

UNIVERSIDAD DON BOSCO

FACULTAD DE INGENIERÍA

ESCUELA DE ELECTRÓNICA



TRABAJO DE GRADUACION:

**DISEÑO DE UN SISTEMA DE IDENTIFICACIÓN DIGITAL PARA
LA OPTIMIZACIÓN DEL SERVICIO EN EL ÁMBITO
HOSPITALARIO**

PRESENTADO POR:

**EDGARD ADONAY AGUIRRE LÓPEZ
ALEXANDER SALVADOR CASTILLO LUCERO
RODRIGO ARNOLDO NAJARRO ALVAREZ**

PARA OPTAR AL GRADO DE:

INGENIERÍA EN TELECOMUNICACIONES

MARZO DE 2008



1	CAPITULO I	5
1.1	DESCRIPCIÓN DEL TEMA	5
1.2	JUSTIFICACIÓN DEL TEMA	7
1.3	OBJETIVOS	8
1.3.1	Objetivo General	8
1.3.2	Objetivos Específicos.....	8
1.4	ALCANCES	9
1.5	LIMITACIONES.....	10
1.6	VALIDACIÓN.....	11
1.6.1	Validación técnica.	11
1.6.2	Validación operativa	13
1.6.3	Validación económica	13
1.7	PLAN DE TRABAJO.....	14
1.7.1	Disponibilidad de tiempo	14
1.7.2	Recursos.....	14
1.7.3	Organización de las actividades	15
1.8	CRONOGRAMA	16
2	CAPITULO II	19
2.1	TECNOLOGÍAS DE ACCESO.....	19
2.1.1	CÓDIGO DE BARRAS	20
2.1.1.1	Características	21
2.1.1.2	Tipos de código de barras.....	22
2.1.1.3	Arquitectura	24
2.1.1.4	Ventajas.....	29



2.1.1.5	Desventajas	29
2.1.1.6	Aplicaciones.....	30
2.1.2	TARJETAS INTELIGENTES (SMART CARD).....	31
2.1.2.1	Características	33
2.1.2.2	Tipos de tarjetas inteligentes	33
2.1.2.3	Arquitectura	36
2.1.2.4	Ventajas.....	38
2.1.2.5	Desventajas	39
2.1.2.6	Aplicaciones.....	39
2.1.3	TECNOLOGIA DE ACCESO BIOMETRICO	41
2.1.3.1	Características	41
2.1.3.2	Tipos de Biometría.....	43
2.1.3.3	Arquitectura	44
2.1.3.4	Ventajas.....	50
2.1.3.5	Desventajas	51
2.1.3.6	Aplicaciones de los sistemas biométricos	52
2.1.4	BANDA MAGNÉTICA.....	52
2.1.4.1	Características	53
2.1.4.2	Tipos de Banda Magnética	55
2.1.4.3	Arquitectura	57
2.1.4.4	Ventajas.....	58
2.1.4.5	Desventajas	59
2.1.4.6	Aplicaciones de Banda Magnética.....	59
2.1.5	IDENTIFICACIÓN POR RADIO FRECUENCIA (RFID).....	60
2.1.5.1	Características	61
2.1.5.2	Tipos de etiquetas RFID.....	62
2.1.5.3	Arquitectura	63
2.1.5.4	Ventajas.....	65
2.1.5.5	Desventajas	66
2.1.5.6	Aplicaciones.....	66
2.2	CUADRO RESUMEN COMPARATIVO.....	68



3	CAPITULO III.....	70
3.1	PROBLEMAS A NIVEL NACIONAL.....	72
3.1.1	Problema Económico.....	72
3.1.2	Problema de Recurso Humano.....	73
3.1.3	Problema Tecnológico.....	74
3.1.4	Problemas Estructurales.....	75
3.2	ESTUDIO DE LA MUESTRA.....	77
3.3	SELECCIÓN DE LA MUESTRA.....	78
3.4	HOSPITAL NACIONAL ESPECIALIZADO “ROSALES”.....	79
3.4.1	Problema Económico.....	81
3.4.2	Problema de Recurso Humano.....	82
3.4.3	Problema de Infraestructura.....	84
3.4.4	Problema Tecnológico.....	86
4	CAPITULO IV.....	89
4.1	CONSULTA EXTERNA.....	90
4.1.1	ANÁLISIS TÉCNICO.....	95
4.1.2	ANÁLISIS ECONÓMICO.....	95
4.1.2.1	Hardware.....	96
4.1.2.2	Software.....	101
4.1.3	TOTAL DE INVERSION EN EQUIPO COMÚN.....	101
4.2	COMPARATIVA DE COSTOS DE IMPLEMENTACION DE TECNOLOGÍAS.....	102
5	CAPITULO V.....	104
5.1	HARDWARE.....	104
5.2	SOFTWARE.....	104



5.3	APLICACIÓN	105
5.3.1	SISTEMA DE IDENTIFICACION DIGITAL DE PACIENTES (SIDP)	106
5.3.1.1	Procedimiento Nº 1: Ejecución.....	106
5.3.1.2	Procedimiento Nº 2: Autenticación.....	107
5.3.1.3	Procedimiento Nº 3. Operación.	109
5.3.2	PAGINA WEB	125
5.3.3	TABLAS COMPARATIVAS ENTRE SISTEMA ACTUAL Y SISTEMA PROPUESTO	129
5.3.4	ESCRITURA EN TARJETAS RFID	134
6	CAPITULO VI.....	136
6.1	MARCO POLÍTICO.....	136
6.2	MARCO LEGAL.....	137
6.3	TRATAMIENTOS DE DATOS SOBRE SALUD	137
6.3.1	Ficheros de hospitales.....	138
6.3.1.1	Tratamientos de datos sanitarios	138
7	CONCLUSIONES	142
8	RECOMENDACIONES	145
9	ANEXOS.....	148
10	REFERENCIAS.....	439



1 CAPITULO I

1.1 DESCRIPCIÓN DEL TEMA

Indudablemente uno de los mayores problemas que afronta la humanidad es la desatención de la salud personal y social, siendo este uno de los factores que más inciden en el bienestar de los ciudadanos. La falta de salud no permite a las personas acceder a educación, ni mucho menos a oportunidades de trabajo, esto dificulta el pleno desarrollo de cada uno de los individuos.

En nuestro país, el servicio de salud presenta muchas deficiencias entre las cuales se puede mencionar la falta y calidad del medicamento, la infraestructura inadecuada para la prestación del servicio, el bajo presupuesto al sector de acuerdo a sus necesidades, la atención prestada al usuario, principalmente en la rapidez con la que se recauda su información personal tal como: llenado de formulario, búsqueda de expediente, asignación de citas, etc.

En base a los problemas mencionados y con el desarrollo de las nuevas tecnologías, se ha pensado en una solución la cual vendrá a optimizar el sistema de identificación actual en dicho servicio.

Para alcanzar esta finalidad, se llevará a cabo una recopilación de datos estadísticos de los pacientes que asisten diariamente a los diferentes hospitales y unidades de salud, esto como base para visualizar la problemática, desde un punto de vista más objetivo.

A partir de esto se seleccionará como muestra un centro de salud en particular, donde se analizarán sus procesos integrados administrativos, tomándose en cuenta todos los procedimientos legales, de integridad y confidencialidad,



sometiéndose a las normativas estipuladas por los organismos existentes que regulan el manejo de datos, y así evaluar los recursos necesarios que serán requeridos para el diseño del sistema de identificación.

Posteriormente, tomando como referencia las diferentes tecnologías de acceso implementadas en los servicios de salud en otros países, se realizará una comparativa entre ellas, para determinar cuál es la alternativa más indicada de acuerdo a las características que cada una posee.

Para finalizar se pretende diseñar una aplicación que sea capaz de almacenar la información primordial de cada uno de los usuarios que utilizan este servicio, y a través de la misma poder acceder de forma inmediata a una base de datos local donde se encuentra su historial clínico detalladamente.

El sistema de identificación, entrará en operación cuando una persona se acerque a la recepción de un centro de asistencia médica en particular. Este será identificado instantáneamente a través de un dispositivo que reconoce la tarjeta que se le asignó en su primera visita. Una vez se efectuó este proceso, se tendrá acceso al expediente clínico del paciente, con lo cual se determinará la acción a tomar.

Después de esto el médico asignado tendrá a su disposición el expediente del paciente, el cual se le hará llegar, por medio de una red que tendrá acceso a una base de datos centralizada. Luego de la consulta el médico podrá asignar citas, prescribir medicamentos, realización de exámenes, y actualizar la información en la misma base de datos. Posteriormente el paciente podrá retirar los medicamentos en la farmacia utilizando el mismo mecanismo de identificación al salir del hospital.



1.2 JUSTIFICACIÓN DEL TEMA

Uno de los sectores que presentan un grave problema en eficiencia, tanto para gobiernos e individuos, es el relacionado con la salud de los ciudadanos.

El servicio de la salud es considerado una de las prestaciones más importantes, y que diariamente concentra una alta afluencia de personas en los diferentes centros de salud de nuestro país, lo que revela la necesidad de un sistema de servicio eficaz.

Este tipo de problemas y el vertiginoso avance de las telecomunicaciones, nos lleva a la búsqueda de nueva soluciones.

Las telecomunicaciones nos han brindado en los últimos años un sin número de herramientas y es sin duda una de las alternativas que nos ofrecen una variedad de soluciones, por las múltiples aplicaciones en donde se puede utilizar.

El sistema de identificación permitirá el intercambio de la información clínica del paciente de manera más eficaz mediante el uso de las comunicaciones inalámbricas. Evitando que los pacientes tengan que llenar algún tipo de formulario o en la espera de la búsqueda de su expediente, además de agilizar la entrega de los medicamentos.



1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Diseñar un sistema de identificación digital de pacientes que permita agilizar los procesos de registro de datos empleados en una consulta médica.

1.3.2 OBJETIVOS ESPECÍFICOS

1. Brindar una herramienta que permita modernizar el proceso de identificación de los pacientes, en el área de la salud.
2. Recopilar y analizar la información estadística de los diferentes centros de salud a nivel nacional.
3. Seleccionar un centro de salud, como muestra para el diseño del sistema de identificación.
4. Evaluar las deficiencias de infraestructura y servicio en la muestra seleccionada, para determinar los recursos que se necesitarán en el diseño del sistema de identificación.
5. Comparativa del servicio propuesto con el modelo utilizado actualmente.
6. Estudio de tecnologías de acceso para el desarrollo de aplicaciones.
7. Investigar las tecnologías de acceso utilizadas en el área de la salud, a nivel mundial.



1.4 ALCANCES

- Determinar los centros de salud que poseen mayor afluencia de pacientes.
- Investigación de los diferentes aspectos legales, razones operativas, y de infraestructura en el sector salud.
- Tener la capacidad de lectura y escritura en tarjetas de acceso.
- Diseño de una herramienta que tenga la capacidad de agilizar el proceso de identificación del paciente, en el área de la salud.
- Se desarrollará un sistema de inventario en el área de farmacia, para tener un mejor control de la cantidad de medicamento disponible en el hospital.
- Portar en el dispositivo acceso la información personal y clínica más relevante.



1.5 LIMITACIONES

- Los dispositivos de acceso (en el caso de RFID y banda magnética) tendrán una limitada capacidad de almacenamiento por lo cual, no se podrá almacenar todo el expediente clínico.
- El poco conocimiento del personal médico acerca de esta tecnología.



1.6 VALIDACIÓN

Por medio de la validación determinaremos las posibilidades reales de finalización del proyecto, el cual necesitará los siguientes aspectos para poder desarrollarse:

- Validación Técnica.
- Validación Operativa.
- Validación Económica.

1.6.1 VALIDACIÓN TÉCNICA.

A través de la validación técnica se verán las posibilidades que se tienen de adquirir la tecnología necesaria para desarrollar lo que se ha propuesto, y así determinar el lenguaje de programación más adecuado para la manipulación del proyecto, además de la facilidad con que los usuarios tendrán acceso a su información desde la Web.

En este sistema es esencial evaluar el hardware y el software que se utilizaran en el desarrollo del mismo desde el punto de vista de la calidad en el servicio.

Esto quiere decir que se buscará un hardware y un lenguaje de programación que cumpla en su mayoría las siguientes características:

- Claridad y sencillez.
- Claridad de la estructura del programa.



- Sencillez de aplicación.
- Facilidad de ampliación.
- Facilidad de corrección y mantenimiento.
- Eficacia.
- Transportabilidad sobre cualquier equipo informático.

HARDWARE

El hardware son todos aquellos dispositivos que nos permitirán la transmisión de datos entre los diferentes elementos que interactúan en el sistema. Así como, la administración de la información de los usuarios.

Los dispositivos que se utilizarán para desarrollar este sistema son:

- Computadoras.
- Lector / Escritor de dispositivos de acceso.
- Tarjetas de acceso.

LENGUAJES DE PROGRAMACIÓN (SOFTWARE)

Los lenguajes de programación permitirán desarrollar aplicaciones a la medida del hardware seleccionado, para satisfacer las necesidades que el usuario presente. Dentro de los lenguajes de programación en los cuales se puede



desarrollar el sistema para poder manipular la información de los usuarios se encuentran:

- Visual Basic
- Visual Basic .NET
- Visual C++
- PHP
- ASP
- SQL Server
- MySQL

1.6.2 VALIDACIÓN OPERATIVA

Este tipo de validación nos permitirá demostrar que con el uso de nuestro sistema el tiempo de identificación de un paciente se verá reducido, en comparación con el tiempo utilizado en el proceso actual.

1.6.3 VALIDACIÓN ECONÓMICA

La rentabilidad de este sistema se basa en la gran cantidad de pacientes que asisten diariamente a los centros de salud, razón por la cual surge la necesidad de implementar dicho sistema, pues este vendrá a reducir en gran medida los costos de operación (Papel, impresiones, tinta, fotocopias, etc.) en los que se incurren actualmente para satisfacer la demanda de salud en nuestro país.



1.7 PLAN DE TRABAJO

1.7.1 DISPONIBILIDAD DE TIEMPO

El tiempo que cada integrante del grupo, aportará semanalmente será de 12 horas, esto se da porque hay un integrante del grupo trabajando, y esto impide poder dedicarle un mayor tiempo al proyecto, y los demás integrantes es cierto que no poseen un trabajo pero se espera que puedan integrarse a uno, en un corto plazo, por esto y por otras actividades que cada uno posee, es como se llegó a la conclusión en el tiempo que cada uno puede aportar.

1.7.2 RECURSOS

Hardware

- Computadoras.
- Lector / Escritor de dispositivos de acceso.

Software

- Apache
- Power Builder
- PHP
- MySQL
- Html



Materiales

- Tarjetas de acceso.
- Cable UTP categoría 5E.
- Conectores RJ 45.
- Switch 10/100 Mbps de 8 puertos.

Fuentes de Consulta:

- Uso del Internet como fuente de información y obtención de material bibliográfico.
- Realización de entrevistas a profesionales de la red hospitalaria para verificar todo el proceso de consulta.

1.7.3 ORGANIZACIÓN DE LAS ACTIVIDADES

Las actividades que hemos clasificado como de mayor relevancia, para el desarrollo del proyecto son las siguientes:

- Investigación sobre la red hospitalaria a nivel nacional.
- Investigación sobre las tecnologías de acceso utilizadas en el área de la salud, a nivel mundial.
- Análisis de la información recaudada.
- Investigación sobre tecnología de acceso.
- Selección y Adquisición de equipos de acceso.



- Desarrollo de aplicación para la transmisión de datos sobre tarjetas de acceso.
- Diseño de Base de Datos para expediente clínico.
- Diseño de Entorno WEB.
- Elaboración y configuración de una red de datos.

1.8 CRONOGRAMA

ACTIVIDAD	MESES/SEMANAS																																			
	Mes 1				Mes 2				Mes 3				Mes 4				Mes 5				Mes 6				Mes 7				Mes 8							
	Seman a				Seman a				Seman a				Seman a				Seman a				Seman a				Seman a											
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Investigación y recopilación de la información de los diferentes centros de salud a nivel nacional y a nivel mundial.	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■																				
Análisis de la información recaudada.	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■																				
Comparativa entre las diferentes tecnologías de acceso.													■	■	■	■																				
Primera Evaluación. (Lectura y Escritura en tarjetas de acceso).													■	■	■	■																				
Correcciones.																	■	■	■	■																
Selección y Adquisición de equipos.																					■	■	■	■												
Implementación y montaje de equipo.																									■	■	■	■								
Realización de pruebas.																													■	■	■	■				
Desarrollo de aplicación para transmisión de datos.																																	■	■	■	■
Elaboración y configuración de una LAN.																																				
Diseño de Base de Datos.																																				
Diseño de Entorno WEB.																																				
Realización de pruebas.																																				
Integración del sistema.																																				
Realización de pruebas.																																				



CAPITULO II

TECNOLOGIAS DE ACCESO



2 CAPITULO II

A diario identificamos de forma natural un número muy elevado de objetos, para esto utilizamos una gran variedad de sentidos como el oído, la vista, el tacto, el olfato entre otros, sin embargo el desarrollo tecnológico ha permitido que la tarea de identificación sea efectuada de forma automática por medio de dispositivos que hacen que esta tarea sea mucho más precisa y especializada. Es así como surgen las diferentes tecnologías de acceso con las que se cuentan actualmente.

Por ello, es que en este capítulo se definirán las tecnologías de acceso más comunes, sus estándares, características y aplicaciones; base teórica que será utilizada a lo largo del documento.

2.1 TECNOLOGÍAS DE ACCESO

Las tecnologías de acceso son un conjunto de dispositivos que ayudan a la identificación de objetos físicos por ejemplo, personas, los productos que pasan por las diferentes etapas de una cadena de suministros, así como para proporcionar información acerca de los mismos a través de la captura de datos automática. Estos datos deben digitalizarse para poderlos introducir en los sistemas informáticos. El objetivo de la mayoría de los sistemas de identificación automática consiste en incrementar la eficiencia, reducir los errores de entrada de datos y liberar al personal para que puedan realizar funciones de mayor relevancia.

Son muchas las tecnologías que se encuentran implícitas en los sistemas de identificación automática. Para efectos de estudio se analizarán cinco tecnologías de acceso:



- Códigos de barras.
- Tarjetas inteligentes (smart card).
- Determinadas tecnologías biométricas (las exploraciones de retina, reconocimiento de huellas digitales, por ejemplo).
- Banda Magnética.
- Identificación por radiofrecuencia (RFID).

2.1.1 CÓDIGO DE BARRAS

El código de barras es la representación de una determinada información mediante un conjunto de líneas paralelas verticales de diferente grosor y espaciado. El código de barras sirve para reconocer rápidamente un artículo en un comercio o en un punto de la cadena logística¹. Actualmente, el código de barras está masivamente implantado de forma internacional hasta el punto de que prácticamente la totalidad de los productos de consumo lo incorporan.

Sus orígenes se dan a partir de 1949 cuando patentaron lo que se denominó "método de clasificación de productos mediante reconocimiento de formas". Pero fue hasta 1952 que se le bautizó con el nombre de código de barras, el cual fue registrado por Joseph Woodland y Bernard Silver [1].

En 1966 se utilizó comercialmente por primera vez, pero pronto se dieron cuenta de que era necesario definir un estándar para el sistema. En 1970 se creó el primer estándar como código universal de identificación de los productos de las

¹ Conjunto de procesos para la comercialización de un producto, desde su origen hasta su entrega al usuario final.



tiendas de alimentación. Luego el uso se extendió y se creó un estándar universal de identificación de productos [1].

La correspondencia o mapeo entre los mensajes que representan y el código de barras se denomina simbología. Las simbologías pueden ser clasificadas en dos grupos, atendiendo a dos criterios diferentes:

- **Continuo frente a discreto:** los caracteres en las simbologías continuas, comienzan con un espacio y el siguiente comienza con una barra, o viceversa. Los caracteres en las simbologías discretas comienzan y terminan con barras; el espacio entre caracteres es ignorado, en cuanto no es lo suficientemente ancho para parecerse al final del código.
- **Bidimensional frente a multidimensional:** las barras en las simbologías bidimensionales son anchas o estrechas; el grosor de estas no importa y pueden variar de un carácter al siguiente. El ancho de las barras en las simbologías multidimensionales son múltiplos de una determinada medida llamada X; generalmente, se emplean barras con un ancho X, 2X, 3X, y 4X.

2.1.1.1 Características

Un símbolo de código de barras puede tener a su vez, varias características, entre las cuales podemos nombrar:

- *Densidad:* Es la anchura del elemento (barra o espacio) más angosto dentro del símbolo de código de barras. Está dado en mils (milésimas de pulgada). Un código de barras no se mide por su longitud física sino por su densidad.

- *WNR*: (Wide to Narrow Ratio) Es la razón del grosor del elemento más angosto contra el más ancho. Usualmente es 1:3 o 1:2.
- *Quiet Zone*: Es el área blanca al principio y al final de un símbolo de código de barras. Esta área es necesaria para una lectura conveniente del símbolo. Lo cual se puede observar en la figura 2.1:



Fig. 2.1 Estructura General. [2]

2.1.1.2 Tipos de código de barras

Los códigos de barras se dividen en dos grandes grupos:

- Códigos de barras lineales.
- Códigos de barras de dos dimensiones.

CÓDIGOS DE BARRAS LINEALES (1-D)

Se utiliza para almacenar códigos, ya sean números o letras, usando una combinación lineal de barras y espacios en diferentes medidas. Algunos de los códigos de barra lineales mas usados en el mercado, y que comúnmente se



utilizan para la identificación de productos se muestran en la siguiente tabla (Tabla 1):

CÓDIGO	DESCRIPCIÓN
CODIGO 39	Este código permite la codificación de caracteres numéricos, letras mayúsculas y algunos símbolos como -, ., \$, /, +, % y "espacio". Se utilizan sólo dos grosores tanto para barras como para espacios.
CODABAR	Es una simbología de longitud variable que codifica solo números. Utiliza dos tipos de grosores para barras y espacios y su densidad es similar a la del Código 39.
INTERCALADO 2 de 5	Se basa en la técnica de intercalar caracteres permitiendo un código numérico que utiliza dos grosores. El primer carácter se representa en barras, y el segundo por los espacios que se intercalan en las barras del primero.
CODIGO 128	El Código 128 utiliza 4 diferentes grosores para las barras y los espacios, este código tiene una densidad muy alta, ocupando en promedio sólo el 60% del espacio requerido para codificar información.
UPC (Código de Universal de Producto)	UPC es la simbología más utilizada en el comercio minorista de EEUU, pudiendo codificar solo números.
EAN (Sistema de Numeración Europeo)	El EAN-13 codifica 13 dígitos. Utiliza un dígito que se combina con uno o dos de los otros dígitos para representar un código de país, indicando el origen de la mercancía.

Tabla 1. Descripción de Códigos de barras lineales.

CÓDIGO DE BARRA DE DOS DIMENSIONES (2-D)

Este tipo de código de barras ha empezado a usarse en documentos para controlar su envío o en seguros médicos y en general, en documentos que requieren la inserción de mensajes más grandes (de hasta 2,725 dígitos) como un expediente clínico completo. En la siguiente tabla se muestran algunos ejemplos de este tipo de código de barra:

CÓDIGO	DESCRIPCIÓN
CÓDIGO PDF417	Es un código multifilas, continuo, de longitud variable, que tiene alta capacidad de almacenamiento de datos. El código consiste en un patrón de marcas (17,4).
CODIGO DATA MATRIZ	Esta hecho por módulos cuadrados organizados dentro de un modelo descubridor de perímetro. Cada símbolo tiene regiones de datos, que contienen un juego de módulos cuadrados nominales en un arreglo regular.
CÓDIGO QR (Quick Response Barcode)	Es un código bidimensional con una matriz de propósito general diseñada para un escaneo rápido de información. QR es eficiente para codificar caracteres Kanji, es una simbología muy popular en Japón. El código QR es de forma cuadrada y puede ser fácilmente identificado por su patrón de cuados oscuros y claros en tres de las esquinas del símbolo.

Tabla 2. Códigos de barra bidimensionales.

2.1.1.3 Arquitectura

El sistema de código de barras es un sistema simple, que se basa en tres componentes (Fig. 2.2):

- *Lectores de código de barras.*
- *Computadora.*
- *Etiquetas.*



Fig. 2.2 Arquitectura del código de barras.

LECTORES DE CÓDIGO DE BARRAS

Es un dispositivo o escáner que toma la luz reflejada sobre el código de barras y la convierte en una señal eléctrica. Los códigos de barras se leen pasando un pequeño punto de luz sobre el símbolo del código de barras impreso. Únicamente se observa una fina línea roja emitida desde el escáner láser, lo que provoca que las barras oscuras absorban la fuente de luz del escáner y la misma se refleje en los espacios vacíos. El láser del escáner (fuente de luz) comienza a



leer el código de barras en un espacio blanco (la zona fija) antes de la primera barra y continúa pasando hasta la última barra, para finalizar en el espacio blanco que sigue a ésta. Debido a que el código no se puede leer si se pasa el escáner fuera de la zona del símbolo, las alturas de las barras se eligen de manera tal de permitir que la zona de lectura se mantenga dentro del área del código de barras. Mientras más larga sea la información a codificar, más largo será el código de barras necesario. A medida que la longitud se incrementa, también lo hace la altura de las barras y los espacios a leer.

Interfaces de los lectores de código de barras: Todas las aplicaciones pueden aceptar la salida que produce un lector de código de barras, siempre y cuando se posea el equipo necesario. Los lectores de códigos de barras se encuentran con distintos interfaces de conexión a la computadora. Existen modelos de lectores que tienen solamente una interfaz integrada, pero hay algunos de ellos que aceptan varias interfaces. Basta con un simple cambio de cables y una reconfiguración para utilizar una interfaz u otra.

- **Interfaz de teclado:** Cuando se requiere que el decodificador sea de teclado se utiliza lo que se conoce como keyboard wedge², el cual se conecta a la entrada de teclado de la computadora. Este tipo de lectores se conectan directamente al puerto del teclado y ofrecen una salida idéntica a la de éste. Cuando se lee un código de barras el lector envía a la computadora los datos como si hubiesen sido escritos con el teclado (el número que corresponde al código de barras leído), lo que hace que su utilización sea muy sencilla con cualquier programa que espere una entrada de teclado. Sin embargo, este tipo de interfaz tiene algunos

² Keyboard wedge: Dispositivo que crea una bifurcación o “Y” entre el teclado y la computadora para que los datos sean interpretados como entradas de teclado.



inconvenientes. Por ejemplo, la escritura del código será siempre completa, es decir, no se puede dividir el código en varias partes. El lector no es capaz de devolver cuatro cifras, y luego el resto. Obviamente, siempre hay que asegurarse que el cursor del sistema este sobre la casilla que se quiere rellenar, el lector no se preocupa de eso y devolverá su salida donde se esta situado.

- **Interfaz RS-232:** Los escáneres que se conectan a la interfaz RS-232 (o interfaz serie) necesitan utilizar un software especial que recupera la información enviada por el escáner de códigos de barras y la coloca donde se le indique. Esta interfaz es algo más sofisticada que la de teclado, y nos ofrece un mejor control sobre el destino de la lectura del código.
- **Interfaz USB:** Son lectores de última generación. Envían la información más rápidamente que los anteriores y su conexión es más simple. No necesitan alimentación añadida, pues la que obtienen por esta interfaz es suficiente.

Tipos de lectores: Existen cuatro tipos principales de lectores:

- Lápiz óptico.
- Láser de pistola.
- CCD (Charge Coupled Device).
- Láser omnidireccional.

Tanto los lectores láser, como los CCD y los omnidireccionales se configuran leyendo comandos de programación impresos en menús de códigos de barras.



Hay algunos que se configuran con interruptores dip, o enviándoles los comandos de programación vía línea serie.

COMPUTADORA

Es el dispositivo donde se procesa toda la información capturada por el lector de cada uno de los productos con los que se comercializan en un establecimiento determinado.

ETIQUETAS

Una etiqueta es un elemento que se adhiere a un objeto, y es donde se estampa el código para luego identificar dicho producto.

Existen diferentes etiquetas para múltiples aplicaciones, cada una de ellas con diferentes características. Las principales características que definen una etiqueta de código de barras son las siguientes:

- Numéricas o alfanuméricas.
- De longitud fija o de longitud variable.
- Discretas o continuas.
- Número de anchos de elementos.
- Auto verificación.



2.1.1.4 Ventajas

El código de barras nace de la necesidad de agilizar la lectura de los artículos en las cajas de cobros de los diferentes centros de comercio y evitar errores de digitación. Otras ventajas que se pueden destacar de este sistema son:

- Agilidad en etiquetar precios pues no es necesario hacerlo sobre cada artículo.
- Rápido control de los productos en existencia.
- El consumidor obtiene una relación de artículos en el ticket de compra lo que permite su comprobación y eventual reclamación.
- Reducen costos de operación gracias a la capacidad de los sistemas informáticos para desarrollar estas tareas en forma rápida y sin errores.
- Se elabora a bajos costos.
- Permite porcentajes muy bajos de error.
- Los equipos de lectura e impresión de código de barras son flexibles y fáciles de conectar e instalar.
- Se mejora la competitividad.
- Se mejora la exactitud de los datos.

2.1.1.5 Desventajas

- Necesita visibilidad para funcionar. Es decir, el código de barras debe ser visible ante el lector para que el producto pueda ser identificado.



- El código de barras tradicionalmente identifica un tipo de producto, no una unidad de dicho producto.
- Un código de barras se daña o se rompe fácilmente, porque normalmente se adhiere a la superficie del producto y si este se rompe no podrá ser leído por el lector.

2.1.1.6 Aplicaciones

Las aplicaciones de los códigos de barras cubren prácticamente cualquier tipo de actividad humana, tanto en industria, comercio, instituciones educativas, instituciones médicas, gobierno, etc. Entre algunas aplicaciones tenemos:

- **Control de Inventarios:** Es altamente recomendado controlar los inventarios con esta tecnología ya que se puede conocer en tiempo real y con exactitud la existencia de un producto para facilitar la toma de decisiones.
- **Punto de Venta:** Si se requiere controlar las transacciones de un comercio se puede utilizar el código de barras conjuntamente con una caja registradora para llevar un control de las ventas, efectivo, compras a crédito, todo esto asociado a un control de inventarios.
- **Credencialización:** Permite llevar eficientemente el control de asistencia de empleados, acceso a centros deportivos. Dependiendo de las necesidades facilita por ejemplo el pago de nomina, vacaciones, registro de acceso a un evento.



- **Etiquetado:** Llevar el control de producción con la generación de etiquetas de código de barras que permiten llevar el control de la producción, almacenaje, rastreabilidad.
- **Recepción de Materiales:** Registrar la entrada de productos conociendo en tiempo real los datos generales del transportista, orden de embarque, además de saber la ubicación física de almacenaje del producto.
- **Rastreabilidad:** Es de suma importancia saber el estatus de un producto, su localización y en sí, el trayecto o movimiento que ha tenido un determinado producto.
- **Control de Recorridos:** Es la mejor forma de saber si la vigilancia cumple cabalmente los recorridos de seguridad dentro de las instalaciones de una empresa.
- **Control de Activos Fijos:** Esta tecnología facilita el control de activos fijos de un empresa para saber con plenitud en donde esta un activo y facilitar el manejo de las cuentas contables.

2.1.2 TARJETAS INTELIGENTES (SMART CARD).

Una tarjeta inteligente (smart card), conocida también como tarjeta de circuitos integrados (TCI), se define como una tarjeta de bolsillo con circuitos integrados incrustados dentro de la tarjeta. Aunque existe un diverso rango de aplicaciones, hay dos categorías principales de TCI: las tarjetas de memoria que contienen sólo componentes de memoria no volátil y posiblemente alguna lógica de seguridad y las tarjetas microprocesadoras que contienen memoria y microprocesadores (Fig. 2.3).



Fig. 2.3 Tarjeta inteligente (smart card)

Las tarjetas inteligentes fueron inventadas y patentadas en los setenta. Existen algunas discusiones de quién es el "inventor" original; entre los que se encuentran Juergen Dethloff de Alemania, Arimura de Japón y Moreno de Francia. El primer uso masivo de las tarjetas fue para el pago telefónico público en Francia en 1983 [3].

El mayor auge de las tarjetas inteligentes fue en los noventa, con la introducción de las tarjetas SIM (Modulo de Identidad del Subscriptor) utilizadas en la telefonía móvil GSM (Sistema Global para las Comunicaciones Móviles) en Europa [3].



Las firmas internacionales MasterCard, Visa, y Europay publicaron un estándar de interoperabilidad para el pago con tarjetas inteligentes. Este estándar, llamado EMV (Europay MasterCard VISA) se ha introducido mundialmente de manera gradual, con la esperanza de reemplazar las tarjetas basadas en cintas magnéticas.

2.1.2.1 Características

Las principales características con las que cuenta esta tecnología son las siguientes:

- **Inteligencia:** Es capaz de almacenar cualquier tipo de información, además es autónoma en la toma de decisiones al momento de realizar transacciones.
- **Utiliza clave de acceso o PIN:** Para poder utilizarse es necesario digitar un número de identificación personal.
- **Actualización de cupos:** Después de agotado el cupo total de la tarjeta inteligente es posible volver a cargar un nuevo cupo.

2.1.2.2 Tipos de tarjetas inteligentes

Las tarjetas inteligentes se dividen en cuatro grandes grupos:

- Según la interfaz.
- Según el formato (tamaño).
- Según la estructura de su sistema operativo.



- Según la capacidad de almacenamiento.

TARJETAS SEGÚN LA INTERFAZ

En la siguiente tabla se mostraran los diferentes tipos de tarjetas inteligentes según la interfaz y sus principales características:

TIPO	DESCRIPCIÓN
Tarjeta inteligente de contacto	Las tarjetas de contacto se insertan dentro de un lector de tarjetas inteligentes, haciendo contacto físico con el lector. Las tarjetas de este tipo no contienen baterías, así que la energía es suministrada por el dispositivo lector.
Tarjetas Inteligentes sin Contacto	Las tarjetas sin contacto no necesitan ser insertadas en un lector, sino que deben estar cerca de una antena para llevar a cabo las transacciones ya que hacen uso de la tecnología RFID.
Tarjetas combicards o tarjetas híbridas	Una tarjeta híbrida comienza con una tarjeta sin contacto a la cuál se le agrega un segundo chip de contacto. El chip sin contacto es generalmente usado en aplicaciones que requieren transacciones rápidas, mientras que el chip de contacto es generalmente utilizado en aplicaciones que requieren de alta seguridad.

Tabla 3. Tarjetas inteligentes según la interfaz

TARJETAS SEGÚN EL FORMATO (TAMAÑO)

En el estándar ISO/IEC 7816 se definen los formatos para tarjetas inteligentes los cuales se muestran en la siguiente tabla:

CÓDIGO	DESCRIPCIÓN
ID 000	Se usa en tarjetas SIM para teléfonos móviles GSM.
ID 00	Es de tamaño intermedio poco utilizado comercialmente.
ID 1	Es el tamaño más habitual utilizado en tarjetas de crédito.

Tabla 4. Tarjetas inteligentes según el formato

TARJETAS SEGÚN LA ESTRUCTURA DE SU SISTEMA OPERATIVO

En la siguiente tabla se mostrarán los diferentes tipos de tarjetas inteligentes según la estructura de su sistema operativo y sus principales características:

TIPO	DESCRIPCIÓN
Basadas en sistemas de archivos, aplicaciones y comandos	Estas tarjetas disponen del equivalente a un sistema de archivos compatible con el estándar ISO/IEC 7816 y un sistema operativo en el que se incrustan una o más aplicaciones que exponen una serie de comandos que se pueden invocar a través de APIs ³
Java Cards	Una Java Card es una tarjeta capaz de ejecutar mini-aplicaciones Java. En este tipo de tarjetas el sistema operativo es una pequeña máquina virtual Java (JVM) y en ellas se pueden cargar dinámicamente aplicaciones desarrolladas específicamente para este entorno.

Tabla 5. Tarjetas inteligentes según la estructura de su sistema operativo

³ API (Application Program Interface): Conjunto de convenciones de programación que definen cómo se invoca un servicio desde un programa.

TARJETAS SEGÚN LA CAPACIDAD DE ALMACENAMIENTO

Según las capacidades de almacenamiento de su chip interno, así como su seguridad las tarjetas más habituales son:

TIPO	DESCRIPCIÓN
Memoria	Tarjetas que únicamente son un contenedor de archivos pero que no albergan aplicaciones ejecutables. Éstas se usan generalmente en aplicaciones de identificación y control de acceso sin altos requisitos de seguridad.
Microprocesadas	Tarjetas con una estructura análoga a la de una computadora. Éstas albergan archivos y aplicaciones y suelen usarse para identificación y pago con monederos electrónicos.
Criptográficas	Tarjetas microprocesadas avanzadas en las que hay módulos hardware para la ejecución de algoritmos usados en cifrados y firmas digitales. En estas tarjetas se puede almacenar de forma segura un certificado digital y firmar documentos o autenticarse con la tarjeta sin que el certificado salga de la misma.

Tabla 6. Tarjetas inteligentes según sus capacidades

2.1.2.3 Arquitectura

El sistema de tarjetas inteligentes es un sistema que se basa en tres componentes (Fig. 2.4):

- Lectores/escriores de tarjetas inteligentes.
- Computadora.

- Tarjetas inteligentes.



Fig. 2.4 Arquitectura de Tarjetas Inteligentes

LECTORES/ESCRITORES DE TARJETAS INTELIGENTES

Como el propio nombre lo indica, un lector de tarjetas es una interfaz que permite la comunicación entre una tarjeta y otro dispositivo. Los terminales se diferencian unos de otros en la conexión con la computadora, la comunicación con la tarjeta y el software que poseen.

Existen lectores/escritores de tarjetas inteligentes para cada aplicación y los podemos dividir en:

- **Lectores/escritores conectados a una PC:** Como su nombre lo indica son lectores/escritores fabricados para ser usados conectándolo a una computadora, esta conexión puede ser a través de un puerto serie, USB, PCMCIA, etc.
- **Lectores/escritores conectados a un equipo específico:** Son lectores /escritores que se pueden instalar (previo fabricación y diseño) en un



aparato determinado para cumplir con una función. Estos lectores se pueden instalar en:

- Cajeros automáticos.
 - Parquímetros.
 - Puertas (control de acceso).
- **Lectores Portátiles:** Son equipos que no necesitan de otro aparato para cumplir su función. Generalmente poseen todos los recursos integrados como baterías, memoria, pin pad, etc.

COMPUTADORA

Dispositivo donde se procesa toda la información capturada por el lector.

TARJETAS INTELIGENTES

Es una tarjeta de plástico convencional combinada con un circuito integrado, donde se almacena la información que se desea transportar.

2.1.2.4 Ventajas

- Altos niveles de seguridad.
- Facilidad de uso:
 - Comodidad para usuario.



- Liquidez, compras por red.
- Estándares específicos.
- Evolución constante:
 - Capacidad de memoria.
 - Capacidad de procesamiento.
- Proporciona un soporte activo en el cual es posible actualizar la información sin necesidad de emitir una nueva tarjeta.

2.1.2.5 Desventajas

- Sensible a fluidos.
- Dispositivos de lectura/escritura sin estandarización.
- Costo de producción elevado.

2.1.2.6 Aplicaciones

- **Programas de Fidelización:** Cada vez más programas de fidelización en sectores como líneas aéreas, hoteles, restaurantes de comida rápida y grandes almacenes, utilizan las tarjetas inteligentes, que registran los puntos y premios que ofrecen datos detallados sobre los hábitos y experiencias de los clientes a los operadores de dichos programas, a fin de elaborar sus campañas de marketing con mayor precisión.



- **Monederos Electrónicos:** Estas tarjetas electrónicas de pago permiten evitar los problemas de encontrar el cambio exacto, ya que cargan el efectivo en un monedero electrónico que puede usarse para las compras de todos los días en tiendas, kioscos, billetes de transporte, parquímetros, teléfonos, etc.

- **Aplicaciones en Grupos Cerrados:** Las tarjetas inteligentes están muy extendidas entre los grupos cerrados de usuarios (residentes de una ciudad, personal de una universidad, personal de una compañía, aficionados de un equipo deportivo, clientes de un parque de atracciones, etc.), los cuales pueden utilizar tarjetas con múltiples aplicaciones para pagar sus cuotas o acceder a servicios (por ejemplo, descuento en compras, entrada para partidos, máquinas expendedoras, credenciales de biblioteca, parques de atracciones, estacionamientos, etc.), de forma ilimitada de acuerdo con la autorización y circunstancias personales

- **Tarjetas de asistencia Médica (Clínicas y Seguros):** Las tarjetas inteligentes o smart cards, pueden almacenar expedientes médicos, información para casos de emergencia y situación en materia de seguros de enfermedad.

- **Documentos y Credenciales (Seguridad y Control de Acceso):** Los gobiernos que deseen reducir costos y papeleo pueden utilizar las tarjetas inteligentes para emitir carnés de conducir, pasaportes y visas. Las tarjetas inteligentes pueden programarse para permitir el acceso a edificios o datos, dependiendo del cargo y del nivel de autorización.

- **Débito-Crédito:** En muchos países, las versiones para cinta magnética de estas aplicaciones residen conjuntamente con las tarjetas inteligentes,



pero las nuevas normas EMV (EuroPay, MasterCard y Visa) implican que los adquirentes deben actualizar sus redes, tarjetas y terminales.

2.1.3 TECNOLOGIA DE ACCESO BIOMETRICO

El concepto biometría proviene de las palabras bio (vida) y metría (medida), es una tecnología de seguridad, que consiste en la verificación automática de la identidad, basada en el reconocimiento de características biológicas, de comportamiento o conductuales de la persona, como por ejemplo, la huella dactilar, el iris, la mano, el sonido de la voz, la forma del rostro, entre otras.

