </pre>
Objeto	#Página2 #Botón27
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) var tit string endvar num="" num view("Número del artículo") if locate("Titulo",tit) then edit() else message("No se encuentra el artículo") endif endmethod </pre>
Objeto	#Página2 Salir
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) self endedit() formreturn("Salir") endmethod </pre>
Objeto	#Página2 #Botón26
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) var num number endvar num=0 num view("Número del artículo") if locate("Numero",num) then edit() else message("No se encuentra el artículo") endif endmethod </pre>

Método	pushButton
Fuente	method pushButton(var eventInfo Event) skip(-1) endmethod

Objecto	#Página2 #Botón23
---------	-------------------

Método	pushButton
Fuente	method pushButton(var eventInfo Event) skip(1) endmethod

```

Objeto      F_Captura

Método      Type
Fuente      „Registro básico de captura
Type
    reg_pac=record
    clave          number
    nombre         string
    fecha          date
    Tratamiento string
    Paso           string
endrecord
endType

Objeto      F_Captura

Método      Var
Fuente      „Declaracion del registro básico de captura como variable
Var
    datos reg_pac
endVar

Objeto      F_Captura

Método      open
Fuente      method open(var eventInfo Event)
    „Al abrir la ficha maximizar la ventana
    „se verifica que no ocurra el evento prefilter porque
    „ esto significaría heredar el metodo a todos los objetos
    if not(eventInfo Isprefilter()) then
        hidespeedbar()
        maximize()
    endIf
endmethod

Objeto      F_Captura

Método      arrive
Fuente      „Al entrar en la ficha mostrar el menú adecuado
method arrive(var eventInfo MoveEvent)
    var
        prin menu          „menu principal
        dat,mant,pac,hor popupmenu
    endvar
    maximize()

    hor addtext("&Clinica &A")
    hor addtext("&Clinica &B")
    hor addtext("&Clinica &C")

    mant addtext("&Materias")
    mant addtext("&Tratamientos")
    mant addpopup("&Horarios",hor)

    pac addtext("&General")
    pac addtext("&Cirugia")
    pac addtext("&Parodontia")
    pac addtext("&Pr&ostodoncia")
    pac addtext("&Exodoncia")
    pac addtext("&Placa y sarro")
    pac addtext("&Periodontal")

    dat addpopup("&Pacientes",pac)
    dat addtext("&Alumnos")
    „dat addtext("&Calificaciones")
    „dat addtext("&Catedráticos")

    prin addpopup("&Datos personales",dat)
    prin addpopup("&Ajustes del sistema",mant)
    prin addtext("&Inventario")
    „prin addtext("&Revista")
    prin addtext("&Salir")
    prin show()
endmethod

Objeto      F_Captura

Método      menuAction
Fuente      method menuAction(var eventInfo MenuEvent)
    var
        vent form
        tab tableview
        opc string
    endvar
    opc=eventInfo menuchoice()
    switch
        „Registrar los datos de los pacientes

```

```

if vent open("reg_pac") then
    vent wait()
    vent close()
else
    beep()
    message("Problemas al abrir un archivo")
endif
„Registrar los datos de los alumnos
case opc="&Alumnos"
if vent open("datos") then
    vent wait()
    vent close()
else
    beep()
    message("Problemas al abrir un archivo")
endif
„Registrar o modificar datos de los catastrales
case opc="&Catedráticos"
if vent open("nuev_cat") then
    vent wait()
    vent close()
else
    beep()
    message("Problemas al abrir un archivo")
endif
„Modificar las materias
case opc="&Materias"
if vent open("reg_mate") then
    vent wait()
    vent close()
else
    beep()
    message("Problemas al abrir un archivo")
endif
„Las siguientes tres opciones modifican los horarios
case opc="Clinica &A"
if vent open("ho_cli_a") then
    vent wait()
    vent close()
else
    beep()
    message("Problemas al abrir un archivo")
endif
case opc="Clinica &B"
if vent open("ho_cli_b") then
    vent wait()
    vent close()
else
    beep()
    message("Problemas al abrir un archivo")
endif
case opc="Clinica &C"
if vent open("ho_cli_c") then
    vent wait()
    vent close()
else
    beep()
    message("Problemas al abrir un archivo")
endif
„Actualizar datos de los tratamientos
case opc="&Tratamientos"
if tab open("trats") then
    tab wait()
    tab close()
else
    beep()
    message("Problemas al abrir un archivo")
endif
„Capturar el inventario inicial
case opc="&Inventario"
if vent open("compras") then
    vent wait()
    vent close()
else
    beep()
    message("Problemas al abrir un archivo")
endif
„Cerrar la ficha
case opc="&Salir" formreturn("Salir")
endswitch
endmethod

```

Objecto	F_Captura #Text06	Página: 34
Método	mouseEnter	
Fuente	method mouseEnter(var eventInfo MouseEvent) endmethod	

Objeto	F_Compras
Método	Var
Fuente	<pre> „Variable global para determinar si es un .. producto nuevo o se está editando uno anterior Var nuevo logical endVar </pre>
Objeto	F_Compras
Método	open
Fuente	<pre> method open(var eventInfo Event) Var prin menu endvar „Maximizar y mostrar el menú adecuado maximize() hidespeedbar() prin addtext("&Salir") prin show() edit() nuevo=false endmethod </pre>
Objeto	F_Compras
Método	close
Fuente	<pre> „Finalizar la edición de tabla para evitar pérdida de información method close(var eventInfo Event) endedit() endmethod </pre>
Objeto	F_Compras
Método	menuAction
Fuente	<pre> „Gestión del menú method menuAction(var eventInfo MenuEvent) var opc string endvar opc=eventInfo menuchoice() if opc="&Salir" then formreturn("&Salir") endif endmethod </pre>
Objeto	F_Compras B_Salir
Método	pushButton
Fuente	<pre> „Cerrar la ficha cuando se oprima esa tecla method pushButton(var eventInfo Event) formreturn("&Salir") endmethod </pre>
Objeto	F_Compras B_Nuevo
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) „Moverse al último registro y „preparar la variable "nuevo" para poder insertar skip(nrecords()) edit() nuevo=true endmethod </pre>
Objeto	F_Compras B_Buscar
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) var matric number endvar matric=0 „Obtener la clave a buscar </pre>

```

edit()
nuevo=false
else
beep()
message("Esa clave no existe")
endif
endmethod

```

Objecto	F_Compras B_Archivar
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) ,,Moverse hacia adelante un registro ,, al hacerlo se grabarán los datos editados skip(1) endmethod </pre>
Objecto	F_Compras #Botón16
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) existencia=existencia-1 endmethod </pre>
Objecto	F_Compras #Botón17
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) existencia=existencia+1 endmethod </pre>
Objecto	F_Compras Precio_promedio
Método	depart
Fuente	<pre> method depart(var eventInfo MoveEvent) endmethod </pre>

Objeto	F_ConteoCasos
Método	arrive
Fuente	<pre> method arrive(var eventInfo MoveEvent) var prin menu endvar „Al entrar en la ficha, maximizar y mostrar el menú if not(eventInfo isprefilter())then maximize() hideshow() prin addtext("&Imprimir") prin addtext("&Salir") prin show() endif endmethod </pre>
Objeto	F_ConteoCasos
Método	menuAction
Fuente	<pre> „Gestión del menú method menuAction(var eventInfo MenuEvent) var opc string endvar opc=eventInfo menuchoice() switch case opc="&Imprimir" self action(dataprint) case opc="&Salir" formreturn("Salir") endswitch endmethod </pre>

Objeto	F_ImprHojaCtrl
Método	open
Fuente	<pre> method open(var eventInfo Event) var tab_int tcursor endvar „Al abrir la ficha (siempre desde otra) se busca el registro „ que se debe imprimir, el cual se ha puesto previamente „ en la tabla "Interno" if not(eventInfo Isprefilter())then if tab_int open("Interno") then skip(1) if tab_int numero<>0 then if self.setfilter(tab_int numero,tab_int numero) then else message("No se puede determinar el paciente") endif endif endif endif endmethod </pre>
Objeto	F_ImprHojaCtrl EXPEDIENTE1

Método	open
Fuente	<pre> method open(var eventInfo Event) var tab_dat tcursor endvar if expediente Isblank() then if tab_dat open("Ident") then self.value=tab_dat nombre tab_dat close() endif endif endmethod </pre>

Objeto	#Página2 #Botón6
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) var tab,t1 tcursor valor string num_ant number endvar num_ant=0 if tab open("rec_hist.dbf") then tab home() tab edit() num_ant=tab numero skip(1) while not(tab eof()) reg=tab recno() if tab numero=num_ant then tab deleteRecord() else num_ant=tab numero tab skip(1) endif endwhile tab compact() tab close() msgInfo("FIN","Proceso terminado") else message("No se puede abrir el archivo") endif endmethod </pre>
Objeto	#Página2 #Botón3
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) var tab,t1 tcursor valor string num_ant number endvar num_ant=0 if tab open("rec_hist.dbf") and t1 open("Recibos.dbf")then tab home() tab edit() t1 edit() while not(tab eof()) reg=tab recno() if TAB fecha=date() then t1 insertRecord() t1 numero=tab numero t1 fecha=tab fecha t1 num_exp=tab num_exp t1 nom_pac=tab nom_pac t1 concepto=tab concepto t1 precio=tab precio t1 importe=tab importe t1 mat_op=tab mat_op t1 mes=tab mes t1 postRecord() tab deleteRecord() else tab skip(1) endif endwhile tab compact() tab close() t1.close() msgInfo("FIN","Proceso terminado") else message("No se puede abrir el archivo") endif endmethod </pre>

```

Objecto      F_Corte

Método      open
Fuente      method open(var eventInfo Event)
            var
            prin menu
            endvar
            maximiza()
            hideshow()
            prin addtext("&Salir")
            prin.show()
            endmethod

Objecto      F_Corte

Método      menuAction
Fuente      method menuAction(var eventInfo MenuEvent)
            var
            opc string
            endvar
            opc=eventInfo menuchoice()
            if opc="&Salir" then
                formreturn("&Salir")
            endif
            endmethod

Objecto      F_Corte B_Salir

Método      pushButton
Fuente      method pushButton(var eventInfo Event)
            formreturn("&Salir")
            endmethod

Objecto      F_Corte B_SalirHistoricos

Método      pushButton
Fuente      method pushButton(var eventInfo Event)
            var
            t1 cursor
            t1 tcursor
            opc string
            t2 table
            endvar
            „Notificar que se cerrarán los archivos del día
            opc=mesyescancel("Atención","Debe hacer el reporte diario, y cualquier
            otro reporte o movimiento antes de ejecutar esta acción, porque los
            datos se borrarán; DESEA CONTINUAR?")
            if opc="Yes" then
                setmouseshape(mousewall)
                „Abrir las bases adecuadas
                if t1 open("recibos dbf") THEN
                    if t1 open("rec_hist dbf") then
                        t1 edit()
                        „Copiar cada registro de una tabla a otra
                        while not(t eof())
                            t1 end()
                            reg=t recno()
                            t1 insertafterrecord()
                            t1 numero=t numero
                            t1 num_exp=t num_exp
                            t1 nom_pac=t nom_pac
                            t1 fecha=t fecha
                            t1 concepto=t concepto
                            t1 cantidad=t cantidad
                            t1 precio=t precio
                            t1 importe=t importe
                            t1 mat_op=t mat_op
                            t1 mes=month(t fecha)
                            t skip(1)
                            t1 postrecord()
                        endwhile
                        t1 endedit()
                        t1 close()
                        t empty()
                        t compact()
                        if t2 attach("rec_hist dbf") then
                            sort t2
                            on "fecha","numero"
                            endsort
                        endif
                        formreturn("&Cerrar")
                    else
                        beep()
                        message("No se puede guardar la informcion")
                    endif
                endif
            endif

```

```

beep()
message("No se pueda abrir el archivo")
endif
else
message("Cierre de históricos cancelado")
endif
setmouseshape(mousearrow)
endmethod

```

Página: 41

Objeto	F_Corte B_Imprimir
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) ..Enviar los datos a la impresora action(dataprint) endmethod </pre>

Objecto	F_menu_prin
Método	open
Fuente	<pre> method open(var eventInfo Event) „ Maximizar la ventana al entrar a la ficha (al activarla) „ Se verifica que no ocurra el evento "prefilter", ya que esto „ significaría heredar este metodo a todos los objetos if eventInfo Isprefilter() then else maximize() hidespeedbar() beep() endif endmethod </pre>
Objecto	F_menu_prin
Método	arrive
Fuente	<pre> method arrive(var eventInfo MoveEvent) var prin menu endvar „Mostrar la opcion salir en el menu cada vez que se llegue a la ficha if not(eventInfo Isprefilter())then prin addtext("&Salir") PRIN SHOW() maximize() hidespeedbar() endif hidespeedbar() endmethod </pre>
Objecto	F_menu_prin
Método	setFocus
Fuente	<pre> method setFocus(var eventInfo Event) „maximizar la ventana cuando se pulse un boton del raton maximize() hidespeedbar() endmethod </pre>
Objecto	F_menu_prin
Método	menuAction
Fuente	<pre> method menuAction(var eventInfo MenuEvent) var opc string endvar „ Gestion del menu opc=eventInfo menuchoice() „al escoger salir, se cierra paradox if opc="&Salir" then exit() endif endmethod </pre>
Objecto	F_menu_prin B_consulta
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) var vent form endvar „Activar la ficha de alumnos if vent open("alumnos") then vent wait() vent close() else message("No se puede activar esa opcion") endif endmethod </pre>
Objecto	F_menu_prin B_salir
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) „ Salir de paradox exit() endmethod </pre>
Objecto	F_menu_prin B_ajustes
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) var vent form endvar </pre>

```

if vent.open("ajustes") then
    vent.wait()
    vent.close()
else
    message("No se puede activar esa opcion")
endif
endmethod