La biométrica no se puso en práctica en las culturas occidentales hasta finales del siglo XIX, pero era utilizada en China desde el siglo XIV.

En 1883 se desarrollo el primer sistema biométrico, el sistema Bertillon, que fue adoptado extensamente en occidente hasta que aparecieron defectos en el sistema [4].

2.1.3.1 Características

CARACTERISTICAS DE UN INDICADOR BIOMETRICO

Un indicador biométrico es alguna característica física con la cual se puede realizar biometría. Cualquiera que sea el indicador, debe cumplir los siguientes requerimientos [5]:



- **Universalidad:** característica física que todas las personas deben de tener en común.
- **Unicidad:** existe una baja probabilidad de que existan dos personas con una característica idéntica.
- **Permanencia:** la característica no cambia en el tiempo.
- **Cuantificación:** la característica puede ser medida en forma cuantitativa.

Los requerimientos anteriores sirven como criterio para descartar o aprobar alguna característica como *indicador biométrico*. Luego de seleccionar algún indicador que satisfaga los requerimientos antes señalados, es necesario imponer restricciones prácticas sobre el sistema que tendrá como misión recibir y procesar estos indicadores.

CARACTERISTICAS DE UN SISTEMA BIOMETRICO PARA IDENTIFICACION PERSONAL

Las características básicas que un sistema biométrico debe cumplir, pueden expresarse mediante las restricciones que deben ser satisfechas. Ellas apuntan, básicamente, a la obtención de un sistema biométrico con utilidad práctica. Las restricciones antes señaladas apuntan a que el sistema considere:

- **El desempeño:** se refiere a la exactitud, la rapidez y la robustez alcanzada en la identificación, además de los recursos invertidos y el efecto de factores ambientales y/u operacionales
- **La aceptabilidad:** indica el grado en que la gente está dispuesta a aceptar un sistema biométrico en su vida diaria. Es claro que el sistema

no debe representar peligro alguno para los usuarios y debe inspirar "confianza" a los mismos.

- **La fiabilidad:** refleja cuán difícil es burlar al sistema. El sistema biométrico debe reconocer características de una persona viva, pues es posible crear dedos de látex, grabaciones digitales de voz, prótesis de ojos, etc. Algunos sistemas incorporan métodos para determinar si la característica bajo estudio corresponde o no a la de una persona viva. Los métodos empleados son ingeniosos y usualmente más simples de lo que uno podría imaginar.

2.1.3.2 Tipos de Biometría

En la siguiente figura (Fig. 2.5) se muestra los tipos de biometría según sus características:

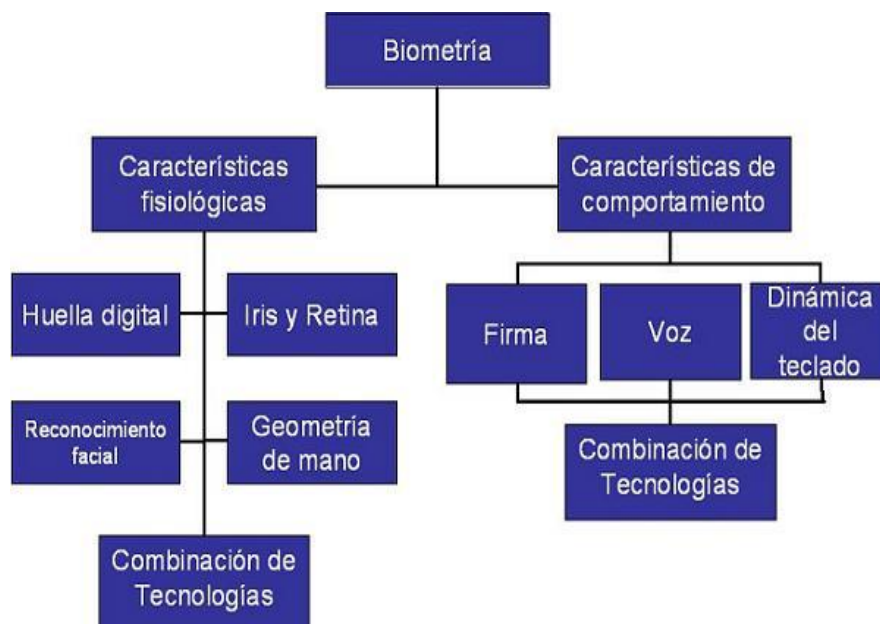


Fig. 2.5 Biometría según sus características [6]



Considerando las características anteriores la biometría se puede dividir en:

- Biometría estática (características fisiológicas).
- Biometría dinámica (características psicológicas).

BIOMETRIA ESTÁTICA

Es cuando se lleva a cabo una identificación en base a la anatomía de la persona, es decir, en base a las características físicas de los individuos, por ejemplo:

- Imagen facial.
- Imagen del iris.
- Escáner de retina.

BIOMETRIA DINÁMICA

Es cuando la identificación se lleva a cabo a partir de su comportamiento. En otras palabras se estudia el comportamiento del individuo, por medio de la voz, la forma de caminar y el análisis gestual, entre otras.

2.1.3.3 Arquitectura

Los dispositivos biométricos poseen dos componentes básicos, esto se muestra en la siguiente figura (Fig. 2.6):

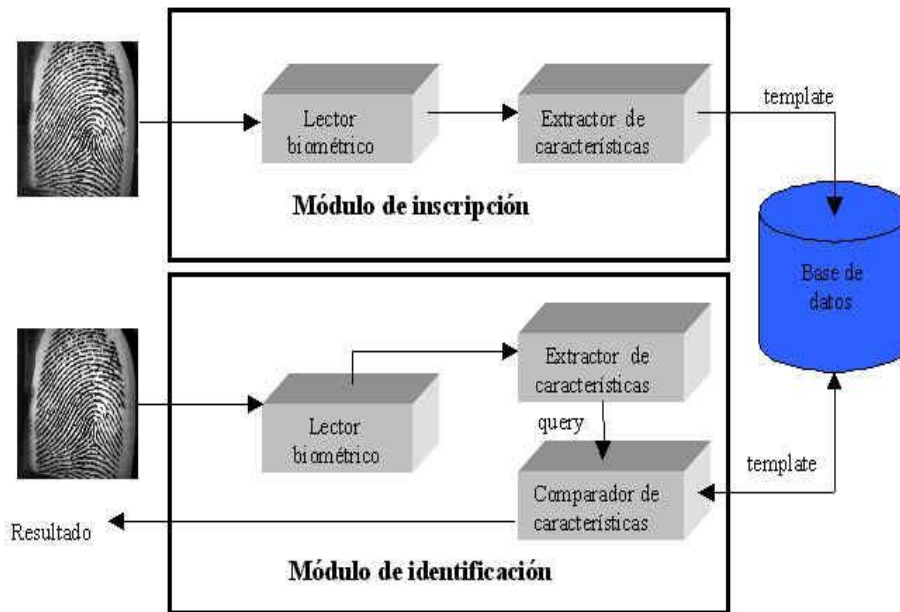


Fig. 2.6 Arquitectura de un sistema biométrico para identificación personal, ejemplificado con huellas dactilares [7]

El primero se encarga de la adquisición análoga o digital de algún indicador biométrico de una persona, como por ejemplo, la adquisición de la imagen de una huella dactilar mediante un escáner. El segundo maneja la compresión, procesamiento, almacenamiento y comparación de los datos adquiridos (en el ejemplo una imagen) con los datos almacenados.

SISTEMAS BIOMETRICOS ACTUALES

En la actualidad existen sistemas biométricos que basan su acción en el reconocimiento de diversas características. Las técnicas biométricas más conocidas están basadas en los siguientes indicadores:

1. Rostro.
2. Huellas dactilares.
3. Geometría de la mano.
4. Iris.
5. Patrones de la retina.
6. Voz.
7. Firma.

IDENTIFICACIÓN DE HUELLA DACTILAR

Este método se basa en el reconocimiento de la huella digital que es única por persona, identificando sus características principales como el arco, lazo y la espiral. La imagen captada posteriormente es procesada. En la siguiente figura (Fig. 2.7) se muestra un equipo biométrico que efectúa este tipo de reconocimiento.



Fig. 2.7 Lector óptico de huella digital [8]

IDENTIFICACION DE LA GEOMETRIA DE LA MANO

Este método de identificación toma como referencia las dimensiones, forma de la mano y dedos, sus posiciones relativas. En la siguiente figura (Fig. 2.8) se muestra un equipo biométrico que efectúa este tipo de reconocimiento.



Fig. 2.8 Lector de geometría de mano [8]

IDENTIFICACION DE LA RETINA

Este método reconoce los sistemas vasculares de la retina que son únicos en cada persona. En la siguiente figura (Fig. 2.9) se muestra un equipo biométrico que efectúa este tipo de reconocimiento.



Fig. 2.9 Lector de retina [8]

IDENTIFICADOR DEL IRIS

Este método toma como muestra las características del iris, que son únicas por persona, esta técnica es posterior al reconocimiento de retina. En la Fig. 2.10 se muestra un equipo biométrico que efectúa este tipo de reconocimiento.



Fig. 2.10 Lector de iris [8]

IDENTIFICACION DEL ROSTRO

Es el método más común entre personas, y toma como referencia la forma geométrica así como la posición de las características de la cara, (persona parada o en movimiento). En la siguiente figura (Fig. 2.11) se muestra un equipo biométrico que efectúa este tipo de reconocimiento.



Fig. 2.11 Identificador de rostro [8]

IDENTIFICACION DE VOZ

Este método reconoce patrones lingüísticos y del hablante, que reflejan la anatomía bucal y faríngea (tamaño y forma), así como su comportamiento (estilo y tono). En la siguiente figura (Fig. 2.12) se muestra un equipo biométrico que efectúa este tipo de reconocimiento.



Fig. 2.12 Identificador de voz [8]

AUTENTICACIÓN DE LA FIRMA

Su método de identificación se centraliza en dos componentes:

- Estática: escrito
- Dinámica: proceso de escritura
 - Ángulo de lápiz.
 - Presión en lápiz.
 - Presión sobre superficie.
 - Velocidad y aceleración.



2.1.3.4 Ventajas

HUELLAS DACTILARES	GEOMETRÍA DE LA MANO	RETINA	IRIS	IDENTIFICACIÓN DEL ROSTRO	IDENTIFICACIÓN DE VOZ	AUTENTICACIÓN DE LA FIRMA
<ul style="list-style-type: none"> • Tecnología madura. • Fácil de usar, no invasivo. • Precisión alta. • Estabilidad grande. • Posible dar alta a varios dedos. • Costo bajo. 	<ul style="list-style-type: none"> • Tecnología madura. • No invasivo. • Aceptación grande. • Sin asociaciones negativas. • Posible aprendizaje. • Patrón pequeño. 	<ul style="list-style-type: none"> • Precisión alta. • Resistente a impostores. • Estabilidad grande. • Protegida del exterior. • Independencia genética. • Proceso de alta rápido. 	<ul style="list-style-type: none"> • Precisión alta. • Resistente a impostores. • Estabilidad grande. • Rápido. • No afectada por gafas, lentes de contacto. 	<ul style="list-style-type: none"> • No invasivo. • Coste bajo. • Cámara convencional. • Posible funcionamiento encubierto. 	<ul style="list-style-type: none"> • Infraestructura existente. • Fácil de usar, no invasivo. • Sin asociación negativa. 	<ul style="list-style-type: none"> • Resistente a falsificación. • Muy aceptado. • No invasivo. • Sin grabación de firma.

Tabla 7. Ventajas de Técnicas Biométricas



2.1.3.5 Desventajas

HUELLAS DACTILARES	GEOMETRÍA DE LA MANO	RETINA	IRIS	IDENTIFICACIÓN DEL ROSTRO	IDENTIFICACIÓN DE VOZ	AUTENTICACIÓN DE LA FIRMA
<ul style="list-style-type: none"> • Imposible dar de alta a algunos. • Afectado por condiciones de piel. • Afectado por suciedad en sensor. • Asociado con aplicaciones forenses y delincuencia. 	<ul style="list-style-type: none"> • Precisión baja. • Coste alto. • Lectores relativamente grandes. • Difícil de usar para algunos. 	<ul style="list-style-type: none"> • Difícil de usar. • Invasivo. • Caro. • Afectado por gafas y cataratas. • Percibido como amenaza a la salud. 	<ul style="list-style-type: none"> • Invasivo (aunque imagen puede captarse a distancia). • Posible detector del estado de Caro. 	<ul style="list-style-type: none"> • Precisión baja. • Ataque de gemelos. • Potencial violación de privacidad. • Muy afectado por apariencia. • Posible funcionamiento encubierto. 	<ul style="list-style-type: none"> • Ataque con grabación. • Variabilidad de la voz. • Afectado por ruido ambiental. • Plantilla grande (5K a 10K). • Precisión baja. 	<ul style="list-style-type: none"> • Inconsistencias en firma. • Difícil de usar. • Plantilla grande (1K a 3K). • Firmas sencillas son problemáticas.

Tabla 8. Desventajas de Técnicas Biométricas



2.1.3.6 Aplicaciones de los sistemas biométricos

- Acceso físico:
 - Dispositivos: teléfono, PDA, computadora portátil, coche.
 - Zona de seguridad: habitación, edificio.
- Acceso a servicio:
 - Comercio electrónico.
 - Prestación de Seguridad Social.
- Seguimiento (no necesariamente encubierto):
 - Control de empleados.
 - Control de ciudadanos.

2.1.4 BANDA MAGNÉTICA

Una banda magnética (llamada a veces *magstripe* como abreviación de *magnetic stripe*) es toda aquella banda oscura presente en tarjetas de crédito, abonos de transporte público o carnés personales que está compuesta por partículas ferromagnéticas bañadas de resina (generalmente epoxi⁴) y que almacenan cierta cantidad de información mediante una codificación determinada que polariza dichas partículas. La banda magnética es grabada o

⁴ Epoxi: Una Resina Epoxi o poliepóxido es un polímero termoestable que se endurece cuando se mezcla con un agente catalizador o "endurecedor".

leída mediante contacto físico pasándola a través de una cabeza lectora/escritora, gracias al fenómeno de la inducción magnética [9].

Las primeras tarjetas con banda magnética fueron usadas desde principios de los sesentas en el transporte público, London Transit Authority instaló un sistema de tarjeta con banda magnética en el sistema de tren London Underground, en Londres.

2.1.4.1 Características

En la Fig. 2.13 se observan sus características principales.

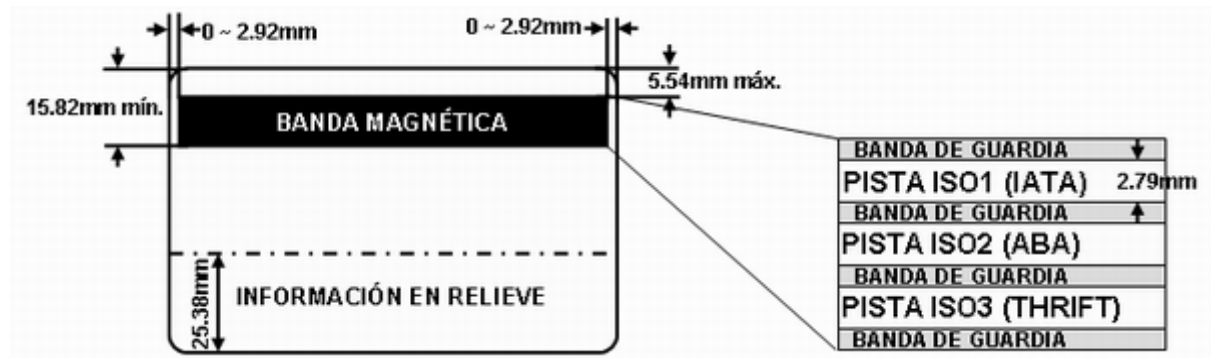


Fig. 2.13 Características Banda Magnética [11]

La norma ISO 7810/7811 define las características de la posición de la banda magnética en la tarjeta, la técnica de grabación y cifrado de caracteres:

- PISTA 1: 79 caracteres alfanuméricos.
- PISTA 2: 40 caracteres numéricos.

- PISTA 3: 107 caracteres numéricos.

Cada carácter que es codificado en la banda esta hecho de un número de bits, donde la polaridad de las partículas define cada bit. Los esquemas más comunes de codificación son F2F (Aiken BiPhase) y MFM (Modified Frequency Modulation).

Las medidas que presenta una tarjeta de banda magnética se muestran en la Fig. 2.14.

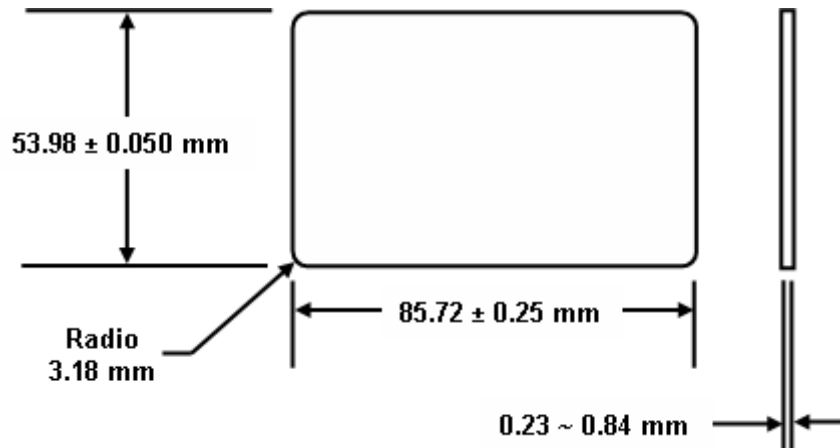


Fig. 2.14 Dimensiones convencionales de una tarjeta de banda magnética [10]

La norma internacional ISO/IEC define las características físicas de la tarjeta en 3 formatos básicos:

- CR-80: 86 x 54mm. (estándar tarjeta de crédito), con un grosor de 0,76mm.
- CR-90: 92 x 60mm. (CNI format).



- CR-100: 95 x 67mm.

2.1.4.2 Tipos de Banda Magnética

Existen 2 tipos de banda magnética conforme a la norma ISO 7811 [10]:

- Baja coercitividad.
- Alta coercitividad.

La diferencia es la cantidad de Oersteds⁵ de la que está compuesta, siendo para la de baja de 300 Oe y de 2700 y 4000 Oe para la de alta coercitividad. Sin embargo, la mayor cantidad de Oersteds la hace más resistente a una posible pérdida de información producida principalmente por la acción de campos magnéticos.

- **Banda magnética LO-CO (Baja coercitividad 300 Oersteds):** Indicada para tarjetas de fidelización como tarjetas comerciales, promocionales, bancarias, etc. Requiere una coercitividad de 300 Oersted. Muy apta para empresas provistas de lectores-grabadores. Es la más utilizada.
- **Banda magnética HI-CO (Alta coercitividad 2700/4000 Oersteds):** Banda magnética de alta densidad (desde 2700 hasta 4000 Oersted) y de gran resistencia a campos magnéticos. Especialmente utilizada para el control de acceso y presencia, parkings y uso continuo, empleada en sistemas que requieren un uso más frecuente como sistemas de control del personal. Una vez gravada es prácticamente imposible su decodificación.

⁵ Oersted: es la medida de que tan difícil es la codificación de la información en una cinta magnética.



En función de la aplicación donde se vaya a usar es recomendable una u otra coercitividad. La utilización de una u otra banda magnética condicionará al lector que deberá ser compatible con la misma. En la tabla 9 se hace referencia a los múltiples estándares de la norma ISO, que hacen referencia a esta tecnología.

ESTÁNDAR	DEFINICIÓN
ISO/IEC 4909	Tarjetas de Identificación. Tarjetas para transacción financiera.
ISO/IEC 7810	Tarjetas de Identificación. Características físicas
ISO/IEC 7811	Tarjetas de Identificación. Técnica de grabación.
ISO/IEC 7812	Tarjetas de Identificación. Identificación de emisores.
ISO/IEC 7813	Tarjetas de Identificación. Tecnología de Información. Tarjetas de transacción financiera.
ISO/IEC 8484	Bandas magnéticas en libros de ahorros
ISO/IEC 15457	Tarjetas de identificación. Tarjetas flexibles delgadas

Tabla 9. Estándares ISO.

2.1.4.3 Arquitectura



Fig. 2.15 Arquitectura de banda magnética [10]

El sistema de banda magnética es un sistema que se basa en dos componentes:

- Escritores / Lectores de banda magnética.
- Tarjetas de banda magnética.

LECTORES/ ESCRITORES DE BANDA MAGNETICA

La escritura, se efectúa por medio de un pequeño electroimán (la cabeza de escritura) que transforma una señal eléctrica emitida por el sistema electrónico de codificación en campo magnético variable Norte-Sur o Sur-Norte. El material



magnético se polariza según el campo y conserva así el rastro de la señal. La lectura se efectúa según el mismo principio.

TARJETAS DE BANDA MAGNETICA

Es una tarjeta de plástico que posee una banda magnética, donde se almacena la información codificada que se desea transportar.

2.1.4.4 Ventajas

- Fácil de utilizar.
- Portabilidad de la información.
- Completa gestión y control de accesos (puertas, garajes, acceso a empleados, etc.).
- Ofrece control de accesos con un costo de mantenimiento muy bajo.
- Permite la edición, copia y cancelación inmediata en caso de robo o pérdida.
- Tecnología probada.
- Medio barato.
- Puede ser combinado con impresiones.
- Soporta un gran número de usos.
- La información puede ser reescrita y actualizada.



2.1.4.5 Desventajas

- Requiere de equipamiento complejo.
- Frecuente mantenimiento.
- Susceptible a borrado de información accidental.
- Baja seguridad para almacenar datos confidenciales.
- Baja capacidad para almacenar datos.
- Baja resistencia a la fricción, radiaciones, campos magnéticos, etc.
- Incapacidad para realizar firmas digitales y cifrado de mensajes.
- El contenido de la banda magnética por la tecnología que implica, puede ser leído y aunque no es sencillo, puede ser manipulado por personas con conocimientos y medios adecuados.
- Susceptible al polvo, al agua y a temperaturas excesivas.

2.1.4.6 Aplicaciones de Banda Magnética

- Tarjetas comerciales.
- Tarjetas de control de accesos.
- Tarjetas de débito.
- Tarjetas de control horario.
- Tarjetas para parkings.



- Tarjetas "llave" para hoteles.
- Tarjetas de asistencia en viaje.
- Tarjetas identificación.
- Carnés de socio.
- Tarjetas V.I.P.

2.1.5 IDENTIFICACIÓN POR RADIO FRECUENCIA (RFID)

RFID (Radio Frequency Identification) es una nueva tecnología para la identificación de objetos a distancia sin necesidad de contacto. Se requiere lo que se conoce como etiqueta o tag RFID que consiste en un microchip que va adjunto a una antena de radio y sirve para identificar al elemento portador de la etiqueta.

El origen de la RFID está tristemente relacionado con la guerra, concretamente con la II Guerra Mundial, en la que el uso del radar permitía la detección de aviones a kilómetros de distancia, pero no su identificación. El ejército alemán descubrió que si los pilotos balanceaban sus aviones al volver a la base cambiaría la señal de radio reflejada de vuelta. Este método hacía así distinguir a los aviones alemanes de los aliados y se convirtió en el primer dispositivo de RFID pasivo. [12]

Los sistemas de radar y de comunicaciones por radiofrecuencia avanzaron en las décadas de los 50 y los 60 en que los científicos de los países más avanzados trabajaban para explicar cómo identificar objetos remotamente. Las compañías pronto comenzaron a trabajar con sistemas antirrobo que usando ondas de radio determinaban si un objeto había sido pagado o no a la salida de



las tiendas. Se utiliza con una etiqueta en la que un único bit decide si se ha pagado o no por el objeto en cuestión. La etiqueta pitará en los sensores colocados a la salida si el objeto no se ha pagado.

En 1973, Charles Walton recibió la patente para un sistema RFID pasivo que abría las puertas sin necesidad de llaves. Una tarjeta con un chip comunicaba una señal al lector de la puerta que cuando validaba la tarjeta desbloqueaba la cerradura.

El gobierno americano también trabajaba sobre esta tecnología en los años 70 y montó sistemas parecidos para el manejo de puertas en las centrales nucleares, cuyas puertas se abrían al paso de los camiones que portaban materiales para las mismas que iban equipados con un chip. También se desarrolló un sistema para el control del ganado que había sido vacunado insertando bajo la piel de los animales una etiqueta RFID pasiva con la que se identificaba los animales que habían sido vacunados y los que no.

Después han ido llegando mejoras en la capacidad de emisión y recepción, así como en la distancia, lo cual ha llevado a extender su uso en ámbitos tanto domésticos como de seguridad nacional, como sucede con el pasaporte expedido en la actualidad en los EEUU que lleva asociadas etiquetas RFID.

2.1.5.1 Características

La comunicación entre las etiquetas y los lectores RFID depende de la frecuencia de operación. Esta frecuencia determina la velocidad de comunicación y la distancia de lectura, normalmente a mayor frecuencia, mayor rango de lectura. Los sistemas de baja frecuencia, LF (Low Frequency), que trabajan a 125 kHz, son apropiadas para aplicaciones de corto alcance como

identificación de animales y sistemas antirrobo, como llaves de coche con RFID. Los sistemas que operan a 13,56 MHz se caracterizan como sistemas de alta frecuencia. Una de las aplicaciones básicas de HF son las tarjetas inteligentes, de acceso a edificios o como forma de pago en transporte público en países como Chile y Holanda [13].

La tercera frecuencia de operación, UHF (ultra HF), se contempla para uso generalizado en los mayoristas, que están luchando por que los proveedores rotulen las cajas con etiquetas UHF. Estas etiquetas trabajan alrededor de los 900 MHz. Sin embargo los sistemas UHF son más sensibles a las condiciones ambientales como el agua que absorbe la energía e impide la comunicación con el lector. En la tabla 10 podemos observar, los diferentes rangos de frecuencias en las que trabajan las etiquetas RFID y las distancias de lectura.

BANDA DE FRECUENCIA	FRECUENCIA DE TRABAJO	DISTANCIA DE LECTURA
LF	125 khz	Pocos centímetros
HF	13.56 Mhz	< 1 mt
UHF	915 Mhz, 433.92 Mhz. o 315 Mhz	< 5 mts

Tabla 10. Bandas de Frecuencia de RFID

2.1.5.2 Tipos de etiquetas RFID

- **Las etiquetas pasivas:** no tienen alimentación propia, no tienen batería, y no inician la comunicación. El lector debe interrogar primero la etiqueta pasiva, enviando ondas electromagnéticas que se acoplen a la antena de la etiqueta. Acorde con la autenticación, comprobación y cifrado, la etiqueta responde a la interrogación enviando vía radio la información

contenida en ella. Las etiquetas pasivas ya se están utilizando en una gran variedad de aplicaciones, entre ellas las tarjetas de acceso a edificios, billetes de transporte público y cada vez más en el seguimiento de productos a través de la cadena de suministro. Dependiendo de la sofisticación del chip, cuanta memoria tiene o su capacidad de cifrado.

- **Las etiquetas RFID semi-pasivas:** son muy similares a las pasivas, salvo que incorporan además una pequeña batería. Esta batería permite al circuito integrado de la etiqueta estar constantemente alimentado. Además, elimina la necesidad de diseñar una antena para recoger potencia de una señal entrante. Las etiquetas RFID semi-pasivas responden más rápidamente, por lo que son más fuertes en el radio de lectura comparadas con las etiquetas pasivas.
- **Las etiquetas RFID activas,** por otra parte, deben tener una fuente de energía, y pueden tener rangos mayores y memorias más grandes que las etiquetas pasivas, así como la capacidad de poder almacenar información adicional enviada por el transmisor-receptor. Actualmente, las etiquetas activas más pequeñas tienen un tamaño aproximado de una moneda. Muchas etiquetas activas tienen rangos prácticos de diez metros, y una duración de batería de hasta varios años.

2.1.5.3 Arquitectura

Un sistema básico RFID tiene tres elementos primarios: el lector y la antena RFID (que vienen integrados en un mismo dispositivo), la computadora a la que se conecta el lector en la mayoría de casos para integrarla a un sistema con una base de datos y la etiqueta. La Fig. 2.15 muestra la configuración básica de un sistema RFID.

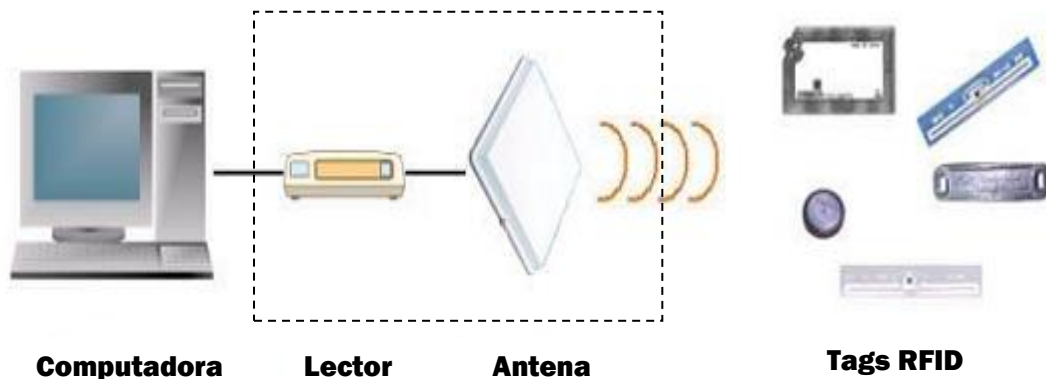


Fig. 2.16 Componentes de un sistema RFID [14]

- **Lector de RFID:** compuesto por una antena, un transmisor-receptor y un decodificador. El lector envía periódicamente señales para ver si hay alguna etiqueta en sus alrededores. Cuando capta una señal de una etiqueta (la cual contiene la información de identificación de ésta), extrae la información y se la envía al subsistema de procesamiento de datos.
- **Computadora:** sistema donde se almacena la información sobre los elementos etiquetados.
- **Etiqueta RFID:** Las etiquetas RFID toman multitud de formas y tamaños según los diferentes entornos donde deben utilizarse, esta característica de adaptación proporciona diversos tipos de etiquetas.

Además estas etiquetas pueden estar encapsuladas en diferentes tipos de material. Hay etiquetas que se encapsulan en plástico (normalmente PVC), o botones para obtener mayor durabilidad, sobretodo en aplicaciones de ciclo cerrado donde se tiene que reutilizar o en ambientes hostiles. También pueden estar insertadas en tarjetas de plástico como las de crédito, este tipo se denominan “contactless smart cards”, cómo

último destacamos los encapsulados de cerámica especialmente idóneos en entornos corrosivos, líquidos o para incrementar la protección de las etiquetas.

Las etiquetas RFID están compuestas por:

- Circuito Codificador/Decodificador: se utiliza para transformar un tipo de datos en otro diferente.
- Memoria: Dispositivo electrónico en el que se almacenan los datos dentro de la etiqueta.
- Antena: Dispositivo metálico encargado de radiar y recibir ondas de radio y que adapta la entrada/ salida del receptor/transmisor al medio. La antena va unida a la memoria y es la responsable de la transmisión de información de las etiquetas, hacia el lector RFID. Normalmente cuanto mayor es la antena mayor es el rango de lectura o distancia de identificación.
- Fuente de alimentación: Dispositivo que suministra la potencia en forma de tensión y corriente.

2.1.5.4 Ventajas

- Gran capacidad de almacenaje de información.
- Información almacenada en la etiqueta puede ser actualizada.
- Capacidad de recoger información de muchas etiquetas al mismo tiempo.
- Recolección de datos sin contacto directo o visibilidad de las etiquetas.



- Gran precisión en la recuperación de datos.
- Las etiquetas pasivas, pueden ocultarse fácilmente y colocarse en diversos productos.
- Bajo riesgo de obstrucción al no tener que insertarse en ningún tipo de ranura.
- Bajo mantenimiento de equipo al no tener que reparar sensores de contacto.
- Seguridad de funcionamiento en condiciones de inclemencia (suciedad, polvo, humedad, temperatura).
- Reducción de errores humanos.

2.1.5.5 Desventajas

- Cortas distancias de los tags pasivos
- Riesgo de adulteración o reprogramación de los tags programables o actualizables.
- Requiere alta inversión inicial en hardware, sistemas y configuración de los puntos de control.

2.1.5.6 Aplicaciones

- Control de calidad, producción y distribución.



- Localización y seguimiento de objetos.
- Control de accesos.
- Identificación de materiales.
- Control de fechas de caducidad.
- Detección de falsificaciones.
- Almacenaje de datos.
- Automatización de los procesos de fabricación.
- Información al consumidor.
- Identificación de personas.
- Reducción de colas a la hora de pasar por caja.
- Identificación y localización de animales perdidos.
- Elaboración de censos de animales.
- Identificación y control de equipajes en los aeropuertos.
- Inventario automático.



2.2 CUADRO RESUMEN COMPARATIVO

TECNOLOGIA CARACTERISTICAS	CODIGO DE BARRAS	TARJETAS INTELIGENTES (CONTACT CARDS)	TECNOLOGIAS BIOMETRICAS	BANDA MAGNETICA	RFID
Lectura a larga distancia			✓		✓
Lectura multiple					✓
Unicidad			✓	✓	✓
individualidad	✓	✓	✓	✓	✓
Almacenamiento de informacion		✓		✓	✓
Resistencia a interperie		✓	✓		✓
Fiabilidad		✓	✓		✓
Permanencia	✓	✓	✓	✓	✓
Aceptabilidad	✓	✓		✓	
Lectura sin linea vista					✓

TECNOLOGIA CARACTERISTICAS	TARJETAS INTELIGENTES (CONTACT CARDS)	BANDA MAGNETICA	RFID
Capacidad de Memoria	1k bytes-256K bytes [15]	1k bytes-4K bytes [16]	64 bytes-64K bytes [17]



CAPITULO III

SITUACIÓN ACTUAL DEL SECTOR SALUD EN EL SALVADOR



3 CAPITULO III

El Salvador, es un país ubicado en América Central que tiene una extensión territorial de 21,040.79 Km² y 6,990,657 habitantes para el año 2006 [18], lo que muestra una densidad demográfica de 332.24 habitantes por Km², convirtiéndolo en uno de los países más densamente poblados de América Latina. Políticamente, el país está dividido en 14 departamentos y 262 municipios.

La concentración de población está principalmente en el área urbana, la cual es mayor en zonas marginales; si bien la expectativa de vida ha aumentado no se ha mejorado sustancialmente su calidad.

Cerca de la mitad de la población vive bajo la línea de pobreza, concentrándose ésta en el campo y en áreas marginales. Pese a los avances en materia de educación, un quinto de la población es analfabeta; la escolaridad promedio de los(as) salvadoreños(as) es baja sin alcanzar niveles de educación que le permitan prevenir enfermedades y cuidar su salud en general.

Aunque las cifras oficiales estiman que el desempleo es 5.7% y 31.5% de la población está subempleada [19], la PEA⁶ cotizante a la seguridad social apenas es un 21.5% [16], lo cual significa que 2 millones de la misma población no tiene empleo estable.

Para el año 2006 El Salvador se ubicó en la posición 101 en el Índice de Desarrollo Humano de un total de 174 países [20], cifras que lo catalogan como país de desarrollo mediano.

⁶ PEA: Población Económicamente Activa. En general se considera PEA al conjunto de personas, de uno u otro sexo, que están dispuestas a aportar su trabajo para la producción de bienes y servicios económicos.



Los cambios demográficos y sociales de nuestro país inciden en el perfil epidemiológico, el cual presenta una alta prevalencia de enfermedades infecciosas relacionadas estrechamente con el ambiente, carencias nutricionales, hábitos de consumo, falta de educación en salud y estilos de vida de la población sumado a la falta de servicios básicos.

De ahí que El Salvador se enfrente a un doble reto, terminar con el atraso representado por las enfermedades del subdesarrollo, tales como las infecciones comunes, las muertes maternas; y enfrentar los problemas emergentes asociados con el incremento en la esperanza de vida, la industrialización, la urbanización y la violencia.

El gasto de salud anual per cápita es de \$168 muy por abajo del per cápita promedio para América Latina [20]. El gasto de los hogares en salud es mayor que el gasto público lo cual denota el enorme esfuerzo que las familias realizan para el cuidado de la salud, a costa de otras necesidades.

El Sector Salud está conformado por dos subsectores: i) el público integrado por el MSPAS, el Instituto Salvadoreño del Seguro Social (ISSS), el Instituto Salvadoreño de Rehabilitación Integral (ISRI), Bienestar Magisterial, Sanidad Militar y otras instituciones autónomas y ii) el privado que incluye en entidades lucrativas y no lucrativas, que actúan descoordinadamente, con recursos concentrados en lo urbano, lo que duplica esfuerzos, y deja descubiertas amplias zonas del país.

La cobertura provista entre los dos subsectores posiblemente alcanza a dos tercios de la población, quedando un tercio de la población sin cobertura regular de servicios de salud. Por otra parte la utilización de la infraestructura de servicios de salud disponible no es óptima, esta presenta diversos patrones que van desde la saturación hasta la subutilización. La disponibilidad de recursos



físicos es también muy variada. Existe además un insuficiente e inadecuado abastecimiento de medicamentos, lo cual conlleva a una inconformidad entre los usuarios.

Las deficiencias de gestión y administración son temas que han sido discutidos en diversas ocasiones, poniéndose en manifiesto estos problemas en las distintas instituciones públicas del país.

Existe deficiente cobertura de los programas de atención en salud y limitada calidad de los servicios; el énfasis en las acciones curativas y los modelos actuales de prestación de los servicios de salud no han permitido la implementación de nuevos enfoques que mejoren los resultados y la satisfacción de los usuarios.

La Gestión Administrativa de los sistemas de suministros, financieros, de personal, apoyo técnico e infraestructura necesitan mejorarse debido a que repercute en una baja capacidad de servicio.

Esto se ve reflejado en problemas económicos, manejo de recurso humano, tecnológicos e infraestructurales.

3.1 PROBLEMAS A NIVEL NACIONAL

3.1.1 PROBLEMA ECONÓMICO

El presupuesto para el año 2006 en el Ramo de Salud Pública y Asistencia Social (MSPAS) asciende a la cantidad de \$293,070,830.00 lo que significa un 8.8% del Presupuesto General de la Nación que totaliza \$3,338 millones [21].



El presupuesto se distribuye en un 6% para el funcionamiento de la dirección y administración institucional, 80% al funcionamiento de los hospitales nacionales y 14% que procede de préstamos externos, una parte está destinada al área de modernización y el resto a la reconstrucción de hospitales y ampliación de los servicios de salud (subcontratación) [22].

Una importante consideración de este presupuesto se concentra en una mayor proporción en el pago de remuneraciones, aproximadamente en un 70%, a las que se suman los gastos de funcionamiento siendo este el 30% restante (servicios básicos, combustibles y lubricantes, papelería, útiles de escritorio, etc.), lo cual indica que existe un alto grado de rigidez en la asignación de los recursos financieros disponibles, quedando un escaso margen para solventar otros rubros no menos críticos en salud (medicamentos, insumos, tecnología, infraestructura, capacitaciones, etc.) [20].

3.1.2 PROBLEMA DE RECURSO HUMANO

El problema de recurso humano, surge como consecuencia del bajo presupuesto otorgado al sector salud, en donde el jefe, coordinadores o encargados de sección, técnicos, obreros, secretarias, no se ven incentivados a desempeñar un servicio eficiente por la poca motivación salarial, este factor obliga a los empleados a emigrar a otros lugares en la búsqueda de un mejor salario, lo que ocasiona que las plazas que quedan vacantes sean sustituidas por personal, que por lo general, no llenan los requisitos de un perfil adecuado para el puesto. Por otra parte, la equívoca distribución de personal, coloca a empleados en áreas donde se sub-utiliza este recurso, dejando a otras áreas con una deficiencia de personal operativo.



Esto incurre en un servicio deficiente, y arraiga aun más el rígido sistema actual, obstaculizando la apertura a nuevas tecnologías y formas de servicio.

El recurso humano es un factor importante puesto que es quien ejecuta todas las funciones operativas dentro de una institución, desde el personal administrativo encargado de gestionar los procedimientos, hasta los empleados subalternos ejecutores. De ahí que el bajo nivel de inversión en la administración genera frustraciones e inicia un ciclo de ineficiencia como producto de la frustración, ocasionando que se pierdan oportunidades de desarrollo personal y profesional.

3.1.3 PROBLEMA TECNOLÓGICO

Los cambios en el mundo actual exigen de todas las organizaciones, ya sean estas públicas o privadas, una capacidad de adaptación e innovación constante, estas exigencias y tecnologías nuevas generan presiones para establecer nuevas formas de proceder y así aumentar el impacto de la institución en la comunidad.

Los hospitales como prestadores de servicios de relevancia social deben estar atentos al desarrollo de tecnologías, no solo para adaptaciones temporales para circunstancias favorables, sino para acciones gerenciales ya que el aprovechamiento al máximo de los conocimientos mejorará la eficiencia al prestar los servicios.

Sin embargo en el país, estos cambios tecnológicos se ven frenados obstaculizando el salto a nuevas tecnologías de servicio, esto por factores económicos como principal punto, al no haber un porcentaje significativo del presupuesto otorgado al sector salud que proporcione una liquidez para la compra de equipos modernos.



En cuanto al adiestramiento no existe un proceso de capacitación constante en la cual los nuevos equipos sean utilizados por el recurso médico, como consecuencia de una falta de apoyo a la dirección del hospital que origina el que no se preste una atención continua a las prácticas y métodos de acción, cayendo en una subutilización de los equipos haciendo que se vuelvan obsoletos.

Y como último punto, la infraestructura primitiva que obstaculiza la incorporación de nuevas tecnologías por ofrecer una inadecuada ambientación para la implementación de ellas.

3.1.4 PROBLEMAS ESTRUCTURALES

El Ministerio de Salud Pública y Asistencia Social (MSPAS) es el responsable directo de proveer servicios de salud al 80% de la población, y en función de ello dispone de una red de 623 establecimientos (30 hospitales, 369 unidades de salud, 179 casas de salud, 50 centros rurales de Salud y nutrición, 2 clínicas y 2 centros de atención de emergencias) y 4,673 camas [24] (Ver Anexo N° 1).

En la actualidad el país enfrenta muchos problemas a nivel de infraestructura en casi la mayoría de hospitales en todo el territorio. Estos problemas se vieron acentuados aún más con los terremotos sufridos en los años 1986 y 2001 en donde la infraestructura pública sufrió severos daños, incluido más de la mitad de los hospitales. Para el terremoto de 1986 entre los hospitales dañados se encuentra la pérdida total del Hospital General del ISSS, el Hospital Bloom, con colapso parcial en una réplica y evacuado un 100%, el Hospital de Maternidad (MSPAS) con daño en la lavandería y quirófanos, 100% evacuado, el Hospital Rosales con daño en quirófanos y el Hospital Neumológico y General con daño



en quirófanos. Además se evacuaron los hospitales de San Rafael (MSPAS), Maternidad, 1° Mayo (ISSS) y Militar de San Salvador.

Para el año 2001 El Salvador sufrió el efecto de dos terremotos en los que murieron 1,260 personas, se destruyeron ocho hospitales; y 113 de los 361 establecimientos de salud, fueron destruidos o gravemente dañados. En total, el 55% de la capacidad de entrega de servicios de salud fue afectada. Sin embargo, a pesar del tiempo transcurrido desde entonces, los fondos destinados a la salud no han logrado suplir las deficiencias estructurales de estos hospitales.

Es el hospital la infraestructura que más debe resistir en una situación de emergencia, para poder dar asistencia a las gentes afectadas o a los heridos como consecuencia de la catástrofe. Por lo tanto es en estos centros de salud, ubicados en los países en zonas sísmicas, donde los requerimientos técnicos deben normarse, caso que no sucede en nuestro país. Dichos requerimientos no solamente deben estar referidos a las condicionantes del cálculo estructural, sino que han de comprender a las diferentes profesiones, desde la programación arquitectónica, el anteproyecto de arquitectura y de las demás especialidades, llámense médicas, de ingeniería eléctrica o mecánica y sanitaria.

Los problemas de infraestructura no solo radican en daños físicos a las instalaciones, sino también en la inadecuada distribución de las mismas.

Por lo tanto, el progreso de los hospitales puede ser visto como resultado de la capacidad gerencial de responder a los desafíos impuestos por los cambios de la sociedad, económicos, políticos y tecnológicos.

Estos son los problemas que enfrenta el sector salud a nivel nacional, sin embargo el atacar estos problemas desde un punto de vista macro puede resultar muy difícil para plantearles una solución coherente, por ello se ha optado



por tomar una muestra en donde se abordaran los mismos pero de forma más profunda.

3.2 ESTUDIO DE LA MUESTRA

En El Salvador los hospitales se encuentran ubicados en los lugares con mas concentración de población lo que facilita el acceso a ellos, no obstante hay diversos aspectos a tomar en cuenta para la selección de la muestra, entre estas variables podemos mencionar: la cercanía de cada centro de salud, la accesibilidad y disponibilidad de información, la cantidad de población atendida, los tipos de servicio, entre otras.

En base a estas variables se han tomado como referencia las siguientes tres alternativas:

- Hospital Nacional San Bartolo “Enf. Angelica Vidal de Najarro”.
- Hospital Nacional Neumológico "Dr. José Antonio Saldaña”.
- Hospital Nacional Especializado “Rosales”.

A continuación se describe cada una de ellas:

a) ALTERNATIVA A

Hospital Nacional San Bartolo “Enf. Angélica Vidal de Najarro”.

Se encuentra ubicado en la zona oeste del departamento de San Salvador, Final Calle Francisco Menéndez, Contiguo a Zona Franca San Bartolo, Ilopango, La población del municipio es de 152,465 habitantes



[25], y en el hospital se atiende una población promedio de 514 personas diariamente (ver anexo 2).

b) ALTERNATIVA B

Hospital Nacional Neumológico "Dr. José Antonio Saldaña".

Se encuentra ubicado en la zona sur del departamento de San Salvador, Kilómetro 8½, Carretera Los Planes de Renderos, San Salvador, La población del municipio es de 507,665 habitantes [25], y en el hospital se atiende una población promedio de 176 personas diariamente (ver Anexo 3).

c) ALTERNATIVA C

Hospital Nacional Especializado "Rosales"

Se encuentra ubicado en el centro de San Salvador, Final Calle Arce y 25 Av. Norte, San Salvador, La población del municipio es de 507,665 habitantes [25], y en el hospital se atiende una población promedio de 927 personas diariamente (ver Anexo 4).

3.3 SELECCIÓN DE LA MUESTRA

Después de realizar una evaluación sobre los posibles sitios para la selección de la muestra, se determinó que el lugar más factible para el desarrollo de nuestro proyecto es la alternativa C, Hospital Nacional Especializado "Rosales". Este hospital cuenta con fácil acceso, en relación a las otras dos instituciones, otro punto relevante en la toma de esta decisión es que poseen gran apertura a proyectos universitarios, lo que contribuye a la rápida obtención de información.

No podemos dejar de mencionar que en esta institución es donde a diario se atiende el mayor número de pacientes, lo que pondrá a prueba la eficiencia de nuestro proyecto.

3.4 HOSPITAL NACIONAL ES3(\$,\$2526\$)/(6



Fig. 3.1 Imagen de la fachada del Hospital Nacional Especializado Rosales.

Este hospital debe su nombre a un ex presidente de la república, cuyo gobierno se dio en 1885, su nombre fue José Rosales, este al verse sin herederos dejó en su testamento el deseo que su fortuna fuese repartida entre su esposa y el Hospital de El Salvador. El Sr. Rosales murió en 1891, y ese mismo año se colocó la primera piedra para su construcción, terminándose de construir en 1902, justo hace 105 años. En agradecimiento, se nombró al hospital con el apellido del benefactor.



La estructura organizativa del hospital está dividida en estas tres grandes áreas de acuerdo a su desempeño laboral:

- Área Administrativa.
- Área Médica.
- Área Paramédica.

Área Administrativa

Esta área es la encargada de proporcionar apoyo para la ejecución de las diversas actividades del quehacer hospitalario, y está compuesta por los siguientes departamentos: dirección y administración, recursos humanos, la unidad financiera, la unidad de adquisiciones y contrataciones, el almacén de medicamentos, el cual es de suma importancia en la planificación y gestión de recursos básicos necesarios para el trabajo asistencial. (Ver Anexo 5)

Área Médica

Esta área se encarga de brindar atención médica a toda la población que así lo solicite, ofrece servicios como: consulta externa, emergencias, medicina interna, cirugías y enfermería. (Ver Anexo 5)

Área Paramédica

Esta área se encarga de prestar servicios complementarios dentro de la institución, está compuesta por dos departamentos: departamento de diagnóstico y departamento de servicios de apoyo. (Ver Anexo 5)

El contar con grandes instalaciones y una variedad de servicios especializados, genera un alto nivel de complejidad en el funcionamiento interno, lo que conduce



a una serie de problemas que afectan a la población que requiere de estos servicios.

Como se mencionó con anterioridad los hospitales se ven envueltos en cuatro problemas fundamentales y este caso en particular no es la excepción.

3.4.1 PROBLEMA ECONÓMICO

Anualmente el ministerio de hacienda establece un presupuesto destinado para el sector salud, posteriormente esta partida es distribuida por el ministerio de salud en todos los hospitales de la red nacional.

Todos los hospitales formulan un plan de trabajo anual, con un estimado de gastos e inversiones, pero este plan no es tomado en cuenta en el momento de asignar la partida presupuestaria a cada institución, la partida es establecida según el criterio de la unidad financiera del MSPAS y por lo general es un fondo menor al que se tenía contemplado en el plan de trabajo.

Cada institución posee un comité el cual está conformado por el administrador y asesores y es el que se encarga de distribuir este reducido presupuesto en los diferentes rubros, tomando como prioridad los medicamentos, los insumos médicos y la alimentación de los pacientes, quedando en un segundo plano las necesidades de carácter estructural como tecnológico.

El Hospital Nacional Rosales se le asignó un presupuesto \$25,441,055 [22] para el año 2007, con él se pretende suplir todas las necesidades que el hospital posee, sin embargo este objetivo no es alcanzado, debido a que este presupuesto no se basa en las necesidades reales, mas bien es estipulado arbitrariamente.



Este corto presupuesto afecta a la institución de manera que no cubre con las demandas programadas, cirugías, atención ambulatoria, además que no logran ejecutar recursos, capacitaciones administrativas, incentivos personales, perdiéndose de vista la idea de mejorar los procesos.

Una consecuencia que trae el corto presupuesto asignado es la carencia de medicamentos, como muestra de ello en Abril de 2006, el Hospital Rosales, centro de mayor referencia en el país, registró escasez de 170 medicamentos.

Otro ejemplo es la construcción de las salas de quirófanos que fueron inauguradas hasta mayo de 2007, instalaciones que quedaron gravemente dañadas por lo terremotos de enero del 2001, haciéndolas inoperables por lo que fue necesario la creación de quirófanos provisionales.

Esto denota que el problema financiero es la causa principal de donde se derivan la mayor parte de problemas de tipo estructural, tecnológico y de recurso humano.

3.4.2 PROBLEMA DE RECURSO HUMANO

El recurso humano forma una de las partes más importantes dentro de los hospitales ya que los demás recursos, materiales y técnicos, están sujetos a este, de aquí la importancia de mantener al personal siempre capacitado, motivado, dentro de un ambiente de cordialidad y de respeto.

En el Hospital Nacional Especializado Rosales, existen diferentes tipos de problemas relacionados al recurso humano, los cuales pueden resumirse brevemente en: incapacidad en la selección de personal, capacitación, reconocimiento y remuneración.



Todos estos problemas, radican en la insuficiencia de recursos económicos, la magnitud del trabajo que existe no es proporcional a los salarios devengados por cada uno de los empleados, por esta misma razón no se tiene la capacidad de contratar nuevos recursos que cubran las necesidades que el hospital requiere.

En algunas ocasiones cuando se solicita contratar personal, el problema radica, en que la selección de este, se ve influenciada por personas que poseen un alto grado de jerarquía ya sea dentro como fuera de la institución, sin importar la capacidad, ni la experiencia que la persona a contratar posea.

Por otra parte, el constante cambio en las tecnologías exige un plan que permita al personal ser partícipe de estos procesos, para que puedan ser implementados en esta institución.

En la realidad, presupuesto para capacitaciones no existe, la mayoría de personal que labora en esta institución no ha recibido capacitaciones para el manejo de equipo, por lo que el usuario constantemente solicita ayuda a los técnicos y eso disminuye la eficiencia. Una medida que ha optado la institución para cubrir esta deficiencia es la implementación de proyectos con universidades y alumnos que hacen horas sociales para capacitar al personal de la institución en las diferentes áreas, todo esto por la falta de presupuesto para este rubro.

Una vez se consigue la capacitación para el personal surge una polémica y es la actitud que toman los usuarios ante la apertura a nuevos cambios. Estas reacciones suelen ser de dos tipos:

- Empleados abiertos a los cambios tecnológicos
- Empleados que se oponen a los cambios.



Estos últimos que se oponen, son personas que tienen mucho tiempo de laborar en la institución y cuyo temor radica en causar algún tipo de daño a los equipos.

Estos problemas tienen sus antecedentes en un sistema que hace imposible ofrecer estímulos económicos al personal, sea por buen desempeño, asistencia, sentido de responsabilidad u otras razones, pues las remuneraciones están fijadas por ley y son iguales para todos los funcionarios según criterios de grado y antigüedad. Al no existir incentivos económicos, el único reconocimiento que puede ofrecérselo a un buen trabajador es cambiarlo a otra área donde este puede desenvolverse mejor.

Por otra parte, es prácticamente imposible despedir a un mal empleado, pues los funcionarios titulares son casi dueños de sus puestos y se requiere comprobar, mediante 2 o 3 años de trámites, que alguien ha cometido delitos muy graves antes de poder despedirlo. Mientras tanto, le quedan disponibles numerosas formas de apelación.

3.4.3 PROBLEMA DE INFRAESTRUCTURA

El Hospital Rosales cuenta con un terreno de 57,000 mts², con un área efectiva de 151 mts por 109.73 mts (Ver Anexo 6). Este tiene a disposición 450 camas hospitalarias, 55 consultorios médicos, 6 consultorios odontológicos, 9 quirófanos, 1 sala U.C.I., 5 salas de rayos X, 1 laboratorio clínico, 1 laboratorio ginecológico y 1 sala de ultrasonografía.

El Hospital Nacional Especializado Rosales es el centro asistencial más grande y el único hospital especializado de referencia nacional para cirugía y medicina en el país. Está previsto que este hospital será el punto de referencia nacional en caso de desastres y emergencias mayores.



Sin embargo este hospital no está preparado para eventos de gran magnitud, prueba de ello fue que después del sismo del 13 de enero se observaron en el área de hospitalización daños severos en el sistema de cielos falsos, grietas diagonales en los muros de relleno y desprendimiento del revestimiento en algunos de ellos.

Estos daños no representaban peligro de colapso de la estructura, pero si impedían el normal funcionamiento hasta que los escombros fuesen removidos.

En la torre quirúrgica se produjeron graves desperfectos en los componentes no estructurales y daños leves en los componentes estructurales; un alto porcentaje de los cielos falsos colapsó y en los muros de relleno se observaron grietas importantes. También resultó severamente deteriorado el sistema de movilización de los ascensores y se observaron grietas en las fundaciones del edificio. Algunos de estos daños, y sobre todo, la no disponibilidad de ascensores provocó que las 15 salas de cirugía quedasen fuera de servicio, con la suspensión de todas las operaciones programadas.

Para atender las urgencias se improvisaron quirófanos en las salas del hospital que no resultaron dañadas. Dos meses más tarde se construyeron quirófanos provisionales para recuperar la capacidad resolutive. Los cilindros de gases clínicos se volcaron, obstaculizando el proceso de evacuación de las personas que se encontraban en el edificio y poniendo en peligro al hospital y a sus ocupantes.

Desde entonces el proceso de reconstrucción a sido lento, por la limitante que todos los hospitales enfrentan que es el corto presupuesto asignado.

Como muestra existe una sala de consulta externa provisional, la que posee una sala de espera de pacientes y otra de preparación, una estación de enfermeras, un espacio para información y orientación de usuarios, área de vacunación,



consultorios médicos y de odontología; cuenta con cubículos para consejería de programas preventivos y epidemiológicos, un cubículo para atención de diabéticos por ASADI y el área de servicios sanitarios. La iluminación es pobre, así como la ventilación; los espacios para la atención de pacientes son reducidos en general, no obstante se está trabajando en la construcción de las nuevas instalaciones del servicio de consulta externa, la cual poseerá infraestructura más moderna, mayor número de consultorios, y una distribución más adecuada lo que contribuirá a un ambiente más propicio para la atención al público (Ver Anexo 7).

No podemos dejar de mencionar, que el hospital por ser una infraestructura antigua se ha convertido en un patrimonio cultural por lo que no se pueden realizar modificaciones estructurales en las instalaciones.

3.4.4 PROBLEMA TECNOLÓGICO

El área tecnológica de un hospital se compone básicamente de equipos médicos, equipos informáticos y de comunicación. El número de equipos y la complejidad de la comunicación difiere de un centro a otro en relación al nivel de atención, el Hospital Nacional Rosales es un hospital de tercer nivel lo que demanda la existencia de equipos médicos especializados, mayor número de recurso informático, y una red de comunicación adecuada a las grandes instalaciones.

Sin embargo, el hospital enfrenta algunos problemas en esta área, parte de los equipos médicos no se encuentran en condiciones óptimas y han sufrido severos daños a consecuencia de los terremotos o por falta de un mantenimiento preventivo lo que incurre en un deterioro más acelerado de los aparatos.



El recurso informático cubre aproximadamente un 70% del hospital, abarcando principalmente el sector administrativo, quedando sin este recurso el área médica.

A partir del año 2003 se puso en marcha la instalación de una red local, y en la actualidad ya se cuenta con servicio de Internet, no obstante aun existen áreas aisladas lo que inhabilita la comunicación entre departamentos.

En la institución existe una jerarquía de adquisición que establece como principal punto el abastecimiento de medicamentos y alimentación de pacientes, como segunda instancia los insumos médicos, y por último el área tecnológica. Esto dificulta aun más el salto a nuevas tecnologías, que podrían mejorar la eficiencia en servicios, comunicaciones, etc., pues requieren de un alto grado de inversión para poder implementarlas.



CAPITULO IV

ANALISIS TECNICO FINANCIERO



4 CAPITULO IV

En capítulos anteriores se habló acerca de las características, ventajas, desventajas y aplicaciones de las tecnologías de acceso más conocidas en el mercado. Posteriormente se hizo mención de la problemática que afrontan la mayor parte de los hospitales a nivel nacional, problemas de tipo tecnológico y de recurso humano que arraigados a un problema financiero y de infraestructura terminan en una ineficiencia en el servicio personalizado a pacientes. Es así como se hace posible la búsqueda de una solución mediante técnicas aplicativas que permita la ejecución de un sistema que optimice el proceso actual en el servicio general a pacientes.

Cabe mencionar que dentro de los servicios que el hospital otorga a los ciudadanos están: emergencia, consulta externa, cirugía general, medicina interna entre otros. Sin embargo, es el servicio de consulta externa el que requiere mayor cantidad de recursos por la afluencia de personas que diariamente solicitan esta asistencia. Es así como se ha optado por brindar una alternativa para que el servicio de consulta externa a pacientes sea más ágil y personalizado que el sistema actual.

En este capítulo se procederá a un análisis del proceso actual de consulta, identificando los puntos que pueden ser mejorados. Una vez establecidos estos indicadores, se tomarán como base para el diseño de la solución que conforma, la elaboración de un escenario propuesto y la evaluación de cada tecnología dentro de este espacio, es decir, una valoración de los equipos y componentes dentro de los parámetros existentes en el entorno de desarrollo del proyecto, con el propósito de seleccionar la mejor alternativa que ofrezca un balance técnico y financiero que conlleve al diseño del sistema final.



4.1 CONSULTA EXTERNA

La consulta externa es el servicio en el cual se imparte atención médica a los enfermos no internados y cuyo padecimiento les permite acudir al hospital. La atención médica en consulta externa puede ser de diversa índole pero principalmente consiste en el interrogatorio y examen que conducen al diagnóstico y a la prescripción de un tratamiento.

Este procedimiento se puede representar en un bosquejo o diagrama de flujo, con el objetivo de visualizar las trayectorias, variables y toma de decisiones que participan significativamente en el proceso.

En el diagrama de flujo (Fig. 4.1) se representa el recorrido de un paciente que solicita el servicio de consulta, desde que ingresa al hospital en el área de consulta externa, hasta su retiro de las instalaciones.

Como primer paso, si el solicitante es un paciente nuevo se registra efectuándose la toma de sus datos personales (nombre, dirección, etc.) y se crea su expediente, si ya está registrado se identifica y se determina el por qué de su visita (consulta, retiro de medicamentos). En el caso de ser una evaluación médica y dependiendo de su situación clínica se asigna un médico (general, oftalmólogo, etc.). Posteriormente se le hace entrega de un número de ficha para solicitar el expediente clínico en el departamento de Archivo, consecutivamente el cuadro clínico se le envía al médico asignado. Luego de referido el interno, este efectúa la revisión médica al paciente, y de ser necesario, él mismo establece la toma de exámenes y prescripción de medicamentos. Por otro lado, después de la consulta, el paciente está en la facultad de retirar medicinas y resultados de laboratorio en cualquier instante.

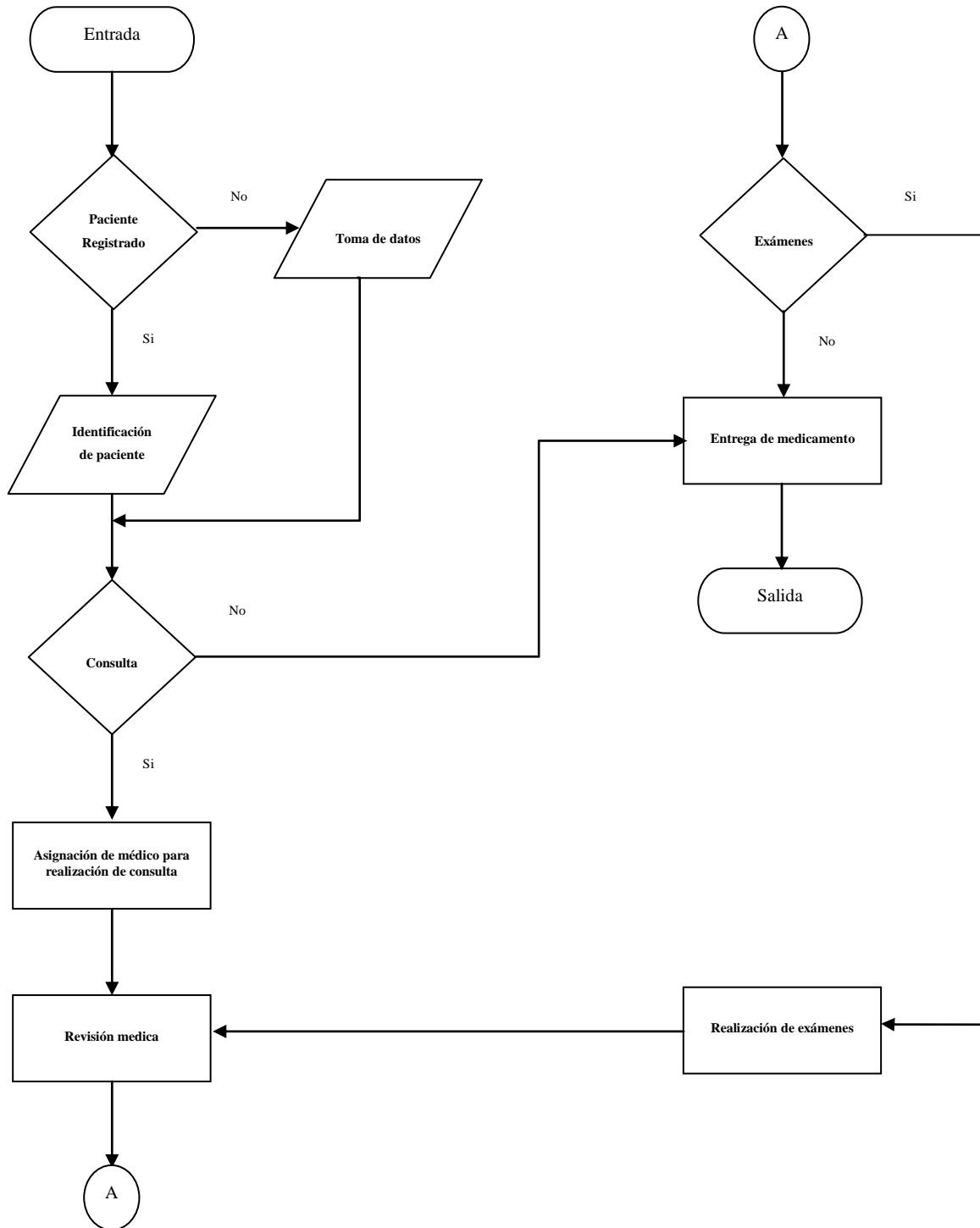


Fig. 4.1 Diagrama de flujo de la trayectoria de un paciente en una consulta médica.



Como se observa en el diagrama, el primer paso para realizar el proceso de consulta de un paciente es la identificación de este, dicha identificación se efectúa en la sección de Registro, lugar donde se entrega una ficha, la cual es necesaria para la solicitud de expedientes en la unidad de Archivo.

El problema frecuente en estos dos departamentos es la conglomeración de personas o lo que comúnmente se conoce como “colas”: En Registro esta concentración de pacientes se origina por la metodología utilizada en la recopilación de datos que se ejecuta de forma manual, de igual forma en el departamento de Archivo la búsqueda del expediente se hace manualmente produciendo que este proceso sea efectuado de manera lenta y engorrosa. Otros problemas implícitos de esta modalidad es el tiempo que tarda el personal en llevar cada expediente al internista que efectuará la revisión médica y la probabilidad que alguno de los expedientes no se encuentren ubicados en los ficheros correspondientes producto de alguna equivocación del personal de Archivo. Mientras tanto en el chequeo médico existe el inconveniente del desconocimiento por parte del doctor de los medicamentos disponibles en el almacén de farmacia, lo que imposibilita al internista, en caso de no existir un tipo de medicamento, la prescripción de un sustituto.

Para finalizar el proceso de consulta se efectúa la entrega de medicamentos en farmacia, presentándose en este sitio nuevamente el problema de conglomeración de personas, esto como resultado de la espera del paciente por la medicina previamente recetada.

Ahora que se han establecido los problemas en el actual proceso de consulta, es posible elaborar un contexto que proponga una solución a esta problemática. En la fig. 4.2 se muestra un escenario que puede servir como base para la implementación de un sistema.

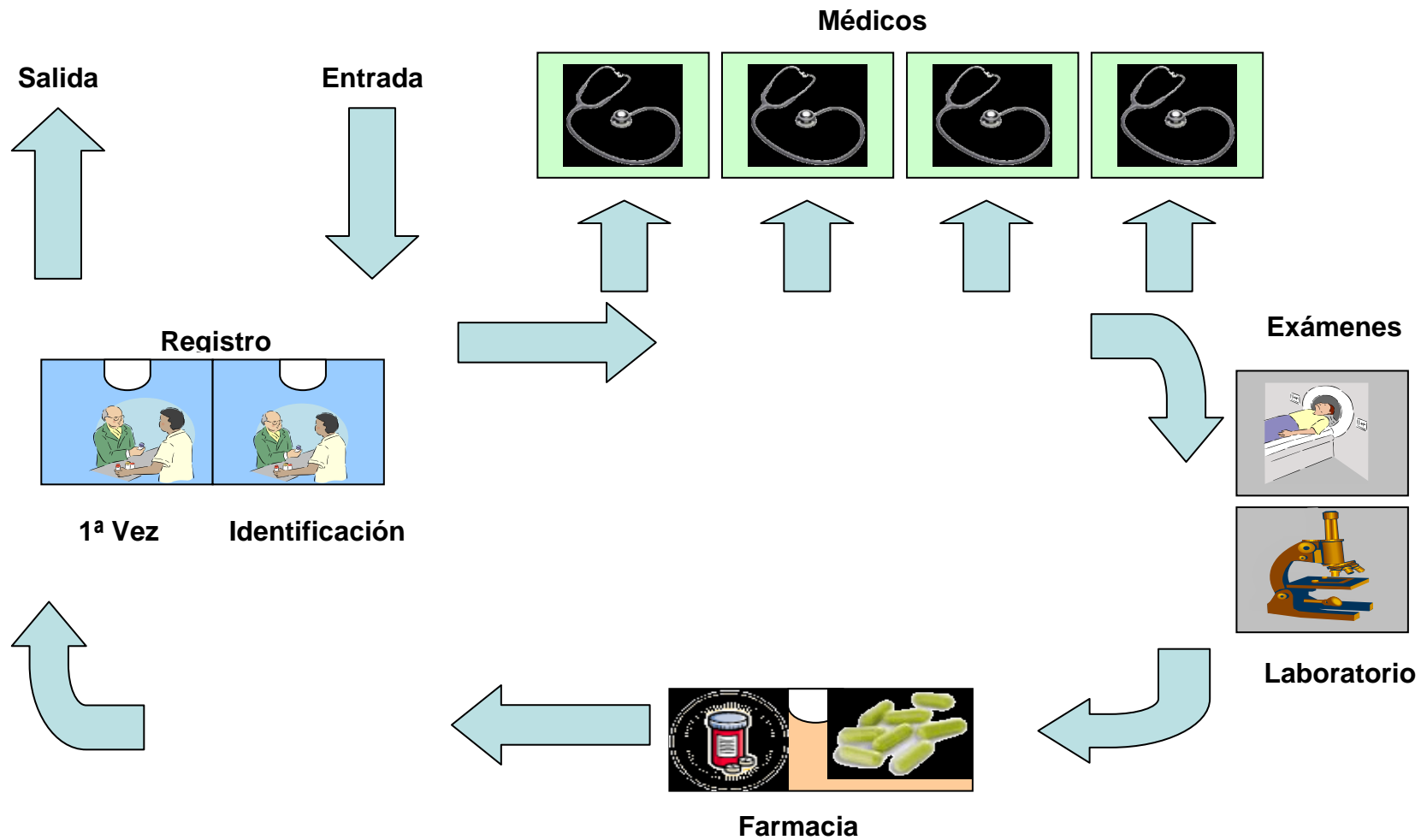


Fig. 4.2 Escenario propuesto del servicio de consulta externa.



En el escenario mostrado en la figura anterior se pretende solucionar el problema de lentitud en el proceso identificación del paciente mediante técnicas de identificación digital, con el fin de agilizar y hacer más eficiente este proceso.

Por otra parte, para eliminar el problema de la demora en la búsqueda de expedientes y del extravío de cuadros clínicos, se propone desarrollar una base de datos que contenga toda la información relacionada con el paciente (datos personales, historial clínico, resultado de exámenes, etc.) y de esta manera facilitar la búsqueda de esta información con la ayuda de programas de consulta.

Para dar solución al problema en el traslado de expedientes desde el departamento de Archivo hasta el consultorio, se planea la implementación de una infraestructura de red que posibilite a cada médico tener acceso a la información clínica del paciente, así como tener acceso a una base de datos donde se lleve el control de los medicamentos existentes en el área de farmacia, esto de manera inmediata.

Y como último punto se pretende evitar la aglomeración de pacientes en el área de farmacia, mediante el uso de una pantalla donde se muestre la solicitud de medicamentos, después que el médico haya realizado la prescripción, puesto que la recolección de estos podrá realizarse antes que el paciente se haga presente en dicho lugar.

Para alcanzar este objetivo es necesario tomar en cuenta todas las variables involucradas, instalación, software, costo, eficiencia, etc. Con el fin de realizar un análisis técnico y económico en el que se muestre como comparativa las diversas tecnologías y así determinar la opción que ofrezca la mayor utilidad a un considerable costo.



4.1.1 ANÁLISIS TÉCNICO

En capítulos anteriores se contrastaron las diferentes tecnologías propuestas en base a las características comunes y su funcionalidad, características como permanencia, fiabilidad, seguridad entre otras; y es a partir de sus ventajas y desventajas que se establece la tecnología de Identificación por Radio Frecuencia (RFID) como la mejor alternativa en el área, por ofrecer un perfil adecuado para el ambiente hospitalario.

4.1.2 ANÁLISIS ECONÓMICO

Algunas veces la tecnología que presenta mayor eficiencia, seguridad, es en términos de costo la menos rentable, para ello es imprescindible elaborar un presupuesto con cada tecnología con el fin de establecer cual de estas propone un equilibrio económico-tecnológico que no sacrifique de manera significativa ninguno de los dos rubros. No obstante en la implementación entran en función varios componentes con los que se relaciona cada una de ellas, de esta manera es que se ha considerado el dividir estos elementos en dos categorías:

Hardware: conjunto de componentes físicos que incluye dispositivos electrónicos y electromecánicos, cables, periféricos de todo tipo y cualquier otro elemento físico involucrado.

Software: el conjunto de programas y procedimientos necesarios para hacer posible la realización del sistema. Esto incluye herramientas informáticas tales como administrador de base de datos y software de desarrollo de aplicaciones que permitirá al resto de programas funcionar adecuadamente, facilitando la interacción con los componentes físicos y el resto de aplicaciones.



4.1.2.1 Hardware

Para la implementación de cualquiera de estas tecnologías es necesario contar con una infraestructura de red de comunicación local, igualmente es indispensable la adjudicación de equipos informáticos incluidos en el sistema de aplicación.

DISTANCIAS PARA CABLE DE RED

Tomando como referencia el plano de consulta externa (Fig. 4.3), se obtuvieron las distancias existentes entre las computadoras ubicadas en los 29 consultorios y el cuarto de comunicaciones. Estas distancias se detallan en la tabla 11:

CONSULTORIO	LONGITUD (MTS)
Consultorio Hematología 1	22
Consultorio Hematología 2	18
Consultorio Infectología	14
Consultorio Ortopedia 1	12
Consultorio Ortopedia 2	16
Consultorio Cirugía Plástica	24
Consultorio Proctología	32
Consultorio Urología 1	38
Consultorio Urología 2	42
Consultorio Nefrología 1	52

Tabla 11. Respectivas distancias de cada consultorio al gabinete informático



CONSULTORIO	LONGITUD (MTS)
Consultorio Nefrología 2	61
Consultorio Medicina interna 1	45
Consultorio Medicina interna 2	31
Consultorio Medicina interna 3	27
Consultorio Medicina interna 4	23
Clínica Hipertensión	19
Dermatología 1	17
Dermatología 2	21
Clínica de EPOC	25
Neumología	29
Neurología Siquiatría	33
Neurología	37
Consultorio Cirugía 1	41
Consultorio Cirugía 2	45
Cirugía Anestesia	49
Consultorio Otorrino	14
Consultorio Alergología	20
Consultorio Psicología	27
Farmacia	57
Servidor central	100
Total	991

Tabla 11. Respectivas distancias de cada consultorio al gabinete informático

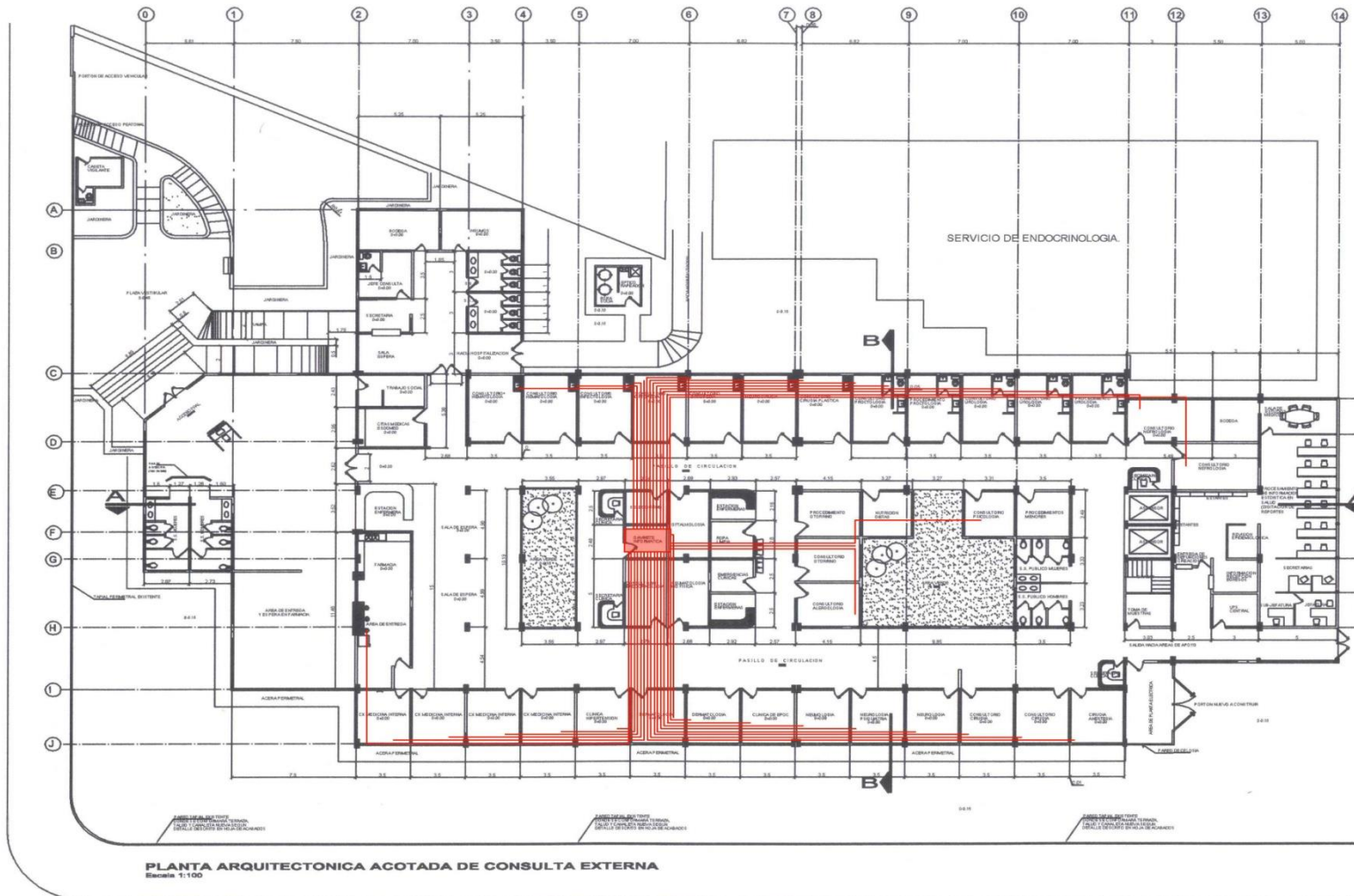


Fig. 4.3 Hospital Nacional Especializado Rosales, Diseño de red en el área destinada a Consulta Externa



A partir de la tabla 11 se determinó la longitud de cable para la implementación de la red, sin embargo para efectos de diseño es recomendable tomar un porcentaje de apreciación de aproximadamente el 20%, dando como resultado un total de 1,189.2 mts.

EQUIPO DE RED

Para el manejo de información, almacenamiento y comunicación se hará uso de materiales y equipos que se describen en la tabla 12 (Ver Anexo N°9):

DESCRIPCIÓN	CANTIDAD	PRECIO UNITARIO	PRECIO TOTAL
Servidor	1	\$ 2514.25	\$ 2,514.25
Computadora Personal	29	\$ 902.87	\$ 26,183.23
Switch de 48 puertos	1	\$ 450.87	\$ 450.87
Patch panel de 48 puertos	1	\$ 110.00	\$ 110.00
Patch cord de 1 mt.	29	\$ 1.50	\$ 43.50
Patch cord de 10 pies	29	\$ 2.50	\$ 72.50
Keystone	29	\$ 2.25	\$ 65.25
Wallplate	29	\$ 1.00	\$ 29.00
Bobina de Cable UTP 5e (300 mts)	4	\$ 75.00	\$ 300.00
RACK 19"	1	\$ 200	\$ 200
		Total	\$29,968.60

Tabla 12. Descripción de equipos para la implementación de la red local.

El costo del equipo de acceso que incluye lectores, lectores/escritores y tarjetas se detalla en la tabla 12. Para efectos de una mejor apreciación financiera se



optó por fijar la misma cantidad de recurso por cada tecnología (Ver Anexo N° 10).

EQUIPO DE ACCESO	CANTIDAD	PRECIO UNIDAD	PRECIO TOTAL
Código de barra			
Lector	2	\$ 459.00	\$ 918.00
Impresor de tarjetas PVC	1	\$ 1739.60	\$ 1739.60
Tarjeta credencial PVC	500	\$ 0.11	\$ 56.00
Total			\$ 2713.00
Tarjetas Inteligentes			
Lector	2	\$ 29.14	\$ 58.28
Lector/Escritor	1	\$ 32.90	\$ 32.90
Tarjeta	500	\$ 1.38	\$ 691.95
Total			\$ 783.13
Sistema Biométrico			
Lector Huella Dactilar	3	\$ 367.94	\$ 1103.82
Total			\$ 1103.82
Banda magnética			
Lector	2	\$ 115.99	\$ 231.98
Lector/Escritor	1	\$ 574.57	\$ 574.57
Tarjeta	500	\$ 0.26	\$ 127.08
Total			\$ 933.63
RFID			
Lector	2	\$ 180.00	\$ 360.00
Lector/Escritor	1	\$ 220.00	\$ 220.00
Tarjeta	500	\$ 1.50	\$ 750.00
Total			\$ 1330.00

Tabla 13. Comparativa del costo por tecnología de acceso.



4.1.2.2 Software

Para el almacenamiento de la información de los usuarios se hará uso de SQL SERVER, este será el sistema de administración para la Base de Datos. Por otra parte, para la elaboración de la interfaz gráfica se utilizará el entorno de desarrollo Power Builder, finalmente para realizar las consultas en línea se hará uso del programa Apache con el lenguaje PHP, en conjunto permiten conectarse a la base de datos y así realizar las consultas de los usuario registrados. El costo de los programas de aplicación utilizados para la interfaz gráfica, el almacenamiento de información en la base de datos, la comunicación entre equipos y el entorno WEB se detalla en la tabla 14 (Ver Anexo N°11):

APLICACIÓN	CANTIDAD	PRECIO	TOTAL
Power Builder 10	1	\$394.383	\$394.38
SQL Server 2000	1	\$163.90	\$163.90
PHP	1	Gratis	\$ 0.00
Apache	1	Gratis	\$ 0.00
Total			\$ 558.28

Tabla 14. Costo por programas de aplicación.

4.1.3 TOTAL DE INVERSION EN EQUIPO COMÚN

En la tabla 15 se detalla el monto total de inversión del equipo en común para todas las tecnologías.