```

Objeto F_menu_prin B_captura

Método ,
Fuente

```

pushButton
method pushButton(var eventInfo Event)
var
    vent form
endvar
if vent.open("captura") then
    vent.wait()
    vent.close()
else
    message("No se puede activar esa opcion")
endif
endmethod

```

Objeto F_menu_prin B_director

Método
Fuente

```

pushButton
method pushButton(var eventInfo Event)
var
    vent form
endvar
,, Activar la ficha de director
if vent.open("director") then
    vent.wait()
    vent.close()
else
    message("No se puede activar esa opcion")
endif
endmethod

```

Objeto F_menu_prin B_caja

Método
Fuente

```

arrive
method arrive(var eventInfo MoveEvent)
,, Activar maximizar para la ficha en curso
maximize()
hidespeedbar()
endmethod

```

Objeto F_menu_prin B_caja

Método
Fuente

```

pushButton
method pushButton(var eventInfo Event)
var
    vent form
endvar
,, Activar la ficha cajas
if vent.open("caja") then
    vent.wait()
    vent.close()
else
    message("No se puede activar esa opcion")
endif
endmethod

```

```

Objeto      F_cobro

Método      Var
Fuente      .. Las variables declaradas aqui seran utiles en todos los
            .. metodos de la ficha, incluyendo los de los objetos
            Var
            guardado logical
            endVar

Objeto      F_cobro

Método      open
Fuente      method open(var eventInfo Event)
            var
            prin menu
            tab_pac tcursor
            endvar
            .. Maximizar la ventana al entrar a la ficha (al activarla)
            .. Se verifica que no ocurra el evento "prefilter", ya que esto
            .. significaria heredar este metodo a todos los objetos
            if not(eventInfo isprefilter())then
            maximize()
            prin addtext("&Salir")
            prin show()
            ..Abrir la tabla de recibos y tomar el numero del ultimo registro
            if tab_pac open("Recibos.dbf") then
            if tab_pac nrecords()>0 then
            num_rec=tab_pac cmax("numero")+1
            tab_pac close()
            else
            tab_pac close()
            if tab_pac open("rec_hist.DBF")then
            num_rec=tab_pac cmax("numero")+1
            tab_pac close()
            endif
            endif
            else
            beep()
            message("No se puede abrir el archivo de recibos")
            endif
            cantidad=1
            fecha=date()
            endif
            endmethod

Objeto      F_cobro

Método      menuAction
Fuente      .. Gestion del menu
            method menuAction(var eventInfo MenuEvent)
            var
            opc string
            endvar
            .. Recibir el mensaje de opcion del menu
            opc=eventInfo menuchoice()
            .. Si se escoge "salir", cerrar la ventana
            if opc="&Salir" then
            formreturn("Salir")
            endif
            endmethod

Objeto      F_cobro B_serv_grat

Método      depart
Fuente      method depart(var eventInfo MoveEvent)
            .. Al salir, ir al campo "numrec"
            num_rec moveto()
            endmethod

Objeto      F_cobro B_serv_grat

Método      pushButton
Fuente      .. Este metodo funciona igual al del boton B_aceptar
            .. pero opera sobre la tabla de servicio gratuito
            method pushButton(var eventInfo Event)
            var
            tab,tab_pac,tab_dat tcursor
            vent form
            rep report
            opc string
            opti string
            nume number
            endvar
            setmouseshape(mousewait)
            message("Abriendo expedientes y actualizando")
            if not eventInfo isprefilter()

```

```

if not(tab_pac locate("expediente",expediente)) then
tab_pac edit()
tab_pac insertrecord()
tab_pac expediente=expediente
tab_pac.nombre=nombre
tab_pac.fecha=date()
tab_pac.postrecord()
else
tab_pac edit()
tab_pac.expediente=expediente
tab_pac.nombre=nombre
tab_pac.fecha=date()
endif
tab_pac.close()
endif
endif
if tab_dat.open("rec_grat.db") then
tab_dat.edit()
tab_dat.insertrecord()
tab_dat.numero=tab_dat.cmax("numero")+1
tab_dat.num_exp=expediente
tab_dat.fecha=fecha
tab_dat.concepto=concepto.value
tab_dat.cantidad=cantidad
tab_dat.precio=importe/cantidad
tab_dat.nom_pac=nombre
tab_dat.importe=importe
tab_dat.mat_op=matr_op
tab_dat.nom_op=nom_op
tab_dat.postrecord()
tab_dat.endedit()
endif
if tab_dat.open("control.db") then
tab_dat.edit()
tab_dat.insertrecord()
tab_dat.expediente=expediente
tab_dat.fecha=date()
tab_dat.trat=concepto
tab_dat.operador=matr_op
tab_dat.postrecord()
tab_dat.close()
else
beep()
message("No se puede registrar en hoja de control")
endif
opc=msgquestion("HISTORIA CLINICA","¿Desea Imprimir la historia clinica?")
if opc="Yes" then
if tab_dat.open("Interno") then
tab_dat.skip(1)
tab_dat.edit()
tab_dat.numero=expediente.value
tab_dat.close()
if vent.open("Iden2") then
if vent.wait()="Impreso" then
vent.close()
rep.attach("planf")
rep.print()
endif
endif
tab_dat.open("Interno")
tab_dat.skip(1)
tab_dat.edit()
tab_dat.numero=0
tab_dat.close()
endif
endif
expediente=""
nombre=""
matr_op=""
nom_op=""
concepto=""
cantidad=1
importe=""
setmouseshape(mousearrow)
guardado=true
endmethod

```

Objeto	F_cobro B_lista
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) var vent form endvar .. Activar la ficha de lista de expedientes if vent.open("lista_p") then vent.wait() vent.close() else message("No se puede abrir el archivo") endif endmethod </pre>
Objeto	F_cobro B_buscar_num
Método	pushButton
Fuente	<pre> .. Este metodo busca el numero de un expediente basado .. en el nombre del paciente method pushButton(var eventInfo Event) var nom,opc string tabla tcursor succ logical endvar nom="" .. Obtener el nombre nom.view("Teclee el apellido o nombre del paciente") setmouseshape(mousewait) nom=nom+" " message("buscando "+nom) .. Abrir la tabla if tabla.open("ident.db") then tabla.home() .. Si se encuentra el registro, se toman los datos necesarios if tabla.locatepattern(3,nom) then opc="Yes" succ=true while opc="Yes" opc=msgyesnocancel("Registro encontrado",tabla.nombre+" ¿DESEA CONTINUAR BUSCANDO?") .. Puede continuarse la busqueda if opc="Yes" and succ then if tabla.locatenextpattern(3,nom) then succ=true else succ=false endif else if opc="Yes" then if tabla.locatepriorpattern(3,nom) then succ=false else succ=true endif endif endif endwhile expediente=tabla.expediente endif else message("No se puede abrir el archivo") endif setmouseshape(mousearrow) endmethod </pre>
Objeto	F_cobro B_cancelar
Método	pushButton
Fuente	<pre> .. Vaciar los datos y cerrar la ficha method pushButton(var eventInfo Event) expediente="" nombre="" matr_op="" nom_op="" concepto="" cantidad=1 importe="" formreturn("Cancelar") endmethod </pre>
Objeto	F_cobro cantidad
Método	depart
Fuente	<pre> method depart(var eventInfo MoveEvent) .. Al salir de este campo se puede calcular el importe en base .. al precio y cantidad de servicios </pre>

```

    importe value=number(cantidad value)*number(importe value)
endif
endmethod

```

Página: 47

Objeto F_cobro C_operador matr_op

Método canDepart

```

Fuente method canDepart(var eventInfo MoveEvent)
var
    tabla tcursor
endvar
message("Buscando al operador")
.. Antes de salir del campo se valida que no este vacio
if not(self.isBlank()) then
    .. Luego se busca el nombre del alumno en la tabla
    if tabla open("alumnos db") then
        if tabla locate("matricula",matr_op) then
            nom_op value=tabla "nombre"
        else
            .. Si no se encuentra el nombre se evita la salida
            .. enviando el mensaje de "no se permite salir"
            eventInfo setErrorcode(cannotdepart)
            beep()
            message("Operador no encontrado")
        endif
        tabla close()
    else
        message("No se encuentra un archivo")
    endif
endif
endmethod

```

Objeto F_cobro NUM_REC

Método canDepart

```

Fuente method canDepart(var eventInfo MoveEvent)
.. Antes de salir de este campo se informa que
.. este registro no se ha guardado
guardado=false
endmethod

```

Objeto F_cobro #Cuadro14.CONCEPTO

Método depart

```

Fuente method depart(var eventInfo MoveEvent)
var
    tab tcursor
endvar
.. Al salir de este campo se puede determinar
.. el precio del servicio buscandolo en la
.. tabla de tratamientos
if tab open("trats") then
    if tab locate("tratamiento",self.value) then
        importe=tab costo
        precio=tab costo
    endif
    tab close()
else
    beep()
    message("No se puede determinar el precio")
endif
endmethod

```

Objeto F_cobro B_aceptar

Método depart

```

Fuente method depart(var eventInfo MoveEvent)
.. Al salir de este boton nos movemos
.. al campo numero de recibo
num_rec moveto()
endmethod

```

Objeto F_cobro B_aceptar

Método pushButton

```

Fuente method pushButton(var eventInfo Event)
var
    tab,tab_pac,tab_dat tcursor
    vent form
    rep report
    opc string
    opci string
    nume,SIG number
endvar
setmousesshape(mousewait)
message("Abriendo expedientes y actualizando")

```

```

If not(EXPEDIENTE ISBLANK()) then
;; Abrir la tabla y buscar el expediente o crear uno nuevo
If tab_pac open("Iden2.dbf") then
,, Si NO se encuentra el expediente se crea uno nuevo
If not(tab_pac locate("expediente",expediente)) then
tab_pac edit()
tab_pac insertrecord()
tab_pac expediente=expediente
tab_pac nombre=nombre
tab_pac fecha=date()
tab_pac.postrecord()
,, Se brinda la opcion de imprimir la historia clinica del paciente
opc=msgquestion("HISTORIA CLINICA","¿Desea imprimir la historia
clinica?")
If opc="Yes" then
,, Para imprimir la solo se pone el numero de expediente en la tabla
,, "Interno", de ahi la tomara la fecha de impresion al activarse
If tab_dat open("Interno")then
tab_dat skip(1)
tab_dat edit()
tab_dat numero=expediente value
tab_dat close()
If vent open("Iden2") then
If vent wait()="Impreso" then
vent close()
rep attach("plang")
rep.print()
endif
endif
tab_dat open("Interno")
tab_dat skip(1)
tab_dat edit()
tab_dat numero=0
tab_dat close()
endif
endif
else
,, Si encuentra el expediente solo se actualizan los datos
tab_pac edit()
tab_pac expediente=expediente
tab_pac nombre=nombre
tab_pac fecha=date()
endif
tab_pac.close()
endif
endif
,, Buscar el numero del recibo en la tabla adecuada
If tab_dat open("recibos.dbf") then
,, Si se encuentra puede ser que se requiera corregirlo
If tab_dat.locate("numero",num_rec)then
OPC="Yes"
opc=msgquestion("RECIBO REPETIDO","Ese recibo ya esta registrado,
¿desea borrar el anterior y poner este? Si tiene ese numero oprima [SI]")
If opc="Yes" then
tab_dat edit()
tab_dat "numero"=num_rec
tab_dat "num_exp"=expediente
tab_dat "fecha"=fecha
tab_dat "concepto"=concepto value
tab_dat "cantidad"=cantidad
tab_dat "precio"=number(importe)/number(cantidad)
tab_dat "nom_pac"=nombre
tab_dat "importe"=importe
tab_dat "mat_op"=matr_op
tab_dat "nom_op"=nom_op
tab_dat.postrecord()
tab_dat endedit()
else
,, Si no se elige corregir, el numero del recibo está
,, mal escrito, debe crearse o modificarse el correcto
nume=num_rec
nume.view("Teclee el numero correcto")
tab_dat edit()
tab_dat insertrecord()
tab_dat "numero"=num_rec
tab_dat "num_exp"=expediente
tab_dat "fecha"=fecha
tab_dat "concepto"=concepto value
tab_dat "cantidad"=cantidad
tab_dat "precio"=importe/cantidad
tab_dat "nom_pac"=nombre
tab_dat "importe"=importe
tab_dat "mat_op"=matr_op
tab_dat "nom_op"=nom_op
tab_dat.postrecord()
tab_dat endedit()
endif
else
,, Si no se encuentra el recibo se crea el nuevo
tab_dat edit()