HARDWARE	SOFTWARE	TOTAL
\$29,968.60	\$ 558.28	\$ 30,526.88

Tabla 15. Monto total de inversión en equipos comunes.

4.2 COMPARATIVA DE COSTOS DE IMPLEMENTACION DE TECNOLOGÍAS

TECNOLOGÍA	COSTO	EQUIPO EN COMÚN	TOTAL
Código de barras	\$ 2713.00	\$ 30,526.88	\$ 33,239.88
Tarjetas inteligentes	\$ 783.13	\$ 30,526.88	\$ 31,310.01
Sistema Biométrico	\$ 1103.82	\$ 30,526.88	\$ 31,630.70
Banda magnética	\$ 933.63	\$ 30,526.88	\$ 31,460.51
RFID	\$ 1330.00	\$ 30,526.88	\$ 31,856.88

Tabla 16. Monto total de inversión en equipos comunes.⁷

En base a la información provista en las tablas anteriores, producto de una investigación financiera, se optó como mejor alternativa por su equilibrio técnico-económico la tecnología de acceso Identificación por Radio Frecuencia (RFID). Técnicamente esta tecnología posee ventajas de funcionalidad, seguridad y desempeño haciéndola sobresalir sobre las demás tecnologías por estas características. Financieramente esta ubicada en el rango de tecnologías con mayor costo de inversión, no obstante esta diferencia no es significativa en relación al costo-beneficio que esta tecnología proporciona.

⁷ El análisis financiero realizado es únicamente costo-monto inicial, basado en equipo tecnológico; no se considera costo operativo.



CAPITULO V

APLICACIÓN



5 CAPITULO V

En el presente capitulo se presenta el prototipo para la implementación del sistema de identificación, con lo cual se pretende demostrar que es posible construir un sistema real utilizando una tecnología que ofrece estabilidad, seguridad, aceptabilidad y un sin número de características, que hacen en conjunto con programas de aplicación adecuados, una herramienta gráfica fácil de usar.

5.1 HARDWARE

En el proceso de identificación se hará uso de un lector y tarjetas de tipo RFID, para efectos demostrativos se utilizará como lector el modelo MF1 RW 232 D2, este posee una distancia máxima de autenticación de aproximadamente 5 cms, estableciendo su comunicación en la banda de 13.56 MHz, sus especificaciones técnicas se mencionan mas detalladamente en la sección de Anexos (Ver Anexo N°12). Las tarjetas utilizadas como dispositivo portador de identificación son del modelo ISO 14443A, poseen una capacidad de memoria de 64K bytes y son compatibles en la banda de 13.56 MHz (Ver Anexo N° 13).

5.2 SOFTWARE

Como programa para elaborar la interfaz grafica se seleccionó a Power Builder versión 10. Power Builder es un entorno de programación en ambiente gráfico que permite desarrollar poderosas aplicaciones con acceso a bases de datos,



igualmente proporciona todas las herramientas necesarias para la construcción de aplicaciones complejas.

Este software facilita el desarrollo de sistemas por su característica de visualización y manejo de controles, lo que permite al usuario una amigable navegación dentro de los sistemas elaborados.

Para la administración de los expedientes se seleccionó SQL Server por ser un gestor de base de datos que posee múltiples herramientas de análisis y gestión de almacenamiento, además proporciona un nivel de confiabilidad aceptable en la seguridad de la información.

Por otra parte se ha implementado un portal Web dándose a conocer información general del hospital, servicios que este brinda a los pacientes y que el usuario tenga la posibilidad de acceder a una sección donde pueda corroborar la fecha de su próxima visita. Para este propósito se hizo uso de los programas de aplicación Apache, que es un servidor Web gratuito, potente y que ofrece un servicio estable, sencillo de mantener y configurar; y PHP⁸ como herramienta de programación estructurada para el manejo de consultas de base de datos.

5.3 APLICACIÓN

En las siguientes líneas se procederá a describir la interfaz gráfica del Sistema de Identificación Digital de Pacientes (SIDP), indicando los procedimientos a seguir por parte del usuario según su cargo (repcionista, personal de laboratorio, médico o personal de farmacia). Posteriormente se describirá el

⁸ **PHP:** es un lenguaje de programación usado normalmente para la creación de páginas web dinámicas. PHP es un acrónimo recursivo que significa "**PHP** **H**ypertext **Pre**-processor", (inicialmente PHP Tools, o, *Personal Home Page Tools*)

servicio de control de citas en el entorno Web desarrollado para los usuarios a través de Internet.

5.3.1 SISTEMA DE IDENTIFICACION DIGITAL DE PACIENTES (SIDP)

5.3.1.1 Procedimiento N° 1: Ejecución.

Para inicializar la interfaz grafica del sistema es necesario ejecutar el archivo SIDP.EXE, inmediatamente se desplegará en pantalla la siguiente ventana de presentación (Fig. 5.1).



Fig. 5.1 Pantalla de presentación del Sistema de Identificación Digital de Pacientes

5.3.1.2 Procedimiento Nº 2: Autenticación.

En este paso cada usuario procede a autenticarse ante el sistema, es decir, identificarse dependiendo del cargo que desempeña, por ejemplo: personal de farmacia, personal de recepción entre otros. La ventana desplegada es la que se muestra en la siguiente figura (Fig. 5.2).

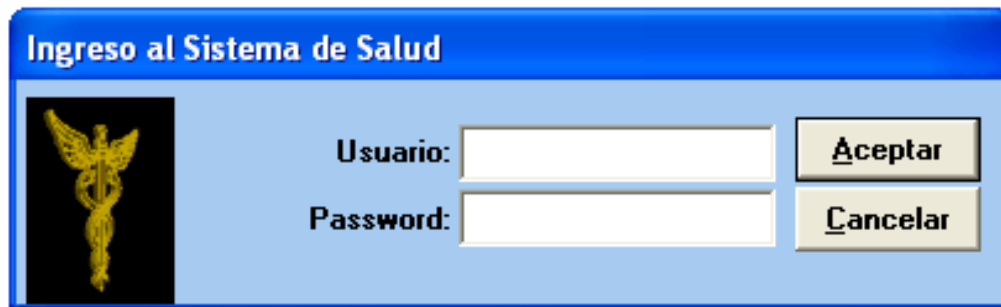


Fig. 5.2 Autenticación del Sistema de Identificación Digital de Pacientes

Los usuarios autorizados para ingresar al sistema son los que laboran en los siguientes establecimientos:

- ✓ Recepción
- ✓ Consultorios
- ✓ Laboratorio
- ✓ Farmacia

Cada usuario tiene asignado un nombre y una contraseña, con los cuales podrá tener acceso únicamente a las operaciones que le corresponden, restringiendo el ingreso a procedimientos no autorizados. Para efectos demostrativos en este



sistema se han fijado los siguientes nombres de usuario con sus respectivas contraseñas:

- **Recepción:** Departamento donde se efectúa la identificación de los pacientes y la asignación de médicos.

Nombre de usuario: RECEPCION

Contraseña: RECEPCION

- **Consultorio:** Departamento donde se lleva a cabo la revisión médica por parte de los internistas.

Nombre de usuario: CONSULTORIO

Contraseña: CONSULTORIO

- **Laboratorio:** Departamento donde se realizan los exámenes a los pacientes previamente autorizados por el médico.

Nombre de usuario: LABORATORIO

Contraseña: LABORATORIO

- **Farmacia:** Departamento donde se hace entrega de los medicamentos prescritos por el médico.

Nombre de usuario: FARMACIA

Contraseña: FARMACIA

Si el nombre de usuario y contraseña no pertenecen a los usuarios autorizados para el uso de este sistema se desplegará el siguiente mensaje de error (Fig. 5.3).

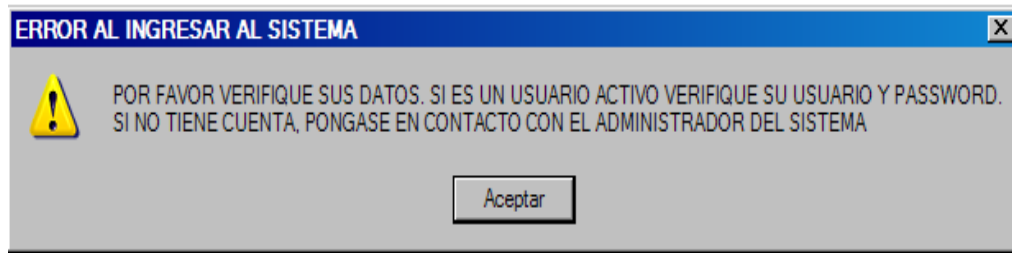


Fig. 5.3 Ventana de mensaje de error de ingreso al sistema

5.3.1.3 Procedimiento N° 3. Operación.

Como siguiente paso está la operación básica del sistema. Cada departamento posee diferentes funciones, por lo que esta aplicación presenta en pantalla únicamente las opciones para las cuales está autorizado cada usuario, es decir, el usuario tendrá acceso exclusivamente a ciertas operaciones acorde al trabajo delegado.

Sin embargo antes de entrar en detalle en la operatividad del sistema, es necesario describir opciones generales utilizadas en cada pantalla. A continuación se menciona cada una de ellas con sus respectivas funciones:

Barra de herramientas: se utiliza para actualizar registros. Se muestra en la mayoría de pantallas del sistema (Fig. 5.4).

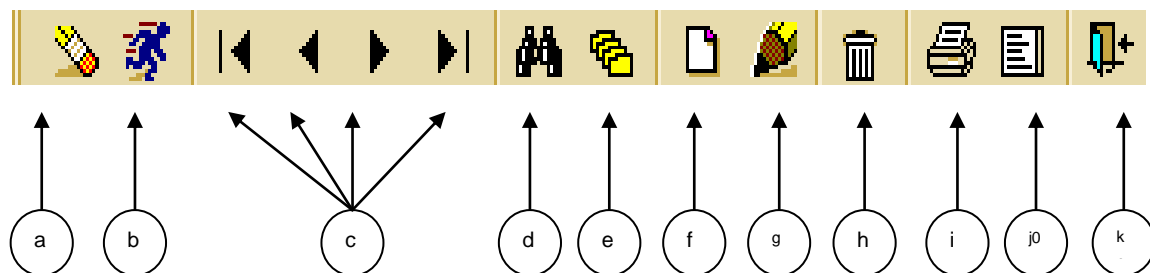


Fig. 5.4 Barra de herramientas para actualizar registros



- a) Se utiliza para limpiar los criterios de búsqueda en la barra de filtro (Fig. 5.5).
- b) Se utiliza para actualizar los datos con los criterios de búsqueda que se han ingresado en la barra de filtro (Fig. 5.5), en caso de no llenarse ningún campo la búsqueda proporcionará como resultado la lista de todos los pacientes localizados en la base de datos.
- c) Flechas de desplazamiento utilizadas para desplazarse de un registro a otro. Desplazamiento al primer registro, al anterior registro, al siguiente registro y al último registro respectivamente.
- d) Botón para realizar una búsqueda más avanzada.
- e) Ordenar alfabéticamente los registros de acuerdo a un campo seleccionado.
- f) Crea un nuevo registro (por Ej.: expediente, diagnóstico).
- g) Modifica un registro.
- h) Elimina un registro.
- i) Imprimir un reporte de los datos mostrados en la pantalla.
- j) Imprime el listado que se ha mostrado en la pantalla.
- k) Salida de todo reporte o del sistema.

Barra de filtro: La barra de filtro (Fig. 5.5) se utiliza como criterio de búsqueda dentro de una pantalla. Para efectuar la búsqueda debe llenarse los campos en blanco especificando el registro a buscar, ya sea realizando la exploración por código, nombre, fecha de consulta o médico. Luego se presiona la tecla <TAB> y

subsiguientemente se oprime el botón de actualización (Fig. 5.4.b) para ejecutar la búsqueda.

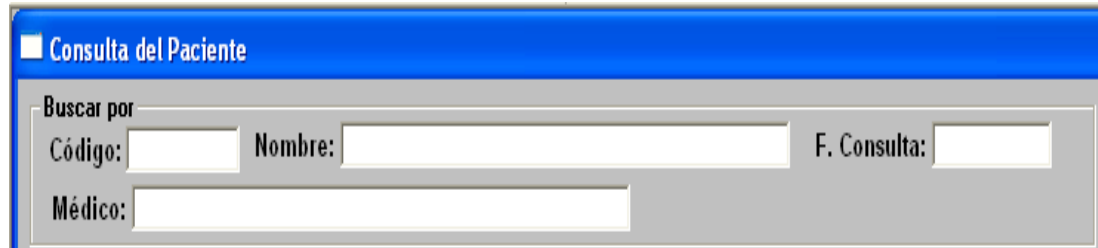


Fig. 5.5 Barra de filtro para efectuar búsquedas de campo

Barra de Herramientas para impresión de reporte: Envía la información desplegada en el reporte en gestión o pantalla, a la impresora predeterminada del sistema (Fig. 5.6).

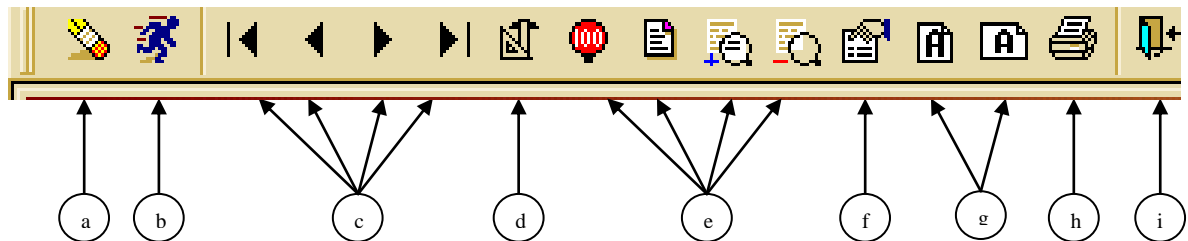


Fig. 5.6 Barra de filtro para efectuar búsquedas de campo

- Se utiliza para limpiar los criterios de búsqueda en la barra de filtro (Fig. 5.5).
- Se utiliza para actualizar los datos con los criterios de búsqueda que se han ingresado en la barra de filtro (Fig. 5.5), además actualiza los datos que se encuentran en el detalle del reporte y el encabezado.

- c) Botones de desplazamiento para moverse dentro de las páginas del reporte según convenga.
- d) Muestra una regla en el reporte.
- e) Eligen el tamaño de visualización del reporte en pantalla: visualización del 100%, visualización en pantalla completa o escalas de visualización.
- f) Cambia las propiedades de la página del reporte.
- g) Elige la orientación de visualización del reporte: horizontal y vertical.
- h) Imprime el reporte.
- i) Salir del reporte

RECEPCIÓN

El personal de recepción tendrá acceso a la pantalla que se muestra en la fig. 5.7, los campos disponibles para el recepcionista son el de Expediente y Consulta que se encuentran ubicados en el menú de operaciones de la barra de herramientas.

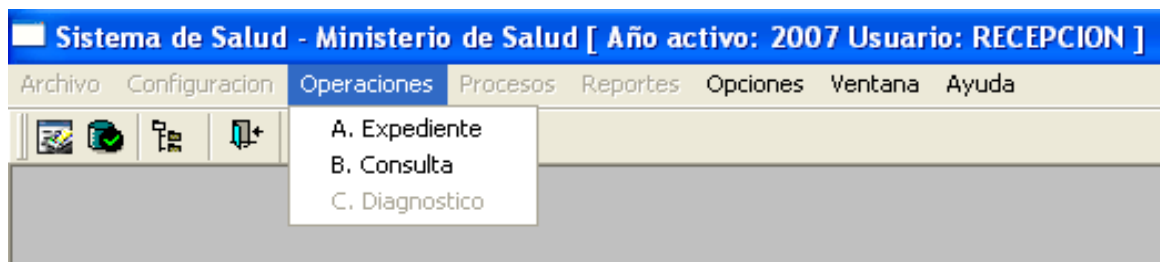
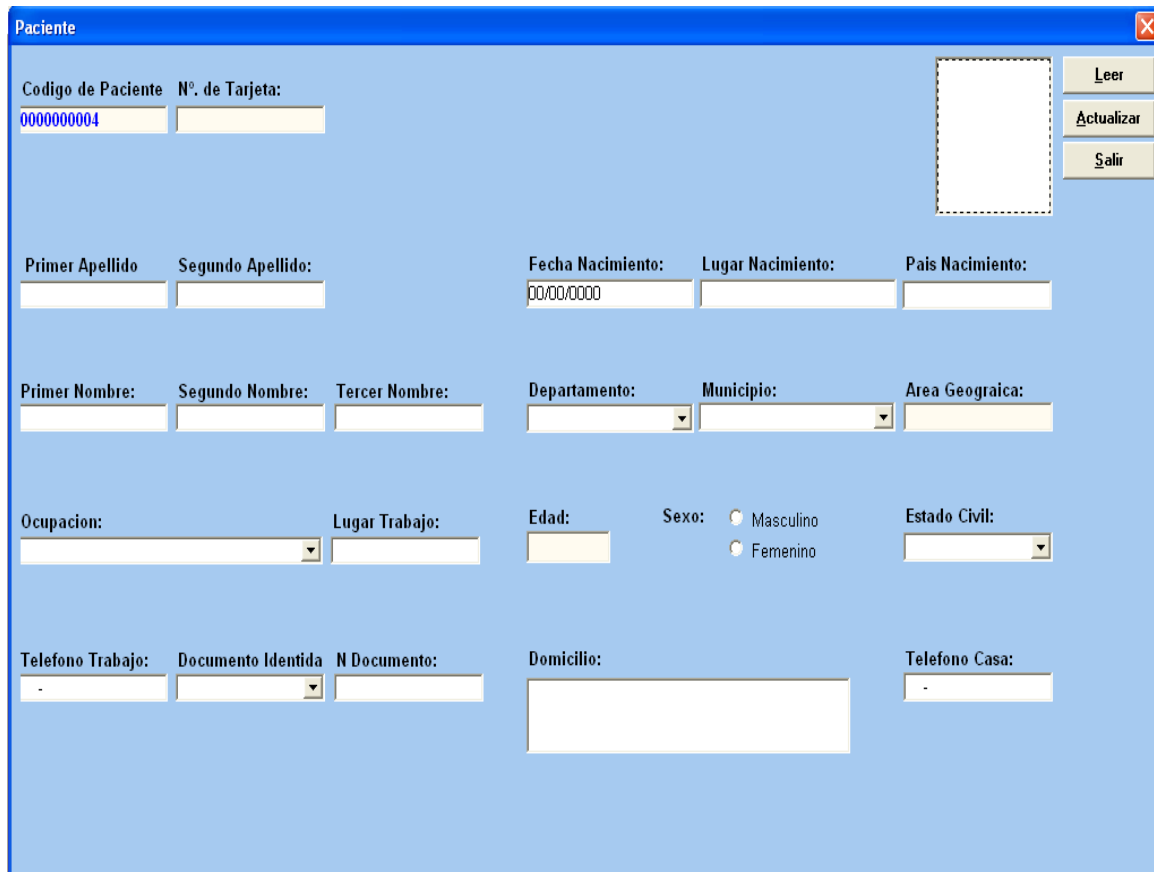


Fig. 5.7 Menú principal para el personal de recepción

- **Expediente:** En esta opción se tiene acceso al expediente médico que será asignado a cada persona (Fig. 5.8). Este formulario consta de campos en donde se introducirá la información básica de cada paciente, por ejemplo: el nombre, dirección, fecha de nacimiento, entre otros.

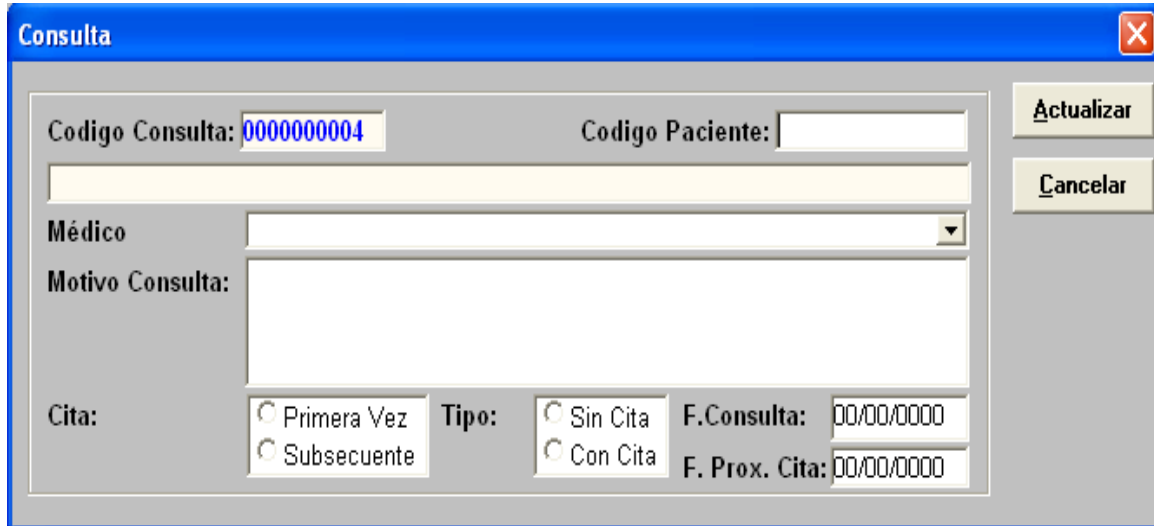
Todos los campos aparecerán vacíos cuando es la primera vez que el paciente visita el centro hospitalario. Una vez se haya introducido toda la información se presiona el botón actualizar y es de esta forma como se crea el expediente clínico que posteriormente es almacenado en la base de datos del hospital.



5.8 Formulario del expediente clínico del paciente.

- **Consulta:** En esta opción se lleva el control de consultas del hospital, asignándose un número correlativo o código cada vez que un paciente solicita una consulta, además se asigna el médico que atenderá al paciente dependiendo del motivo de su visita.

Esta ventana (Fig. 5.9) posee un campo vacío donde es posible describir algunos de los síntomas de las enfermedades que los usuarios presentan en su visita al hospital, previo a la revisión médica del internista. También es posible visualizar si el paciente posee una cita previa y de ser necesario fijar la próxima cuando el caso lo amerite.

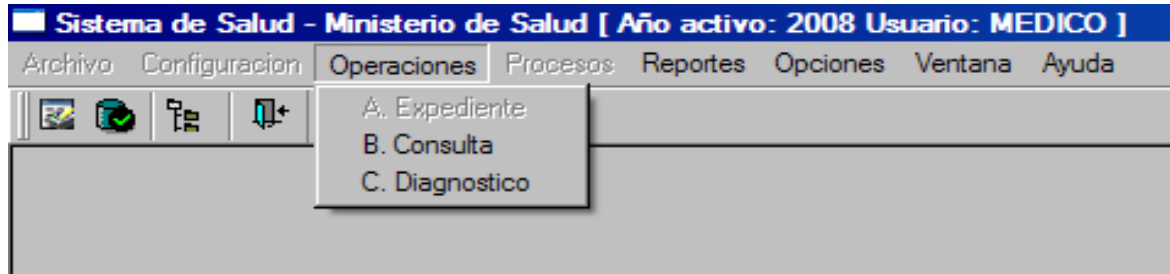


5.9 Formulario de consulta de paciente

CONSULTORIO

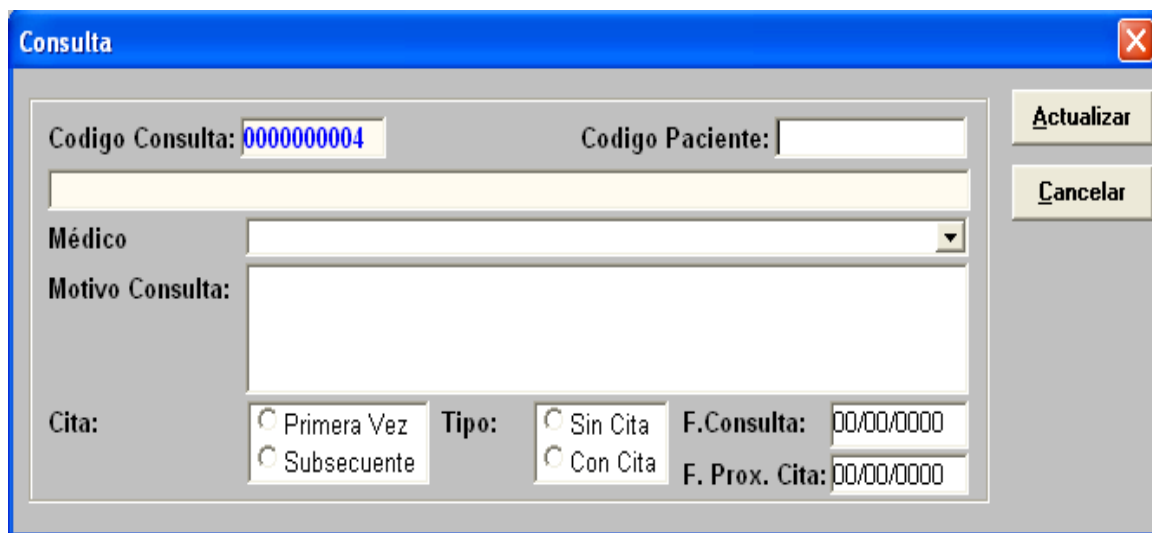
Los médicos ubicados en los consultorios, después de su identificación ante el SIDP, tendrán acceso a la pantalla que se muestra en la fig. 5.10. Los campos

disponibles para el doctor en el menú de operaciones son el de Consulta y Diagnóstico.



5.10 Formulario de consulta de paciente

- **Consulta:** En esta opción el médico tendrá la facultad de visualizar previamente los nombres de los pacientes que le corresponde, ver su expediente y enterarse de los síntomas que el paciente presenta en su visita al centro médico (Fig. 5.11).



5.11 Formulario de consulta de paciente

- **Diagnóstico:** en esta opción el médico confirma el diagnóstico posterior a la revisión médica. En esta ventana (Fig. 5.12) el internista identifica la enfermedad del paciente, determina los exámenes a realizar y asigna el medicamento de acuerdo al padecimiento del solicitante.

Diagnostico ✖

Código Consulta: Código Cita: Fecha Cita:

Paciente: **EDGARD ADONAY AGUIRRE LOPEZ**

Médico:

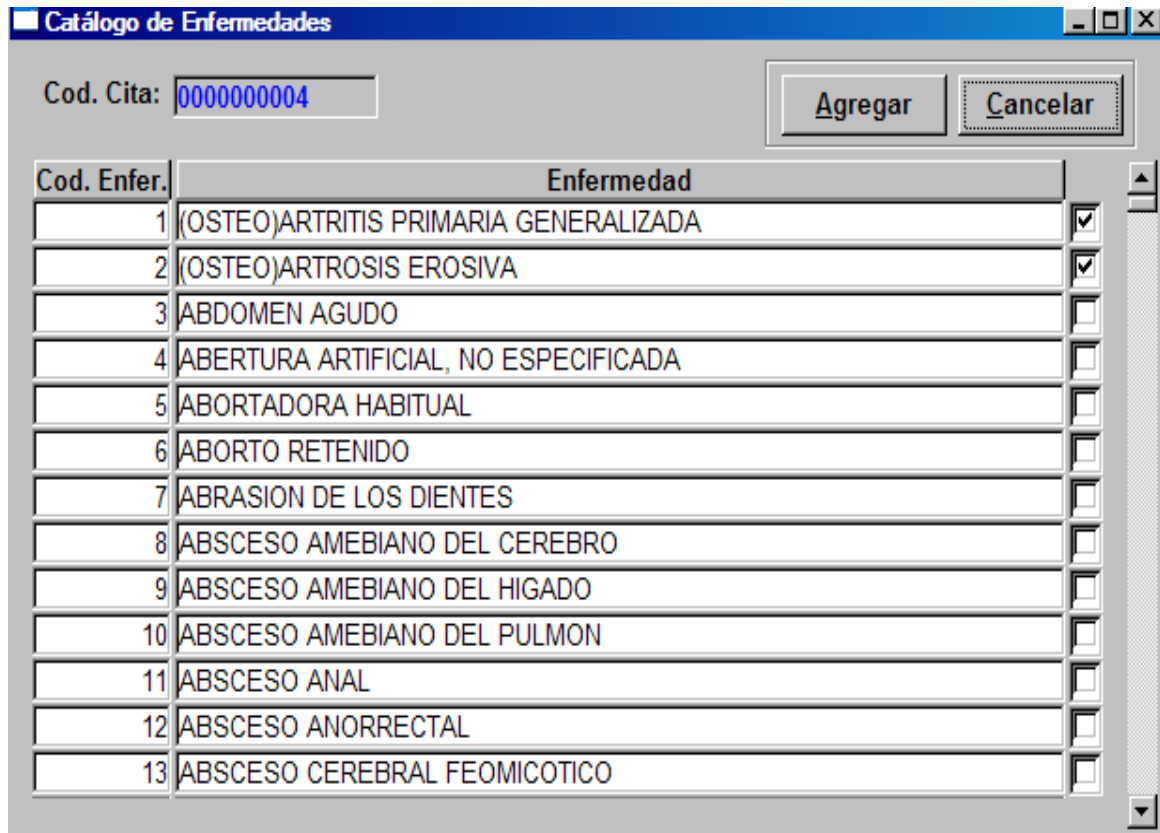
Cod.Enfermedad	Enfermedad
1	(OSTEO)ARTRITIS PRIMARIA GENERALIZADA
2	(OSTEO)ARTROSIS EROSIVA

Id Examen	Examen	Resultado
1	A. C. T. H. (HORMONA ADRENOCORTICOTROPICA)	NEGATIVO
5	ACIDO FÓLICO (FOLATO)	POSITIVO

Medicamentos Asignados				
Cod. Medicamento	Nombre Medicamento	Composición	Dosis	Cantidad
20-01-01015	5-FLUOROURACILO	SOL. INYECTABLE	3 ML C/DIA	15
01-01-01002	ACETAMINOFEN	JARABE 120MG/5M	1 CHTDA C/8 HORAS	10

5.12 Pantalla de Diagnóstico

Dentro de la pantalla de diagnóstico el doctor estará en la facultad de agregar el tipo de enfermedad diagnosticada al paciente, para ello el internista tendrá a disposición un catálogo de enfermedades (Fig. 5.13).



Catálogo de Enfermedades

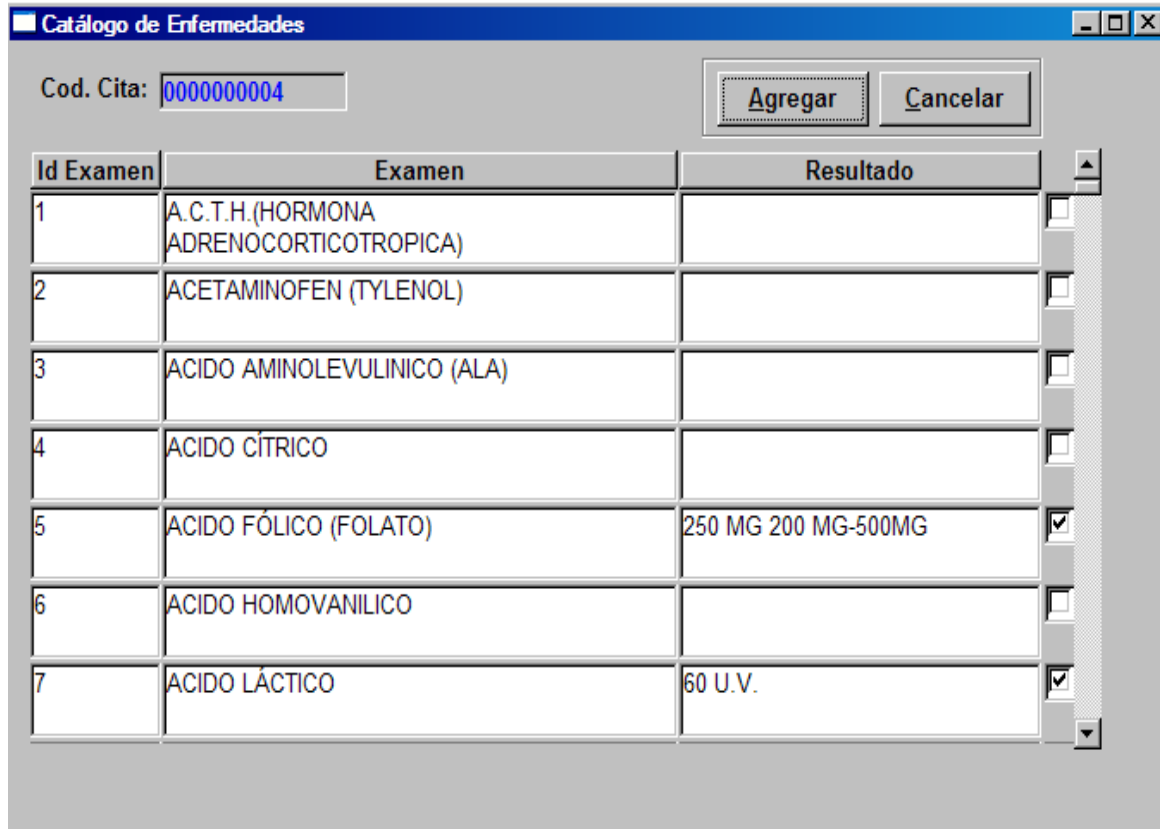
Cod. Cita:

Cod. Enfer.	Enfermedad	
1	(OSTEO)ARTRITIS PRIMARIA GENERALIZADA	<input checked="" type="checkbox"/>
2	(OSTEO)ARTROSIS EROSIVA	<input checked="" type="checkbox"/>
3	ABDOMEN AGUDO	<input type="checkbox"/>
4	ABERTURA ARTIFICIAL, NO ESPECIFICADA	<input type="checkbox"/>
5	ABORTADORA HABITUAL	<input type="checkbox"/>
6	ABORTO RETENIDO	<input type="checkbox"/>
7	ABRASION DE LOS DIENTES	<input type="checkbox"/>
8	ABSCESO AMEBIANO DEL CEREBRO	<input type="checkbox"/>
9	ABSCESO AMEBIANO DEL HIGADO	<input type="checkbox"/>
10	ABSCESO AMEBIANO DEL PULMON	<input type="checkbox"/>
11	ABSCESO ANAL	<input type="checkbox"/>
12	ABSCESO ANORRECTAL	<input type="checkbox"/>
13	ABSCESO CEREBRAL FEOMICOTICO	<input type="checkbox"/>

5.13 Catálogo de Enfermedades

Además el médico podrá prescribir los exámenes que él estime necesarios de acuerdo a los sistematología del paciente.

En el formulario de diagnóstico (Fig. 5.12), el doctor ingresa a un catálogo donde se visualiza una lista de todos los exámenes que tiene a su disposición en el hospital (Fig. 14), así como un campo donde es posible redactar observaciones a los resultados de los exámenes ya realizados



Id Examen	Examen	Resultado
1	A.C.T.H.(HORMONA ADRENOCORTICOTROPICA)	
2	ACETAMINOFEN (TYLENOL)	
3	ACIDO AMINOLEVULINICO (ALA)	
4	ACIDO CÍTRICO	
5	ACIDO FÓLICO (FOLATO)	250 MG 200 MG-500MG
6	ACIDO HOMOVANILICO	
7	ACIDO LÁCTICO	60 U.V.

5.14 Catálogo de Exámenes

En la lista de medicamentos únicamente aparecen en detalle las medicinas disponibles en el departamento de farmacia (Fig. 5.15). El médico podrá visualizar directamente el inventario del almacén, es decir, el nombre del medicamento, así como la cantidad en existencia.

En esta ventana el doctor tendrá la facultad de escribir la dosis del medicamento recetado y la composición de este. Una vez recetada, la orden es enviada al personal ubicado en el departamento de farmacia, y automáticamente se resta de la cantidad total almacenada.



Cod Medica.	Nombre Medicamento	Composición	Dosis	Stock	Cantida
20-01-01015	5-FLUOROURACILO	SOL. INYECTABLE 50MG	1 CADA 24 HORAS	85	15
10-06-06001	ACEITE MINERAL	EMULSION		100	0
01-01-01002	ACETAMINOFEN	JARABE 120MG/5ML	1 CDTA CADA 4 HORAS	90	10
01-01-01037	ACETAMINOFEN	SUPOSITORIO 150-300MG		85	0
01-01-01003	ACETAMINOFEN	GOTAS 60 MG		100	0

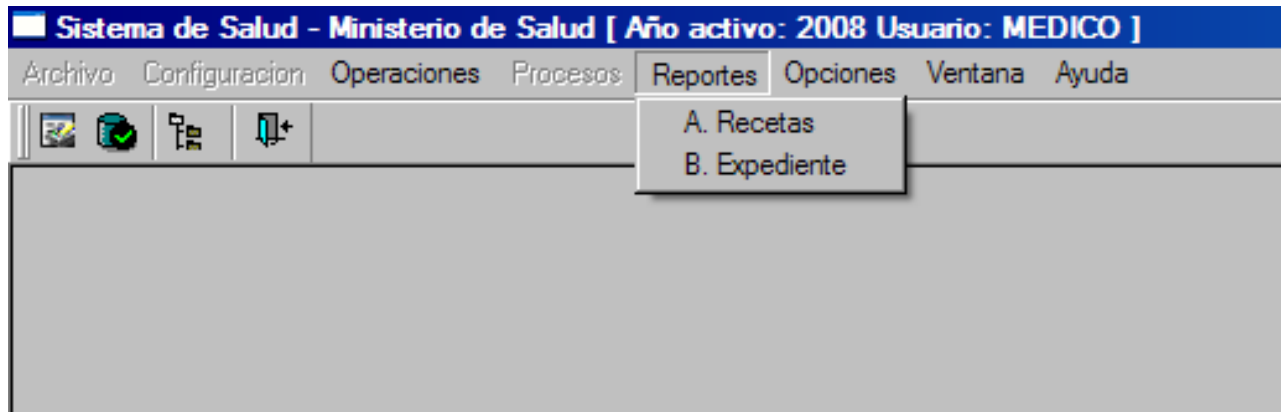
5.15 Catalogo de Medicamentos

En esta pantalla se receta el nombre del medicamento, la composición, la dosis y la cantidad a otorgar. Esta orden es enviada al departamento de farmacia para su posterior reclamo.

REPORTES

El modulo de reportes presenta todos los reportes que se encuentran en operación en el sistema (Fig. 5.16). Consta de dos opciones:

- ✓ Receta
- ✓ Historial Clínico



5.16 Módulo de reportes del sistema

- **Recetas:** Este menú se presenta como una alternativa en caso de no existir medicamento en el almacén de farmacia del hospital.

El médico ingresa a esta opción y visualiza el formato con la receta prescrita (Fig. 5.17). La receta muestra el número de receta, el número de cita, y el nombre del paciente. Además describe la dosis y el médico responsable de la orden.

Una vez prescrita, se imprime la receta en papel para que el paciente pueda adquirirla en otro establecimiento farmacéutico.



MINISTERIO DE SALUD
Servicio Médico Hospitalario
RECETA MEDICA A DESPACHAR No. 16

Código Departamento: 01

Cita: 0000000001

Nombre usuario: EDGARD ADONAI AGUIRRE LOPEZ

Nº Afiliación: 0000000002

Medicamento	Cantidad
01-01-01001 ACETAMINOFENTAB. 500 MG	12

(PARA USO EXCLUSIVO DEL MEDICO)

(F.) Dr. _____

Nombre Dr: MFA - DINA MAGDALENA PINEDA HIDALGO

Código del Dr: 01012

Fecha de emisión: 18/11/2007

5.17 Formato de receta impresa

- **Historial Clínico:** En este menú el médico tiene la facultad de revisar el historial clínico del paciente desde su primera visita al hospital (Fig. 5.18).

Historial Clínico

Paciente: EDGARD ADONAY AGUIRRE LOPEZ

Cod.Enfermedad	Enfermedad
Código de la cita: 000000004 Fecha Cita: 13/01/2008 Médico: MES - ROBERTO ARTURO VALDIVIESO LOPEZ	
2	(OSTEO)ARTROSIS EROSIVA
1	(OSTEO)ARTRITIS PRIMARIA GENERALIZADA

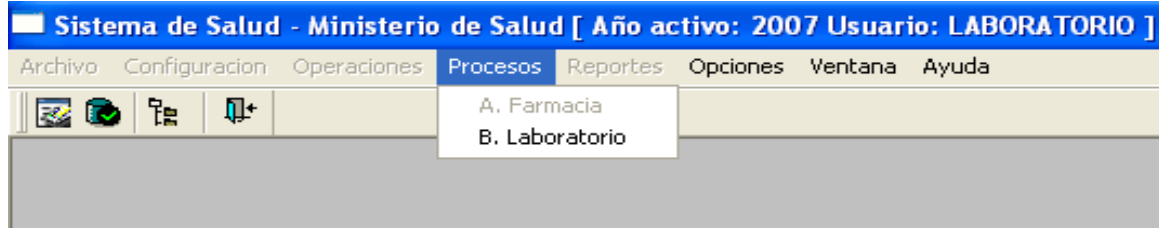
Id Examen	Examen	Resultado
Código de la cita: 000000004 Fecha Cita: 13/01/2008 Médico: MES - ROBERTO ARTURO VALDIVIESO LOPEZ		
1	A.C.T.H.(HORMONA ADRENOCORTICOTROPICA)	NEGATIVO
5	ACIDO FÓLICO (FOLATO)	POSITIVO
358	PROTEÍNAS	DEFICIENTE

Medicamentos Asignados				
Cod. Medicamento	Nombre Medicamento	Composición	Dosis	Cantidad
Código de la cita: 000000004 Fecha Cita: 13/01/2008 Médico: MES - ROBERTO ARTURO VALDIVIESO LOPEZ				
01-01-01002	ACETAMINOFEN	JARABE 120MG/5M	1 CHTDA C/8 HORAS	10
20-01-01015	5-FLUOROURACILO	SOL. INYECTABLE	3 ML C/DIA	15
16-01-01022	INSULINA	CARTUCHO 100 UI	1 C/TRES DIAS	7

5.18 Historial clínico del paciente

LABORATORIO

El personal de laboratorio al iniciar el SIDP, luego de identificarse ante el sistema, solamente tendrá acceso dentro del menú de procesos, a la opción de laboratorio. Desplegando en pantalla la ventana que se muestra en la fig. 5. 19.