```

```

tab_dat.insertrecord()
tab_dat.numero=num_rec
tab_dat.num_exp=expediente
tab_dat.fecha=fecha
tab_dat.concepto=concepto value
tab_dat.cantidad=cantidad
tab_dat.precio=importe/cantidad
tab_dat.nom_pac=nombre
tab_dat.importe=importe
tab_dat.mat_op=matr_op
tab_dat.nom_op=nom_op
tab_dat.postrecord()
tab_dat.endedit()
endif
SIG=tab_dat.cmax("numero")+1
tab_dat.close()
endif
.. Se crea o modifica una hoja de control de citas
if tab_dat.open("control db")then
tab_dat.edit()
tab_dat.insertrecord()
tab_dat.expediente=expediente
tab_dat.fecha=date()
tab_dat.trat=concepto
tab_dat.operador=matr_op
tab_dat.postrecord()
tab_dat.close()
else
beep()
message("No se puede registrar en hoja de control")
endif
.. Se brinda la opcion de imprimir el recibo
OPC=msgquestion("R E C I B O","¿Desea imprimir el recibo?")
if opc="Yes" then
.. Para imprimir solo se pone el numero de expediente en la tabla
.. "Interno", de ahí la tomara la ficha de impresion al activarse
if tab_dat.open("Interno")then
tab_dat.skip(1)
tab_dat.edit()
tab_dat.numero=num_rec value
tab_dat.postrecord()
tab_dat.close()
if vent.open("Impr_rec") then
vent.wait()
vent.close()
endif
tab_dat.open("Interno")
tab_dat.skip(1)
tab_dat.edit()
tab_dat.numero=0
tab_dat.postrecord()
tab_dat.close()
endif
endif
.. Se vacían los campos de datos
num_rec=sig
expediente=""
nombre=""
matr_op=""
nom_op=""
concepto=""
cantidad=1
importe=""
setmouseshape(mousearrow)
guardado=true
endmethod

```

```
method canDepart(var eventInfo MoveEvent)
var
    tab_pac tcursor
    opc string
endvar
,, Antes de salir debe verificarse que no se deje vacío este campo
,, es posible dejarlo vacío para consultas externas
message("Buscando expediente")
BLANKASZERO(FALSE)
IF not(SELf ISBLANK()) THEN
,, Si no está vacío se abre la tabla de expedientes para buscarlo
if tab_pac open("ident.dbf") then
,, En caso de que el valor sea 0 significa que el usuario desea
,, que el sistema le indique el número del expediente
if self.value=0 then
    self.value=tab_pac cmax("expediente")+1
else
,, Si no es cero, debe buscarse el expediente y mostrar el nombre
if tab_pac locate("expediente",self.value) then
    nombre=tab_pac nombre
else
,, Si no se encuentra el expediente debe notificarse
beep()
opc=msgyesnocancel("PRECAUCION","Ese número de expediente no
existe, ¿Desea crear uno nuevo?")
if opc="Cancel" then
    eventInfo seterrorcode(cannotdepart)
endif
if opc="No" then
    self blank()
BEEP()
,, Si se deja en blanco el campo no se registrará al paciente
opc=MSGYESNOCANCEL("C U I D A D O","No se registrará el
paciente")
if opc<>"Yes" then
    eventInfo seterrorcode(cannotdepart)
endif
endif
message(opc)
endif
endif
tab_pac close()
endif
ELSE
,, Si se deja vacío el campo no se registrará el paciente
BEEP()
opc=MSGYESNOCANCEL("C U I D A D O","No se registrará el paciente")
if opc<>"Yes" then
    eventInfo seterrorcode(cannotdepart)
endif
message(opc)
ENDIF
endmethod
```

Objecto F_NuevoAlumno
 Método open
 Fuente


```

method open(var eventInfo Event)
var
  prin menu
endvar
„Al abrir, maximizar, mostrar el menú y activar el modo de edición
hidespeedbar()
prin addtext("&Salir")
prin show()
maximize()
self edit()
endmethod
    
```

Objecto F_NuevoAlumno
 Método close
 Fuente


```

method close(var eventInfo Event)

endmethod
    
```

Objecto F_NuevoAlumno
 Método menuAction
 Fuente


```

method menuAction(var eventInfo MenuEvent)
var
  opc string
endvar
opc=eventInfo menuchoice()
if opc="&Salir" then
  endedit()
  formreturn("&Salir")
endif
endmethod
    
```

Objecto F_NuevoAlumno m6
 Método changeValue
 Fuente


```

method changeValue(var eventInfo ValueEvent)

endmethod
    
```

Objecto F_NuevoAlumno m5
 Método changeValue
 Fuente


```

method changeValue(var eventInfo ValueEvent)

endmethod
    
```

Objecto F_NuevoAlumno m4
 Método changeValue
 Fuente


```

method changeValue(var eventInfo ValueEvent)

endmethod
    
```

Objecto F_NuevoAlumno m3
 Método changeValue
 Fuente


```

method changeValue(var eventInfo ValueEvent)

endmethod
    
```

Objecto F_NuevoAlumno m2
 Método changeValue
 Fuente


```

method changeValue(var eventInfo ValueEvent)

endmethod
    
```

Objecto F_NuevoAlumno m1
 Método changeValue
 Fuente


```

method changeValue(var eventInfo ValueEvent)

endmethod
    
```

Objecto F_NuevoAlumno grupo

Fuente	<pre> method depart(var eventInfo MoveEvent) var mates tcursor endvar if mates open("Materias") and mates locate("Grupo",self.value) then m1=mates mat1 m2=mates mat2 m3=mates mat3 m4=mates mat4 m5=mates mat5 m6=mates mat6 endif endmethod </pre>
Objecto	F_NuevoAlumno grupo
Método	changeValue
Fuente	<pre> method changeValue(var eventInfo ValueEvent) endmethod </pre>
Objecto	F_NuevoAlumno B_Salir
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) „Desactivar el modo de edición y cerrar la ficha endedit() formreturn("Salir") endmethod </pre>
Objecto	F_NuevoAlumno B_Aceptar
Método	depart
Fuente	<pre> method depart(var eventInfo MoveEvent) matricula moveto() endmethod </pre>
Objecto	F_NuevoAlumno B_Aceptar
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) var tabla tcursor endvar „Abrir la tabla y guardar el registro del alumno setmousesshape(mousewait) if tabla open("alumnos.dbf") and tabla locate("matricula",matricula.value) then msgstop("Error","Alumno registrado antes") „Validar que no se haya registrado anteriormente else tabla edit() tabla insertrecord() tabla matricula=matricula tabla nombre=nombre tabla semestre=grupo tabla postrecord() tabla endedit() tabla close() matricula="" nombre="" grupo="" dirección="" teléfono="" endif setmousesshape(mousearrow) endmethod </pre>
Objecto	F_NuevoAlumno Matricula
Método	changeValue
Fuente	<pre> method changeValue(var eventInfo ValueEvent) var tabla tcursor endvar do default if not tabla open("alumno") then beep() msgstop("error","no se puede abrir la tabla") else if tabla nrecords()>0 then if tabla locate("matricula",self.value) or tabla matricula=self then msgstop("Error","Ese alumno ya esta registrado") endif endif endif </pre>

```

Objecto      #Página7
Método      Type

Fuente .     Type
            reg_pac=record
            clave          number
            nombre         string
            fecha          date
            Tratamiento string
            Paso            string
            endrecord
            endType

Objecto      #Página7
Método      Var

Fuente       Var
            datos reg_pac
            endVar

Objecto      #Página7
Método      open

Fuente       method open(var eventInfo Event)
            hidespeedbar()
            maximize()
            endmethod

Objecto      #Página7
Método      close

Fuente       method close(var eventInfo Event)
            endmethod

Objecto      #Página7
Método      arrive

Fuente       method arrive(var eventInfo MoveEvent)
            var
                prin menu          ,menu principal
                estas,inves,cajas,cch      popupmenu
            endvar
            maximize()

            estas addtext("&Conteo de Casos")
            estas addtext("&Gráfica por Casos")
            estas addtext("Gráfica por &Población")
            estas addtext("Gráfica de &Inventario")

            inves addtext("&Compras y cambios")
            „inves addtext("&Modificaciones")
            „inves addtext("&Revisión")
            inves addtext("&Valoración")

            cajas.addtext("&Efectivo")
            cajas addtext("&Históricos del mes")
            cajas addtext("Históricos &Por fecha")

            cch addtext("&Egresos")
            cch addtext("&Reposiciones")
            cch addtext("&Contrarecibos")
            cch addtext("&Saldo actual")

            prin addpopup("&Estadísticas",estas)
            prin addpopup("&Caja",cajas)
            prin addpopup("&Inventario",inves)
            prin addtext("&Recibos")
            prin addpopup("C c&hica",cch)
            „prin addtext("&Revista")
            prin addtext("&Salir")
            prin show()
            endmethod

Objecto      #Página7
Método      menuAction

Fuente       method menuAction(var eventInfo MenuEvent)
            var
                vent form
                opc,nom string
                num,recs number
                tab form
                t1,t2 tcursor
                fl logical
            endvar

```

```

endvar
opc=eventInfo menuchoice()
switch
case opc="&Conteo de Casos"
if vent open("conteo") then
vent wait()
vent close()
else
beep()
message("Problemas al abrir un archivo")
endif
case opc="&Gráfica por Casos"
if vent open("graf_cas") then
vent wait()
vent close()
else
beep()
message("Problemas al abrir un archivo")
endif
case opc="Gráfica por &Población"
if vent open("graf_pob") then
vent wait()
vent close()
else
beep()
message("Problemas al abrir un archivo")
endif
case opc="Gráfica de &Inventario"
if vent open("graf_inv") then
vent wait()
vent close()
else
beep()
message("Problemas al abrir un archivo")
endif
case opc="&Compras y cambios"
if vent open("compras") then
vent wait()
vent close()
else
beep()
message("Problemas al abrir un archivo")
endif
.. "&Modificaciones"
.. "&Revisión"
case opc="&Valuación"
if vent open("valua_in") then
vent wait()
vent close()
else
beep()
message("Problemas al abrir un archivo")
endif
case opc="&Efectivo"
if vent open("efectivo") then
vent wait()
vent close()
else
beep()
message("Problemas al abrir un archivo")
endif
case opc="&Históricos del mes"
showspeedbar()
if rep.open("r_hist") then
rep.maximize()
showspeedbar()
else
beep()
message("Problemas al Imprimir")
endif
hidespeedbar()
case opc="Históricos &Por fecha"
fech.blank()
fech.view("Fecha que desea Imprimir")
ff=true
if not(fech.isBlank()) then
if t1.open("rec_hist.dbf") and t2.open("rec_conc.dbf") then
t2.edit()
t2.empty()
t2.edit()
recs=0
setmouseshape(mousewait)
while ff
message(t1.numero)
if t1.locatenext("fecha",fech)then
ff=True
recs=recs+1
t2.insertrecord()

```

```

t2 num_exp=t1 num_exp
t2 fecha=t1 fecha
t2 concepto=t1 concepto
t2 precio=t1 precio
t2 cantidad=t1 cantidad
t2 importe=t1 importe
t2 mat_op=t1 mat_op
t2 postrecord()
else
ff=false
endif
endwhile
t2 endedit()
t2 close()
setmousesshape(mousearrow)
if recs>0 then
if tab open("r_hist") then
tab wait()
tab close()
endif
endif
else
message("No se puede abrir un archivo")
endif
endif
case opc="&Revista"
if vent open("ctrl_rev") then
else
beep()
message("Problemas al abrir un archivo")
endif
case opc="&Recibos"
setmousesshape(mousewait)
num=0
nom=""
OPC=msgquestion("Donde buscar","¿Sabe el NUMERO DE
EXPEDIENTE?")
if opc="Yes" then
num view("Teclee el número de expediente")
else
nom view("Teclee el nombre del paciente")
if t1 open("ident dbf") then
if t1 locate("nombre",nom) then
num=t1 numero
else
message("No se encuentra ese paciente")
num=0
endif
endif
endif
if num>0 then
ff=true
if t1 open("rec_hist dbf") and t2 open("rec_cons dbf") then
t2 edit()
t2 empty()
t2 edit()
recs=0
while ff
if t1 locatenext("num_exp",num) then
ff=true
recs=recs+1
t2 insertrecord()
t2 numero=t1 numero
t2 num_exp=t1 num_exp
t2 fecha=t1 fecha
t2 concepto=t1 concepto
t2 precio=t1 precio
t2 cantidad=t1 cantidad
t2 importe=t1 importe
t2 mat_op=t1 mat_op
t2 postrecord()
else
ff=false
endif
endwhile
t2 endedit()
t2 close()
if recs>0 and tab open("rec_cons") then
tab wait()
tab close()
endif
else
message("No se puede abrir un archivo")
endif
endif
setmousesshape(mousearrow)
case opc="&Egresos"
if vent open("reg_egr") then

```

```

vent close()
else
  beep()
  message("Problemas al abrir un archivo")
endif
case opc=="&Reposicioneo"
if vent open("reg_ingr") then
  vent wait()
  vent close()
else
  beep()
  message("Problemas al abrir un archivo")
endif
case opc=="&Contrarecbos"
if vent open("reg_conf") then
  vent wait()
  vent close()
else
  beep()
  message("Problemas al abrir un archivo")
endif
case opc=="&Saldo actua"
if vent open("saldocc") then
  vent wait()
  vent close()
else
  beep()
  message("Problemas al abrir un archivo")
endif
case opc=="&Salir" formreturn("Salir")
endswitch
endmethod