5.19 Menú principal para el personal de laboratorio

Una vez se ingrese al menú de laboratorio, el personal de este departamento visualizará en pantalla (Fig. 5.20), una lista de las órdenes de exámenes enviadas por su respectivo médico, posterior a la revisión médica. Esta pantalla identifica el nombre del paciente y el tipo de prueba a realizarse.



5.20 Ventana de visualización de lista de exámenes.

FARMACIA

Luego de darse de alta frente al sistema se tendrá acceso al menú que se muestra en la fig. 5.21.

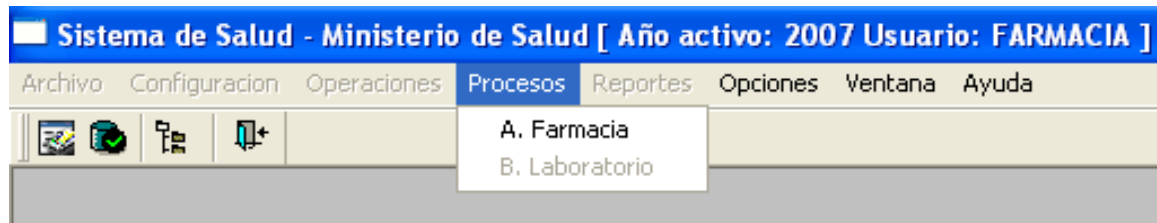


Fig. 5.21 Menú principal para el personal de laboratorio

En este departamento se hará entrega de los medicamentos correspondientes a cada paciente, el sistema recibirá la orden del médico y será enviada vía red, permitiendo al personal de farmacia visualizar la orden en pantalla como se muestra en la fig. 5.22.



5.22 Ventana de visualización de lista de entrega de medicamento

En esta pantalla es posible observar la lista de los medicamentos por paciente, el médico que receta la medicina, el nombre de la medicina y la cantidad a entregar. Después de ser atendido el paciente, el personal de farmacia determina que el paciente ha sido despachado.

5.3.2 PAGINA WEB

Como una aplicación adicional se ha desarrollado un entorno Web, en donde el usuario tendrá acceso a un sitio en Internet que proporciona: Información general del hospital, guía de servicios, consulta de citas y medios de contacto con la institución.

Es de esta manera como el paciente puede verificar la fecha de su próxima visita y enterarse de los servicios que la institución presta. En la fig. 5.23 se muestra la pantalla de inicio del sitio con sus respectivos links.

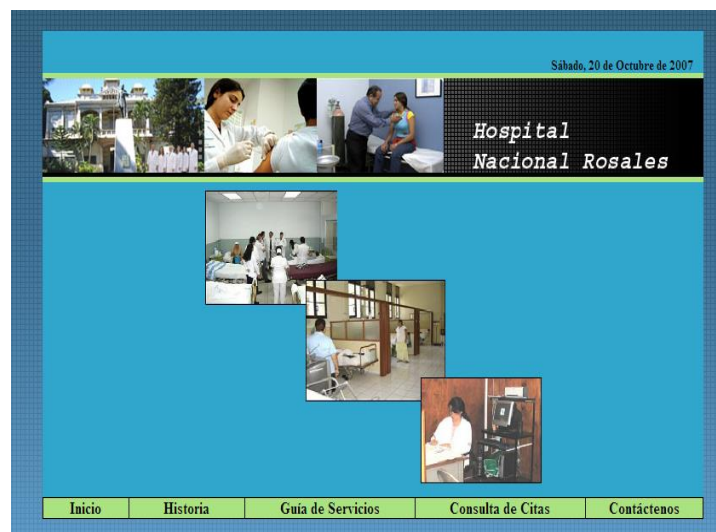


Fig. 5.23 Página de inicio del sitio Web

En la siguiente figura (Fig. 5.24) se detalla brevemente la historia del hospital, desde su fundación hasta la actualidad



Fig. 5.24 Pagina de historia de fundación del Hospital Rosales

El hospital consta de varios servicios que el paciente puede visualizar en el sitio Web, entre ellos (Fig. 5.25):

- ✓ Cardiología
- ✓ Dermatología
- ✓ Cirugía
- ✓ Urología
- ✓ Ortopedia
- ✓ Reumatología, etc.



Fig. 5.25 Pagina Guía de Servicios del Hospital Rosales

Además el usuario puede realizar consultas de la fecha y hora de su próxima cita (Fig. 5.27) mediante el ingreso de su número de identificación personal (DUI) y el número de tarjeta (Fig. 5.26), otorgando así un nivel de privacidad al usuario.



Fig. 5.26 Pagina de ingreso a consulta de citas.



Fig. 5.27 Pagina de visualización de consulta de citas

El Hospital ofrece un link de contacto para obtener información por vía telefónica o correo electrónico (Fig. 5.28).

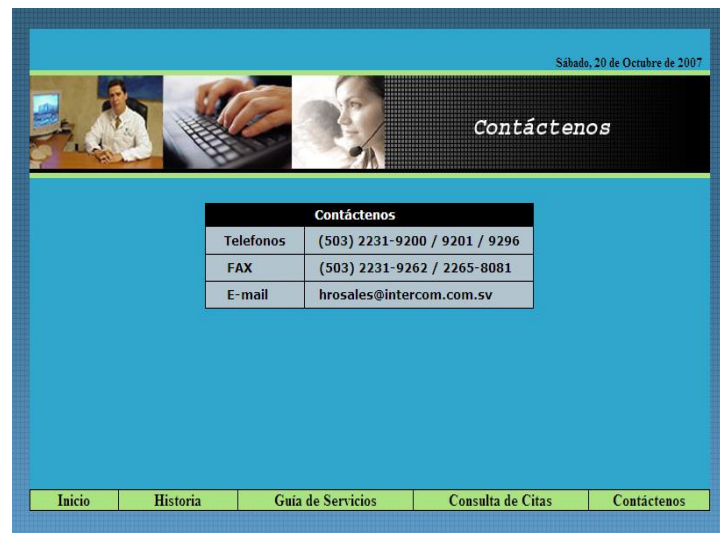


Fig. 5.28 Pagina de Contacto del Hospital Rosales



5.3.3 TABLAS COMPARATIVAS ENTRE SISTEMA ACTUAL Y SISTEMA PROPUESTO

CITA		
PROCEDIMIENTO	SISTEMA ACTUAL	SISTEMA PROPUESTO
Espera en cola. Entrada de Hospital	Solo a primera hora	Puede evitarse
Espera en cola. Departamento de Registro	20 min. – 25 min.	5 min. – 10 min.
Toma de datos. Departamento de Archivo	20 min. – 25 min.	-----
Espera del médico en sala. Área de Consulta	1 hora	1 hora
Chequeo Médico. Consultorio	15 min. – 20 min.	10 min. – 15 min.
Retiro de Medicamentos. Farmacia	15 min. – 20 min.	5 min. – 10 min.
Tiempo estimado	2 horas y 15 min. (Promedio aprox.)	1 hora y 20 min. (Promedio aprox.)

Tabla 17. Comparativa del consumo de tiempo entre el sistema actual y el sistema propuesto.⁹

⁹ Los tiempos han sido tomados un día de semana, dependen de la afluencia y época del año, así se ve incrementado o disminuido.



CRITERIOS	SISTEMA ACTUAL		SISTEMA PROPUESTO	
	Ventajas	Desventajas	Ventajas	Desventajas
Toma de datos en departamento de Registro		Se efectúa manualmente lo que implica mayor tiempo de ejecución.	Se efectúa digitalmente en un formulario electrónico, consumiendo menos tiempo de ejecución.	
Solicitud de expedientes en el departamento de Archivo		El paciente tiene que dirigirse a este departamento, ubicado en otra sección.	No es necesario que exista este departamento, la información esta almacenada en una base de datos central.	Por la cantidad de pacientes que generan un registro nuevo, es necesario equipos con alta capacidad de almacenamiento.
Manejo de la información del paciente (Expediente).		Los expedientes se almacenan en archiveros, exponiéndolos a factores ambientales (humedad, deterioro, etc.), y extravió de información.	El personal de registro al igual que los médicos tiene acceso directamente a los cuadros clínicos de los pacientes a través de una red.	Problemas de comunicación en la red pueden dificultar el acceso a expedientes.

Tabla 18. Comparativa de ventajas y desventajas entre el sistema actual y el sistema propuesto.



CRITERIOS	SISTEMA ACTUAL		SISTEMA PROPUESTO	
	VENTAJAS	DESVENTAJAS	VENTAJAS	DESVENTAJAS
Introducción de la información en el formulario (Expediente) por parte del médico interno durante la revisión médica.	El médico escribe directamente en el expediente clínico,		El sistema ofrece un ambiente amigable y ordenado para la introducción de datos, en el formulario de diagnóstico.	El médico diagnostica el nombre de enfermedades ingresando a menús para tener acceso a la base de datos del sistema.
Redacción de recetas.		llegibilidad en la escritura, o mala redacción de nombre de medicamentos	Al redactarse en formato digital, la receta impresa favorece la lectura por parte del paciente o personal de farmacia.	
Prescripción de exámenes.		El personal de laboratorio prepara equipo y recurso hasta que le paciente solicita el examen.	La orden es recibida inmediatamente desde el consultorio, agilizando la preparación de material para la realización de exámenes.	

Tabla 18. Comparativa de ventajas y desventajas entre el sistema actual y el sistema propuesto.



CRITERIOS	SISTEMA ACTUAL		SISTEMA PROPUESTO	
	VENTAJAS	DESVENTAJAS	VENTAJAS	DESVENTAJAS
Prescripción de medicamentos		Desconocimiento del medicamento existente en el almacén de farmacia.	Control del abastecimiento de medicina en el departamento de farmacia.	
Aceptación por parte personal laboral	Es un procedimiento tradicional.			Al ser un sistema estrictamente nuevo, facilita el rechazo por parte del personal antiguo, requiere capacitaciones.
Aceptación por parte del usuario	Es un procedimiento tradicional			Desconfianza a cambios, influencia cultural.

Tabla 18. Comparativa de ventajas y desventajas entre el sistema actual y el sistema propuesto.



CRITERIOS	SISTEMA ACTUAL		SISTEMA PROPUESTO	
	VENTAJAS	DESVENTAJAS	VENTAJAS	DESVENTAJAS
Consumo de tiempo total en el proceso de consulta (Eficiencia).		Por ser un proceso fundamentalmente manual, implica mayor consumo de tiempo en los subprocesos (Identificación, prescripción, búsqueda, etc.)	Los subprocesos se ven optimizados por el uso herramientas y equipos tecnológicos, disminuyendo el tiempo de realización.	
Mantenimiento	No es necesario debido a que la información es manejada en papel y almacenada en archiveros.		En el caso de la tecnología RFID, por sus características no requiere de un mantenimiento regular.	Es necesario un mantenimiento regular de equipos y software. (Depende de la tecnología utilizada).

Tabla 18. Comparativa de ventajas y desventajas entre el sistema actual y el sistema propuesto.

5.3.4 ESCRITURA EN TARJETAS RFID

Las tarjetas RFID son capaces de almacenar cierta cantidad de información, esta capacidad depende del tipo de tarjeta. En el caso de las tarjetas utilizadas en el SIDP son del modelo ISO 14443A, poseen una capacidad de memoria de 64K bytes, por lo que es posible almacenar en ellas información básica, como por ejemplo: Nombre, apellido, dirección, tipo de sangre, etc. (Fig.5.29).

Es posible usar estas tarjetas en diferentes servicios del hospital, para identificación de médicos, acceso a áreas restringidas, etc. Todo depende del sistema a evaluar y que permita la optimización a través del uso de tecnologías de identificación digital.

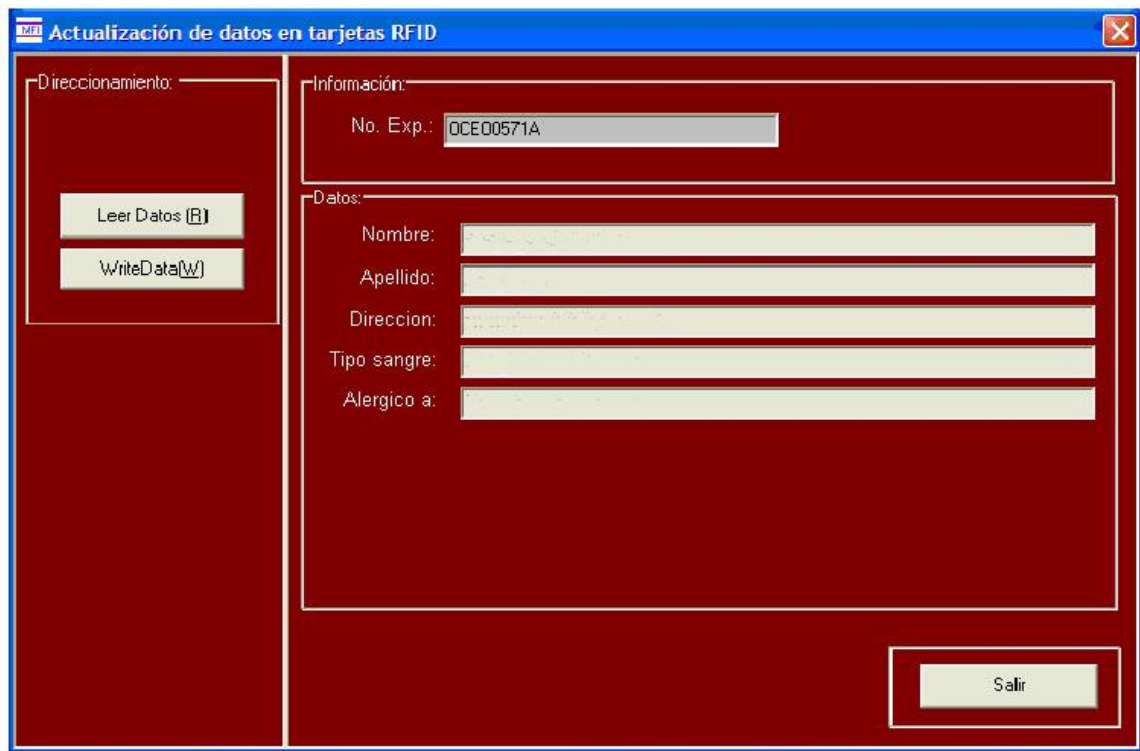


Fig. 5.29 Ventana de escritura en tarjetas RFID.



CAPITULO VI

MARCO POLITICO Y LEGAL DE LA PROTECCIÓN DE DATOS EN EL SALVADOR



6 CAPITULO VI

El marco político, legal y jurisprudencial en el cual se inserta una regulación específica de la protección de datos, en El Salvador, es favorable al nacimiento de este derecho, pese a que, en este momento, no se encuentra suficientemente protegido. En tal sentido, son rescatables los proyectos políticos de estímulo de la sociedad de la información los pronunciamientos jurisprudenciales existentes, que ya reconocen este derecho, y la preocupación social que se plasma en casos jurisprudenciales, donde la libertad informática es reclamada.

6.1 MARCO POLÍTICO

En El Salvador, la informática se considera una “palanca del proceso de modernización del país, en donde convergen las tecnologías de las telecomunicaciones, ciencias de la computación, las microelectrónica y otras; y como el factor que haga la diferencia en la competitividad de los diferentes actores de la vida nacional”¹⁰. El gobierno de El Salvador apuesta por el uso de nuevas tecnologías y su regulación, como se expresa en el documento *Estrategia Nacional de Gobierno Electrónico*, formulado por la Secretaría de la Presidencia de la República de El Salvador en febrero de 2004.

La Política Nacional de Informática, aprobada en junio de 2000¹¹, tiene, como uno de sus ejes fundamentales, el manejo y la administración de la información.

¹⁰ Reglamento Interno del Comité Nacional de Informática.

San Salvador, 1998.

¹¹ Política Nacional de Informática. Comité Nacional de Informática, San Salvador, julio de 2000.

<http://cni.org.sv/pni2000.html>



Entre sus objetivos incluye “promover el mantenimiento de la confidencialidad y privacidad de la información de las personas, así como la propiedad intelectual y la seguridad entre transacciones, bases de datos, e información empresarial e industrial”. La confidencialidad y la privacidad son claras referencias a la necesidad de un marco jurídico para la libertad informática. Ello se concreta en una línea de acción específica, la de “formular y presentar un proyecto de ley relativo a establecer un marco legal para proteger la confidencialidad y privacidad de la información de las personas en los medios electrónicos”.

6.2 MARCO LEGAL

El principal apoyo jurídico del derecho a la protección de datos se encuentra en el artículo 2 de la Constitución Nacional, que recoge la facultad que tiene toda persona para sustraerse de intromisiones extrañas, y esta expresado en el inciso segundo del mismo artículo: “se garantiza el derecho a honor, a la intimidad personal y familiar y a la propia imagen”

6.3 TRATAMIENTOS DE DATOS SOBRE SALUD

Los datos relativos a la salud son informaciones especialmente protegidas en las leyes de protección de datos. En el Convenio 108 del Consejo de Europa¹², se dispone que los datos relativos a la salud no podrán ser tratados de manera automática, a menos que el derecho interno prevea garantías apropiadas. Esto

¹² Art. 6 Convenio 108



establece la exigencia de una ley (principio de legalidad) para la tutela de la persona¹³.

6.3.1 FICHEROS DE HOSPITALES

Los datos de salud se entregan en hospitales y consultas hospitalarias y se emplean con diversos fines.

6.3.1.1 Tratamientos de datos sanitarios

Un dato sanitario es un dato sensible que, además, debe reunir las siguientes características:

Confidencialidad: esto es, el derecho del afectado a dar su consentimiento para la captura y tratamiento de estos datos.

Integridad: exactitud, actualidad y, en determinados casos, su disponibilidad.

El Consejo de Europa ha prestado atención particular a este tipo de datos en varias recomendaciones. El concepto de dato sanitario es amplio, pues comprende los datos presentes, futuros y pasados referentes a la salud física y mental. Además, en este sector, la elaboración de códigos deontológico resulta especialmente recomendable, pues completan y perfeccionan el conjunto normativo aplicable. En estas recomendaciones, se dispone que, si bien, en principio, los datos médicos deben tomarse de la persona interesada, también

¹³ Esta particularidad de los datos de salud y otros relativos al origen racial, las opiniones políticas, las convicciones religiosas u otras convicciones también son categorías de datos especialmente protegidos en la Directiva 95/46/CE de protección de datos.



podrán recopilarse de otras fuentes. Por lo tanto, si una legislación lo prevé, los datos médicos pueden recopilarse y tratarse, siempre que se haga para proteger la salud pública, prevenir un peligro concreto, reprimir una infracción penal determinada o para otros fines de interés público de importancia.

Una ley también podrá autorizar la recopilación y el tratamiento de datos para fines médicos preventivos, de diagnóstico o terapéuticos referidos a la persona interesada o de un pariente de línea genética, con el fin de salvaguardar intereses vitales de la persona interesada o de una tercera persona, respetar una obligación contractual específica o para fines de comprobación, ejercicio o defensa de un derecho judicialmente.

Los profesionales sanitarios, o los individuos u órganos que trabajen en representación de los mismos, son los únicos legitimados para recoger y procesar datos médicos. Cuando un paciente proporciona sus datos a un profesional sanitario para que este complete su ficha médica, en una consulta externa o ingreso hospitalario, puede exigir al médico información sobre lo que va a pasar con esos datos.

El paciente debe ser informado, de modo expreso, preciso e inequívoco, de que sus datos se van a recoger en un fichero, de la finalidad de recogida y de los destinatarios de los mismos. De igual forma, el paciente debe poder ejercitar los derechos de acceso, rectificación, cancelación y oposición, y para ello debe ser informado de la identidad del responsable del tratamiento y de la dirección del mismo.

En El Salvador, la desprotección del ciudadano o de la ciudadana en lo que respecta a sus datos sanitarios es enorme, y para las potenciales agresiones a la autodeterminación del individuo no se cuenta con ningún tipo de previsión legal.



Se insta, desde aquí, a que el legislador aborde cuanto antes esta cuestión, teniendo en cuenta el carácter sensible que tienen los datos de la salud.

La implantación de nuevas tecnologías en El Salvador es una prioridad gubernamental, porque de ello depende el progreso social y la competitividad de su economía. Si la información constituye la base de cualquier decisión, la capacidad de acceder a la misma es un elemento importante para la educación, la libre formación de la personalidad y el desarrollo de las relaciones sociales, políticas y jurídicas que sustentan las comunidades democráticas. De hecho, la política de Ciencia y Tecnología se centra fundamentalmente en el acceso a las TIC (Informática y Comunicaciones) como herramienta que posibilita “el conocimiento, la información y la comunicación fundamentales para el progreso y el bienestar de la humanidad”¹⁴.

En nuestro país, el derecho de acceso a la información está resguardado y/o garantizado no por una, sino por diversas disposiciones que coinciden en los objetivos y protección del contenido de acceso a la información. En este sentido, es posible que se relacione el acceso a la información con el derecho contenido en el artículo 6 de la Constitución Salvadoreña. Por otra parte, el derecho de acceso a la información está sustentado también en el artículo 2 de la Constitución salvadoreña, en tanto que el escrutinio y la disposición de la información en las instituciones públicas debe ser del conocimiento de la ciudadanía, especialmente cuando esta información la pueden conocer terceros, es decir, su manejo “podría incidir en la esfera de su intimidad personal y familiar y en la propia imagen”.

¹⁴ CONACYT, “Fortalezcamos la Ciencia y la Tecnología en EL Salvador para ayudar a cerrar la brecha en Educación y Tecnología”. San Salvador, jueves 18 de septiembre de 2003.



Asimismo se debe señalar que el acceso a la información es la concreción y el objetivo último de los derechos que permiten incitar a los organismos públicos en la obtención de una respuesta, de un dato o para proporcionar información sobre determinados aspectos que son el conocimiento de las instituciones. De ahí que cuando se tiene acceso a la información, su contenido este protegido o tutelado a partir de la Constitución Salvadoreña.



7 CONCLUSIONES

- Actualmente El Salvador cuenta con recurso tecnológico a través de los centros de educación superior (Universidades), es importante que existan convenios educativos-gubernamentales que permitan la investigación, estudio e implementación de alternativas tecnológicas para la optimización de los servicios públicos
- Es necesario que exista una infraestructura de comunicación entre las diferentes instituciones de salud pública a nivel nacional, que permita la movilización de la información clínica desde centrales y de esta manera evitar la utilización de recursos tecnológico, humano y físico.
- Por medio de la combinación de recursos técnicos y a través de un análisis financiero es posible diseñar sistemas automatizados que se adecuen a las necesidades de las instituciones públicas de salud, haciendo posible mejorar el proceso de los servicios ofrecidos a los pacientes.
- Las tecnologías están a la vanguardia, sin embargo, los dirigentes del país no gestionan en sus planes de gobierno metodologías de inversión que permitan mejorar los sistemas actuales.
- El sector Salud presenta problemas de administración y de organización y mientras siga existiendo políticas de no apertura ante propuestas de cambio, la salud pública en el país seguirá enfrentando complicaciones en el servicio a los pacientes promoviendo el subdesarrollo de la nación.
- El Salvador carece de un marco normativo general respecto al acceso a la información pública o privada y al régimen de protección de datos, pese a



que, en este país, existen numerosos registros públicos o privados, informáticos o manuales, que contienen información personal.

- Mediante las tecnologías de identificación se comprobó que se puede agilizar los diferentes procesos que existen en el ámbito de la salud, para garantizar un mejor servicio a los usuarios que hacen uso de este rubro.
- El uso de estas tecnologías requiere de una inversión inicial elevada, sin embargo el retorno de la inversión se verá reflejado en el beneficio que cada uno de los usuarios tendrán con la agilización del proceso.
- Analizando todas las necesidades que presenta el Hospital Nacional Rosales, se observó que cada una de estas se pueden solventar, desarrollando diferentes aplicaciones utilizando la tecnología RFID, dependiendo del servicio que este brinda en sus diferentes áreas.
- RFID es una tecnología que puede utilizarse en muchas aplicaciones de reconocimiento y sobre todo en aquellas donde el objetivo principal es la identificación individual de objetos o personas.
- Una ventaja que tiene la tecnología RFID es la de poder almacenar una gran cantidad de información en las tarjetas que esta tecnología utiliza, a la vez esta información puede ser modificada cuantas veces sea necesario.
- La tecnología RFID ofrece la facilidad de identificar objetos a largas distancias, sin importar los obstáculos que existan entre el lector y las etiquetas; siendo necesario revisar antes la factibilidad legal de la implementación en el mercado salvadoreño.



- Realizando una comparativa entre las diferentes tecnologías de reconocimiento, se observa que RFID posee una superioridad sobre estas debido a que la recolección de datos se realiza sin contacto directo o visibilidad de las etiquetas, lo que produce un bajo riesgo de obstrucción al no tener que insertar las tarjetas en algún tipo de ranura disminuyendo los costos de mantenimiento.
- Una de las características que diferencia a RFID sobre las demás tecnologías de reconocimiento es que esta posee un alto nivel de seguridad, ya que la información que se maneja en sus etiquetas puede ser cifrada para evitar, el uso inadecuado de la misma.
- Comparando todas las tecnologías de reconocimiento, se determino que RFID evita las falsificaciones, debido a que en otras tecnologías como por ejemplo con una simple fotocopia se puede clonar una etiqueta de identificación, no así los tags RFID.



8 RECOMENDACIONES

- Brindar capacitaciones al personal, que hará uso de este sistema, para evitar cualquier tipo de inconvenientes en el manejo de la información, en cada una de las áreas involucradas.
- Hacer conciencia a las personas que requieren de este servicio, en cuanto a la importancia y el cuidado que se debe tener con el manejo de las tarjetas que se les han asignado, para minimizar costos de operación en las que el hospital puede incurrir.
- Mantener al menos una persona, que brinde soporte al sistema para resolver los problemas de mayor envergadura y que no puedan ser solventados por el personal que lo utiliza, como también para que proporcione mantenimiento al equipo que conforma todo el sistema de identificación.
- Construir una infraestructura de red a nivel nacional para poder trabajar con una base de datos centralizada, para que los pacientes puedan asistir a cualquier centro hospitalario sin la necesidad, de crearles un nuevo expediente clínico cada vez que estos visiten un diferente centro de salud.
- Implementar un sistema de seguridad que proteja la integridad y la fiabilidad de la información de cada uno de los pacientes, que se maneja dentro del hospital.
- Instalar un sistema de redundancia y de respaldo de la información almacenada para evitar regresar al engorroso sistema actual y al mismo tiempo impedir la pérdida total de la información.



- Para mejorar el sistema propuesto, se pueden adquirir más dispositivos de reconocimiento e instalarlos en cada una de las áreas de consulta externa con la finalidad de agilizar aun más el proceso.
- Establecer convenios con empresas que financien la realización de proyectos de graduación buscando con esto que todos los proyectos a realizar tengan una aplicación práctica.



ANEXOS

9 ANEXOS

ANEXO Nº 1

DISTRIBUCION POBLACIONAL Y ESTABLECIMIENTOS





ANEXO N° 2

Hospital: Ilopango

Año = 2006 Meses = Enero – Diciembre

			Emergencias								
Preventivas y Curativas			Por Medico (a) ¹				Por Enfermería ²			Referidos	
Año	Consultas Medicas	Atenciones por Enfermeria	Total (1+2)	Total (a+b)	Pacientes MSPAS ^a	Pacientes ISSS ^b	Total (c+d)	Pacientes MSPAS ^c	Pacientes ISSS ^d	A otros establecimientos	De otros establecimientos
2006	84,846	12,063	40,426	40,426	40,426	0	0	0	0	1,950	4,564
Totales	84,846	12,063	40,426	40,426	40,426	0	0	0	0	1,950	4,564

Fuente:

Unidad de Información, Monitoreo y Evaluación. (MSPAS)

$$\frac{84,846 + 12,063}{240} + \frac{40,426}{365} = 514 \text{ pacientes atendidos diariamente.}$$



ANEXO N° 3

RVSLWDO1HXPROyJLFRU-RVp\$QWRQLR6DOGDxD´

Año = 2006 Meses = Enero – Diciembre

			Emergencias								
Preventivas y Curativas			Por Médico (a) ¹				Por Enfermería ²			Referidos	
Año	Consultas Médicas	Atenciones por Enfermería	Total (1+2)	Total (a+b)	Pacientes MSPAS ^a	Pacientes ISSS ^b	Total (c+d)	Pacientes MSPAS ^c	Pacientes ISSS ^d	a establecimientos	otros de establecimientos
06	24,074	0	27,845	27,845	27,845	0	0	0	0	1,083	2,683
Totales	24,074	0	27,845	27,845	27,845	0	0	0	0	1,083	2,683

Fuente:

Unidad de Información, Monitoreo y Evaluación. (MSPAS)

$$\frac{24,074}{240} + \frac{27,845}{365} = 176 \text{ pacientes atendidos diariamente.}$$



ANEXO N° 4

RVSLWDO(VSHFLDOL]DGR5RVDOHV´

Año = 2006 Meses = Enero – Diciembre

		Emergencias									
		Preventivas y Curativas		Por Médico (a) ¹				Por Enfermería ²		Referidos	
Año	Consultas Médicas	Atenciones por Enfermería	Total (1+2)	Total (a+b)	Pacientes MSPAS ^a	Pacientes ISSS ^b	Total (c+d)	Pacientes MSPAS ^c	Pacientes ISSS ^d	a otros establecimientos	de otros establecimientos
06	192,584	0	45,505	45,505	45,505	0	0	0	0	0	25,281
Totales	192,584	0	45,505	45,505	45,505	0	0	0	0	0	25,281

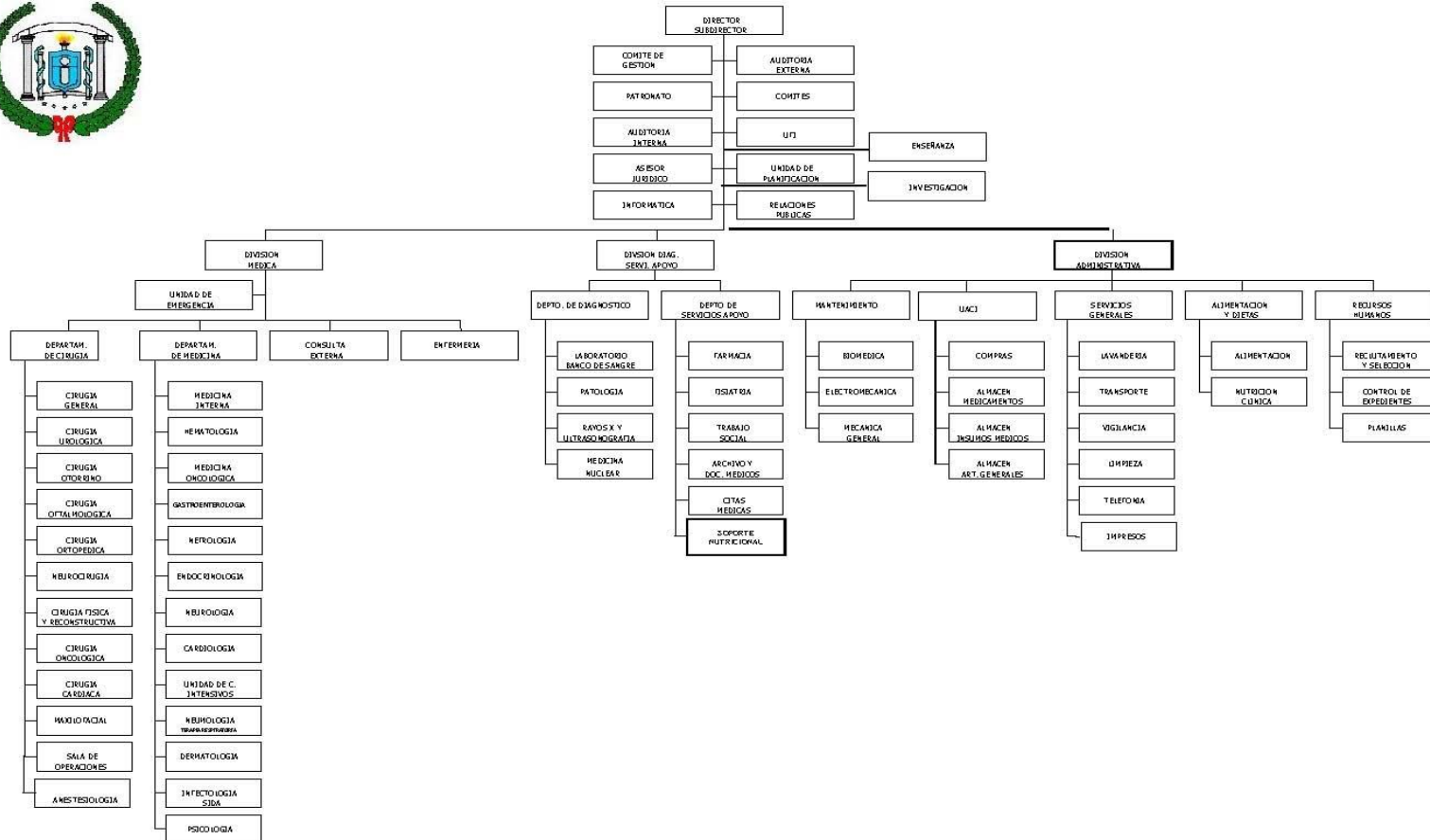
Fuente:

Unidad de Información, Monitoreo y Evaluación. (MSPAS)

$$\frac{192,584}{240} + \frac{45,505}{365} = 927 \text{ pacientes atendidos diariamente.}$$

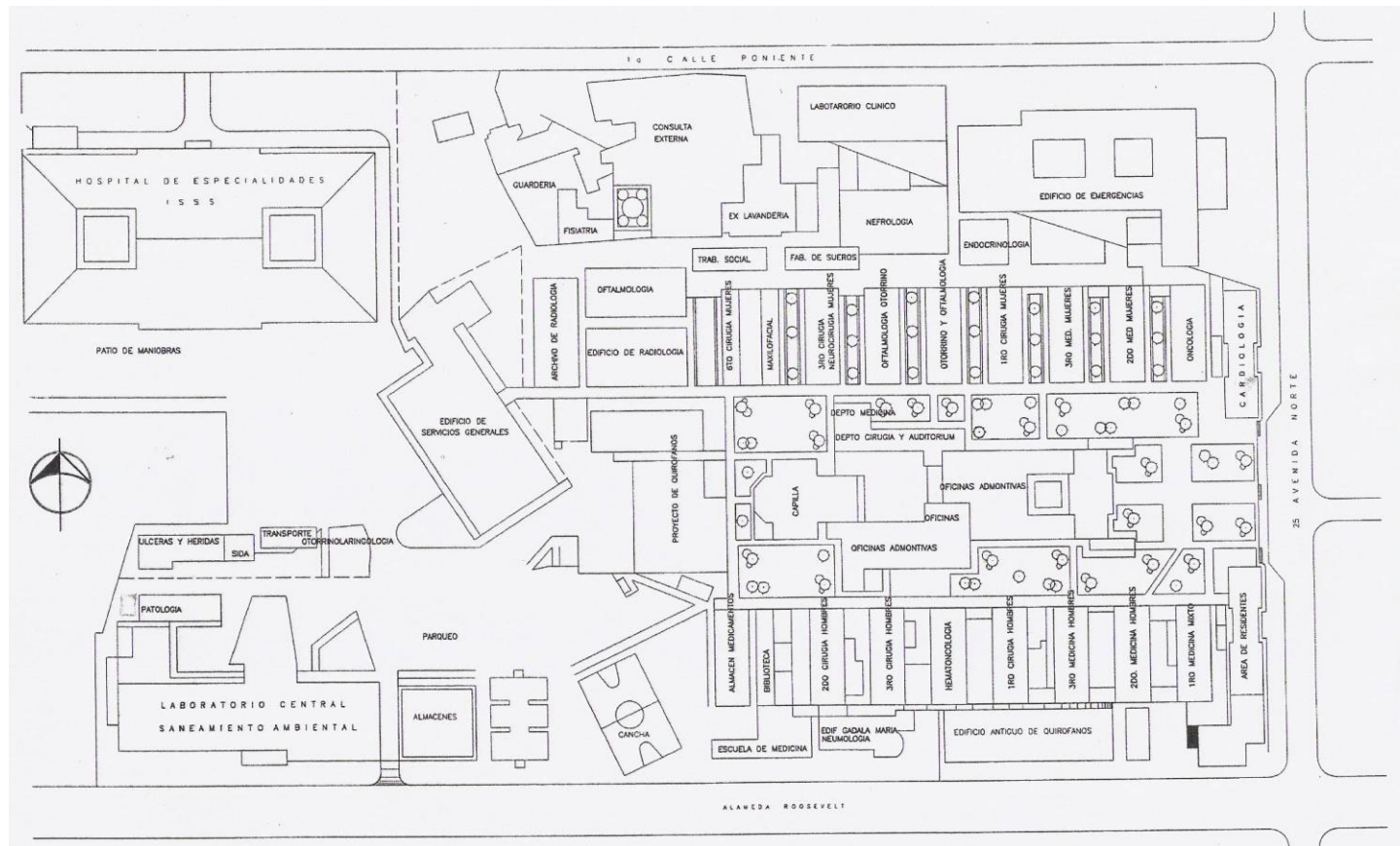
ANEXO Nº 5

ESTRUCTURA ORGANIZATIVA DEL HOSPITAL NACIONAL ESPECIALIZADO ROSALES



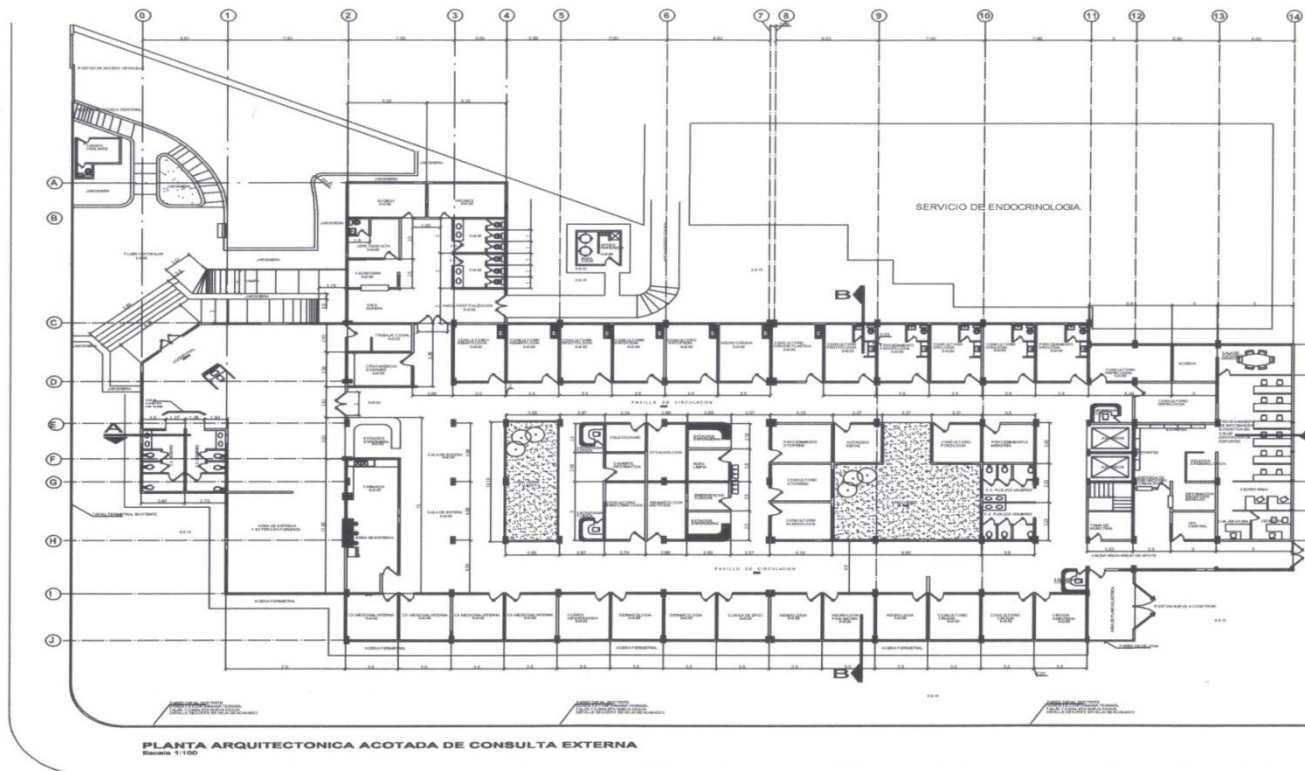
ANEXO Nº 6

HOSPITAL NACIONAL ESPECIALIZADO ROSALES, ESQUEMA ACTUAL DE UBICACIÓN DE EDIFICACIONES



ANEXO Nº 7

HOSPITAL NACIONAL ESPECIALIZADO ROSALES, NUEVA ÁREA DESTINADA A CONSULTA EXTERNA





ANEXO Nº 8

Universidad Don Bosco

Población: Personal medico administrativo que labora en el Hospital Nacional Especializado Rosales

Objetivo: Recopilar información general de las condiciones actuales en las que se labora en el hospital.