```

Objeto	#Página2
Método . Fuente	<pre> open method open(var eventInfo Event) var prin menu endvar maximize() hideprogressbar() prin addtext("&Salir") prin show() endmethod </pre>
Objeto	#Página2
Método Fuente	<pre> menuAction method menuAction(var eventInfo MenuEvent) var opc string endvar opc=eventInfo menuchoice() if opc="&Salir" then formreturn("Salir") endif endmethod </pre>
Objeto	#Página2.#Botón20
Método Fuente	<pre> pushButton method pushButton(var eventInfo Event) formreturn("Sin cerrar") endmethod </pre>

Objecto	#Página2
Método	open
Fuente	<pre>method open(var eventInfo Event) var prin menu endvar prin addtext("&Imprimir") prin addtext("&Salir") maximize() hidespeedbar() prin show() endmethod</pre>
Objecto	#Página2
Método	menuAction
Fuente	<pre>method menuAction(var eventInfo MenuEvent) var opc string endvar opc=eventInfo menuchoice() if opc="&Salir" then formreturn("Salir") endif if opc="&Imprimir" then self action(dataprint) endif endmethod</pre>

Objeto	F_GrafInv
Método	open
Fuente	<pre>method open(var eventInfo Event) var prin menu endvar „Al abrir, maximizar y mostrar el menú adecuado prin addtext("&Salir") maximize() hidespeedbar() prin show() endmethod</pre>
Objeto	F_GrafInv
Método	menuAction
Fuente	<pre>method menuAction(var eventInfo MenuEvent) var opc string endvar „La única opción del menú es salir opc=eventInfo menuchoice() if opc="&Salir" then formreturn("&Salir") endif endmethod</pre>

Objeto	F_grafica
Método	open
Fuente .	<pre> method open(var eventInfo Event) var prin menu endvar „Maximizar y mostrar el menú adecuado prin addtext("&Salir") maximize() hideshow() prin show() endmethod </pre>
Objeto	F_grafica
Método	menuAction
Fuente	<pre> „Gestión de l menú method menuAction(var eventInfo MenuEvent) var opc string endvar opc=eventInfo menuchoice() if opc="&Salir" then formreturn("&Salir") endif endmethod </pre>

Objeto	F_Históricos
Método	open
Fuente	<pre> method open(var eventInfo Event) var prin menu endvar „Al abrir la ficha, mostrar el menú if not(eventInfo Isprefilter())then maximize() hidespeedbar() self edit() prin addtext("&Imprimir") prin addtext("&Borrar registro") prin addtext("&Añadir registro") prin addtext("&Salir") prin show() endit endmethod </pre>
Objeto	F_Históricos
Método	menuAction
Fuente	<pre> „Gestión del menú method menuAction(var eventInfo MenuEvent) var opc,op string INFO REPORT datos reportprintInfo endvar opc=eventInfo menuchoice() switch case opc="&Imprimir" datos name="R_hist" info open(datos) case opc="&Borrar registro" op=msgquestion("Desea borrarlo?",string(rec_hist numero)) if op="Yes" and deleterecord() then message("Registro eliminado") endif case opc="&Añadir registro" self edit() if insertrecord() then message("Introduzca los datos") endif case opc="&Salir" formreturn("Salir") endswitch endmethod </pre>

Objecto	F_HorarioA
Método	open
Fuente	<pre> method open(var eventInfo Event) var prin menu endvar prin addtext("&Salir") prin show() maximize() locate("fecha",date()) hidespeedbar() endmethod </pre>
Objecto .	F_HorarioA
Método	menuAction
Fuente .	<pre> method menuAction(var eventInfo MenuEvent) var opc string endvar opc=eventInfo menuchoice() if opc="&Salir" then formreturn("&Salir") endif endmethod </pre>
Objecto	F_HorarioA salir
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) formreturn("&Salir") endmethod </pre>

Objecto	F_HorarioB
Método	open
Fuente	<pre>method open(var eventInfo Event) var prin menu endvar prin addtext("&Salir") prin show() maximize() locate("fecha",date()) hidespeedbar() endmethod</pre>
Objecto	F_HorarioB
Método	menuAction
Fuente	<pre>method menuAction(var eventInfo MenuEvent) var opc string endvar opc=eventInfo menuchoice() if opc="&Salir" then formreturn("Salir") endif endmethod</pre>
Objecto	F_HorarioB.salir
Método	pushButton
Fuente	<pre>method pushButton(var eventInfo Event) formreturn("Salir") endmethod</pre>

Objeto	F_HistClinica
Método	open
Fuente	<pre> method open(var eventInfo Event) var tab_int tcursor prin menu endvar „Al abrir la ficha se debe buscar en la tabla "Interno" si „ hay un numero registrado, de ser así, significa que „ alguna ficha envió el mensaje de imprimir una historia „ clínica, debe localizarse el registro e imprimirse if not(eventInfo Isprefilter())then do default if tab_int open("Interno")then tab_int movetorecno(2) if (tab_int numero<>0) then „Si hay un número, localizarlo if locate("expediente",tab_int numero)then „Si se localiza, imprimirlo action(dataprint) endif formreturn("impreso") endif endif prin addtext("&Imprimir") prin addtext("&Salir") prin show() endif endmethod </pre>
Objeto	F_HistClinica
Método	menuAction
Fuente	<pre> method menuAction(var eventInfo MenuEvent) var opc string endvar „Obtener la seleccion del menu opc=eventInfo menuchoice() switch „Imprimir los datos de pantalla case opc="&Imprimir" action(dataprint) „Cerrar la ficha case opc="&Salir" formreturn("Salir") endswitch endmethod </pre>
Objeto	F_HistClinica.vac
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) var pos smallint s1 string endvar pos=0 s1="" pos=antecedentes search("U") if pos=0 then antecedentes=antecedentes+"U" else s1=antecedentes substr(1,pos-1) if pos<antecedentes size() then antecedentes=antecedentes substr(pos+1,antecedentes size()-pos) antecedentes=s1+antecedentes else antecedentes=s1 endif endif endmethod </pre>
Objeto	F_HistClinica.ein

Método pushButton

Fuente

```
method pushButton(var eventInfo Event)
var
  pos smallInt
  s1 string
endvar
pos:=0
s1=""
pos:=antecedentes search("Ñ")
if pos=0 then
  antecedentes:=antecedentes+"Ñ"
else
  s1:=antecedentes substr(1,pos-1)
  if pos<antecedentes size() then
    antecedentes:=antecedentes substr(pos+1,antecedentes size()-pos)
    antecedentes:=s1+antecedentes
  else
    antecedentes:=s1
  endif
endif
endmethod
```

Objecto F_HistClinica.emb

Método pushButton

Fuente

```
method pushButton(var eventInfo Event)
var
  pos smallInt
  s1 string
endvar
pos:=0
s1=""
pos:=antecedentes search("B")
if pos=0 then
  antecedentes:=antecedentes+"B"
else
  s1:=antecedentes substr(1,pos-1)
  if pos<antecedentes size() then
    antecedentes:=antecedentes substr(pos+1,antecedentes size()-pos)
    antecedentes:=s1+antecedentes
  else
    antecedentes:=s1
  endif
endif
endmethod
```

Objecto F_HistClinica.dia

Método pushButton

Fuente

```
method pushButton(var eventInfo Event)
var
  pos smallInt
  s1 string
endvar
pos:=0
s1=""
pos:=antecedentes search("D")
if pos=0 then
  antecedentes:=antecedentes+"D"
else
  s1:=antecedentes substr(1,pos-1)
  if pos<antecedentes size() then
    antecedentes:=antecedentes substr(pos+1,antecedentes size()-pos)
    antecedentes:=s1+antecedentes
  else
    antecedentes:=s1
  endif
endif
endmethod
```

Objecto F_HistClinica.ane

Método pushButton

Fuente

```
method pushButton(var eventInfo Event)
var
  pos smallInt
  s1 string
```

```

s1=""
pos=antecedentes search("I")
if pos=0 then
    antecedentes=antecedentes+"I"
else
    s1=antecedentes substr(1,pos-1)
    if pos<antecedentes size() then
        antecedentes=antecedentes substr(pos+1,antecedentes size()-pos)
        antecedentes=s1+antecedentes
    else
        antecedentes=s1
    endif
endif
endmethod

```

Objeto **F_HistClinica.asm**

Método
Fuente

```

pushButton
method pushButton(var eventInfo Event)
var
    pos smallint
    s1 string
endvar
pos=0
s1=""
pos=antecedentes search("A")
if pos=0 then
    antecedentes=antecedentes+"A"
else
    s1=antecedentes substr(1,pos-1)
    if pos<antecedentes size() then
        antecedentes=antecedentes substr(pos+1,antecedentes size()-pos)
        antecedentes=s1+antecedentes
    else
        antecedentes=s1
    endif
endif
endmethod

```

Objeto . **F_HistClinica.tra**

Método
Fuente

```

pushButton
method pushButton(var eventInfo Event)
var
    pos smallint
    s1 string
endvar
pos=0
s1=""
pos=antecedentes search("N")
if pos=0 then
    antecedentes=antecedentes+"N"
else
    s1=antecedentes substr(1,pos-1)
    if pos<antecedentes size() then
        antecedentes=antecedentes substr(pos+1,antecedentes size()-pos)
        antecedentes=s1+antecedentes
    else
        antecedentes=s1
    endif
endif
endmethod

```

Objeto **F_HistClinica.ten**

Método
Fuente

```

pushButton
method pushButton(var eventInfo Event)
var
    pos smallint
    s1 string
endvar
pos=0
s1=""
pos=antecedentes search("G")

```

```

    antecedentes=antecedentes+"G"
else
    s1=antecedentes substr(1,pos-1)
    if pos<antecedentes size() then
        antecedentes=antecedentes substr(pos+1,antecedentes size()-pos)
        antecedentes=s1+antecedentes
    else
        antecedentes=s1
    endif
endif
endmethod

```

Objecto **F_HistClinica.pre**

Método **pushButton**

Fuente **method pushButton(var eventInfo Event)**

```

var
    pos smallInt
    s1 string
endvar
pos=0
s1=""
pos=antecedentes search("P")
if pos=0 then
    antecedentes=antecedentes+"P"
else
    s1=antecedentes substr(1,pos-1)
    if pos<antecedentes size() then
        antecedentes=antecedentes substr(pos+1,antecedentes size()-pos)
        antecedentes=s1+antecedentes
    else
        antecedentes=s1
    endif
endif
endmethod

```

Objecto **F_HistClinica.fie**

Método **pushButton**

Fuente **method pushButton(var eventInfo Event)**

```

var
    pos smallInt
    s1 string
endvar
pos=0
s1=""
pos=antecedentes search("F")
if pos=0 then
    antecedentes=antecedentes+"F"
else
    s1=antecedentes substr(1,pos-1)
    if pos<antecedentes size() then
        antecedentes=antecedentes substr(pos+1,antecedentes size()-pos)
        antecedentes=s1+antecedentes
    else
        antecedentes=s1
    endif
endif
endmethod

```

Objecto **F_HistClinica.ren**

Método **pushButton**

Fuente **method pushButton(var eventInfo Event)**

```

var
    pos smallInt
    s1 string
endvar
pos=0
s1=""
pos=antecedentes search("R")

```

```

    antecedentes=antecedentes+"R"
  else
    s1=antecedentes substr(1,pos-1)
    if pos<antecedentes size() then
      antecedentes=antecedentes substr(pos+1,antecedentes size()-pos)
      antecedentes=s1+antecedentes
    else
      antecedentes=s1
    endif
  endif
endmethod

```

Objeto **F_HistClinica.eve**

Método **pushButton**

Fuente **method pushButton(var eventInfo Event)**

```

var
  pos smallInt
  s1 string
endvar
pos=0
s1=""
pos=antecedentes search("V")
if pos=0 then
  antecedentes=antecedentes+"V"
else
  s1=antecedentes substr(1,pos-1)
  if pos<antecedentes size() then
    antecedentes=antecedentes substr(pos+1,antecedentes size()-pos)
    antecedentes=s1+antecedentes
  else
    antecedentes=s1
  endif
endif
endmethod

```

Objeto **F_HistClinica.tub**

Método **pushButton**

Fuente **method pushButton(var eventInfo Event)**

```

var
  pos smallInt
  s1 string
endvar
pos=0
s1=""
pos=antecedentes search("T")
if pos=0 then
  antecedentes=antecedentes+"T"
else
  s1=antecedentes substr(1,pos-1)
  if pos<antecedentes size() then
    antecedentes=antecedentes substr(pos+1,antecedentes size()-pos)
    antecedentes=s1+antecedentes
  else
    antecedentes=s1
  endif
endif
endmethod

```

Objeto **F_HistClinica.epi**

Método **pushButton**

Fuente **method pushButton(var eventInfo Event)**

```

var
  pos smallInt
  s1 string
endvar
pos=0
s1=""

```

```

    antecedentes=antecedentes+"E"
  else
    s1=antecedentes substr(1,pos-1)
    if pos<antecedentes size() then
      antecedentes=antecedentes substr(pos+1,antecedentes size()-pos)
      antecedentes=s1+antecedentes
    else
      antecedentes=s1
    endif
  endif
endmethod