- ¿Sabemos que la mayoría de centros de salud, cuentan con un limitado presupuesto, es esta la situación en este centro?
- ¿Esta contemplado, alguna parte del presupuesto para capacitar el personal, si es "no" como gestionan esto?
- ¿Esta contemplado, alguna parte del presupuesto para compra de equipo, si es "no" como gestionan esto?
- ¿Que podría mencionar acerca del problema financiero, por qué se da este problema?
- ¿Piensa que existe algún tipo de problema con el recurso humano en el hospital de ser así que tipo de problema?
- Si la pregunta anterior es afirmativa ¿Cuál cree que sea la causa de que exista dicho problema?
- ¿Que tipo de conocimientos tienen sobre computadoras, las enfermeras, recepcionistas, personal de farmacia, que trabajan en el área de consulta?



- ¿Existen incentivos por parte del hospital para los empleados que sobresalen haciendo bien sus tareas?
- ¿Opina que los empleados están abiertos a cambios en lo que respecta al enfoque tecnológico, ven esto con buenos ojos o presentan algún tipo de temor?
- ¿Tiene el problema financiero relación alguna con el problema de recurso humano? ¿Por que?
- ¿Cuentan con computadoras en la institución?
- Si la pregunta anterior es afirmativa ¿En que área hacen uso de ellas?
- ¿Posee el hospital una red local?
- ¿Posee el hospital Internet?
- ¿Existe algún sistema para automatizar procesos dentro del hospital?
- ¿Para dar el servicio de consulta externa con que equipos cuenta el hospital?
- ¿Piensa que el recurso tecnológico, puede ayudar para realizar procesos mas eficientes de los que existen actualmente sean estos procesos consulta, entrega de medicamentos, búsqueda de expedientes, respuesta de exámenes, etc.?
- ¿Tiene el problema financiero relación alguna con el problema Tecnológico?
- ¿Qué tipos de problema infraestructural presenta el hospital?.

ANEXO Nº 9

LISTA DE EQUIPOS DE RED

Nuevo: La última tecnología

Comparar Productos



PowerEdge™ 2900
Servidor en torre de dos sockets

Desde
\$2,225
Precio sin IVA, gastos de envío excluidos
(salvo disposición en contrario)
Precio incluye descuentos

[Configurar y Cotizar](#)

[Detalles del PowerEdge 2900](#)

Procesador
Hasta 2 procesadores de Quad-Core Intel®
Xeon® serie 5300 o Dual-Core Intel® Xeon®
serie 5100

Memoria
Hasta 48 GB de memoria FBD SDRAM de 667
MHz con ECC

Almacenamiento
Unidades SAS o SATA II Hasta 3TB¹ de
capacidad máxima de almacenamiento interno

[Configurar y Cotizar](#)

[Detalles del PowerEdge 2900](#)

PowerEdge™ 2900
Servidor en torre de dos sockets

Comparar Productos



OptiPlex™ 320
Computadoras de escritorio
fáciles de usar para oficinas
en crecimiento.

Desde
\$799
Precio sin IVA, gastos de envío
excluidos (salvo disposición en
contrario)
Precio incluye descuentos




[Configurar y Cotizar](#)

[Más Detalles](#)

EXCELENTE PARA

- Usuarios empresariales e institucionales que necesitan funciones básicas de manejabilidad para una inversión inicial de bajo costo.

	<h2>PowerConnect 3448</h2> <h3>\$399</h3> <p>Precio sin IVA, gastos de envío excluidos (salvo disposición en contrario) Precio incluye descuentos</p> <hr/> <p> Imprimir resumen</p> <hr/> <p> Revisión del resumen</p> <p> Agregar a artículos guardados Compare, almacene y recupere sus configuraciones de sistema</p> <p> Agregar al carrito de compras</p>
<p>PowerConnect™ 3448 Switch Ethernet 10/100 de 48 Puertos + 2 Puertos Gigabit Ethernet</p> <p>Haga clic en el botón verde para últimos precios o llámenos.</p> <p> Configurar y Cotizar</p> <p> Detalles del PowerConnect 3448</p> <p>EXCELENTE PARA</p> <ul style="list-style-type: none"> • Switch con 48 puertos 10/100 Ethernet + 2 puertos Gigabit Ethernet 2 puertos Gbe de fibra vía transceptores SFP • Switch administrable flexible, de formato rack que cumple con el estandar CLI y L3/L4 • Compatible con la ultima tecnología de redes inalámbricas 	

<h2>CAT 6, 110 / KRN HYBRID</h2>										
<ul style="list-style-type: none"> • Standard Spacing for Mounting on 19" EIA Standard Racks. • 110 Shell, Krone Contact Hybrid Terminals • 110 Tool Can Be Used to Terminate • Gold plated Modular Jack Contacts. • Wiring: Color Coded for Both 568A & 568B Codes. • Port Label with Plastic Covers <table border="0"> <tr> <td>1K0-PP6-168</td> <td>16 Ports, 1U</td> </tr> <tr> <td>1K0-PP6-248</td> <td>24 Ports, 1U</td> </tr> <tr> <td>1K0-PP6-328</td> <td>32 Ports, 2U</td> </tr> <tr> <td>1K0-PP6-488</td> <td>48 Ports, 2U</td> </tr> </table>  <p>Back View with Port Number and Color Code Label</p>	1K0-PP6-168	16 Ports, 1U	1K0-PP6-248	24 Ports, 1U	1K0-PP6-328	32 Ports, 2U	1K0-PP6-488	48 Ports, 2U		
1K0-PP6-168	16 Ports, 1U									
1K0-PP6-248	24 Ports, 1U									
1K0-PP6-328	32 Ports, 2U									
1K0-PP6-488	48 Ports, 2U									
<p># 110-PP5E-488 RJ45 48 Ports 2U <input type="text" value="1"/> <input type="button" value="ORDER"/> \$110.00</p> <p># 110-PP5E-968 RJ45 96 Ports 4U <input type="text" value="1"/> <input type="button" value="ORDER"/> \$249.00</p>										

Wall Plates



Keystone Jack



RACK

19" EIA RACK-MOUNT - #10-32 TAPPED HOLES

Item Number Bolt-Down	Item Number Free-Standing	Rack Units (U) (in.)	Overall Dimensions		Inside Width (in.)	Hole-to-Hole Centers (in.)	Shipping Weight (lbs.)
			H (in.)	W (in.)			
MNF1942D*	FSF1942D*	24U 42.00	47.25	20.18	17.81	18.31	24
MNF1961D*	FSF1961D*	35U 61.25	66.50	20.18	17.81	18.31	28
MNF1970D*	FSF1970D*	40U 70.00	75.25	20.18	17.81	18.31	32
MNF1978D*	FSF1978D*	45U 78.75	84.00	20.18	17.81	18.31	34
MNF1989D*	FSF1989D*	51U 89.25	95.88	20.18	17.81	18.31	37

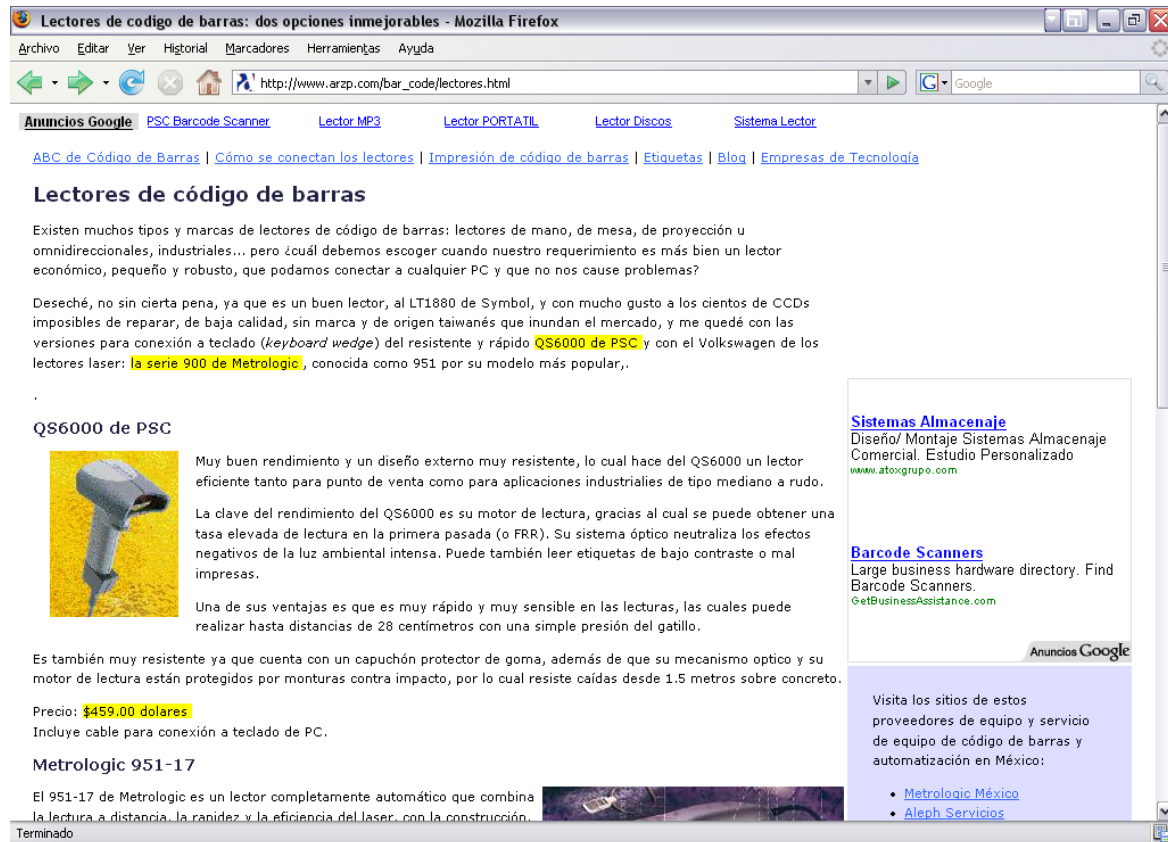
* Specify color: "M"=Black "U"=Mill Example: MNF1978DM



ANEXO N°10

LECTOR DE CODIGO DE BARRAS

http://www.arzp.com/bar_code/lectores.html



The screenshot shows a Mozilla Firefox browser window with the URL http://www.arzp.com/bar_code/lectores.html. The page content includes a navigation menu with links like 'PSC Barcode Scanner', 'Lector MP3', 'Lector PORTATIL', 'Lector Discos', and 'Sistema Lector'. The main heading is 'Lectores de código de barras'. The text describes various types of barcode readers and highlights two models: the QS6000 de PSC and the Metrologic 951-17. The QS6000 section includes an image of the device and its price of \$459.00. The Metrologic 951-17 section includes a small image of the device. There are also Google AdSense ads on the right side of the page.

QS6000 de PSC

Precio: \$459.00 dólares.

Incluye cable para conexión a teclado de PC.

IMPRESORA CODIGO DE BARRAS EN TARJETAS PVC

<http://www.mbcestore.com.mx/zebra/p330i.htm>

The screenshot shows the MBCEstore.com website. The main content area displays the Zebra P330i printer. The product name is "Zebra P330i" with part number "P330I-0000A-ID0". The price is listed as "Precio normal: \$2895.00" with a discount of "\$1155.4", resulting in a "Precio final: \$1739.60 Dlls*". The availability is "En Stock". There is a "COMPRAR" button and a quantity selector set to "1". The page also features a sidebar with navigation links, a search bar, and a "Centro de Atención a Clientes" section. The footer includes logos for Zebra, Symbol, CHHP, Eltron, and Datamax.

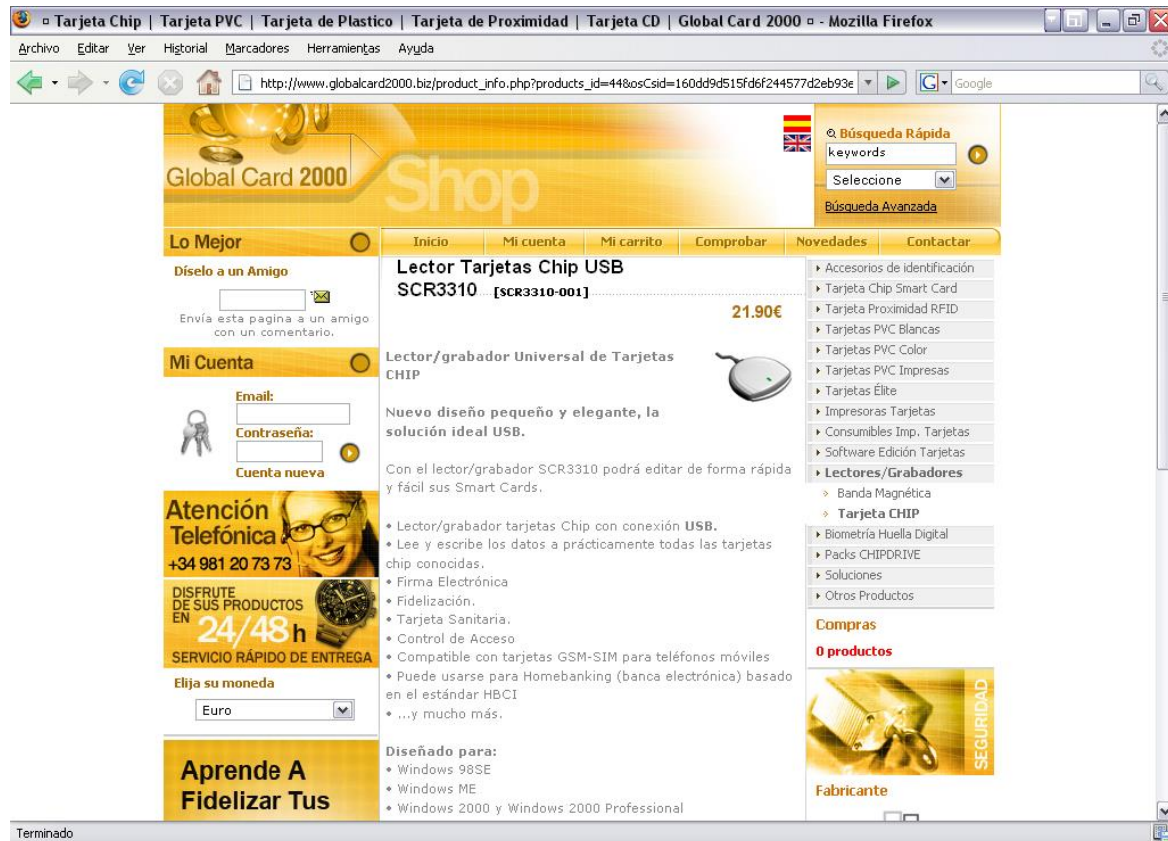
Zebra P330i

Precio final: \$1739.60

Zebra P330i Impresora Credenciales, impresión a una cara, modelo básico.
Interfaz USB.

LECTOR DE TARJETAS INTELIGENTES

http://www.globalcard2000.biz/product_info.php?products_id=44&osCsid=160dd9d515fd6f244577d2eb93ea68e6



SCR3310

- Lector/grabador tarjetas Chip con conexión USB.
- Lee y escribe los datos a prácticamente todas las tarjetas chip conocidas.
- Firma Electrónica

Precio: \$29.14

LECTORES/ESCRITOR DE TARJETAS INTELIGENTES

<http://www.c3po.es/ltc31.html>



Lector de tarjetas inteligentes LTC31

Lector y grabador externo de tarjeta chip conectado a PC

Características

- Dispositivo lector y grabador de tarjeta chip conectado a PC.
- Instalación en modo Plug and Play para Windows.
- Diseño de reducidas dimensiones y poco peso que facilita su portabilidad.
- Base plana para su posible sujeción a otros dispositivos (monitor, teclado, etc.) (se suministra adhesivo de doble cara).
- Carcasa de PVC ligero y resistente a golpes.
- Presentaciones y acabados opcionales para grandes volúmenes.
- Diseñado especialmente para su utilización en entornos de certificación con firma digital.

Qué incluye

Presentación OEM:

- Lector y grabador externo
- CD con manuales, drivers, librerías y utilidades
- Hoja de garantía
- Adhesivo de doble cara para sujeción.
- Embalaje

Presentación Kit:

- Lector y grabador externo
- Adhesivo de doble cara para sujeción.

Compra en-línea

LTC31 (USB):
23,28 € [Comprar](#)

LTC31 serie (RS232):
23,28 € [Comprar](#)

IVA y gastos envío no incluidos.

Fabricantes y organizaciones:

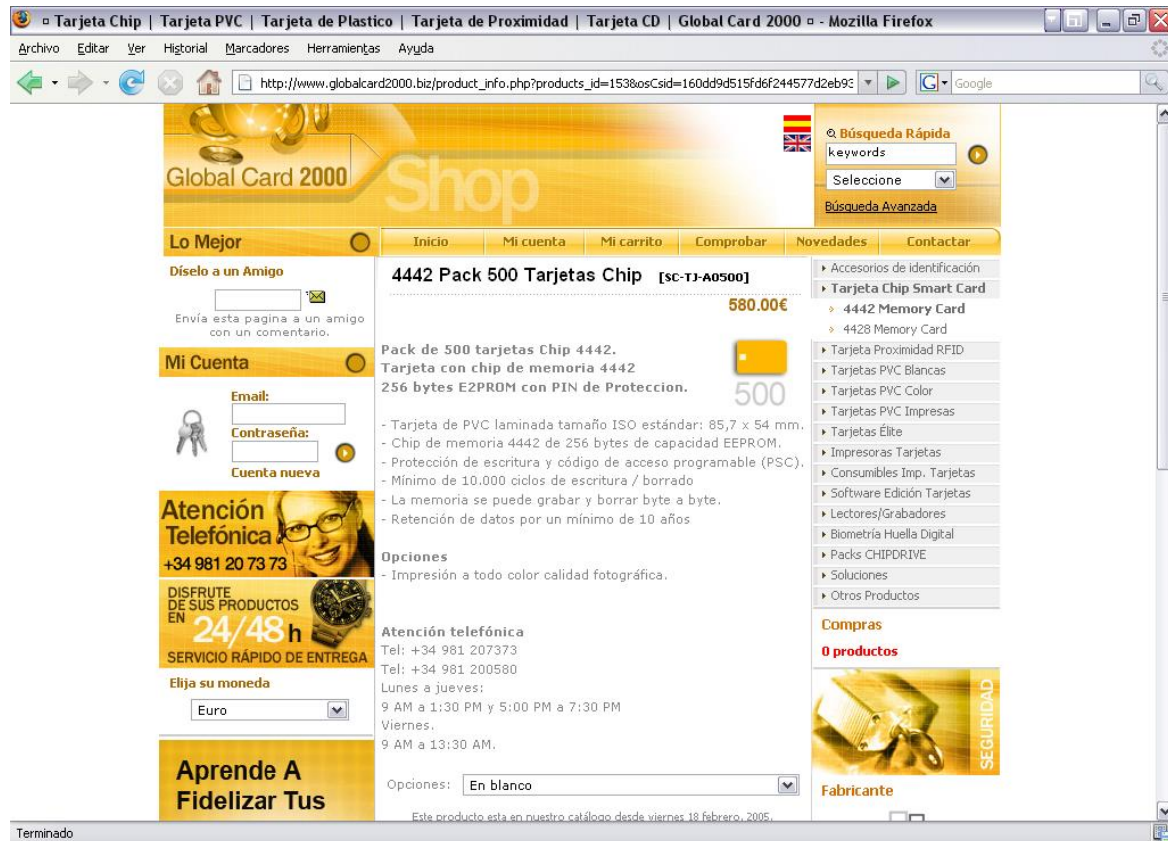
LTC31

- a) Dispositivo lector y grabador de tarjeta chip conectado a PC.
- b) Instalación en modo Plug and Play para Windows.
- c) Diseño de reducidas dimensiones y poco peso que facilita su portabilidad.

Precio: \$32.90

TARJETAS INTELIGENTES

http://www.globalcard2000.biz/product_info.php?products_id=153&osCsid=160d9d515fd6f244577d2eb93ea68e6



The screenshot shows a web browser window displaying the 'Global Card 2000 Shop' website. The main product featured is the '4442 Pack 500 Tarjetas Chip' (model SC-TJ-A0500) priced at 580.00€. The product description includes: 'Pack de 500 tarjetas Chip 4442. Tarjeta con chip de memoria 4442 256 bytes E2PROM con PIN de Protección.' Key features listed are: 'Tarjeta de PVC laminada tamaño ISO estándar: 85,7 x 54 mm.', 'Chip de memoria 4442 de 256 bytes de capacidad EEPROM.', 'Protección de escritura y código de acceso programable (PSC).', 'Mínimo de 10.000 ciclos de escritura / borrado.', 'La memoria se puede grabar y borrar byte a byte.', and 'Retención de datos por un mínimo de 10 años.' The page also features a 'Mi Cuenta' section with fields for email and password, a 'Atención Telefónica' section with a phone number (+34 981 20 73 73) and a 24/48h service promise, and a 'Compras' section showing 0 products in the cart. The website has a yellow and orange color scheme.

SC-TJ-A0500

- Tarjeta de PVC laminada tamaño ISO estándar: 85,7 x 54 mm.
- Chip de memoria 4442 de 256 bytes de capacidad EEPROM.
- Protección de escritura y código de acceso programable (PSC).

Precio: \$1.38

LECTOR/ESCRITOR BANDA MAGNETICA

<http://www.tpvmarket.com/infoarticulo172.html>

The screenshot shows a web browser window displaying the product page for the 'Lector / Grabador de Tarjetas de Banda Magnética MSE630-T'. The page features a navigation menu with categories like 'Impresoras', 'Scanners', and 'Monitores'. The product description includes details about its functionality, interface options, and pricing. A prominent offer states a price of 349,95 € (405,94€ IVA Incl.) with a 'PRECIO MÍNIMO GARANTIZADO' badge. The page also includes a 'Comprar este artículo' button and a 'Volver' button.

Lector / Grabador de Tarjetas de Banda Magnética MSE630-T

Lector y Grabador de tarjetas de banda magnética.

Lector destinado a hacer lectura o grabación de tarjetas de pago electrónico, tarjetas de fidelización de clientes y tarjetas de descuento.

Interface disponibles: Serie

Se suministra con software de grabación.

La grabación y la lectura se realizan mediante el puerto serie, el grabador viene acompañado del software y los drives necesarios, graba tarjetas de baja coercitividad.

Puede comprar tarjetas de baja coercitividad en nuestra tienda, en la sección de Tarjetas de banda magnética

349,95 €
405,94€ IVA Incl.

PRECIO MÍNIMO GARANTIZADO

Mejoramos cualquier oferta que encuentre en internet...
CLICK AQUÍ

Necesita Ayuda
Envíenos su consulta y le responderemos lo antes posible.
[Contactar](#)

Envío en 24 horas.
Compre hoy y reciba su pedido mañana mismo.
[Condiciones](#)

Devolución.
Si no está satisfecho le enviamos un

Comprar este artículo

Volver

MSE630-T

Lector destinado a hacer lectura o grabación de tarjetas de pago electrónico, tarjetas de fidelización de clientes y tarjetas de descuento.

Interfaz disponible: Serie

Precio: \$574.57

LECTOR BANDA MAGNETICA

<http://www.tpvmarket.com/infoarticulo169.html>

tpvmarket.com - Lector de Tarjetas de Banda Magnética POS900 - PVP: 81.95 EUR - Mozilla Firefox

Inicio | Impresoras | Scanners | Monitores | Cajones | Equipos TPV | Moneda y Billetes | Visores | Balanzas y Tec. Prog | Tarjetas e Ident. | Consumible impresoras

Lector de Tarjetas de Banda Magnética POS900

Lector de tarjetas de banda magnética.

Lector destinado a hacer lectura de tarjetas de pago electrónico, tarjetas de fidelización de clientes y tarjetas de descuento.

Interface disponibles: PC AT/XT/PS2 compatible.

Conexión al teclado, al leer una tarjeta los datos serán enviados por el puerto ps2 como si los hubieramos escrito mediante un teclado.

No requiere configuración ni drivers, conectar y listo!

81,95 €
95,06€ IVA Incl.

PRECIO MÍNIMO GARANTIZADO
Mejoramos cualquier oferta que encuentre en internet...
CLICK AQUÍ

Necesita Ayuda
Envíenos su consulta y le responderemos lo antes posible.
[Contactar](#)

Envío en 24 horas
Compre hoy y reciba su pedido mañana mismo.
[Condiciones](#)

15 días
Devolución.
Si no está satisfecho le enviamos un

Comprar este artículo

Volver

POS900

Lector de tarjetas de banda magnética.

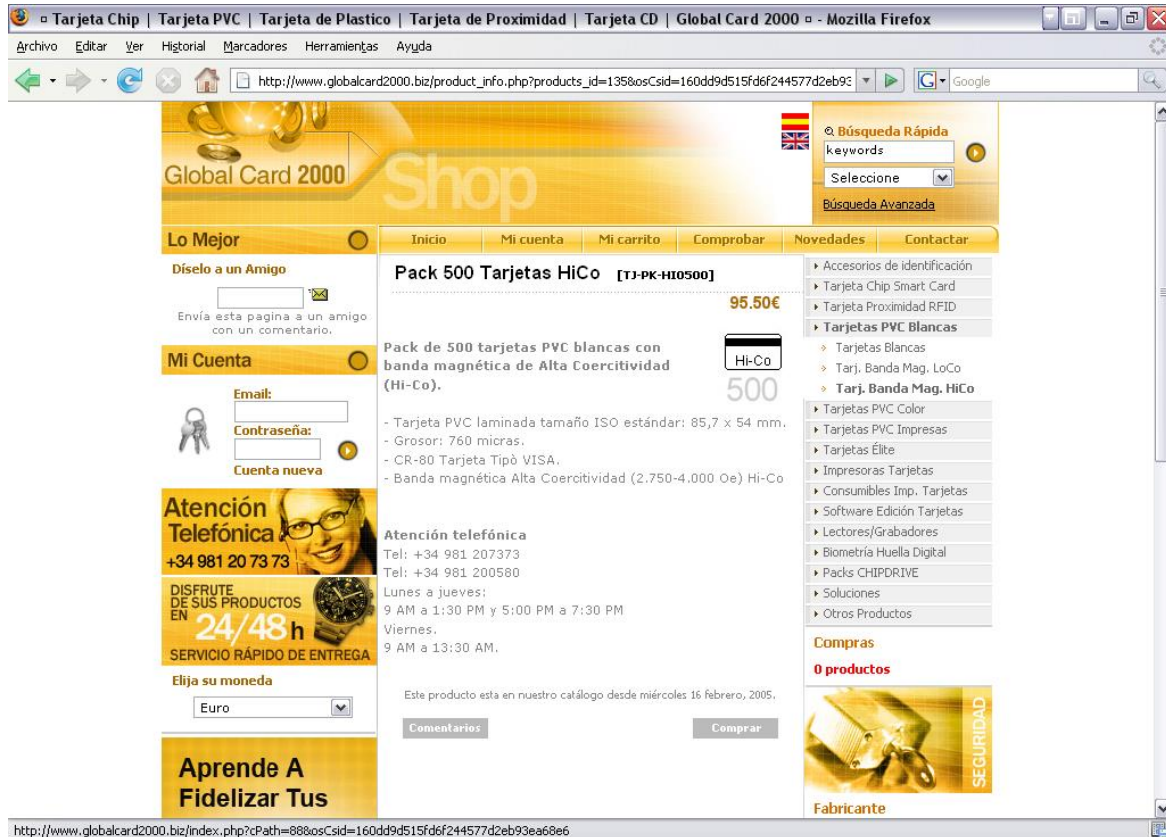
Lector destinado a hacer lectura de tarjetas de pago electrónico, tarjetas de fidelización de clientes y tarjetas de descuento.

Interfaces disponibles: PC AT/XT/PS2 compatible.

Precio: \$115.99

TARJETA HICO

http://www.globalcard2000.biz/product_info.php?products_id=135&osCsid=160dd9d515fd6f244577d2eb93ea68e6



The screenshot shows a web browser window displaying the 'Global Card 2000 Shop' website. The main product featured is the 'Pack 500 Tarjetas HiCo' (TJ-PK-HI0500) priced at 95.50€. The product description includes: 'Pack de 500 tarjetas PVC blancas con banda magnética de Alta Coercitividad (Hi-Co)'. Key features listed are: 'Tarjeta PVC laminada tamaño ISO estándar: 85,7 x 54 mm.', 'Grosor: 760 micras.', 'CR-80 Tarjeta Tipo VISA.', and 'Banda magnética Alta Coercitividad (2.750-4.000 Oe) Hi-Co'. The page also includes a 'Mi Cuenta' section with fields for email and password, a 'Atención Telefónica' section with contact numbers and hours, and a 'Compras' section showing 0 products. The website footer contains the URL: <http://www.globalcard2000.biz/index.php?cPath=88&osCsid=160dd9d515fd6f244577d2eb93ea68e6>

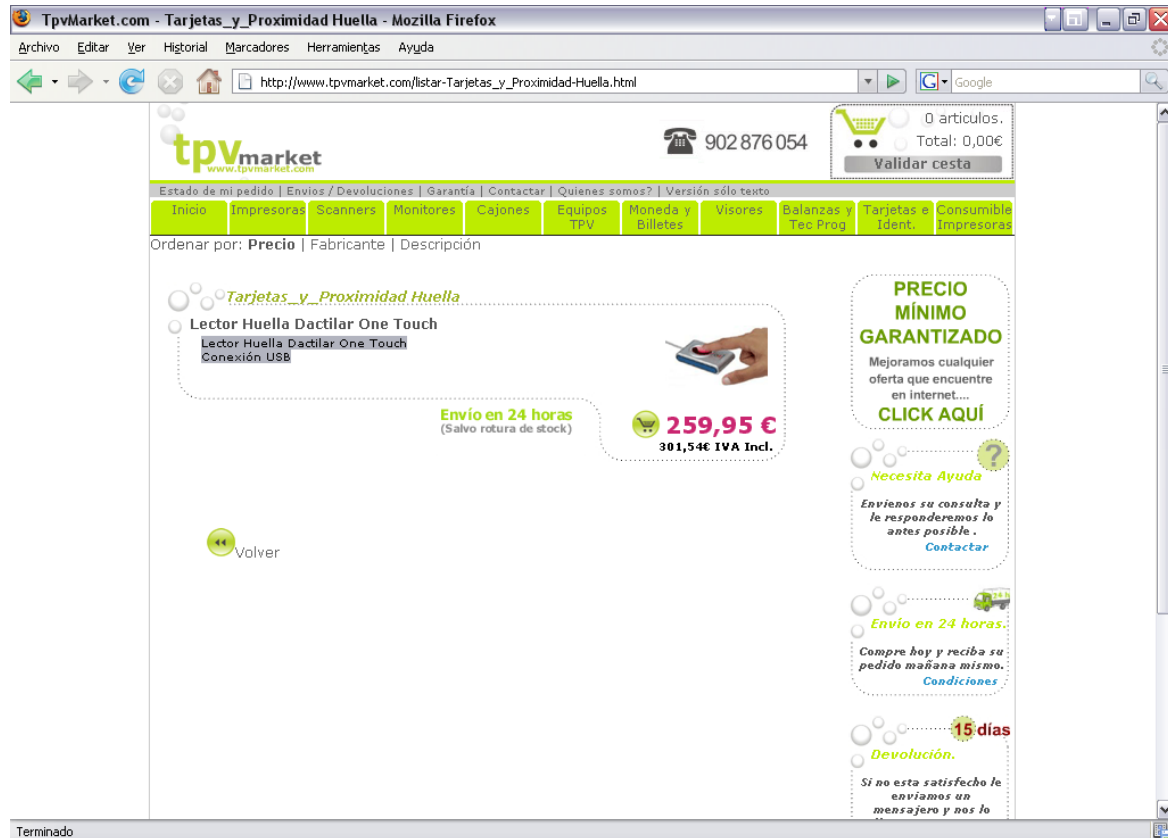
TJ-PK-HI0500

- Tarjeta PVC laminada tamaño ISO estándar: 85,7 x 54 mm.
- CR-80 Tarjeta Tipo VISA.
- Banda magnética Alta Coercitividad (2.750-4.000 Oe) Hi-Co

Precio unidad: \$0.26

LECTOR BIOMETRICO DE HUELLA DACTILAR

http://www.tpvmarket.com/listar-Tarjetas_y_Proximidad-Huella.html



The screenshot shows a web browser window displaying the tpvmarket.com website. The page features a navigation menu with categories like 'Impresoras', 'Scanners', and 'Monitores'. The main content area displays a product listing for 'Lector Huella Dactilar One Touch' with a price of 259,95 € (301,54€ IVA Incl.) and a 'Volver' button. The right sidebar contains promotional messages such as 'PRECIO MÍNIMO GARANTIZADO' and 'Envío en 24 horas'.

ONE TOUCH

Conexión USB

Precio: \$367.94

LECTOR RFID

<http://barmax.com/products.asp?cat=32>

The screenshot shows a web browser window displaying the BarMax website. The page title is "Códigos de Barras, Puntos de Ventas, Sistemas RFID, Impresoras de Tarjetas, Impresoras de Códigos de Barras, Sistemas de Biometría, CCTV." The browser address bar shows "http://barmax.com/products.asp?cat=32". The website layout includes a left sidebar with navigation links (SERVICIOS, DESCARGAR DEMOS, DISTRIBUIDORES, BUSCAR, CAJA, EMAIL) and contact information for BarMax in Monterrey, N.L. The main content area lists three products:

- BMX 802 USB:** Lector / grabador de proximidad HF (High Frequency - Alta Frecuencia) a 13,56 MHz ISO 14443A, ISO 14443B, ISO Plug&Play. Mifare Con interfase USB. Su Precio: \$220.00USD. [Más Detalles](#) [ordenar](#)
- BMX-802:** Lector RFID para tarjetas Mifare ISO 14443a con interfase Serial o USB resiste caídas y Líquidos excelente para aplicaciones de Punto de Venta. Su Precio: \$180.00USD. [ordenar](#)
- HH880 Terminal RFID:** La HH880 Terminal portátil cuenta con un escáner integrado de RFID que trabaja en un Rango de frecuencia de 902-925 MHz que le proporcionan el radio necesario para leer através de una llanta de uso industrial, camiones y automóviles. [Más Detalles](#) [Llamenos](#) [ordenar](#)

The right sidebar contains sections for "CCTV Internet - Ofertas", "Noticias" (Nuevo Sistema para Control de Llantas o Neumáticos RFID), "Credenciales", and "Trabajos de Impresión de Tarjetas con Banda Magnética, Chip Inteligente, Códigos de Barras." The footer includes "home | ayuda | contacto | e-mail" and "Derechos Reservados BarMax de México S.A. de C.V."

BMX-802.

Lector RFID para tarjetas Mifare ISO 14443a con interfase Serial o USB.

Precio: \$180.00

LECTOR/ESCRITOR RFID

<http://barmax.com/products.asp?cat=32>

The screenshot shows the Barmax website interface. At the top, there is a navigation bar with links for 'home', 'buscar', 'ayuda', 'contacto', 'e-mail', and 'tutoriales'. The date 'Martes Enero 15 2008' is displayed on the right. The main content area is titled 'Productos' and features a list of RFID products. The first product is the 'BMX 802 USB', described as a 'Lector / grabador de proximidad HF (High Frequency - Alta Frecuencia) a 13,56 MHz ISO 14443A, ISO 14443B, ISO Plug&Play. Mifare Con interfase USB'. Its price is listed as '\$220.00USD'. The second product is the 'BMX-802', described as a 'Lector RFID para tarjetas Mifare ISO 14443a con interfase Serial o USB resiste caidas y Liquidos excelente para aplicaciones de Punto de Venta'. Its price is '\$180.00USD'. The third product is the 'HH880 Terminal RFID', described as a 'La HH880 Terminal portátil cuenta con un escáner'. The website also includes a sidebar with a menu, contact information for 'Rio Conchos 1722 Mitras Centro Monterrey, N.L.', and logos for MasterCard and Visa. A banner at the top right promotes 'GRATIS Software CAJA EXPRESS CON MONEDERO ELECTRONICO Excelente Oferta!'.

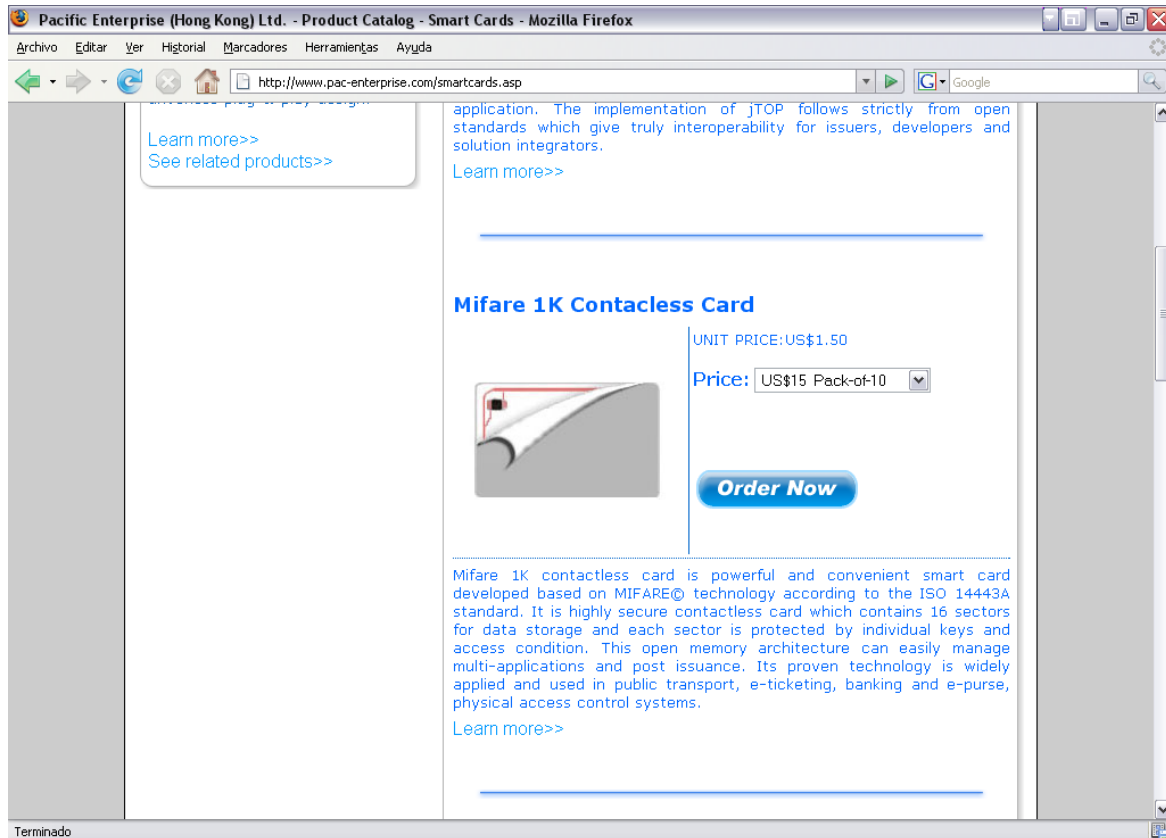
BMX 802 USB

Lector / grabador de proximidad HF (High Frequency - Alta Frecuencia) a 13,56 MHz ISO 14443A, ISO 14443B, ISO Plug&Play. Mifare Con interfase USB.

Precio: \$220.00

TARJETAS RFID

<http://www.pac-enterprise.com/smartcards.asp>



Pacific Enterprise (Hong Kong) Ltd. - Product Catalog - Smart Cards - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

http://www.pac-enterprise.com/smartcards.asp

Learn more>>
See related products>>

application. The implementation of JTOP follows strictly from open standards which give truly interoperability for issuers, developers and solution integrators.
Learn more>>

Mifare 1K Contacless Card

UNIT PRICE:US\$1.50
Price: US\$15 Pack-of-10

Order Now

Mifare 1K contactless card is powerful and convenient smart card developed based on MIFARE® technology according to the ISO 14443A standard. It is highly secure contactless card which contains 16 sectors for data storage and each sector is protected by individual keys and access condition. This open memory architecture can easily manage multi-applications and post issuance. Its proven technology is widely applied and used in public transport, e-ticketing, banking and e-purse, physical access control systems.
Learn more>>

Terminado

Precio: \$1.50

ANEXO Nº 11

Power Builder 10

<http://eshop.sybase.com/eshop/buy?p=PowerBuilder+Desktop&v=10.0&t=Windows+x86&resubmit=select&b=29&s=Software&id=17856>

SQL Server

<http://www.softwaremedia.com/product/1500.html?ovchn=PGR&ovcrn=E32-00001&ovtac=CMP>

The screenshot shows a web browser window with the following content:

- Browser title: Buy Microsoft SQL Server 2000 Developer Edition - Windows Internet Explorer
- Address bar: <http://www.softwaremedia.com/product/1500.html?ovchn=PGR&ovcrn=E32-00001&ovtac=CMP>
- Page header: SoftwareMedia.com, your one stop software shop. FreeShipping NO HASSLE RETURNS
- Search bar and navigation links: view cart | my account | order tracking | contact | about | resellers | policies | faq
- Product title: SQL Server 2000 Developer Edition
- Product details: Product #: E32-00001, Stock: Ships Free Tomorrow
- Price: \$163.90, Add to Cart button
- Description: Microsoft SQL Server 2000 is the complete database and analysis offering for rapidly delivering the next generation of scalable e-commerce, line-of-business and data warehousing solutions.
- Features: Fully Web-Enabled...Query, analyze and manipulate data over the Web. Use XML in SQL Server 2000 to exchange data between loosely coupled data easily and securely from a browser, through firewalls, and perform fast
- Right sidebar: Categories, Downloads, Manufacturers (Acronis, ACT, Adobe, etc.)