```

Objeto **F_HistClinica.ata**

Método **pushButton**
Fuente

```

method pushButton(var eventInfo Event)
  var
    pos smallInt
    s1 string
  endvar
  pos=0
  s1=""
  pos=antecedentes search("C")
  if pos=0 then
    antecedentes=antecedentes+"C"
  else
    s1=antecedentes substr(1,pos-1)
    if pos<antecedentes size() then
      antecedentes=antecedentes substr(pos+1,antecedentes size()-pos)
      antecedentes=s1+antecedentes
    else
      antecedentes=s1
    endif
  endif
endmethod

```

Objeto **F_HistClinica.sop**

Método **pushButton**
Fuente

```

method pushButton(var eventInfo Event)
  var
    pos smallInt
    s1 string
  endvar
  pos=0
  s1=""
  pos=antecedentes search("S")
  if pos=0 then
    antecedentes=antecedentes+"S"
  else
    s1=antecedentes substr(1,pos-1)
    if pos<antecedentes size() then
      antecedentes=antecedentes substr(pos+1,antecedentes size()-pos)
      antecedentes=s1+antecedentes
    else
      antecedentes=s1
    endif
  endif
endmethod

```

Objeto **F_HistClinica.hep**

Método **pushButton**
Fuente

```

method pushButton(var eventInfo Event)
  var
    pos smallInt
    s1 string
  endvar
  pos=0
  s1=""

```

```

    antecedentes=antecedentes+"H"
else
    s1=antecedentes substr(1,pos-1)
    if pos<antecedentes size() then
        antecedentes=antecedentes substr(pos+1,antecedentes size()-pos)
        antecedentes=s1+antecedentes
    else
        antecedentes=s1
    endif
endif
endmethod

```

Objeto **F_HistClinica.ale**

Método **setFocus**

Fuente **method setFocus(var eventInfo Event)**

endmethod

Objeto **F_HistClinica.ale**

Método **pushButton**

Fuente **method pushButton(var eventInfo Event)**
var
 pos smallint
 s1 string
endvar
 pos=0
 s1=""
 pos=antecedentes search("M")
 if pos=0 then
 antecedentes=antecedentes+"M"
 else
 s1=antecedentes substr(1,pos-1)
 if pos<antecedentes size() then
 antecedentes=antecedentes substr(pos+1,antecedentes size()-pos)
 antecedentes=s1+antecedentes
 else
 antecedentes=s1
 endif
 endif
endmethod

Objeto **F_HistClinica.NOMBRE**

Método **setFocus**

Fuente **method setFocus(var eventInfo Event)**
message("Cambio de pantalla")
beep()
 if antecedentes search("M")>0 then
 ale=true
 else
 ale=false
 endif
 if antecedentes search("H")>0 then
 hep=true
 else
 Hep=false
 endif
 if antecedentes search("C")>0 then
 ata=true
 else
 ata=false
 endif
 if antecedentes search("S")>0 then
 sop=true
 else
 sop=false
 endif
endmethod

```

    fle=true
  else
    fle=false
  endif
  if antecedentes search("P")>0 then
    pre=true
  else
    pre=false
  endif
  if antecedentes search("E")>0 then
    epi=true
  else
    epi=false
  endif
  if antecedentes search("T")>0 then
    Tub=true
  else
    tub=false
  endif
  if antecedentes search("V")>0 then
    eve=true
  else
    eve=false
  endif
  if antecedentes search("R")>0 then
    ren=true
  else
    ren=false
  endif
  if antecedentes search("N")>0 then
    tra=true
  else
    tra=false
  endif
  if antecedentes search("G")>0 then
    ten=true
  else
    ten=false
  endif
  if antecedentes search("A")>0 then
    asm=true
  else
    asm=false
  endif
  if antecedentes search("F")>0 then
    ane=true
  else
    ane=false
  endif
  if antecedentes search("D")>0 then
    dia=true
  else
    dia=false
  endif
  if antecedentes search("B")>0 then
    emb=true
  else
    emb=false
  endif
  if antecedentes search("Ñ")>0 then
    ein=true
  else
    ein=false
  endif
  if antecedentes search("U")>0 then
    vac=true
  else
    vac=false
  endif
endmethod

```

```

Objecto      #Página2

MethodName   open

Source       method open(var eventInfo Event)
var
    tab_Int tcursor
    prin menu
endvar
„Al abrir la ficha (siempre desde otra) se busca el registro
„ que se debe imprimir, el cual se ha puesto previamente
„ en la tabla "Interno"
If not(eventInfo Isprefilter())then
    do default
    If tab_Int open("Interno")then
        tab_Int skip(1)
        If (tab_Int numero<>0) then
            If locate("NUMERO",tab_Int numero)then
                SELF action(dataprint)
            endif
            formreturn("Impreso")
        endif
    endif
    prin addtext("&Imprimir")
    prin addtext("&Salir")
    prin show()
endif
endmethod

```

```

Objecto      #Página2

MethodName   menuAction

Source       method menuAction(var eventInfo MenuEvent)
var
    opc string
endvar
opc=eventInfo menuchoice()
If opc="&Imprimir" then
    self action(dataprint)
endif
If opc="&Salir" then
    formreturn("Salir")
endif
endmethod

```

```

Objeto      F_Normal

Método      Var
Fuente      Var
            t1 tcursor
            endVar

Objeto      F_Normal

Método      open
Fuente      method open(var eventInfo Event)
            „Maximizar y abrir la tabla de tratamientos
            if not(eventInfo {sprefilter()}) then
                maximize()
                t1 open("trats")
            end if
        endmethod

Objeto      F_Normal con1

Método      depart
Fuente      method depart(var eventInfo MoveEvent)
        endmethod

Objeto      F_Normal con1

Método      changeValue
Fuente      method changeValue(var eventInfo ValueEvent)
        endmethod

Objeto      F_Normal B_CambConcepto

Método      pushButton
Fuente      method pushButton(var eventInfo Event)
            „Actualizar el concepto del recibo con el de la pantalla
            F_Normal action(databeginedit)
            F_Normal concepto=F_Normal con1
            F_Normal action(datapostrecord)
        endmethod

Objeto      F_Normal B_Otro

Método      pushButton
Fuente      method pushButton(var eventInfo Event)
            var
                num,mat,exp,prec,cant,imp,mes number
                con,nom string
                fec date
            endvar
            „Crear otro recibo con los F_Normal de la pantalla
            num=F_Normal numero
            mat=F_Normal mat_op
            exp=F_Normal num_exp
            con=F_Normal concepto
            prec=F_Normal precio
            cant=F_Normal cantidad
            imp=F_Normal importe
            fec=F_Normal fecha
            mes=F_Normal mes
            „Editar e insertar el nuevo registro
            F_Normal action(databeginedit)
            F_Normal action(datainsertrecord)
            F_Normal numero=num
            F_Normal concepto=F_Normal con1
            F_Normal cantidad=cant
            F_Normal importe=0
            F_Normal precio=prec
            F_Normal fecha=fec
            F_Normal mes=mes
            „Hacer los cambios en disco
            F_Normal action(datapostrecord)
            F_Normal action(dataendedit)
            „Volver a la posición donde ocurrió el error
            F_Normal locate("numero",num)
        endmethod

Objeto      F_Normal B_Continuar

Método      pushButton

```

```
var
  bien logical
endvar
bien=true
while bien
  t1 nombre()
  „Avanzar mientras el concepto registrado esté en la tabla
  „ de tratamientos
  if t1 locate("tratamiento",F_Normal concepto) then
    bien=true
    F_Normal skip(1)
  else
    „Cuando no encuentre un concepto, notificarlo y detenerse
    message("no encuentro "+F_Normal concepto)
    bien=false
  endif
endwhile
endmethod
```

Objeto #Página2

Método open

Fuente *method open(var eventInfo Event)*
 „Al abrir la ficha (siempre desde otra) se busca el registro
 „ que se debe imprimir, el cual se ha puesto previamente
 „ en la tabla "Interno"
endmethod

Objeto #Página2

Método action

Fuente *method action(var eventInfo ActionEvent)*

endmethod

Objeto #Página2 CIRUGIA

Método pushButton

Fuente *method pushButton(var eventInfo Event)*
 var
 pos smallInt
 s1 string
endvar
 pos=0
 s1=""
 pos=parodont search("Q")
 if pos=0 then
 parodont=parodont+"Q"
 else
 s1=parodont substr(1,pos-1)
 if pos<parodont size() then
 parodont=parodont substr(pos+1,parodont size()-pos)
 parodont=s1+parodont
 else
 parodont=s1
 endif
 endif
endmethod

Objeto #Página2 DX_QUIR

Método pushButton

Fuente *method pushButton(var eventInfo Event)*
 var
 pos smallInt
 s1 string
endvar
 pos=0
 s1=""
 pos=parodont search("X")
 if pos=0 then
 parodont=parodont+"X"
 else
 s1=parodont substr(1,pos-1)
 if pos<parodont size() then
 parodont=parodont substr(pos+1,parodont size()-pos)
 parodont=s1+parodont
 else
 parodont=s1
 endif
 endif
endmethod

Objeto #Página2 TER_OCLU

Método pushButton

Fuente *method pushButton(var eventInfo Event)*
 var
 pos smallInt
 s1 string
endvar
 pos=0
 s1=""
 pos=parodont search("A")
 if pos=0 then
 parodont=parodont+"A"
 else
 s1=parodont substr(1,pos-1)
 if pos<parodont size() then
 parodont=parodont substr(pos+1,parodont size()-pos)
 parodont=s1+parodont
 else
 parodont=s1
 endif
 endif
endmethod

```
endlf
endmethod
```

Objeto	#Página2 CIR_PRE
Método	pushButton
Fuente	<pre>method pushButton(var eventInfo Event) var pos smallint s1 string endvar pos=0 s1="" pos=parodont search("C") If pos=0 then parodont=parodont+"C" else s1=parodont substr(1,pos-1) If pos<parodont size() then parodont=parodont substr(pos+1,parodont size()-pos) parodont=s1+parodont else parodont=s1 endif endif endmethod</pre>
Objeto	#Página2 PRO_TEMP
Método	pushButton
Fuente	<pre>method pushButton(var eventInfo Event) var pos smallint s1 string endvar pos=0 s1="" pos=parodont search("T") If pos=0 then parodont=parodont+"T" else s1=parodont substr(1,pos-1) If pos<parodont size() then parodont=parodont substr(pos+1,parodont size()-pos) parodont=s1+parodont else parodont=s1 endif endif endmethod</pre>
Objeto	#Página2 EXODON
Método	pushButton
Fuente	<pre>method pushButton(var eventInfo Event) var pos smallint s1 string endvar pos=0 s1="" pos=parodont search("E") If pos=0 then parodont=parodont+"E" else s1=parodont substr(1,pos-1) If pos<parodont size() then parodont=parodont substr(pos+1,parodont size()-pos) parodont=s1+parodont else parodont=s1</pre>

```
endif
endmethod
```

Página: 77

```
Objecto      #Página2 PROFIL

Método      pushButton
Fuente      method pushButton(var eventInfo Event)
            var
            pos smallint
            s1 string
            endvar
            pos=0
            s1=""
            pos=parodont search("P")
            if pos=0 then
                parodont=parodont+"P"
            else
                s1=parodont substr(1,pos-1)
                if pos<parodont size() then
                    parodont=parodont substr(pos+1,parodont size()-pos)
                    parodont=s1+parodont
                else
                    parodont=s1
                endif
            endif
            endmethod
```

```
Objecto      #Página2 PRES_MED

Método      pushButton
Fuente      method pushButton(var eventInfo Event)
            var
            pos smallint
            s1 string
            endvar
            pos=0
            s1=""
            pos=parodont search("M")
            if pos=0 then
                parodont=parodont+"M"
            else
                s1=parodont substr(1,pos-1)
                if pos<parodont size() then
                    parodont=parodont substr(pos+1,parodont size()-pos)
                    parodont=s1+parodont
                else
                    parodont=s1
                endif
            endif
            endmethod
```

```
Objecto      #Página2 PRU_LAB

Método      pushButton
Fuente      method pushButton(var eventInfo Event)
            var
            pos smallint
            s1 string
            endvar
            pos=0
            s1=""
            pos=parodont search("L")
            if pos=0 then
                parodont=parodont+"L"
            else
                s1=parodont substr(1,pos-1)
                if pos<parodont size() then
                    parodont=parodont substr(pos+1,parodont size()-pos)
                    parodont=s1+parodont
                else
                    parodont=s1
                endif
            endif
            endmethod
```

Objeto	#Página2 REHAB_PRO
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) var pos smallInt s1 string endvar pos=0 s1="" pos=parodont search("I") if pos=0 then parodont=parodont+"I" else s1=parodont substr(1,pos-1) if pos<parodont size() then parodont=parodont substr(pos+1,parodont size()-pos) parodont=s1+parodont else parodont=s1 endif endif endmethod </pre>
Objeto	#Página2 TER_PARO
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) var pos smallInt s1 string endvar pos=0 s1="" pos=parodont search("D") if pos=0 then parodont=parodont+"D" else s1=parodont substr(1,pos-1) if pos<parodont size() then parodont=parodont substr(pos+1,parodont size()-pos) parodont=s1+parodont else parodont=s1 endif endif endmethod </pre>
Objeto	#Página2 ENDODON
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) var pos smallInt s1 string endvar pos=0 s1="" pos=parodont search("N") if pos=0 then parodont=parodont+"N" else s1=parodont substr(1,pos-1) if pos<parodont size() then parodont=parodont substr(pos+1,parodont size()-pos) parodont=s1+parodont else parodont=s1 endif endif endmethod </pre>

```
endif
endmethod
```