Precio: \$163.90



ANEXO Nº12

MF1-RW-232

13.56 Mhz Mifare RFID Reader/Writer for Mifare MF1 S50

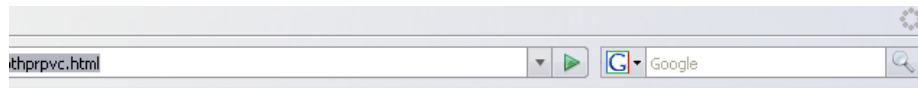
Reader Features:

- Support Mifare MF1 S50 ISO/IEC 14443A RF interface ;
- Operation frequency : 13.56Mhz
- Maximum effective distance up to 5cm;
- Less than 100ms decoding time;
- Low power dissipation with single power supply;
- RS232 output interface;
- Built-in bi-color LED and buzzer

Size: (120x70x22) mm

ANEXO Nº13

<http://www.rfideasstore.com/miisothprpvc.html>



RFIDEAS INC.
Solutions For Identification and Access
Toll-Free 866-439-4884



MIFARE ISO Thick Printable PVC Card (50 Pack)

MIFARE contactless smart cards for use in access control, log-on, ticketing and more.

Frequency 13.56 MHz

Protocol ISO 14443

Read/Write Read/Write

EEPROM Capacity 1024 bits

Dimensions: 85.6 x 53.9 x 0.8 mm

BDG-ISO-MIFARE-1K \$72.50



```
</TD>
</TR>

<!-- --- Imagen --- -->
<TR>
<TD HEIGHT="90" BGCOLOR="#000000">
<IMG SRC="hro/rosales.jpg" WIDTH="748" HEIGHT="90">
</TD>
</TR>

<!-- --- FRANJA VERDE INFERIOR --- -->
<TR>
<TD height="5" bgcolor="#AAE380">
</TD>
</TR>

<TR>
<TD height="300" bgcolor="#30A6CD">
<center>
 </center>
</TD>
</TR>

<!-- --- MENU INFERIOR --- -->
<TR>
<TD>

<TABLE width=776 height=21 border=0 cellpadding=0 cellspacing=1>
<TR align="center" valign="middle" bgcolor="#FFFFFF">
<TD width="100" height="19" bgcolor="#AAE380">
<a href="index.html" style="text-decoration:none;">
<font color="#000000"><strong>
```

Inicio



```
</strong>
```

```
</font>
```

```
</a>
```

```
</TD>
```

```
<TD width="135" bgcolor="AAE380">
```

```
<a href="historia.html" style="text-decoration:none;">
```

```
<font color="#000000"><strong>
```

Historia

```
</strong></font></b>
```

```
</a>
```

```
</TD>
```

```
<TD width="200" bgcolor="AAE380">
```

```
<a href="servicios.html" style="text-decoration:none;">
```

```
<font color="#000000"><strong>
```

Guía de Servicios

```
</strong></font>
```

```
</a>
```

```
</TD>
```

```
<TD width="200" bgcolor="AAE380">
```

```
<a href="slinea.php" style="text-decoration:none;">
```

```
<font color="#000000"><strong>
```

Consulta de Citas

```
</strong></font>
```

```
</a>
```

```
</TD>
```

```
<TD width="137" bgcolor="AAE380">
```



```
<a href="contactenos.html" style="text-decoration:none;">
<font color="#000000"><strong>
```

Contáctenos

```
</strong></font>
```

```
</a>
```

```
</TD>
```

```
</TR>
```

```
</TABLE>
```

```
</TD>
```

```
</TR>
```

```
</TABLE>
```

```
</BODY>
```

```
</HTML>
```

```
*****          *      Página      de      Historia      *
*****
*****
**
```

```
<HTML>
```

```
<BODY BACKGROUND="fondo.jpg">
```

```
<!-- --- Tabla Principal toda la página --- -->
```

```
<TABLE WIDTH="100%" HEIGHT="100%" BORDER="0" CELLPADDING="0"
CELLSPACING="0">
```



</TABLE>

<!-- --- TEXTO JUSTIFICADO --- -->

<table border="0" cellpadding="0" cellspacing="0" bgcolor="#30A6CD" >

<TR>

<TD width="776" height="250" bgcolor="#30A6CD">

<table width="700" border="0" cellpadding="0" cellspacing="0" bgcolor="#30A6CD" >

<TR>

<TD width="75"></TD>

<TD>

<p align="justify">

La edificaci&ocute;n del Hospital inici&ocute;n con la colocaci&ocute;n

de la primera piedra, el 9 de Abril del año 1891. Para su construcci&ocute;n se contrat&ocute;n la Cocieté Forges d'Aiseau, de Bélrica, finalmente el 13 de Julio de 1902, se inaugur&ocute;n el edificio. Durante la ceremonia de inaguraci&ocute;n se deposit&ocute;n una caja metálica que contenía una placa de cobre con la siguiente inscripci&ocute;n: "Bajo la protecci&ocute;n de Dios todopoderoso, y con los cuantiosos recursos donados por Don José Rosales, se comienza la obra de éste Hospital, siendo presidente Don Carlos Ezeta, quien puso la primera piedra; y bendijo la obra el ilustrísimo señor obispo Adolfo Pérez y Aguilar. San Salvador, Abril 9 de 1981";, placa que aún se conserva en la actualidad. </p>

<p align="justify" >

En su centenario, se erigi&ocute;n en el patio principal una estatua en memoria de su fundador el Sr. Jorge Rosales y es asi como después de 100 a&os;os el Hospital Rosales se ha convertido en el m&aximo Centro de Beneficencia.

</p>



```
</font>  
</TD>  
</TR>  
</TABLE>
```

```
</TD>  
</TR>  
<TR>  
<TD height="50" bgcolor="#30A6CD"></TD>  
</TR>
```

```
<!-- --- MENU INFERIOR --- -->  
<TR>  
<TD>
```

```
<TABLE width=776 height=21 border=0 cellpadding=0 cellspacing=1>  
<TR align="center" valign="middle" bgcolor="#FFFFFF">  
<TD width="100" height="19" bgcolor="#AAE380">  
<a href="index.html" style="text-decoration:none;">  
<font color="#000000"><strong>
```

Inicio

```
</strong>  
</font>  
</a>  
</TD>
```

```
<TD width="135" bgcolor="AAE380">  
<a href="historia.html" style="text-decoration:none;">  
<font color="#000000"><strong>
```

Historia



```
</strong></font></b>
```

```
</a>
```

```
</TD>
```

```
<TD width="200" bgcolor="AAE380">
```

```
<a href="servicios.html" style="text-decoration:none;">
```

```
<font color="#000000"><strong>
```

Guía de Servicios

```
</strong></font>
```

```
</a>
```

```
</TD>
```

```
<TD width="200" bgcolor="AAE380">
```

```
<a href="slinea.php" style="text-decoration:none;">
```

```
<font color="#000000"><strong>
```

Consulta de Citas

```
</strong></font>
```

```
</a>
```

```
</TD>
```

```
<TD width="137" bgcolor="AAE380">
```

```
<a href="contactenos.html" style="text-decoration:none;">
```

```
<font color="#000000"><strong>
```

Contáctenos

```
</strong></font>
```

```
</a>
```

```
</TD>
```

```
</TR>
```

```
</TABLE>
```




```
<TR>
<TD HEIGHT="5" BGCOLOR="#AAE380">
</TD>
</TR>

<!-- --- Imagen --- -->
<TR>
<TD HEIGHT="90" BGCOLOR="#000000">
<IMG SRC="hro/servicios.jpg" WIDTH="748" HEIGHT="90">
</TD>
</TR>

<!-- --- FRANJA VERDE INFERIOR --- -->
<TR>
<TD height="5" bgcolor="#AAE380">
</TD>
</TR>

<!-- --- TEXTO JUSTIFICADO --- -->
<TR>
<TD height="250" bgcolor="#30A6CD">
<table height="300" width="700" border="0" cellpadding="0"
cellspacing="0" bgcolor="#30A6CD" >
<TR>
<TD width="90"></TD>
<TD width="200">

<li>Alergolog&iacute;a.</li>
<li>Anestesia General.</li>
<li>Anat. Patolog&iacute;a.</li>
<li>Cardiolog&iacute;a.</li>
<li>Central De Equipo.</li>
<li>Centro Quir&uacute;rgico.</li>
<li>Cirug&iacute;a.</li>
<li>Cirug&iacute;a Pl&aacute;stica.</li>
```



```
<li>Cirug&iacute;a Ambulatoria.</li>
<li>Cirug&iacute;a Oncol&oacute;gica.</li>
<li>Cirug&iacute;as Hombres.</li>
<li>Cirug&iacute;as Mujeres.</li>
<li> Cirug&iacute;a General.</li>
<li>Coloproctolog&iacute;a.</li>
<li>Consulta Interna Gral.</li>
```

```
</font></TD>
```

```
<td width="200">
```

```
<li>Cuidados Coronarios.</li>
<li>Dermatolog&iacute;a.</li>
<li>Electrocardiogramas.</li>
<li>Emergencia.</li>
<li>Endocrinolog&iacute;a.</li>
<li>Endoscopia.</li>
<li>Hematología.</li>
<li>Infectolog&iacute;a.</li>
<li>Medicina F&iacute;sica.</li>
<li>Medicina Nuclear. </li>
<li>Nefrolog&iacute;a.</li>
<li>Neumolog&iacute;a.</li>
<li>Neurocirug&iacute;a.</li>
<li>Neurolog&iacute;a.</li>
<li>Nutrici&oacute;n.</li>
```

```
</font></TD>
```

```
<TD width="200">
```

```
<li>Observ. Medicina.</li>
<li> Odontol&oacute;gica.</li>
<li>Oftalmolog&iacute;a.</li>
```



```
<li>Oncolog&iacute;a.</li>
<li>Ortoped&iacute;a.</li>
<li>Otorrinolaringolog&iacute;a.</li>
<li>Psicol&oacute;gia.</li>
<li>Psiquiatr&iacute;a y Sicolog&iacute;a.</li>
<li>Quir&oacute;fano Emergencia.</li>
<li>Reumatolog&iacute;a.</li>
<li>Terapia Respiratoria.</li>
<li>Traumat-Ortopedia.</li>
<li>Uci. </li>
<li>Ulceras/Heridas.</li>
<li>Urolog&iacute;a.</li>
</TD>

</TR>
</TABLE>

</TD>
</TR>

<!-- --- MENU INFERIOR --- -->
<TR>
<TD>

<TABLE width=776 height=21 border=0 cellpadding=0 cellspacing=1>
<TR align="center" valign="middle" bgcolor="#FFFFFF">
<TD width="100" height="19" bgcolor="#AAE380">
<a href="index.html" style="text-decoration:none;">
<font color="#000000"><strong>

Inicio

</strong>
</font>
</a>
```



</TD>

<TD width="135" bgcolor="AAE380">

Historia

</TD>

<TD width="200" bgcolor="AAE380">

Guía de Servicios

</TD>

<TD width="200" bgcolor="AAE380">

Consulta de Citas

</TD>

<TD width="137" bgcolor="AAE380">



Contáctenos

```
</strong></font>
```

```
</a>
```

```
</TD>
```

```
</TR>
```

```
</TABLE>
```

```
</TD>
```

```
</TR>
```

```
</TABLE>
```

```
</BODY>
```

```
</HTML>
```

```
*****      *   Página   de   Servicio   en   línea   *  
*****  
*****  
**
```

```
<HTML>
```

```
<BODY BACKGROUND="fondo.jpg">
```

```
<!-- --- Tabla Principal toda la página --- -->
```

```
<TABLE WIDTH="100%" HEIGHT="100%" BORDER="0" CELLPADDING="0"  
CELLSPACING="0">
```

```
<TR BORDERCOLOR="#FFFFFF">
```

```
<TD ALIGN="center">
```

```
<!-- --- Tabla del contenido --- -->
```



```
</TR>
```

```
</TABLE>
```

```
<TABLE width="776" height="80" border="0" cellpadding="0" cellspacing="0">
```

```
<TR height="30%">
```

```
<TD width="300" bgcolor="#30A6CD" ></TD>
```

```
<TD width="280" bgcolor="#30A6CD" ></TD>
```

```
<TD bgcolor="#30A6CD" border="0">
```

```
</TD>
```

```
</TR>
```

```
</TABLE>
```

```
<!-- --- MENU INFERIOR --- -->
```

```
<TR>
```

```
<TD>
```

```
<TABLE width=776 height=21 border=0 cellpadding=0 cellspacing=1>
```

```
<TR align="center" valign="middle" bgcolor="#FFFFFF">
```

```
<TD width="100" height="19" bgcolor="#AAE380">
```

```
<a href="index.html" style="text-decoration:none;">
```

```
<font color="#000000"><strong>
```

```
Inicio
```

```
</strong>
```

```
</font>
```

```
</a>
```

```
</TD>
```

```
<TD width="135" bgcolor="AAE380">
```

```
<a href="historia.html" style="text-decoration:none;">
```

```
<font color="#000000"><strong>
```



Historia

</TD>

<TD width="200" bgcolor="AAE380">

Guía de Servicios

</TD>

<TD width="200" bgcolor="AAE380">

Consulta de Citas

</TD>

<TD width="137" bgcolor="AAE380">

Contáctenos

</TD>

</TR>



```
</TD>
</TR>

<!-- --- FRANJA VERDE SUPERIOR --- -->
<TR>
<TD HEIGHT="5" BGCOLOR="#AAE380">
</TD>
</TR>

<!-- --- Imagen --- -->
<TR>
<TD HEIGHT="90" BGCOLOR="#000000">
<IMG SRC="hro/slinea.jpg" WIDTH="748" HEIGHT="90">
</TD>
</TR>

<!-- --- FRANJA VERDE INFERIOR --- -->
<TR>
<TD height="5" bgcolor="#AAE380">
</TD>
</TR>

<!-- --- TABLA AZUL --- -->
<TABLE width="776" border="0" cellpadding="0" cellspacing="0"
bgcolor="#000000">
<TR>
<TD height="40" WIDTH="400" bgcolor="#30A6CD">
</TD>
<TD bgcolor="#30A6CD"><H4>

<?PHP
    $var = $_POST['nombre'];
    $var2 = $_POST['dui'];
    $connectID = mssql_connect("192.168.10.10","sa","");
    $counter=1;
```



```
mssql_select_db("Minis_Salud");
$result=mssql_query("select* from EXPEDIENTE where
CODIGO_PACIENTE='$var' AND N_Documento='$var2'");
$result2=mssql_query("select* from CONSULTA where
CODIGO_PACIENTE='$var'");
?>

</TD>
</TR>
</TABLE>

<!-- --- TABLA CUADRO DE INFORMACIÓN --- -->
<TABLE width="776" height="100" border="0" cellpadding="0"
cellspacing="2" bgcolor="#30A6CD">
<?PHP

    if ($row=mssql_fetch_array($result2)){
        $c4=$row["USUARIO_D"];
        $c5=$row["FECHA_P_CITA"];
        $c7=$row["ID_PROVEEDOR"];
    }
    else{
        $c4="";
        $c5="";
        $c7="";
    }

    $result3=mssql_query("select* from PROVEEDOR where
ID_PROVEEDOR='$c7'");

    if ($row=mssql_fetch_array($result3)){
        $c6=$row["NOMBRE_PROV"];
        $c8=$row["APELLIDO_PROV"];
    }
    else{
```



```

    $c6="";
    $c8="";
}
$c9= "$c6 $c8";

if ($row=mssql_fetch_array($result)){
    $c1=$row["Primer_Nombre"];
    $c2=$row["Primer_Apellido"];
    $c3="Fecha de Proxima Cita:";
}
else{
    $counter="";
    $c1="";
    $c2="";
    $c5=" ningún registro!";
    $c9="";
    $c3=";No se ha encontrado ";
    $_POST['nombre']="";
}

?>
<TR>
<TD>
<CENTER>

<!-- --- DIBUJAMOS EL CUADRO DE INFORMACIÓN DEL DUI --- -->
<TABLE width="600" border="0" cellpadding="0" cellspacing="1"
bgcolor="#000000">
<TR>
<CENTER>
<TD width="340" height="20" bgcolor="#000000" align="right">
<FONT COLOR=#FFFFFF FACE=VERDANA SIZE=2><b>

<?PHP
```



```
        echo $c3;
    ?>

</TD>
<TD height="20" bgcolor="#000000" align="left">
<FONT COLOR=#FFFFFF FACE=VERDANA SIZE=2><b>

<?PHP
        echo $c5;
    ?>

</TD>
</TR>
</TABLE>

<!--      ---  DIBUJAMOS EL CUADRO DE INFORMACIÓN DEL DUI  --- -->
<TABLE  width="600"  border="0"  cellpadding="0"  cellspacing="1"
bgcolor="#000000">
<TR>

<TD bgcolor="#B2C5CF" ALIGN="CENTER" width="15%" height="20" >
<FONT COLOR=#000000 FACE=VERDANA SIZE=2><b>
No de Afiliación
</TD>

TD ALIGN="CENTER" bgcolor="#B2C5CF" width="10%" height="20" >
<FONT COLOR=#000000 FACE=VERDANA SIZE=2><b>
Nombre
</TD>

<TD ALIGN="CENTER" bgcolor="#B2C5CF" width="10%" height="20">
<FONT COLOR=#000000 FACE=VERDANA SIZE=2><b>
Apellido
</TD>
```



```
<TD ALIGN="CENTER" bgcolor="#B2C5CF" width="35%" height="20">
<FONT COLOR=#000000 FACE=VERDANA SIZE=2><b>
Medico Asignado
</TD>
```

```
<TD height="20" bgcolor="#000000" align="CENTER">
<FONT COLOR=#FFFFFF FACE=VERDANA SIZE=2><b>
</TD>
</TR>
```

```
<TR>
<TD bgcolor="#EEEEEE" VALIGN="BOTTOM" ALIGN="CENTER">
<FONT FACE=VERDANA SIZE=2>
```

```
<?PHP
    echo $_POST['nombre'];
?>
</TD>
```

```
<TD bgcolor="#EEEEEE" VALIGN="BOTTOM" ALIGN="CENTER">
<FONT FACE=VERDANA SIZE=2>
```

```
<?PHP
    echo (" $c1 ");
?>
</TD>
```

```
<TD bgcolor="#EEEEEE" VALIGN="BOTTOM" ALIGN="CENTER">
<FONT FACE=VERDANA SIZE=2>
```

```
<?PHP
    echo (" $c2 ");
?>
</TD>
```

```
<TD bgcolor="#EEEEEE" VALIGN="BOTTOM" ALIGN="CENTER">
<FONT FACE=VERDANA SIZE=2>
```

```
<?PHP
```



```
        echo ("%c9");
    ?>

</TD>
</TR>
</TABLE>

</TD>
</TR>
</TABLE>

<TABLE width="776" border="0" cellpadding="0" cellspacing="0"
bgcolor="#000000">
<TR>
<TD height="160" WIDTH="400" bgcolor="#30A6CD" VALIGN="BOTTOM"
ALIGN="RIGHT">
</TD>
</TR>
</TABLE>

<!-- --- MENU INFERIOR --- -->
<TR>
<TD>

<TABLE width=776 height=21 border=0 cellpadding=0 cellspacing=1>
<TR align="center" valign="middle" bgcolor="#FFFFFF">
<TD width="100" height="19" bgcolor="#AAE380">
<a href="index.html" style="text-decoration:none;">
<font color="#000000"><strong>

Inicio

</strong>
</font>
```



</TD>

<TD width="135" bgcolor="AAE380">

Historia

</TD>

<TD width="200" bgcolor="AAE380">

Guía de Servicios

</TD>

<TD width="200" bgcolor="AAE380">

Consulta de Citas

</TD>

<TD width="137" bgcolor="AAE380">



Contáctenos

</TD>

</TR>

</TABLE>

</TD>

</TR>

</TABLE>

</BODY>

</HTML>

```

*****          *          Página          Contactenos          *
*****
*****
**

```

<HTML>

<BODY BACKGROUND="fondo.jpg">

<!-- --- Tabla Principal toda la página --- -->

```

<TABLE WIDTH="100%" HEIGHT="100%" BORDER="0" CELLPADDING="0"
CELLSPACING="0">

```

<TR BORDERCOLOR="#FFFFFF">

<TD><CENTER>

<!-- --- Tabla del contenido --- -->



```
</TD>
<TD bgcolor="#30A6CD"><H4>
</TD>
</TR>
</TABLE>

<!-- --- TABLA CUADRO DE INFORMACIÓN --- -->
<TABLE width="776" height="110" border="0" cellpadding="0"
cellspacing="2" bgcolor="#30A6CD">
<TR>
<TD>
<CENTER>

<!-- --- DIBUJAMOS EL CUADRO DE INFORMACIÓN DEL DUI --- -->
<TABLE width="375" height="20" border="0" cellpadding="0"
cellspacing="1" bgcolor="#000000">
<TR>
<CENTER>

<TD width="340" height="20" bgcolor="#000000" align="center">
<FONT COLOR=#FFFFFF FACE=VERDANA SIZE=2><b>
Contáctenos
</TD>

<TD height="20" bgcolor="#000000" align="left">
<FONT COLOR=#FFFFFF FACE=VERDANA SIZE=2><b>
</TD>
</TR>
</TABLE>

<!-- --- DIBUJAMOS EL CUADRO DE INFORMACIÓN DEL DUI ----->
<TABLE width="375" height="80" border="0" cellpadding="0"
cellspacing="1" bgcolor="#000000">
<TR>
```



```
</TABLE>
```

```
</TD>
```

```
</TR>
```

```
</TABLE>
```

```
<TABLE width="776" border="0" cellpadding="0" cellspacing="0" bgcolor="#000000">
```

```
<TR>
```

```
<TD height="170" WIDTH="400" bgcolor="#30A6CD" VALIGN="BOTTOM" ALIGN="RIGHT">
```

```
</TD>
```

```
</TR>
```

```
</TABLE>
```

```
<!-- --- MENU INFERIOR --- -->
```

```
<TR>
```

```
<TD>
```

```
<TABLE width=776 height=21 border=0 cellpadding=0 cellspacing=1>
```

```
<TR align="center" valign="middle" bgcolor="#FFFFFF">
```

```
<TD width="100" height="19" bgcolor="#AAE380">
```

```
<a href="index.html" style="text-decoration:none;">
```

```
<font color="#000000"><strong>
```

```
Inicio
```

```
</strong>
```

```
</font>
```

```
</a>
```

```
</TD>
```

```
<TD width="135" bgcolor="AAE380">
```

```
<a href="historia.html" style="text-decoration:none;">
```

```
<font color="#000000"><strong>
```



Historia

```
</strong></font></b>
```

```
</a>
```

```
</TD>
```

```
<TD width="200" bgcolor="AAE380">
```

```
<a href="servicios.html" style="text-decoration:none;">
```

```
<font color="#000000"><strong>
```

Guía de Servicios

```
</strong></font>
```

```
</a>
```

```
</TD>
```

```
<TD width="200" bgcolor="AAE380">
```

```
<a href="slinea.php" style="text-decoration:none;">
```

```
<font color="#000000"><strong>
```

Consulta de Citas

```
</strong></font>
```

```
</a>
```

```
</TD>
```

```
<TD width="137" bgcolor="AAE380">
```

```
<a href="contactenos.html" style="text-decoration:none;">
```

```
<font color="#000000"><strong>
```

Contáctenos

```
</strong></font>
```

```
</a>
```

```
</TD>
```



`</TR>`

`</TABLE>`

`</TD>`

`</TR>`

`</TABLE>`

`</BODY>`

`</HTML>`



ANEXO N° 15

PROGRAMA DEL SISTEMA

```
////////////////////////////////////  
//   Autores       :   Edgard Aguirre.  
//                   Alexander Castillo.  
//                   Rodrigo Najarro.  
//   Creación      :   01/01/2008  
//-----  
////////////////////////////////////
```

LIBRERÍA PRINCIPAL (RFYD.PBL)

OBJETO APLICACIÓN

```
forward  
global type rfyd from application  
end type  
global n_transobj sqlca  
global dynamicdescriptionarea sqlda  
global dynamicstagingarea sqlsa  
global error error  
global n_message message  
end forward
```



global variables

```
////////////////////////////////////  
//
```

DECLARACION DE VARIABLES GLOBALES

```
////////////////////////////////////
```

//Variables Globales

```
String      gs_codigo, gs_login, gs_codigo_usuario, gs_estacion, gs_periodo,  
gs_codpac  
String      gs_grupo, gs_nombre_usuario, gs_puesto, gs_i, gs_b, gs_m,  
gs_codcita  
String      gs_idrequerimiento, gs_iddetalle, gs_idproveedor, gs_obtiene,  
gs_idtipo  
string      gs_codigopaciente, gs_enfermedad, gs_examen, gs_medimento  
Integer     gi_anio, gi_codigo_sistema
```

end variables

global type rfyd from application

string appname = "rfyd"

end type

global rfyd rfyd

event

```
open;/**  
*****
```

// Sistema Salud

```
/**  
*****
```



```
// Fecha/Ver      Descripción
// -----  -----
//*****
*****

//Declaración de variables
int rc

//Inicialización de variables
this.toolbartext      = false
this.toolbarusercontrol = true
this.toolbartips      = true

//ensure only get one toolbar
this.toolbarframetitle    = 'Barra Principal'
this.toolbarsheettitle    = 'Barra Secundaria'
this.displayname          = "Sistema de Salud"
this.dwmmessagegetitle    = this.displayname

//open application manager
n_app_mgr = create n_app_mgr

//determine the keycode for the application to print screen
n_app_mgr.uf_setprintscreenkey(keyf12!)

//default application right-mouse support
n_app_mgr.uf_setrmbssupport(1)           //right-mouse active -
0 means disabled
```



//how to launch the edit windows from u_lst objects

//0=share the datawindows,1=pass the sql key clause,2=pass the whole sql clause

n_app_mgr.uf_seteditlaunchmethod(1)

n_app_mgr.uf_setdellaunchmethod(1)

//0 = sheet for mdi application, 1 = window (main,popup,response etc..)

n_win_mgr.uf_setopeneditlaunch(1)

//login options

**n_app_mgr.uf_setdeveloperlogon(0) //yes - logon developers automatically via
ini file**

**//n_app_mgr.uf_setautomaticlogon(0) //no - runtime user must enter
username/password (ignore ini file logid/logpass)**

n_app_mgr.uf_setappinifilename("minisalud.ini")

//default window classnames for standard mRADIAN windows

**//allows convenient override by applications to user there own windows where
necessary**

n_app_mgr.uf_setwin_about("w_acerca_de")

//application about box

**n_app_mgr.uf_setwin_about_system("w_about_system") //windows
environment**

**n_app_mgr.uf_setwin_aboutproduct("w_aboutproduct") //about
product credits**

n_app_mgr.uf_setwin_logon("w_logon_acceso")

//logon

n_app_mgr.uf_setwin_system_error("w_system_error") //system errors



```
n_app_mgr.uf_setwin_userpref("w_userpref")           //user
preferences
n_app_mgr.uf_setwin_print("w_print")                 //print
datawindow
n_app_mgr.uf_setwin_ppv("w_ppv")
    //print preview
n_app_mgr.uf_setwin_printlist("w_printlist")         //print    list
objects

//application defaults have been established ...
//n_connect_mgr.uf_setdfldbms("ODBC")
    //MSS or SYB or ODBC Supported
n_connect_mgr.uf_setdfldbms("MSS")
    //MSS or SYB or ODBC Supported

//setup the application environment based on options/variables established
n_app_mgr.uf_setenvironment()

openwithparm(w_splash,1)                            //hold for n CPU seconds
setpointer(hourglass!)

//logon method could be direct from ini file and so no logon dialog is necessary
rc = n_app_mgr.uf_logon()

//Abortar si el usuario es incorrecto

If trim(gs_codigo_usuario) = "" then halt close
```



```
//default logon method - else n_connect_mgr.uf_logon(user id,password,...) or open
w_logon
n_app_mgr.uf_openframe(w_frame)

//Por el momento estara desabilitada por no tener el archivo de ayuda
//n_app_mgr.uf_setapphelpfilename("RegAcad.hlp")

//Realizar conexion con la base de datos
if n_connect_mgr.uf_is_connected() = 0 then
    if n_app_mgr.uf_is_exe() = 1 then
        messagebox("Sistema de Salud", &
            "Imposible conectarse a la Base de Datos LOCAL" &
            +"~n~r3. Consulte con el soporte técnico del sistema...!",stopsign!)
        halt close
    else
        messagebox("Advertencia","No está conectado a la Base de Datos")
    end if
end if

end event

on rfyd.create
    appname="rfyd"
    message=create n_message
    sqlca=create n_transobj
    sqlda=create dynamicdescriptionarea
    sqlsa=create dynamicstagingarea
    error=create error
end on
```



on rfyd.destroy

destroy(sqlca)

destroy(sqlda)

destroy(sqlsa)

destroy(error)

destroy(message)

end on

event close;// destroying the application manager destroys all registered objects

int rc

//will close all active/registered windows then destroy itself

if isvalid(n_app_mgr) then

rc = n_app_mgr.uf_destroy()

end if

end event

event systemerror;//Control de errores del sistema

window lwin

open(lwin,n_app_mgr.uf_getwin_system_error())

end event

OBJETO MENU

forward

global type m_minisalud_main from m_main

end type

type m_menúnavegador from menu within m_window



```
end type
type m_window from m_main`m_window within m_minisalud_main
m_menúnavegador m_menúnavegador
end type
type m_tablas from menu within m_minisalud_main
end type
type m_operaciones from menucascade within m_minisalud_main
end type
type m_alicitante from menu within m_operaciones
end type
type m_bconsulta from menu within m_operaciones
end type
type m_cfarmacia from menu within m_operaciones
end type
type m_operaciones from menucascade within m_minisalud_main
m_alicitante m_alicitante
m_bconsulta m_bconsulta
m_cfarmacia m_cfarmacia
end type
type m_procesos from menu within m_minisalud_main
end type
type m_farmacia from menu within m_procesos
end type
type m_laboratorio from menu within m_procesos
end type
type m_procesos from menu within m_minisalud_main
m_farmacia m_farmacia
m_laboratorio m_laboratorio
end type
```



```
type m_consultasyreportes from menu within m_minisalud_main  
end type  
type m_aretas from menu within m_consultasyreportes  
end type  
type m_bexpediente from menu within m_consultasyreportes  
end type  
type m_consultasyreportes from menu within m_minisalud_main  
m_aretas m_aretas  
m_bexpediente m_bexpediente  
end type  
global type m_minisalud_main from m_main  
m_tablas m_tablas  
m_operaciones m_operaciones  
m_procesos m_procesos  
m_consultasyreportes m_consultasyreportes  
end type  
end forward
```

```
global type m_minisalud_main from m_main  
m_tablas m_tablas  
m_operaciones m_operaciones  
m_procesos m_procesos  
m_consultasyreportes m_consultasyreportes  
end type  
global m_minisalud_main m_minisalud_main
```

```
type variables
```

```
//u_nv_smael_seguridad
```

```
inv_seguridad
```



end variables

```
on m_minisalud_main.create  
m_minisalud_main=this  
call super::create  
this.text = "m_bmlicitacion_main"  
this.m_tablas=create m_tablas  
this.m_operaciones=create m_operaciones  
this.m_procesos=create m_procesos  
this.m_consultasyreportes=create m_consultasyreportes  
this.Item[UpperBound(this.Item)+1]=this.m_tablas  
this.Item[UpperBound(this.Item)+1]=this.m_operaciones  
this.Item[UpperBound(this.Item)+1]=this.m_procesos  
this.Item[UpperBound(this.Item)+1]=this.m_consultasyreportes  
end on
```

```
on m_minisalud_main.destroy  
call super::destroy  
destroy(this.m_tablas)  
destroy(this.m_operaciones)  
destroy(this.m_procesos)  
destroy(this.m_consultasyreportes)  
end on
```

```
type m_file from m_main`m_file within m_minisalud_main  
end type
```

```
on m_file.create  
call super::create
```



this.enabled = false

end on

on m_file.destroy

call super::destroy

end on

type m_new from m_main`m_new within m_file

end type

on m_new.create

call super::create

this.text = "Usuarios"

this.enabled = false

this.shortcut = 0

this.microhelp = "Tipos de Usuarios"

this.toolbaritemvisible = true

this.toolbaritemtext = "Tipos de Usuarios"

this.toolbaritemorder = 3

end on

on m_new.destroy

call super::destroy

end on

event clicked;mf_opensheet('w_lst_usuario')

end event

type m_open from m_main`m_open within m_file



end type

on m_open.create

call super::create

this.visible = false

this.text = "Configuración de estación"

this.enabled = false

this.shortcut = 0

this.microhelp = "Configuración de estación"

this.toolbaritemtext = "Configuración del Centro Educativo"

this.toolbaritemorder = 4

end on

on m_open.destroy

call super::destroy

end on

type m_close from m_main`m_close within m_file

end type

on m_close.create

call super::create

this.microhelp = "Cerrar"

this.toolbaritemtext = "Cerrar"

end on

on m_close.destroy

call super::destroy

end on



```
type m_mbcf1 from m_main`m_mbcf1 within m_file  
end type
```

```
on m_mbcf1.create  
call super::create  
end on
```

```
on m_mbcf1.destroy  
call super::destroy  
end on
```

```
type m_logon from m_main`m_logon within m_file  
end type
```

```
on m_logon.create  
call super::create  
this.microhelp = "Conexión a la Base de Datos ..."  
this.toolbaritemname = "DBConnect!"  
this.toolbaritemtext = "Conexión a la Base de Datos ..."  
this.toolbaritemorder = 1  
end on
```

```
on m_logon.destroy  
call super::destroy  
end on
```

```
type m_logoff from m_main`m_logoff within m_file  
end type
```



```
on m_logoff.create  
call super::create  
end on
```

```
on m_logoff.destroy  
call super::destroy  
end on
```

```
type m_mbcf2 from m_main`m_mbcf2 within m_file  
end type
```

```
on m_mbcf2.create  
call super::create  
end on
```

```
on m_mbcf2.destroy  
call super::destroy  
end on
```

```
type m_save from m_main`m_save within m_file  
end type
```

```
on m_save.create  
call super::create  
this.visible = false  
end on
```

```
on m_save.destroy
```



```
call super::destroy
```

```
end on
```

```
type m_saveas from m_main`m_saveas within m_file
```

```
end type
```

```
on m_saveas.create
```

```
call super::create
```

```
this.text = "Guardar &Como...~tCtrl+G"
```

```
this.shortcut = 327
```

```
end on
```

```
on m_saveas.destroy
```

```
call super::destroy
```

```
end on
```

```
type m_mbcf3 from m_main`m_mbcf3 within m_file
```

```
end type
```

```
on m_mbcf3.create
```

```
call super::create
```

```
end on
```

```
on m_mbcf3.destroy
```

```
call super::destroy
```

```
end on
```

```
type m_print from m_main`m_print within m_file
```

```
end type
```



```
on m_print.create  
call super::create  
end on
```

```
on m_print.destroy  
call super::destroy  
end on
```

```
type m_printlist from m_main`m_printlist within m_file  
end type
```

```
on m_printlist.create  
call super::create  
end on
```

```
on m_printlist.destroy  
call super::destroy  
end on
```

```
type m_printpreview from m_main`m_printpreview within m_file  
end type
```

```
on m_printpreview.create  
call super::create  
end on
```

```
on m_printpreview.destroy  
call super::destroy
```



end on

**type m_printersetup1 from m_main`m_printersetup1 within m_file
end type**

**on m_printersetup1.create
call super::create
this.toolbaritemname = "Report5!"
end on**

**on m_printersetup1.destroy
call super::destroy
end on**

**type m_mbcf4 from m_main`m_mbcf4 within m_file
end type**

**on m_mbcf4.create
call super::create
end on**

**on m_mbcf4.destroy
call super::destroy
end on**

**type m_properties from m_main`m_properties within m_file
end type**

on m_properties.create



call super::create

end on

on m_properties.destroy

call super::destroy

end on

type m_mbcf5 from m_main`m_mbcf5 within m_file

end type

on m_mbcf5.create

call super::create

end on

on m_mbcf5.destroy

call super::destroy

end on

type m_delete from m_main`m_delete within m_file

end type

on m_delete.create

call super::create

end on

on m_delete.destroy

call super::destroy

end on



```
type m_mbcf6 from m_main`m_mbcf6 within m_file  
end type  
  
on m_mbcf6.create  
call super::create  
end on  
  
on m_mbcf6.destroy  
call super::destroy  
end on  
  
type m_exitsystem from m_main`m_exitsystem within m_file  
end type  
  
on m_exitsystem.create  
call super::create  
this.toolbaritemvisible = true  
this.toolbaritemname = "Exit!"  
this.toolbaritemtext = "Abandonar el Sistema"  
this.toolbaritemorder = 200  
this.toolbaritemspace = 2  
end on  
  
on m_exitsystem.destroy  
call super::destroy  
end on  
  
type m_edit from m_main`m_edit within m_minisalud_main  
end type
```



```
on m_edit.create  
call super::create  
end on
```

```
on m_edit.destroy  
call super::destroy  
end on
```

```
type m_record from m_main`m_record within m_minisalud_main  
end type
```

```
on m_record.create  
call super::create  
end on
```

```
on m_record.destroy  
call super::destroy  
end on
```

```
type m_display from m_main`m_display within m_minisalud_main  
end type
```

```
on m_display.create  
call super::create  
end on
```

```
on m_display.destroy  
call super::destroy
```



end on

**type m_insert from m_main`m_insert within m_minisalud_main
end type**

**on m_insert.create
call super::create
end on**

**on m_insert.destroy
call super::destroy
end on**

**type m_layout from m_main`m_layout within m_minisalud_main
end type**

**on m_layout.create
call super::create
end on**

**on m_layout.destroy
call super::destroy
end on**

**type m_report from m_main`m_report within m_minisalud_main
end type**

**on m_report.create
call super::create**



end on

on m_report.destroy

call super::destroy

end on

type m_run from m_main`m_run within m_minisalud_main

end type

on m_run.create

call super::create

end on

on m_run.destroy

call super::destroy

end on

type m_utilities from m_main`m_utilities within m_minisalud_main

end type

on m_utilities.create

call super::create

end on

on m_utilities.destroy

call super::destroy

end on

type m_toolbaronly from m_main`m_toolbaronly within m_minisalud_main



end type

on m_toolbaronly.create

call super::create

end on

on m_toolbaronly.destroy

call super::destroy

end on

type m_refmenu from m_main`m_refmenu within m_minisalud_main

end type

on m_refmenu.create

call super::create

end on

on m_refmenu.destroy

call super::destroy

end on

type m_options from m_main`m_options within m_minisalud_main

end type

on m_options.create

call super::create

this.shiftright = true

end on



on m_options.destroy

call super::destroy

end on

type m_printscreen from m_main`m_printscreen within m_options

end type

on m_printscreen.create

call super::create

end on

on m_printscreen.destroy

call super::destroy

end on

type m_printwindow from m_main`m_printwindow within m_options

end type

on m_printwindow.create

call super::create

end on

on m_printwindow.destroy

call super::destroy

end on

type m_mbco1 from m_main`m_mbco1 within m_options

end type



```
on m_mbco1.create  
call super::create  
end on
```

```
on m_mbco1.destroy  
call super::destroy  
end on
```

```
type m_previewswitch from m_main`m_previewswitch within m_options  
end type
```

```
on m_previewswitch.create  
call super::create  
end on
```

```
on m_previewswitch.destroy  
call super::destroy  
end on
```

```
type m_mbco2 from m_main`m_mbco2 within m_options  
end type
```

```
on m_mbco2.create  
call super::create  
this.visible = false  
this.enabled = false  
end on
```

```
on m_mbco2.destroy
```



call super::destroy

end on

type m_userpreferences from m_main`m_userpreferences within m_options

end type

on m_userpreferences.create

call super::create

this.visible = false

this.enabled = false

end on

on m_userpreferences.destroy

call super::destroy

end on

type m_window from m_main`m_window within m_minisalud_main

m_menúnavegador m_menúnavegador

end type

on m_window.create

call super::create

this.shiftright = true

this.m_menúnavegador=create m_menúnavegador

this.Item[UpperBound(this.Item)+1]=this.m_menúnavegador

end on

on m_window.destroy

call super::destroy



```
destroy(this.m_menúnavegador)
```

```
end on
```

```
type m_tile from m_main`m_tile within m_window
```

```
end type
```

```
on m_tile.create
```

```
call super::create
```

```
end on
```

```
on m_tile.destroy
```

```
call super::destroy
```

```
end on
```

```
type m_tilehorizontal from m_main`m_tilehorizontal within m_window
```

```
end type
```

```
on m_tilehorizontal.create
```

```
call super::create
```

```
end on
```

```
on m_tilehorizontal.destroy
```

```
call super::destroy
```

```
end on
```

```
type m_cascade from m_main`m_cascade within m_window
```

```
end type
```

```
on m_cascade.create
```