Página: 79

Objeto	#Página2 PREVEN
Método	pushButton
Fuente	<pre>method pushButton(var eventInfo Event) var pos smallInt s1 string endvar pos=0 s1="" pos=parodont search("R") if pos=0 then parodont=parodont+"R" else s1=parodont substr(1,pos-1) if pos<parodont size() then parodont=parodont substr(pos+1,parodont size()-pos) parodont=s1+parodont else parodont=s1 endif endif endmethod</pre>
Objeto	#Página2 INTERCON
Método	pushButton
Fuente	<pre>method pushButton(var eventInfo Event) var pos smallInt s1 string endvar pos=0 s1="" pos=parodont search("") if pos=0 then parodont=parodont+" " else s1=parodont substr(1,pos-1) if pos<parodont size() then parodont=parodont substr(pos+1,parodont size()-pos) parodont=s1+parodont else parodont=s1 endif endif endmethod</pre>
Objeto	#Página2 OPER_DEN
Método	pushButton
Fuente	<pre>method pushButton(var eventInfo Event) var pos smallInt s1 string endvar pos=0 s1="" pos=parodont search("O") if pos=0 then parodont=parodont+"O" else s1=parodont substr(1,pos-1) if pos<parodont size() then parodont=parodont substr(pos+1,parodont size()-pos) parodont=s1+parodont else parodont=s1</pre>

```
endif
endmethod
```

Página: 80

Objeto	#Página2 AJUS_OCLU
Método	pushButton
Fuente	<pre>method pushButton(var eventInfo Event) var pos smallInt s1 string endvar pos:=0 s1="" pos:=parodont search("J") if pos=0 then parodont:=parodont+"J" else s1:=parodont substr(1,pos-1) if pos<parodont size() then parodont:=parodont substr(pos+1,parodont size()-pos) parodont:=s1+parodont else parodont:=s1 endif endif endmethod</pre>
Objeto .	#Página2 MED_ORAL
Método	pushButton
Fuente	<pre>method pushButton(var eventInfo Event) var pos smallInt s1 string endvar pos:=0 s1="" pos:=parodont search("Y") if pos=0 then parodont:=parodont+"Y" else s1:=parodont substr(1,pos-1) if pos<parodont size() then parodont:=parodont substr(pos+1,parodont size()-pos) parodont:=s1+parodont else parodont:=s1 endif endif endmethod</pre>
Objeto	#Página2 ORTODON
Método	pushButton
Fuente	<pre>method pushButton(var eventInfo Event) var pos smallInt s1 string endvar pos:=0 s1="" pos:=parodont search("K") if pos=0 then parodont:=parodont+"K" else s1:=parodont substr(1,pos-1) if pos<parodont size() then parodont:=parodont substr(pos+1,parodont size()-pos) parodont:=s1+parodont else parodont:=s1</pre>

```
endif  
endmethod
```

Objeto	#Página2 PROSTO
Método	pushButton
Fuente	method pushButton(var eventInfo Event) var pos smallInt s1 string enúvar pos=0 s1="" pos=parodont search("U") if pos=0 then parodont=parodont+"U" else s1=parodont substr(1,pos-1) if pos<parodont size() then parodont=parodont substr(pos+1,parodont size()-pos) parodont=s1+parodont else parodont=s1 endif endif endmethod

```

Objeto      #Página2

Método      arrive
Fuente      method arrive(var eventInfo MoveEvent)
            var
            prin menu
            endvar
            if not(eventInfo Isprefilter())then
            maximize()
            hidespeedbar()
            prin addtext("&Imprimir")
            prin addtext("&Salir")
            prin show()
            endif
            endmethod

Objeto      #Página2

Método      menuAction
Fuente      method menuAction(var eventInfo MenuEvent)
            var
            opc string
            endvar
            opc=eventInfo menuchoice()
            switch
            case opc="&Imprimir"
            self action(dataprint)
            case opc="&Salir"
            formreturn("&Salir")
            endswitch
            endmethod

```

R_HIST.FSL

```

Objeto      #Página2

Método      arrive
Source      method arrive(var eventInfo MoveEvent)
            var
            prin menu
            endvar
            if not(eventInfo Isprefilter())then
            maximize()
            hidespeedbar()
            prin addtext("&Imprimir")
            prin addtext("&Salir")
            prin show()
            endif
            endmethod

Objeto      #Página2

Método      menuAction
Source      method menuAction(var eventInfo MenuEvent)
            var
            opc string
            endvar
            opc=eventInfo menuchoice()
            switch
            case opc="&Imprimir"
            self action(dataprint)
            case opc="&Salir"
            formreturn("&Salir")
            endswitch
            endmethod

```

Objecto	#Página2
Método	Var
Fuente	<pre> Var nuevo logical endvar </pre>
Objecto	#Página2
Método	open
Fuente	<pre> method open(var eventInfo Event) var tab tcursor endvar if eventInfo isprefilter()then else if tab open("contrare.db") then numero=tab cmax("numero")+1 fechapag=date() fechaem=date() tab close() else msgstop("ERROR","No se puede abrir el archivo") formreturn("ERROR") endif endif endmethod </pre>
Objecto	#Página2
Método	arrive
Fuente	<pre> method arrive(var eventInfo MoveEvent) var prin menu endvar if not(eventInfo isprefilter())then maximize() hidespeedbar() prin addtext("&Salir") prin show() endif endmethod </pre>
Objecto	#Página2
Método	menuAction
Fuente	<pre> method menuAction(var eventInfo MenuEvent) var opc string endvar opc=eventInfo menuchoice() if opc="&Salir" then formreturn("Salir") endif endmethod </pre>
Objecto	#Página2 #Botón22
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) var tab tcursor num number endvar if tab open("contrare")then num=0 num view("Teclee el numero de movimiento") if num>0 then if tab locate("numero",num) then numero=tab numero fechaem=tab "fecha factura" fechapag=tab "fecha pago" importe=tab importe conce=tab concepto prove=tab proveedor contrarecibo=tab contrarecibo numfac=tab factura numfac move() else beep() message("No se encuentra ese ingreso") endif endif else msgstop("ERROR","No se puede abrir el archivo") </pre>

Objecto	#Página2 #Botón20
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) formreturn("Salir") endmethod </pre>
Objecto	#Página2 #Botón18
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) var tab tcursor andvar if tab open("contrare db") then tab edit() if tab locate("numero",numero) then else tab insertrecord() endif tab numero=numero tab "fecha factura"=fechaem tab "fecha pago"=fechapag tab importe=importe tab concepto=conce tab proveedor=prove tab contrarecibo=contrarecibo tab factura=numfac tab postrecord() numero=tab cmax("numero")+1 tab close() fecha_c=date() importe=0 0 fechaem=date() fechapag=date() conce="" prove="" contrarecibo blank() numfac blank() numfac moveto() else msgstop("ERROR","No se pueden guardar los datos") endif endmethod </pre>

Objeto	F_Egresos
Método	var
Fuente	<pre> var nuevo logical „Bandera para determinar si un egreso es nuevo o „ es modificación de uno ya capturado endvar </pre>
Objeto	F_Egresos
Método	open
Fuente	<pre> method open(var eventInfo Event) var tab tcursor endvar „Al abrir la ficha se busca el numero de registro „ adecuado y se actualizan las variables if not(eventInfo Isprefilter())then „ Se verifica que no ocurra el evento prefilter, porque „ de ser así se heredaría el metodo a todos los objetos if tab open("repuesto DB")then if tab nrecords()<1 then msgstop("ERROR","Debe registrar los ingresos de caja antes de registrar egresos") tab close() formreturn("error") endif tab end() numero1=tab numero fecha1=tab COBRO tab close() endif if tab open("egresos DB") then numero=tab cmax("numero")+1 fecha=date() tab close() else msgstop("ERROR","No se puede abrir el archivo") formreturn("ERROR") endif endif endmethod </pre>
Objeto	F_Egresos
Método	arrive
Fuente	<pre> method arrive(var eventInfo MoveEvent) var prin menu endvar „Cuando se llegue a la ficha maximizar la ventana „ y mostrar el menu adecuado if not(eventInfo Isprefilter())then maximize() hideprogressbar() prin addtext("&Salir") prin show() endif endmethod </pre>
Objeto	F_Egresos
Método	menuAction
Fuente	<pre> method menuAction(var eventInfo MenuEvent) var opc string endvar „Obtener la seleccion del menu opc=eventInfo menuchoice() „Cerrar la ventana si eso se desea if opc="&Salir" then formreturn("Salir") endif endmethod </pre>
Objeto	F_Egresos B_Modificar
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) var tab tcursor num number endvar „Abrir la tabla de egresos </pre>

```

num=0
num view("Teclee el numero de movimiento")
,,Obtener el número de egreso a modificar
if num>0 then
,,Localizarlo y hacer los cambios
if tab locate("numero",num) then
numero=tab numero
fecha=tab fecha
concepto=tab concepto
proveedor=tab proveedor
importe=tab importe
numero1=tab reposicion
fecha1 blank()
else
beep()
message("No se encuentra ese Ingreso")
endif
endif
fecha moveto()
else
msgstop("ERROR","No se puede abrir el archivo")
endif
endmethod

```

Objeto F_Egresos B_Salir

Método
Fuente

```

pushButton
method pushButton(var eventInfo Event)
,,Cerrar la ficha cuando se oprima el boton
formreturn("Salir")
endmethod

```

Objeto F_Egresos B_Guardar

Método
Fuente

```

pushButton
method pushButton(var eventInfo Event)
var
tab tcursor
endvar
,,Registrar el egreso
if tab open("egresos db") then
tab edit()
if tab locate("numero",numero) then
,,Verificar que no esté registrado
else
tab insertrecord()
endif
tab numero=numero
tab fecha=fecha
tab concepto=concepto
tab proveedor=proveedor
tab importe=importe
tab reposicion=numero1+1
tab postrecord()
numero=tab cmax("numero")+1
tab close()
fecha=date()
concepto blank()
proveedor blank()
importe=0 0
,,Obtener el No. de reposición
if tab open("repuesto")then
tab end()
numero1=tab numero
fecha1=tab COBRO
tab close()
endif
fecha moveto()
else
msgstop("ERROR","No se pueden guardar los datos")
endif
fecha moveto()
endmethod

```

Objeto	F_RegIngresos
Método	Var
Fuente	Var nuevo logical endVar
Objeto	F_RegIngresos
Método	open
Fuente	method open(var eventInfo Event) var tab tcursor endvar if eventInfo Isprefilter() then else „Abrir la tabla de repuestos y actualizar el número en la variable if tab open("repuesto db") then numero=tab cmax("numero")+1 fecha_c=date() tab close() else msgstop("ERROR","No se puede abrir el archivo") formreturn("ERROR") endif endif endmethod
Objeto	F_RegIngresos
Método	arrive
Fuente	method arrive(var eventInfo MoveEvent) var pin menu endvar „Maximizar al entrar a la ficha „ Checar que no ocurra el evento "prefiltro" „ ya que ésto significaría heredar el método a todos „ los objetos if not(eventInfo Isprefilter()) then maximize() hidespeedbar() pin addtext("&Salir") pin show() endif endmethod
Objeto	F_RegIngresos
Método	menuAction
Fuente	„Gestión del menú method menuAction(var eventInfo MenuEvent) var opc string endvar opc=eventInfo menuchoice() if opc="&Salir" then formreturn("Salir") endif endmethod
Objeto	F_RegIngresos B_Buscar
Método	pushButton
Fuente	method pushButton(var eventInfo Event) var tab tcursor num number endvar „Abrir la tabla de repuestos if tab open("repuesto") then num=0 num view("Teclee el numero de movimiento") „Obtener el movimiento a buscar y localizarlo if num>0 then if tab locate("numero",num) then „Mostrar datos en pantalla numero=tab numero fecha_c=tab cobro importe=tab importe importe moveto() else beep() message("No se encuentra ese ingreso") endif endif

```

msgstop("ERROR","No se puede abrir el archivo")
end!
endmethod

```

Objecto F_RegIngresos B_Salir

Método
Fuente

```

pushButton
„Cerrar la ficha al oprimir el boton
method pushButton(var eventInfo Event)
formreturn("Salir")
endmethod

```

Objecto F_RegIngresos B_Guardar

Método
Fuente

```

pushButton
method pushButton(var eventInfo Event)
var
tab tcursor
endvar
„Actualizar el número de reposición capturado
if tab open("repuesto db") then
tab edit()
if tab.locate("numero",numero) then
else
tab insertrecord()
endif
„Registrar los datos en el registro insertado
tab numero=numero
tab cobro=fecha_c
tab importe=importe
tab postrecord()
numero=tab cmax("numero")+1
tab close()
fecha_c=date()
importe=0 0
importe moveto()
else
msgstop("ERROR","No se pueden guardar los datos")
end!
endmethod

```

Objeto	F_RegMaterias
Método	open
Fuente	<pre> method open(var eventInfo Event) var prin menu endvar „Al abrir la ficha, maximizar y mostrar el menú if not(eventInfo Isprefilter())then maximize() hidespeedbar() prin addtext("&Salir") prin show() self edit() endif endmethod </pre>
Objeto	F_RegMaterias
Método	menuAction
Fuente	<pre> method menuAction(var eventInfo MenuEvent) var opc string endvar „Gestión del menú opc=eventInfo menuchoice() if opc="&Salir" then formreturn("Salir") endif endmethod </pre>
Objeto	F_RegMaterias #Botón12
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) formreturn("Salir") endmethod </pre>

Objeto	F_RegPaciente
Método	open
Fuente	<pre> method open(var eventInfo Event) „Al abrir la ficha (desde otra), moverse al último registro „ e insertar uno nuevo maximize() hideSpeedbar() end() edit() skip(1) endmethod </pre>
Objeto	F_RegPaciente vac
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) var pos smallint s1 string endvar pos=0 s1="" pos=antecedentes search("U") if pos=0 then antecedentes=antecedentes+"U" else s1=antecedentes substr(1,pos-1) if pos<antecedentes size() then antecedentes=antecedentes substr(pos+1,antecedentes size()-pos) antecedentes=s1+antecedentes else antecedentes=s1 endif endif endmethod </pre>
Objeto	F_RegPaciente ein
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) var pos smallint s1 string endvar pos=0 s1="" pos=antecedentes search("Ñ") if pos=0 then antecedentes=antecedentes+"Ñ" else s1=antecedentes substr(1,pos-1) if pos<antecedentes size() then antecedentes=antecedentes substr(pos+1,antecedentes size()-pos) antecedentes=s1+antecedentes else antecedentes=s1 endif endif endmethod </pre>
Objeto	F_RegPaciente emb
Método	pushButton
Fuente	<pre> method pushButton(var eventInfo Event) var pos smallint s1 string endvar pos=0 s1="" pos=antecedentes search("B") if pos=0 then antecedentes=antecedentes+"B" else s1=antecedentes substr(1,pos-1) if pos<antecedentes size() then antecedentes=antecedentes substr(pos+1,antecedentes size()-pos) antecedentes=s1+antecedentes else antecedentes=s1 endif endif endmethod </pre>
Objeto	F_RegPaciente dia

```

var
  pos smallInt
  s1 string
endvar
pos:=0
s1=""
pos:=antecedentes search("D")
if pos=0 then
  antecedentes:=antecedentes+"D"
else
  s1:=antecedentes substr(1,pos-1)
  if pos<antecedentes size() then
    antecedentes:=antecedentes substr(pos+1,antecedentes size()-pos)
    antecedentes:=s1+antecedentes
  else
    antecedentes:=s1
  endif
endif
endmethod

```

Objecto F_RegPaciente ana

Método
Fuente

```

pushButton
method pushButton(var eventInfo Event)
var
  pos smallInt
  s1 string
endvar
pos:=0
s1=""
pos:=antecedentes search("I")
if pos=0 then
  antecedentes:=antecedentes+"I"
else
  s1:=antecedentes substr(1,pos-1)
  if pos<antecedentes size() then
    antecedentes:=antecedentes substr(pos+1,antecedentes size()-pos)
    antecedentes:=s1+antecedentes
  else
    antecedentes:=s1
  endif
endif
endmethod

```

Objecto F_RegPaciente asm

Método
Fuente

```

pushButton
method pushButton(var eventInfo Event)
var
  pos smallInt
  s1 string
endvar
pos:=0
s1=""
pos:=antecedentes search("A")
if pos=0 then
  antecedentes:=antecedentes+"A"
else
  s1:=antecedentes substr(1,pos-1)
  if pos<antecedentes size() then
    antecedentes:=antecedentes substr(pos+1,antecedentes size()-pos)
    antecedentes:=s1+antecedentes
  else
    antecedentes:=s1
  endif
endif
endmethod

```

Objecto F_RegPaciente tra

Método
Fuente

```

pushButton
method pushButton(var eventInfo Event)
var
  pos smallInt
  s1 string
endvar
pos:=0
s1=""
pos:=antecedentes search("N")
if pos=0 then
  antecedentes:=antecedentes+"N"
else
  s1:=antecedentes substr(1,pos-1)
  if pos<antecedentes size() then
    antecedentes:=antecedentes substr(pos+1,antecedentes size()-pos)
    antecedentes:=s1+antecedentes
  else
    antecedentes:=s1
  endif
endif
endmethod

```

Objeto	F_RegPaciente ten
Método Fuente	<pre> pushButton method pushButton(var eventInfo Event) var pos smallInt s1 string endvar pos=0 s1="" pos=antecedentes search("G") if pos=0 then antecedentes=antecedentes+"G" else s1=antecedentes substr(1,pos-1) if pos<antecedentes size() then antecedentes=antecedentes substr(pos+1,antecedentes size()-pos) antecedentes=s1+antecedentes else antecedentes=s1 endif endif endmethod </pre>
Objeto	F_RegPaciente pre
Método Fuente	<pre> pushButton method pushButton(var eventInfo Event) var pos smallInt s1 string endvar pos=0 s1="" pos=antecedentes search("P") if pos=0 then antecedentes=antecedentes+"P" else s1=antecedentes substr(1,pos-1) if pos<antecedentes size() then antecedentes=antecedentes substr(pos+1,antecedentes size()-pos) antecedentes=s1+antecedentes else antecedentes=s1 endif endif endmethod </pre>
Objeto	F_RegPaciente fle
Método Fuente	<pre> pushButton method pushButton(var eventInfo Event) var pos smallInt s1 string endvar pos=0 s1="" pos=antecedentes search("F") if pos=0 then antecedentes=antecedentes+"F" else s1=antecedentes substr(1,pos-1) if pos<antecedentes size() then antecedentes=antecedentes substr(pos+1,antecedentes size()-pos) antecedentes=s1+antecedentes else antecedentes=s1 endif endif endmethod </pre>

Objecto F_RegPaciente ren

```

Método
Fuente
pushButton
method pushButton(var eventInfo Event)
var
pos smallInt
s1 string
endvar
pos=0
s1=""
pos=antecedentes search("R")
if pos=0 then
antecedentes=antecedentes+"R"
else
s1=antecedentes substr(1,pos-1)
if pos<antecedentes size() then
antecedentes=antecedentes substr(pos+1,antecedentes size()-pos)
antecedentes=s1+antecedentes
else
antecedentes=s1
endif
endif
endmethod

```

Objecto F_RegPaciente eve

```

Método
Fuente
pushButton
method pushButton(var eventInfo Event)
var
pos smallInt
s1 string
endvar
pos=0
s1=""
pos=antecedentes search("V")
if pos=0 then
antecedentes=antecedentes+"V"
else
s1=antecedentes substr(1,pos-1)
if pos<antecedentes size() then
antecedentes=antecedentes substr(pos+1,antecedentes size()-pos)
antecedentes=s1+antecedentes
else
antecedentes=s1
endif
endif
endmethod

```

Objecto F_RegPaciente tub

```

Método
Fuente
pushButton
method pushButton(var eventInfo Event)
var
pos smallInt
s1 string
endvar
pos=0
s1=""
pos=antecedentes search("T")
if pos=0 then
antecedentes=antecedentes+"T"
else
s1=antecedentes substr(1,pos-1)
if pos<antecedentes size() then
antecedentes=antecedentes substr(pos+1,antecedentes size()-pos)
antecedentes=s1+antecedentes
else
antecedentes=s1
endif
endif

```

```

Objecto      F_RegPaciente api
Método
Fuente      pushButton
            method pushButton(var eventInfo Event)
            var
            pos smallint
            s1 string
            endvar
            pos:=0
            s1=""
            pos:=antecedentes search("E")
            if pos=0 then
            antecedentes=antecedentes+"E"
            else
            s1=antecedentes substr(1,pos-1)
            if pos<antecedentes size() then
            antecedentes=antecedentes substr(pos+1,antecedentes size()-pos)
            antecedentes=s1+antecedentes
            else
            antecedentes=s1
            endif
            endif
            endmethod

```

```

Objecto      F_RegPaciente ata
Método
Fuente      pushButton
            method pushButton(var eventInfo Event)
            var
            pos smallint
            s1 string
            endvar
            pos:=0
            s1=""
            pos:=antecedentes search("C")
            if pos=0 then
            antecedentes=antecedentes+"C"
            else
            s1=antecedentes substr(1,pos-1)
            if pos<antecedentes size() then
            antecedentes=antecedentes substr(pos+1,antecedentes size()-pos)
            antecedentes=s1+antecedentes
            else
            antecedentes=s1
            endif
            endif
            endmethod

```

```

Objecto      F_RegPaciente sop
Método
Fuente      pushButton
            method pushButton(var eventInfo Event)
            var
            pos smallint
            s1 string
            endvar
            pos:=0
            s1=""
            pos:=antecedentes search("S")
            if pos=0 then
            antecedentes=antecedentes+"S"
            else
            s1=antecedentes substr(1,pos-1)
            if pos<antecedentes size() then
            antecedentes=antecedentes substr(pos+1,antecedentes size()-pos)
            antecedentes=s1+antecedentes
            else
            antecedentes=s1

```

Objecto	F_RegPaciente hep
Método Fuente	<pre> pushButton method pushButton(var eventInfo Event) var pos smallint s1 string endvar pos:=0 s1="" pos:=antecedentes search("H") if pos=0 then antecedentes=antecedentes+"H" else s1=antecedentes substr(1,pos-1) if pos<antecedentes size() then antecedentes=antecedentes substr(pos+1,antecedentes size()-pos) antecedentes=s1+antecedentes else antecedentes=s1 endif endif endmethod </pre>
Objecto	F_RegPaciente ale
Método Fuente	<pre> setFocus method selfFocus(var eventInfo Event) endmethod </pre>
Objecto	F_RegPaciente ale
Método Fuente	<pre> pushButton method pushButton(var eventInfo Event) var pos smallint s1 string endvar pos:=0 s1="" pos:=antecedentes search("M") if pos=0 then antecedentes=antecedentes+"M" else s1=antecedentes substr(1,pos-1) if pos<antecedentes size() then antecedentes=antecedentes substr(pos+1,antecedentes size()-pos) antecedentes=s1+antecedentes else antecedentes=s1 endif endif endmethod </pre>
Objecto	F_RegPaciente #Botón5
Método Fuente	<pre> pushButton method pushButton(var eventInfo Event) formreturn("Salir") endmethod </pre>
Objecto	F_RegPaciente #Botón3
Método	pushButton

```

skip(1)
ale=false
Hep=false
ata=false
sop=false
fle=false
pre=false
epi=false
tub=false
eve=false
ren=false
tra=false
ten=false
asm=false
ane=false
dia=false
amb=false
eln=false
vac=false
nombre moveto()
endmethod

```

Objecto F_RegPaciente NOMBRE

Método
Fuente

```

selfFocus
method selfFocus(var eventInfo Event)
message("Cambio de pantalla")
beep()
if antecedentes search("M")>0 then
ale=true
else
ale=false
endif
if antecedentes search("H")>0 then
Hep=true
else
Hep=false
endif
if antecedentes search("C")>0 then
ata=true
else
ata=false
endif
if antecedentes search("S")>0 then
sop=true
else
sop=false
endif
if antecedentes search("F")>0 then
fle=true
else
fle=false
endif
if antecedentes search("P")>0 then
pre=true
else
pre=false
endif
if antecedentes search("E")>0 then
epi=true
else
epi=false
endif
if antecedentes search("T")>0 then
Tub=true
else
tub=false
endif
if antecedentes search("V")>0 then
eve=true
else
eve=false
endif
if antecedentes search("R")>0 then
ren=true
else
ren=false
endif
if antecedentes search("N")>0 then
tra=true
else
tra=false
endif
if antecedentes search("Q")>0 then
ten=true
else
ten=false
endif

```

```

asm=true
else
asm=false
endif
if antecedentes search("I")>0 then
ana=true
else
ana=false
endif
if antecedentes search("D")>0 then
dia=true
else
dia=false
endif
if antecedentes search("B")>0 then
emb=true
else
emb=false
endif
if antecedentes search("N")>0 then
eln=true
else
eln=false
endif
if antecedentes search("U")>0 then
vac=true
else
vac=false
endif
endmethod

```

Objecto	#Página2
Método	open
Fuente	<pre> method open(var eventinfo Event) var prin menu endvar maximize() hidespeedbar() prin addtext("&Imprimir") prin addtext("&Salir") prin show() endmethod </pre>
Objecto	#Página2
Método	menuAction
Fuente	<pre> method menuAction(var eventinfo MenuEvent) var opc string endvar opc=eventinfo menuchoice() switch case opc="&Imprimir" self action(dataprint) case opc="&Salir" formreturn("Salir") endswitch endmethod </pre>

```

Objecto      F_SaldoCajChica

MethodName   Var
Source       Var
             nuevo logical
             endVar

Objecto      F_SaldoCajChica

MethodName   open
Source       method open(var eventInfo Event)
             var
             tab tcursor
             endvar
             „Al abrir la ficha, maximizar
             „ se comprueba que no ocurra "prefilter" pues eto significaría
             „ heredar el método a todos los objetos
             if eventInfo isprefilter()then

             else
             maximize()
             hidespeedbar()
             endif
             endmethod

Objecto      F_SaldoCajChica

MethodName   arrive
Source       „Mostrar el menú adecuado
             method arrive(var eventInfo MoveEvent)
             var
             prin menu
             endvar
             if not(eventInfo isprefilter())then
             maximize()
             hidespeedbar()
             prin addtext("&Salir")
             prin show()
             endif
             endmethod

Objecto      F_SaldoCajChica

MethodName   menuAction
Source       method menuAction(var eventInfo MenuEvent)
             var
             opc string
             endvar
             opc=eventInfo menuchoice()
             If opc="&Salir" then
             formreturn("Salir")
             endif
             endmethod

Objecto      F_SaldoCajChica saldo

MethodName   calcField
Source       totegr-totrep

Objecto      F_SaldoCajChica #Botón20

MethodName   pushButton
Source       method pushButton(var eventInfo Event)
             formreturn("Salir")
             endmethod

```

```

Objecto      FICH_INDI

MethodName   Var
Source       Var
            tabla tcursor
            titulos array[] string
            h 1 longint
            posy number
            opcion string
            endVar

Objecto      FICH_INDI

MethodName   open
Source       method open(var eventInfo Event)
            var
            x number
            Inter tcursor
            endvar
            maximize()
            hidespeedbar()
            if tabla open("revista db") and Inter open("interno") then
            x=tabla nrecords()
            if x>=7 then
            titulos setsize(x)
            else
            titulos setsize(7)
            endif
            switch
            case Inter campo="titulo"
            i=0
            tabla home()
            while not(tabla eof())
            titulos[i]=tabla titulo
            i=i+1
            tabla nextrecord()
            endwhile
            case Inter campo="autor"
            i=0
            tabla home()
            while not(tabla eof())
            if not(titulos contains(tabla autor)) then
            i=i+1
            titulos[i]=tabla autor
            endif
            tabla nextrecord()
            endwhile
            case Inter campo="fecha"
            i=0
            tabla home()
            while not(tabla eof())
            if not(titulos contains(string(tabla fecha))) then
            i=i+1
            titulos[i]=string(tabla fecha)
            endif
            tabla.nextrecord()
            endwhile
            endswitch
            Inter close()
            h=1
            if i<7 then
            for h from i+1 to 7
            titulos[h]=" "
            endfor
            endif
            fich_indi tits tit1 value=titulos[1]
            fich_indi tits tit2 value=titulos[2]
            fich_indi tits tit3 value=titulos[3]
            fich_indi tits tit4 value=titulos[4]
            fich_indi tits tit5 value=titulos[5]
            fich_indi tits tit6 value=titulos[6]
            fich_indi tits tit7 value=titulos[7]
            posy=1
            opcion=titulos[1]
            else
            message("No se encuentra el archivo de la revista")
            endif
            endmethod

Objecto      FICH_INDI #Botón39

MethodName   mouseEnter
Source       method mouseEnter(var eventInfo MouseEvent)
            message("Cancelar la operación")
            endmethod

```

Objecto FICH_INDI #Botón39

MethodName mouseExit
Source method mouseExit(var eventInfo MouseEvent)
message("")
endmethod

Objecto FICH_INDI #Botón39

MethodName pushButton
Source method pushButton(var eventInfo Event)
formreturn("Cancelar")
endmethod

Objecto FICH_INDI #Botón37

MethodName mouseEnter
Source method mouseEnter(var eventInfo MouseEvent)
message("Proceder con el renglón marcado")
endmethod

Objecto FICH_INDI #Botón37

MethodName mouseExit
Source method mouseExit(var eventInfo MouseEvent)
message("")
endmethod

Objecto FICH_INDI #Botón37

MethodName pushButton
Source method pushButton(var eventInfo Event)
var
inter tcursor
endvar
inter open("Interno")
inter edit()
inter cadena=opcion
inter endedit()
inter close()
formreturn(opcion)
endmethod

Objecto FICH_INDI tlt5

MethodName proc
Source proc dotonme()
var
posdot point
posme point
endvar
posdot=dot position
posme=self position
posdot sety(posme y()+20)
dot position=posdot
endproc

Objecto FICH_INDI tlt5 #Botón9

MethodName . pushButton
Source method pushButton(var eventInfo Event)
if l>7 and posy+7<=l then
posy=posy+1
fich_indi tlt5 tlt1=ttulos[posy]
fich_indi tlt5 tlt2=ttulos[posy+1]
fich_indi tlt5 tlt3=ttulos[posy+2]
fich_indi tlt5 tlt4=ttulos[posy+3]
fich_indi tlt5 tlt5=ttulos[posy+4]
fich_indi tlt5 tlt6=ttulos[posy+5]
fich_indi tlt5 tlt7=ttulos[posy+6]
else
beep()
endif
endmethod

Objecto FICH_INDI tlt5 #Cuadro3

MethodName mouseClicked
Source method mouseClicked(var eventInfo MouseEvent)
beep()
endmethod

Objecto FICH_INDI tils #Cuadro3 dot
 MethodName mouseClicked
 Source method mouseClicked(var eventInfo MouseEvent)
 message(self position)
 endmethod

Objecto FICH_INDI tils #Cuadro3 TIT7
 MethodName selfFocus
 Source method selfFocus(var eventInfo Event)
 doOnme()
 opcion=self value
 endmethod

Objecto FICH_INDI tils #Cuadro3 TIT7
 MethodName mouseClicked
 Source method mouseClicked(var eventInfo MouseEvent)
 doOnme()
 opcion=self value
 endmethod

Objecto FICH_INDI tils #Cuadro3 TIT6
 MethodName selfFocus
 Source method selfFocus(var eventInfo Event)
 doOnme()
 opcion=self value
 endmethod

Objecto FICH_INDI tils #Cuadro3 TIT6
 MethodName mouseClicked
 Source method mouseClicked(var eventInfo MouseEvent)
 doOnme()
 opcion=self value
 endmethod

Objecto FICH_INDI tils #Cuadro3 TIT3
 MethodName selfFocus
 Source method selfFocus(var eventInfo Event)
 doOnme()
 opcion=self value
 endmethod

Objecto FICH_INDI tils #Cuadro3 TIT3
 MethodName mouseClicked
 Source method mouseClicked(var eventInfo MouseEvent)
 doOnme()
 opcion=self value
 endmethod

Objecto FICH_INDI tils #Cuadro3 TIT4
 MethodName selfFocus
 Source method selfFocus(var eventInfo Event)
 doOnme()
 opcion=self value
 endmethod

Objecto FICH_INDI tils #Cuadro3 TIT4
 MethodName mouseClicked
 Source method mouseClicked(var eventInfo MouseEvent)
 doOnme()
 opcion=self value
 endmethod

Objecto FICH_INDI tils #Cuadro3 TIT5
 MethodName selfFocus
 Source method selfFocus(var eventInfo Event)
 doOnme()
 opcion=self value
 endmethod

Objecto	FICH_INDI tifs #Cuadro3 TIT5
MethodName Source	mouseClick method mouseClick(var eventInfo MouseEvent) do when me() opcion=self.value endmethod
Objecto	FICH_INDI tifs #Cuadro3 TIT2
MethodName Source	selfFocus method selfFocus(var eventInfo Event) do when me() opcion=self.value endmethod
Objecto	FICH_INDI tifs #Cuadro3 TIT2
MethodName Source	mouseClick method mouseClick(var eventInfo MouseEvent) do when me() opcion=self.value endmethod
Objecto	FICH_INDI tifs #Cuadro3 TIT1
MethodName Source	selfFocus method selfFocus(var eventInfo Event) do when me() opcion=self.value endmethod
Objecto	FICH_INDI tifs #Cuadro3 TIT1
MethodName Source	mouseClick method mouseClick(var eventInfo MouseEvent) do when me() opcion=self.value endmethod
Objecto	FICH_INDI tifs #Botón10
MethodName Source	pushButton method pushButton(var eventInfo Event) if posy>=1 then posy=posy-1 fich_indi tifs tit1=titulos[posy] fich_indi tifs tit2=titulos[posy+1] fich_indi tifs tit3=titulos[posy+2] fich_indi tifs tit4=titulos[posy+3] fich_indi tifs tit5=titulos[posy+4] fich_indi tifs tit6=titulos[posy+5] fich_indi tifs tit7=titulos[posy+6] else beep() endif endmethod

```
Objecto      #Página2

Método .     open
Fuente       method open(var eventInfo Event)
             var
             prin menu
             endvar
             maximize()
             hideshowbar()
             prin addtext("&Salir")
             prin show()
             edit()
             endmethod

Objecto      #Página2

Método       menuAction
Fuente       method menuAction(var eventInfo MenuEvent)
             var
             opc string
             endvar
             opc=eventInfo menuchoice()
             if opc="&Salir" then
             formreturn("Salir")
             endif
             endmethod

Objecto      #Página2 #Botón20

Método       pushButton
Fuente       method pushButton(var eventInfo Event)
             formreturn("Salir")
             endmethod
```

Objecto	#Página2
Método	open
Fuente	method open(var eventInfo Event) var prin menu endvar maximize() hidespeedbar() prin addtext("&Imprimir") prin addtext("&Salir") prin show() endmethod
Objecto	#Página2
Método	menuAction
Fuente	method menuAction(var eventInfo MenuEvent) var opc string endvar opc=eventInfo menuchoice() switch case opc="&Imprimir" if action(dataprint) then message("Se imprimió") else beep() message("Problemas de Impresión") endif case opc="&Salir" formreturn("&Salir") endswitch endmethod
Objecto	#Página2 INVENTAR #Registro5 Importe
Método	calcField
Fuente	{INVENTAR Precio_promedio}*{INVENTAR Existencia}
Objecto	#Página2 Precio_actual
Método	calcField
Fuente	sum({INVENTAR Precio_promedio}*{INVENTAR Existencia})

Objecto	#Página2
Método	open
Fuente	<pre> method open(var eventInfo Event) var prin menu endvar „Al abrir la ficha, maximizar y mostrar el menú prin addtext("&Salir") maximize() hidespeedbar() prin show() endmethod </pre>
Objecto	#Página2
Método	menuAction
Fuente	<pre> „Gestión del menú method menuAction(var eventInfo MenuEvent) var opc string endvar opc=eventInfo menuchoice() if opc="&Salir" then formreturn("Salir") endif endmethod </pre>
Objecto	#Página2 #Botón14
Método	pushButton
Fuente	<pre> „Cerrar la ficha al oprimir el botón method pushButton(var eventInfo Event) formreturn("Salir") endmethod </pre>

11. BIBLIOGRAFIA

- Orilia, Lawrence S.
Las Computadoras y la Información
3a. Edición, 1990
Ed. Mc Graw Hill
- Goldratt, Eliyahu M.
El Síndrome del Pajar
Cómo Extraer Información del Océano de Datos
1a. Edición, 1992
Ed. Ediciones Castillo
- Checkand, Peter
Pensamiento de Sistemas, Práctica de Sistemas
1a. Edición, 1993
Ed. Grupo Noriega Editores.
- Norton, Peter/ Peter Aitken/ Richard Wilton
PC. Programmers Bible
3a. Edición, 1993
Ed. Microsoft Press
- Pressman, Roger S.
Ingeniería de Software, Un Enfoque Práctico
3a. Edición, 1993
Ed. Yourdon Press Computing Series
- Date, C.J. Ver. Español Malpica, Jaime
Introducción a los Sistemas de Bases de Datos
5a. Edición, 1990
Ed. Addison/Wesley Iberoamericana
- Kort, Henry F./Silberschatz, Abraham
Fundamentos de las Bases de Datos
2a. Edición, 1993
- Tanenbaum, Andrew S.
Redes de Ordenadores
2a. Edición, 1991
Ed. Prentice Hall
- Temple, Hooper/ Williamson
Diseño de Redes Locales
1a. Edición, 1989
Ed. Addison/Wesley Iberoamericana



- Black, Uyless
Redes de Computadoras
Protocolos, Normas e Interfaces
1a. Edición, 1990
Ed. Macribit
- Mc. Cann, John T./ segal Rick
Netware Lan Manager ToolKit
1a. Edición, 1992
Ed. Sams Publishing
- Cox, Brad J./Novobilski, Andrew J.
Programación Orientada a Objetos
Un Enfoque Evolutivo
2a. Edición, 1993
Ed. addison-Weasley/Diaz de Santos
- Khoshafian, Setrag/Abnous, Razmik
Objet Orientation
Concepts, Languages, Databases User Interfaces
1a. Edición, 1990
Ed. Wiley
- Tsui Bar-Dauij
Objet Oriented Design For C++
1a. Edición, 1993
Ed. Prentice Hall
- Coad, Peter/Nicola, Jill
Objet Oriented Programming
1a. Edición, 1993
Ed. Yourdon Press Computing Series
- Booch, Grady
Objet Oriented Analysis And Design With Applications
2a. Edición, 1994
Ed. The Benjamin/Cummins Publishing Company
- Sally Shlayer/Stephen J. Mellor
Objet Oriented Systems Analysis Modeling the World in Data
1988
Ed. Yourdon Press Computing Series

- Windows Para Trabajo En Grupo
 - Introducción
 - Manual del Usuario
 - Manual del Usuario del Mail
 - Versión 3.1
 - 1992
 - Microsoft Corporation

- Paradox Para Windows
 - Guía de Introducción
 - Guía del Usuario
 - Introducción a ObjetPAL
 - Guía de Referencia Rápida
 - Guía del Programador ObjetPAL
 - Guía de Referencia ObjetPAL
 - Versión 1.0
 - 1992
 - Borland International, Inc.