call super::create

end on

on m_cascade.destroy

call super::destroy

end on

type m_layer from m_main`m_layer within m_window

end type

on m_layer.create

call super::create

end on

on m_layer.destroy

call super::destroy

end on

type m_mbcw1 from m_main`m_mbcw1 within m_window

end type

on m_mbcw1.create

call super::create

end on

on m_mbcw1.destroy

call super::destroy

end on



```
type m_arrangeicons from m_main`m_arrangeicons within m_window  
end type
```

```
on m_arrangeicons.create  
call super::create  
end on
```

```
on m_arrangeicons.destroy  
call super::destroy  
end on
```

```
type m_listall from m_main`m_listall within m_window  
end type
```

```
on m_listall.create  
call super::create  
end on
```

```
on m_listall.destroy  
call super::destroy  
end on
```

```
type m_mbcw2 from m_main`m_mbcw2 within m_window  
end type
```

```
on m_mbcw2.create  
call super::create  
end on
```



```
on m_mbcw2.destroy  
call super::destroy  
end on
```

```
type m_showtoolbar from m_main`m_showtoolbar within m_window  
end type
```

```
on m_showtoolbar.create  
call super::create  
end on
```

```
on m_showtoolbar.destroy  
call super::destroy  
end on
```

```
type m_toolbartext from m_main`m_toolbartext within m_window  
end type
```

```
on m_toolbartext.create  
call super::create  
end on
```

```
on m_toolbartext.destroy  
call super::destroy  
end on
```

```
type m_mbcw3 from m_main`m_mbcw3 within m_window  
end type
```



```
on m_mbcw3.create  
call super::create  
end on
```

```
on m_mbcw3.destroy  
call super::destroy  
end on
```

```
type m_rmbsupport from m_main`m_rmbsupport within m_window  
end type
```

```
on m_rmbsupport.create  
call super::create  
end on
```

```
on m_rmbsupport.destroy  
call super::destroy  
end on
```

```
type m_help from m_main`m_help within m_minisalud_main  
end type
```

```
on m_help.create  
call super::create  
this.shifttoright = true  
end on
```

```
on m_help.destroy  
call super::destroy
```



end on

**type m_helpcontents from m_main`m_helpcontents within m_help
end type**

**on m_helpcontents.create
call super::create
end on**

**on m_helpcontents.destroy
call super::destroy
end on**

**type m_searchforhelpon from m_main`m_searchforhelpon within m_help
end type**

**on m_searchforhelpon.create
call super::create
end on**

**on m_searchforhelpon.destroy
call super::destroy
end on**

**type m_mbch1 from m_main`m_mbch1 within m_help
end type**

**on m_mbch1.create
call super::create**



end on

on m_mbch1.destroy

call super::destroy

end on

type m_about from m_main`m_about within m_help

end type

on m_about.create

call super::create

end on

on m_about.destroy

call super::destroy

end on

type m_menúnavegador from menu within m_window

end type

event clicked;window lw_sheet

if isvalid(lw_sheet) then

opensheetwithparm(w_menu_navigator, lw_sheet.menuid, w_frame)

else

opensheetwithparm(w_menu_navigator, w_frame.menuid, w_frame)

end if

end event

on m_menúnavegador.create



```
call super::create
this.text = "Menú Navegador"
this.microhelp = "Menú Navegador"
this.toolbaritemname = "TreeView!"
this.toolbaritemtext = "Menú Navegador"
this.toolbaritemorder = 190
this.toolbaritemspace = 1
end on

on m_menúnavegador.destroy
call super::destroy
end on

type m_tablas from menu within m_minisalud_main
end type

on m_tablas.create
call super::create
this.text = "&Configuracion"
this.enabled = false
end on

on m_tablas.destroy
call super::destroy
end on

type m_operaciones from menucascade within m_minisalud_main
m_alicitante m_alicitante
m_bconsulta m_bconsulta
```



m_cfarmacia m_cfarmacia

end type

on m_operaciones.create

call super::create

this.text = "&Operaciones"

this.enabled = false

this.toolbaritemorder = 25

this.toolbaritemspace = 1

this.dropdown = false

this.columns = 2

this.m_alicitante=create m_alicitante

this.m_bconsulta=create m_bconsulta

this.m_cfarmacia=create m_cfarmacia

this.Item[UpperBound(this.Item)+1]=this.m_alicitante

this.Item[UpperBound(this.Item)+1]=this.m_bconsulta

this.Item[UpperBound(this.Item)+1]=this.m_cfarmacia

end on

on m_operaciones.destroy

call super::destroy

destroy(this.m_alicitante)

destroy(this.m_bconsulta)

destroy(this.m_cfarmacia)

end on

type m_alicitante from menu within m_operaciones

end type



on m_alicitante.create

call super::create

this.text = "&A. Expediente"

this.enabled = false

end on

on m_alicitante.destroy

call super::destroy

end on

event clicked;mf_opensheet('w_lst_expediente')

end event

type m_bconsulta from menu within m_operaciones

end type

event clicked;mf_opensheet('w_lst_consulta')

end event

on m_bconsulta.create

call super::create

this.text = "&B. Consulta"

this.enabled = false

end on

on m_bconsulta.destroy

call super::destroy

end on



```
type m_cfarmacia from menu within m_operaciones
```

```
end type
```

```
event clicked;mf_opensheet('w_lst_farmacia')
```

```
end event
```

```
on m_cfarmacia.create
```

```
call super::create
```

```
this.text = "&C. Diagnostico"
```

```
this.enabled = false
```

```
end on
```

```
on m_cfarmacia.destroy
```

```
call super::destroy
```

```
end on
```

```
type m_procesos from menu within m_minisalud_main
```

```
m_farmacia m_farmacia
```

```
m_laboratorio m_laboratorio
```

```
end type
```

```
on m_procesos.create
```

```
call super::create
```

```
this.text = "&Procesos"
```

```
this.enabled = false
```

```
this.m_farmacia=create m_farmacia
```

```
this.m_laboratorio=create m_laboratorio
```



```
this.Item[UpperBound(this.Item)+1]=this.m_farmacia  
this.Item[UpperBound(this.Item)+1]=this.m_laboratorio  
end on
```

```
on m_procesos.destroy  
call super::destroy  
destroy(this.m_farmacia)  
destroy(this.m_laboratorio)  
end on
```

```
type m_farmacia from menu within m_procesos  
end type
```

```
on m_farmacia.create  
call super::create  
this.text = "&A. Farmacia"  
this.enabled = false  
end on
```

```
on m_farmacia.destroy  
call super::destroy  
end on
```

```
event clicked;Open(w_cita_examen)
```

```
end event
```

```
type m_laboratorio from menu within m_procesos  
end type
```



```
on m_laboratorio.create  
call super::create  
this.text = "&B. Laboratorio"  
this.enabled = false  
end on
```

```
on m_laboratorio.destroy  
call super::destroy  
end on
```

```
event clicked;Open(w_cita_laboratorio)  
end event
```

```
type m_consultasyreportes from menu within m_minisalud_main  
m_arecetas m_arecetas  
m_bexpediente m_bexpediente  
end type
```

```
on m_consultasyreportes.create  
call super::create  
this.text = "&Reportes"  
this.enabled = false  
this.m_arecetas=create m_arecetas  
this.m_bexpediente=create m_bexpediente  
this.Item[UpperBound(this.Item)+1]=this.m_arecetas  
this.Item[UpperBound(this.Item)+1]=this.m_bexpediente  
end on
```



```
on m_consultasyreportes.destroy
```

```
call super::destroy
```

```
destroy(this.m_aretas)
```

```
destroy(this.m_bexpediente)
```

```
end on
```

```
type m_aretas from menu within m_consultasyreportes
```

```
end type
```

```
on m_aretas.create
```

```
call super::create
```

```
this.text = "&A. Recetas"
```

```
this.enabled = false
```

```
end on
```

```
on m_aretas.destroy
```

```
call super::destroy
```

```
end on
```

```
event clicked;mf_opensheet('w_ppv_receta')
```

```
end event
```

```
type m_bexpediente from menu within m_consultasyreportes
```

```
end type
```

```
event clicked;mf_opensheet('w_edt_consulta_reporte1')
```

```
end event
```

```
on m_bexpediente.create
```



```
call super::create  
this.text = "&B. Historial Clínico"  
this.enabled = false  
end on
```

```
on m_bexpediente.destroy  
call super::destroy  
end on
```

VENTANA W_ACERCA_DE

```
forward  
global type w_acerca_de from w_about  
end type  
end forward
```

```
global type w_acerca_de from w_about  
integer width = 1893  
integer height = 1032  
long bgcolor = 15780518  
end type  
global w_acerca_de w_acerca_de
```

```
on w_acerca_de.create  
call super::create  
end on
```

```
on w_acerca_de.destroy
```



call super::destroy

end on

type st_version from w_about`st_version within w_acerca_de

integer x = 1056

integer y = 640

integer width = 553

long backcolor = 15780518

string text = "Versión 1.0.0"

end type

type st_app_subtitle from w_about`st_app_subtitle within w_acerca_de

integer x = 937

integer y = 496

integer width = 773

integer height = 156

integer textsize = -10

fontcharset fontcharset = ansi!

fontfamily fontfamily = swiss!

string facename = "Arial"

long backcolor = 15780518

string text = "Sistema Salud"

end type

type st_app_title from w_about`st_app_title within w_acerca_de

integer x = 1024

integer y = 356

integer width = 622

integer height = 132



```
integer textsize = -26
fontcharset fontcharset = ansi!
fontpitch fontpitch = fixed!
fontfamily fontfamily = modern!
string facename = "Courier New"
long textcolor = 32768
long backcolor = 15780518
string text = "MiniSalud"
fillpattern fillpattern = bdiagonal!
end type
```

```
type st_app_business_title from w_about`st_app_business_title within w_acerca_de
integer x = 773
integer y = 76
integer width = 1065
integer height = 244
integer textsize = -20
fontcharset fontcharset = ansi!
long textcolor = 0
long backcolor = 15780518
string text = "Hospital Rosales"
end type
```

```
type st_author from w_about`st_author within w_acerca_de
integer x = 1051
integer y = 716
integer width = 553
long backcolor = 15780518
string text = ""
```



end type

type gb_1 from w_about`gb_1 within w_acerca_de

integer x = 23

integer y = 12

integer width = 1833

integer height = 788

long textcolor = 15780518

long backcolor = 15780518

end type

type p_app_icon from w_about`p_app_icon within w_acerca_de

integer y = 436

integer height = 280

string picturename = "GRAPHIC\bienestar.bmp"

boolean border = true

borderstyle borderstyle = styleraised!

end type

type p_company_logo from w_about`p_company_logo within w_acerca_de

integer y = 76

integer width = 315

string picturename = "GRAPHIC\escudo.bmp"

boolean border = true

borderstyle borderstyle = styleraised!

end type

type cb_ok from w_about`cb_ok within w_acerca_de

integer x = 690



integer y = 816

end type

type p_winlogo from w_about`p_winlogo within w_acerca_de

integer x = 69

integer y = 808

end type

type st_builddate from w_about`st_builddate within w_acerca_de

integer x = 87

integer y = 724

integer width = 759

integer height = 68

long backcolor = 15780518

alignment alignment = left!

end type

VENTANA W_FRAME

forward

global type w_frame from w_mainmdi

end type

end forward

global type w_frame from w_mainmdi

string title = "Sistema de Salud - Ministerio de Salud"

string menuname = "m_minisalud_main"



```
boolean maxbox = false
end type
global w_frame w_frame

type variables
//u_nv_siges_procesos          inv_procesos
//Declaracion de instancias
unv_seguridad                  inv_seguridad
n_ds                           ids_sistema
end variables

on w_frame.create
call super::create
if IsValid(this.MenuID) then destroy(this.MenuID)
if this.MenuName = "m_minisalud_main" then this.MenuID = create
m_minisalud_main
end on

on w_frame.destroy
call super::destroy
if IsValid(MenuID) then destroy(MenuID)
end on

event ue_postopen;call super::ue_postopen;

//Declaracion de variables
String      ls_rc, ls_titulo, ls_login, ls_servidor
Integer     li_rc, li_anio
```



//Declaracion de instancias

//inv_procesos = create u_nv_siges_procesos

inv_seguridad = create unv_seguridad

//Verificamos si esta configurado el codigo de la Entidad Educativa

**ls_servidor = trim(ProfileString('minisalud.ini',
"Configuracion", "servidor",""))**

//Validar si existe un servidor configurado

If trim(ls_servidor) = "" then

//Crear el archivo 'sapbm.ini'

End if

//Recuperamos el año lectivo vigente, caso contrario la variable tendra el valor de 0

gs_periodo = String(Year(Today()))

//Habilitamos las opciones para el usuario

li_rc = inv_seguridad.uf_habilita_opciones_menu()

//colocamos titulos

ls_titulo = this.title

**this.Title = ls_titulo + ' [Año activo: ' + trim(gs_periodo) + ' Usuario: ' +
trim(gs_login) + '] '**

end event

VENTANA W_SPLASH



forward

global type w_splash from w_welcome

end type

end forward

global type w_splash from w_welcome

integer x = 673

integer y = 453

integer width = 2130

integer height = 840

long bgcolor = 8421376

end type

global w_splash w_splash

on w_splash.create

call super::create

end on

on w_splash.destroy

call super::destroy

end on

type st_version from w_welcome`st_version within w_splash

integer x = 914

integer y = 684

integer width = 1120

integer height = 60

long textcolor = 15793151

long bgcolor = 8421376



string text = "Versión: 1.0.0"

end type

type st_app_subtitle from w_welcome`st_app_subtitle within w_splash

integer x = 933

integer y = 500

integer width = 992

integer height = 80

integer textsize = -13

fontcharset fontcharset = ansi!

fontfamily fontfamily = swiss!

string facename = "Arial"

long backcolor = 8421376

string text = "Sistema de Salud"

end type

type st_app_title from w_welcome`st_app_title within w_splash

integer x = 878

integer y = 320

integer width = 1166

integer height = 160

integer textsize = -25

fontcharset fontcharset = ansi!

fontpitch fontpitch = fixed!

fontfamily fontfamily = modern!

string facename = "Courier New"

long textcolor = 12639424

long backcolor = 8421376

string text = "MiniSalud"



end type

type st_app_business_title from w_welcome`st_app_business_title within w_splash

integer x = 878

integer width = 1134

integer height = 160

integer textsize = -22

fontcharset fontcharset = ansi!

long textcolor = 12639424

long bgcolor = 8421376

string text = "Hospital Rosales"

end type

type st_author from w_welcome`st_author within w_splash

integer x = 914

integer y = 632

integer width = 1120

integer height = 48

long textcolor = 15793151

long bgcolor = 8421376

string text = "Author: "

end type

type gb_1 from w_welcome`gb_1 within w_splash

integer y = 48

integer height = 736

long textcolor = 12632256

long bgcolor = 8421376

borderstyle borderstyle = StyleLowered!



end type

type p_app_icon from w_welcome`p_app_icon within w_splash

integer y = 464

integer width = 777

integer height = 268

string picturename = "Graphics\salud1.jpg"

boolean border = true

borderstyle borderstyle = StyleLowered!

end type

type p_company_logo from w_welcome`p_company_logo within w_splash

integer x = 128

integer y = 148

integer width = 343

integer height = 292

string picturename = "Graphics\msigno2.gif"

boolean border = true

borderstyle borderstyle = StyleLowered!

end type

LIBRERÍA LOGIN (RFYD_LOGIN.PBL)

VENTANA W_LST_USUARIO

forward

global type w_lst_usuario from w_showdata

end type



```
type dw_2 from u_lst within w_lst_usuario
end type
type dw_1 from u_lst within w_lst_usuario
end type
end forward
```

```
global type w_lst_usuario from w_showdata
integer width = 3575
integer height = 1756
boolean titlebar = false
string menuname = "m_lst"
boolean controlmenu = false
boolean minbox = false
boolean maxbox = false
boolean resizable = false
boolean border = false
dw_2 dw_2
dw_1 dw_1
end type
global w_lst_usuario w_lst_usuario
```

```
type variables
long il_rows
end variables
```

```
on w_lst_usuario.create
int iCurrent
call super::create
if this.MenuName = "m_lst" then this.MenuID = create m_lst
```



```
this.dw_2=create dw_2  
this.dw_1=create dw_1  
iCurrent=UpperBound(this.Control)  
this.Control[iCurrent+1]=this.dw_2  
this.Control[iCurrent+2]=this.dw_1  
end on  
  
on w_lst_usuario.destroy  
call super::destroy  
if IsValid(MenuID) then destroy(MenuID)  
destroy(this.dw_2)  
destroy(this.dw_1)  
end on  
  
type p_rowpoint from w_showdata`p_rowpoint within w_lst_usuario  
end type  
  
type p_linkpoint from w_showdata`p_linkpoint within w_lst_usuario  
end type  
  
type st_popup from w_showdata`st_popup within w_lst_usuario  
end type  
  
type dw_2 from u_lst within w_lst_usuario  
integer y = 648  
integer width = 3461  
integer height = 856  
integer taborder = 10  
boolean bringtotop = true
```



```
string dataobject = "d_lst_usuario"
```

```
end type
```

```
event constructor;call
```

```
super::constructor;this.uf_linktomaster(dw_1,'ret','id_grupousuario','id_grupousua  
rio')
```

```
this.is_w_edt_classname = "w_edt_usuario"
```

```
this.is_rpt_dataobject = "d_rpt_usuario"
```

```
this.ib_rpt_allrecords = true
```

```
this.ii_edtlaunchmethod = 1 // 1. recupera un sólo reg. 2. los recupera todos
```

```
end event
```

```
event retrieveend;call super::retrieveend;//se recuperaron registros?
```

```
il_rows = rowcount
```

```
end event
```

```
type dw_1 from u_lst within w_lst_usuario
```

```
integer x = 5
```

```
integer y = 32
```

```
integer width = 2491
```

```
integer height = 572
```

```
integer taborder = 10
```

```
boolean bringtotop = true
```

```
string dataobject = "d_lst_grupousuario"
```

```
end type
```

```
event constructor;call super::constructor;this.is_w_edt_classname =
```

```
"w_edt_grupousuario"
```



```
this.is_rpt_dataobject = "d_rpt_grupousuario"  
this.ib_rpt_allrecords = true  
this.ii_edtlaunchmethod = 1 // Donde 1. recupera un sólo reg. 2. los recupera todos  
end event
```

VENTANA W_LOGON_USUARIO

```
forward  
global type w_logon_usuario from w_logon  
end type  
type p_1 from picture within w_logon_usuario  
end type  
type sle_usuario from singlelineedit within w_logon_usuario  
end type  
type sle_password from singlelineedit within w_logon_usuario  
end type  
type st_1 from statictext within w_logon_usuario  
end type  
type st_2 from statictext within w_logon_usuario  
end type  
type cb_1 from commandbutton within w_logon_usuario  
end type  
end forward  
  
global type w_logon_usuario from w_logon  
integer width = 1856  
integer height = 452  
string title = "Ingreso al Sistema de Salud"
```



```
long bgcolor = 15780518
p_1 p_1
sle_usuario sle_usuario
sle_password sle_password
st_1 st_1
st_2 st_2
cb_1 cb_1
end type
global w_logon_usuario w_logon_usuario

type variables
//Declaración de instancias
u_nv_cst_encryptor inv_encryptor
n_extfunctwin32          in_fwin32
end variables

on w_logon_usuario.create
int iCurrent
call super::create
this.p_1=create p_1
this.sle_usuario=create sle_usuario
this.sle_password=create sle_password
this.st_1=create st_1
this.st_2=create st_2
this.cb_1=create cb_1
iCurrent=UpperBound(this.Control)
this.Control[iCurrent+1]=this.p_1
this.Control[iCurrent+2]=this.sle_usuario
this.Control[iCurrent+3]=this.sle_password
```



```
this.Control[iCurrent+4]=this.st_1  
this.Control[iCurrent+5]=this.st_2  
this.Control[iCurrent+6]=this.cb_1  
end on
```

```
on w_logon_usuario.destroy  
call super::destroy  
destroy(this.p_1)  
destroy(this.sle_usuario)  
destroy(this.sle_password)  
destroy(this.st_1)  
destroy(this.st_2)  
destroy(this.cb_1)  
end on
```

```
event open;call super::open;
```

```
//Declaración de variables
```

```
String ls_user_logon
```

```
String rc
```

```
//Construcción de instancias
```

```
inv_encryptor      = create u_nv_cst_encryptor
```

```
in_fwin32          = create n_extfunctwin32
```

```
//Realizar conexión con la base de datos
```

```
dw_logon.SetItem(1,'logid','misalud')
```

```
dw_logon.SetItem(1,'logpass','misalud')
```

```
cb_ok.trigger event clicked()
```



```
//Asignación de usuario de NT
in_fwin32.uf_get_logon_name(ls_user_logon)
sle_usuario.text = trim(ls_user_logon)
sle_password.setfocus()

end event

type dw_logon from w_logon`dw_logon within w_logon_usuario
boolean visible = false
integer x = 434
integer y = 416
integer height = 616
integer taborder = 30
end type

type cb_ok from w_logon`cb_ok within w_logon_usuario
boolean visible = false
integer x = 1339
integer y = 44
integer taborder = 0
boolean default = false
end type

event cb_ok::clicked;int rc
string
ls_logid,ls_logpass,ls_servername,ls_database,ls_dbms,ls_dbparm,ls_connectdsn
```



```
if dw_logon.accepttext() < 1 then
```

```
    return 1
```

```
end if
```

```
setpointer(hourglass!)
```

```
//get dw_logon fields if present on dataobject
```

```
if wf_fieldok("logid") = 1 then ls_logid = dw_logon.getitemstring(1,"logid")
```

```
if wf_fieldok("logpass") = 1 then ls_logpass = dw_logon.getitemstring(1,"logpass")
```

```
if wf_fieldok("servername") = 1 then ls_servername =  
dw_logon.getitemstring(1,"servername")
```

```
if wf_fieldok("database") = 1 then ls_database =  
dw_logon.getitemstring(1,"database")
```

```
if wf_fieldok("connectdsn") = 1 then ls_connectdsn =  
dw_logon.getitemstring(1,"connectdsn")
```

```
in_transobj.logid = ls_logid
```

```
in_transobj.logpass = ls_logpass
```

```
in_transobj.servername = ls_servername
```

```
in_transobj.database = ls_database
```

```
in_transobj.is_connectdsn = ls_connectdsn
```

```
if not isvalid(n_connect_mgr) then
```

```
    messagebox("Connection Manager Not Running","Unable to Connect  
Database Services")
```

```
    return 1
```

```
end if
```

```
//keep track of the number of attempts to logon to this transaction object
```



```
parent.ii_nbr_logon_attempts ++
```

```
if parent.ii_nbr_logon_attempts > parent.ii_limit_logon_attempts then
```

```
    parent.event ue_logonlimitreached()           //show message or
```

```
whatever is required
```

```
    closewithreturn(parent,-1)
```

```
    return 1
```

```
end if
```

```
//could already be connected - so will need to disconnect first
```

```
rc = in_transobj.uf_reconnect()
```

```
//check success/failure of dbms connection
```

```
if rc < 0 then
```

```
    rc = n_connect_mgr.event ue_logonfailure()
```

```
    if rc = 1 then           //not handled by connection manager so indicate
```

```
here
```

```
        messagebox("Logon Failed","Unable to connect to  
Database.~n~r"+in_transobj.sqlerrtext)
```

```
    end if
```

```
    return 1
```

```
end if
```

```
//connect succeeded ! - return to application
```

```
setpointer(hourglass!)
```

```
//after logon success take additional action
```

```
n_connect_mgr.event ue_postlogon()
```



```
//closewithreturn(parent,1)
```

```
//return 1
```

```
end event
```

```
type cb_cancel from w_logon`cb_cancel within w_logon_usuario
```

```
integer x = 1440
```

```
integer y = 172
```

```
integer taborder = 50
```

```
end type
```

```
event cb_cancel::clicked;
```

```
//Declaracion de variables
```

```
Integer li_rc
```

```
li_rc = MessageBox('Cancelando ingreso','Ha decidido cancelar su ingreso al  
sistema. Desea '+&
```

```
          'continuar y cerrar el sistema totalmente?.', question!,yesno!,2)
```

```
if li_rc = 1 then halt close
```

```
end event
```

```
type cb_more from w_logon`cb_more within w_logon_usuario
```

```
boolean visible = false
```

```
integer x = 1691
```

```
integer y = 500
```

```
integer taborder = 60
```

```
boolean enabled = false
```



end type

type p_1 from picture within w_logon_usuario

integer x = 18

integer y = 28

integer width = 274

integer height = 336

boolean bringtotop = true

boolean originalsize = true

string picturename = "Graphics\salud.gif"

boolean focusrectangle = false

end type

type sle_usuario from singlelineedit within w_logon_usuario

integer x = 818

integer y = 72

integer width = 585

integer height = 92

integer taborder = 10

boolean bringtotop = true

integer textsize = -8

integer weight = 700

fontcharset fontcharset = ansi!

fontpitch fontpitch = variable!

fontfamily fontfamily = swiss!

string facename = "MS Sans Serif"

long textcolor = 33554432

textcase textcase = upper!

borderstyle borderstyle = stylelowered!



end type

type sle_password from singlelineedit within w_logon_usuario

integer x = 818

integer y = 176

integer width = 585

integer height = 92

integer taborder = 20

boolean bringtotop = true

integer textsize = -8

integer weight = 700

fontcharset fontcharset = ansi!

fontpitch fontpitch = variable!

fontfamily fontfamily = swiss!

string facename = "MS Sans Serif"

long textcolor = 33554432

boolean password = true

textcase textcase = upper!

borderstyle borderstyle = stylelowered!

end type

type st_1 from statictext within w_logon_usuario

integer x = 567

integer y = 88

integer width = 242

integer height = 52

boolean bringtotop = true

integer textsize = -8

integer weight = 700



```
fontcharset fontcharset = ansi!  
fontpitch fontpitch = variable!  
fontfamily fontfamily = swiss!  
string facename = "MS Sans Serif"  
long textcolor = 33554432  
long backcolor = 15780518  
string text = "Usuario:"  
alignment alignment = right!  
boolean focusrectangle = false  
end type
```

```
type st_2 from statictext within w_logon_usuario  
integer x = 526  
integer y = 196  
integer width = 283  
integer height = 52  
boolean bringtotop = true  
integer textsize = -8  
integer weight = 700  
fontcharset fontcharset = ansi!  
fontpitch fontpitch = variable!  
fontfamily fontfamily = swiss!  
string facename = "MS Sans Serif"  
long textcolor = 33554432  
long backcolor = 15780518  
string text = "Password:"  
alignment alignment = right!  
boolean focusrectangle = false  
end type
```



```
type cb_1 from commandbutton within w_logon_usuario
integer x = 1440
integer y = 60
integer width = 343
integer height = 104
integer taborder = 40
integer textsize = -8
integer weight = 700
fontcharset fontcharset = ansi!
fontpitch fontpitch = variable!
fontfamily fontfamily = swiss!
string facename = "MS Sans Serif"
string text = "&Aceptar"
boolean default = true
end type

event clicked;

//Declaracion de variables
String ls_usr, ls_password, ls_usr_digitado, ls_pass_digitado, ls_calculado
String ls_grupo, ls_nombre, ls_apellido, ls_ui, ls_ub, ls_um, ls_gi, ls_gb, ls_gm

//VALIDACIONES*****
*****

//Verificacion si se ha digitado el usuario
ls_usr_digitado = TRIM(sle_usuario.text)
if isnull(ls_usr_digitado) then
    MessageBox('Error','Debe digitar el Usuario')
```



```
        return
    end if

//Verificacion si se ha digitado el Password
ls_pass_digitado = trim(sle_password.text)
if isnull(ls_pass_digitado) then
    MessageBox('Error','Debe digitar el Password')
    return
end if

//QUITAR                                                    EL
ENCRYPTADO*****
*****

//Encriptamos el password [DESHABILITAR EL PASSWORD ENCRYPTADO.
TEMPORALMENTE]
inv_encryptor.of_set(trim(ls_usr_digitado),trim(ls_pass_digitado))
ls_pass_digitado = inv_encryptor.of_getEncrypted()

//BUSCAR            EN            LA            BASE            DE
DATOS*****

//Seleccionamos el Usuario y Password creado por el cliente en la tabla USUARIO
Select u.login,
        u.password,
        u.id_usuario,
        u.id_grupousuario,
        u.nombre_usuario,
        u.apellido_usuario,
        u.departamento,
        u.perm_inserta,
```



**u.perm_borra,
u.perm_modifica,
gu.perm_inserta,
gu.perm_borra,
gu.perm_modifica**

**Into :ls_usr,
:ls_password,
:gs_codigo_usuario,
:gs_grupo,
:ls_nombre,
:ls_apellido,
:gs_puesto,
:ls_ui,
:ls_ub,
:ls_um,
:ls_gi,
:ls_gb,
:ls_gm**

From usuario u

join grupo_usuario gu on gu.id_grupousuario = u.id_grupousuario

Where u.login =:ls_usr_digitado

And u.password =:ls_pass_digitado

And u.estado_usuario = '1' And gu.estado = '1';

**//Verificamos su existencia y regresamos a la pantalla para pedirle nuevamente los
datos**

If sqlca.sqlcode<>0 then

**messagebox('ERROR AL INGRESAR AL SISTEMA','POR FAVOR
VERIFIQUE SUS DATOS.'+&**



' SI ES UN USUARIO ACTIVO VERIFIQUE SU USUARIO Y
PASSWORD. ~r~n'+&

'SI NO TIENE CUENTA, PONGASE EN CONTACTO CON EL
ADMINISTRADOR '+&

"DEL SISTEMA",EXCLAMATION!)

sle_usuario.setfocus()

Return

End if

//Almacenar datos de usuario

gs_login = ls_usr_digitado

gs_nombre_usuario = trim(ls_nombre) + ' , ' + trim(ls_apellido)

//Asignar los permisos de usuario de insertar, borrar y modificar

if ls_gi = "" or isnull(ls_gi) then gs_i = ls_ui Else gs_i = ls_gi

if ls_gb = "" or isnull(ls_gb) then gs_b = ls_ub Else gs_b = ls_gb

if ls_gm = "" or isnull(ls_gm) then gs_m = ls_um Else gs_m = ls_gm

//Salir de la ventana

closewithreturn(parent,1)

//Retornar valor

return 1

end event

VENTANA W_LOGON_ACCESO

forward



```
global type w_logon_acceso from w_logon
end type
type p_1 from picture within w_logon_acceso
end type
type sle_usuario from singlelineedit within w_logon_acceso
end type
type sle_password from singlelineedit within w_logon_acceso
end type
type st_1 from statictext within w_logon_acceso
end type
type st_2 from statictext within w_logon_acceso
end type
type cb_1 from commandbutton within w_logon_acceso
end type
end forward
```

```
global type w_logon_acceso from w_logon
integer width = 1856
integer height = 452
string title = "Ingreso al Sistema de Salud"
long bgcolor = 15780518
p_1 p_1
sle_usuario sle_usuario
sle_password sle_password
st_1 st_1
st_2 st_2
cb_1 cb_1
end type
global w_logon_acceso w_logon_acceso
```



type variables

//Declaración de instancias

u_nv_cst_encryptor inv_encryptor

n_extfunctwin32 in_fwin32

end variables

on w_logon_acceso.create

int iCurrent

call super::create

this.p_1=create p_1

this.sle_usuario=create sle_usuario

this.sle_password=create sle_password

this.st_1=create st_1

this.st_2=create st_2

this.cb_1=create cb_1

iCurrent=UpperBound(this.Control)

this.Control[iCurrent+1]=this.p_1

this.Control[iCurrent+2]=this.sle_usuario

this.Control[iCurrent+3]=this.sle_password

this.Control[iCurrent+4]=this.st_1

this.Control[iCurrent+5]=this.st_2

this.Control[iCurrent+6]=this.cb_1

end on

on w_logon_acceso.destroy

call super::destroy

destroy(this.p_1)

destroy(this.sle_usuario)



```
destroy(this.sle_password)
destroy(this.st_1)
destroy(this.st_2)
destroy(this.cb_1)
end on

event open;call super::open;

//Declaración de variables
String ls_user_logon
String rc
//Construcción de instancias
inv_encryptor      = create u_nv_cst_encryptor
in_fwin32          = create n_extfunctwin32

//Realizar conexión con la base de datos
dw_logon.SetItem(1,'logid','misalud')
dw_logon.SetItem(1,'logpass','misalud')
cb_ok.trigger event clicked()

//Asignación de usuario de NT
in_fwin32.uf_get_logon_name(ls_user_logon)
sle_usuario.text = trim(ls_user_logon)
sle_password.setfocus()

end event
```



```
type dw_logon from w_logon`dw_logon within w_logon_acceso
```

```
boolean visible = false
```

```
integer x = 434
```

```
integer y = 416
```

```
integer height = 616
```

```
integer taborder = 30
```

```
end type
```

```
type cb_ok from w_logon`cb_ok within w_logon_acceso
```

```
boolean visible = false
```

```
integer x = 1339
```

```
integer y = 44
```

```
integer taborder = 0
```

```
boolean default = false
```

```
end type
```

```
event cb_ok::clicked;int rc
```

```
string
```

```
ls_logid,ls_logpass,ls_servername,ls_database,ls_dbms,ls_dbparm,ls_connectdsn
```

```
if dw_logon.accepttext() < 1 then
```

```
    return 1
```

```
end if
```

```
setpointer(hourglass!)
```

```
//get dw_logon fields if present on dataobject
```

```
if wf_fieldok("logid") = 1 then ls_logid =
```

```
dw_logon.getitemstring(1,"logid")
```



```
if wf_fieldok("logpass")      =      1      then      ls_logpass      =
dw_logon.getitemstring(1,"logpass")
if wf_fieldok("servername")   =      1      then      ls_servername   =
dw_logon.getitemstring(1,"servername")
if wf_fieldok("database")     =      1      then      ls_database     =
dw_logon.getitemstring(1,"database")
if wf_fieldok("connectdsn")   =      1      then      ls_connectdsn   =
dw_logon.getitemstring(1,"connectdsn")

in_transobj.logid            = ls_logid
in_transobj.logpass          = ls_logpass
in_transobj.servername      = ls_servername
in_transobj.database        = ls_database
in_transobj.is_connectdsn   = ls_connectdsn
//in_transobj.lock           = "TS"
//messagebox("Lock", in_transobj.lock)

if not isvalid(n_connect_mgr) then
    messagebox("Connection Manager Not Running","Unable to Connect
Database Services")
    return 1
end if

//keep track of the number of attempts to logon to this transaction object
parent.ii_nbr_logon_attempts ++

if parent.ii_nbr_logon_attempts > parent.ii_limit_logon_attempts then
    parent.event ue_logonlimitreached()           //show message or
whatever is required
```



```
closewithreturn(parent,-1)
return 1
end if

//could already be connected - so will need to disconnect first
rc = in_transobj.uf_reconnect()

//check success/failure of dbms connection
if rc < 0 then
    rc = n_connect_mgr.event ue_logonfailure()
    if rc = 1 then //not handled by connection manager so indicate
here
        messagebox("Logon Failed","Unable to connect to
Database.~n~r"+in_transobj.sqlerrtext)
    end if
    return 1
end if

//connect succeeded ! - return to application
setpointer(hourglass!)

//after logon success take additional action
n_connect_mgr.event ue_postlogon()

//closewithreturn(parent,1)
//return 1
end event

type cb_cancel from w_logon`cb_cancel within w_logon_acceso
```



integer x = 1440

integer y = 172

integer taborder = 50

end type

event cb_cancel::clicked;

//Declaracion de variables

Integer li_rc

li_rc = MessageBox('Cancelando ingreso','Ha decidido cancelar su ingreso al sistema. Desea '+&

'continuar y cerrar el sistema totalmente?.', question!,yesno!,2)

if li_rc = 1 then halt close

end event

type cb_more from w_logon`cb_more within w_logon_acceso

boolean visible = false

integer x = 1691

integer y = 500

integer taborder = 60

boolean enabled = false

end type

type p_1 from picture within w_logon_acceso

integer x = 18

integer y = 28

integer width = 274

integer height = 336

boolean bringtotop = true



```
boolean originalsize = true
string picturename = "Graphics\salud.gif"
boolean focusrectangle = false
end type
```

```
type sle_usuario from singlelineedit within w_logon_acceso
integer x = 818
integer y = 72
integer width = 585
integer height = 92
integer taborder = 10
boolean bringtotop = true
integer textsize = -8
integer weight = 700
fontcharset fontcharset = ansi!
fontpitch fontpitch = variable!
fontfamily fontfamily = swiss!
string facename = "MS Sans Serif"
long textcolor = 33554432
textcase textcase = upper!
borderstyle borderstyle = stylelowered!
end type
```

```
type sle_password from singlelineedit within w_logon_acceso
integer x = 818
integer y = 176
integer width = 585
integer height = 92
integer taborder = 20
```



```
boolean bringtotop = true
integer textsize = -8
integer weight = 700
fontcharset fontcharset = ansi!
fontpitch fontpitch = variable!
fontfamily fontfamily = swiss!
string facename = "MS Sans Serif"
long textcolor = 33554432
boolean password = true
textcase textcase = upper!
borderstyle borderstyle = stylelowered!
end type

type st_1 from statictext within w_logon_acceso
integer x = 567
integer y = 88
integer width = 242
integer height = 52
boolean bringtotop = true
integer textsize = -8
integer weight = 700
fontcharset fontcharset = ansi!
fontpitch fontpitch = variable!
fontfamily fontfamily = swiss!
string facename = "MS Sans Serif"
long textcolor = 33554432
long backcolor = 15780518
string text = "Usuario:"
alignment alignment = right!
```



boolean focusrectangle = false

end type

type st_2 from statictext within w_logon_acceso

integer x = 526

integer y = 196

integer width = 283

integer height = 52

boolean bringtotop = true

integer textsize = -8

integer weight = 700

fontcharset fontcharset = ansi!

fontpitch fontpitch = variable!

fontfamily fontfamily = swiss!

string facename = "MS Sans Serif"

long textcolor = 33554432

long backcolor = 15780518

string text = "Password:"

alignment alignment = right!

boolean focusrectangle = false

end type

type cb_1 from commandbutton within w_logon_acceso

integer x = 1440

integer y = 60

integer width = 343

integer height = 104

integer taborder = 40

integer textsize = -8



```
integer weight = 700
fontcharset fontcharset = ansi!
fontpitch fontpitch = variable!
fontfamily fontfamily = swiss!
string facename = "MS Sans Serif"
string text = "&Aceptar"
boolean default = true
end type

event clicked;

//Declaracion de variables
String ls_usr, ls_password, ls_usr_digitado, ls_pass_digitado, ls_calculado
String ls_grupo, ls_nombre, ls_apellido, ls_ui, ls_ub, ls_um, ls_gi, ls_gb, ls_gm

//VALIDACIONES*****
*****

//Verificacion si se ha digitado el usuario
ls_usr_digitado = TRIM(sle_usuario.text)
if isnull(ls_usr_digitado) then
    MessageBox('Error','Debe digitar el Usuario')
    return
end if

//Verificacion si se ha digitado el Password
ls_pass_digitado = trim(sle_password.text)
if isnull(ls_pass_digitado) then
    MessageBox('Error','Debe digitar el Password')
    return
```



end if

```
//QUITAR EL  
ENCRYPTADO*****  
*****
```

```
//Encriptamos el password [DESHABILITAR EL PASSWORD ENCRYPTADO.  
TEMPORALMENTE]  
inv_encryptor.of_set(trim(ls_usr_digitado),trim(ls_pass_digitado))  
ls_pass_digitado = inv_encryptor.of_getEncrypted()
```

```
//BUSCAR EN LA BASE DE  
DATOS*****
```

```
//Seleccionamos el Usuario y Password creado por el cliente en la tabla USUARIO
```

```
Select u.login,  
u.password,  
u.id_usuario,  
u.id_grupousuario,  
u.nombre_usuario,  
u.apellido_usuario,  
u.departamento,  
u.perm_inserta,  
u.perm_borra,  
u.perm_modifica,  
gu.perm_inserta,  
gu.perm_borra,  
gu.perm_modifica  
  
Into :ls_usr,  
:ls_password,  
:gs_codigo_usuario,
```



**:gs_grupo,
:ls_nombre,
:ls_apellido,
:gs_puesto,
:ls_ui,
:ls_ub,
:ls_um,
:ls_gi,
:ls_gb,
:ls_gm**

From usuario u

join grupo_usuario gu on gu.id_grupousuario = u.id_grupousuario

Where u.login =:ls_usr_digitado

And u.password =:ls_pass_digitado

And u.estado_usuario = '1' And gu.estado = '1';

**//Verificamos su existencia y regresamos a la pantalla para pedirle nuevamente los
datos**

If sqlca.sqlcode<>0 then

**messagebox('ERROR AL INGRESAR AL SISTEMA','POR FAVOR
VERIFIQUE SUS DATOS.'+&**

**' SI ES UN USUARIO ACTIVO VERIFIQUE SU USUARIO Y
PASSWORD. ~r~n'+&**

**'SI NO TIENE CUENTA, PONGASE EN CONTACTO CON EL
ADMINISTRADOR '+&**

"DEL SISTEMA",EXCLAMATION!)

sle_usuario.setfocus()

Return



End if

//Almacenar datos de usuario

gs_login = ls_usr_digitado

gs_nombre_usuario = trim(ls_nombre) + ', ' + trim(ls_apellido)

//Asignar los permisos de usuario de insertar, borrar y modificar

if ls_gi = '' or isnull(ls_gi) then gs_i = ls_ui Else gs_i = ls_gi

if ls_gb = '' or isnull(ls_gb) then gs_b = ls_ub Else gs_b = ls_gb

if ls_gm = '' or isnull(ls_gm) then gs_m = ls_um Else gs_m = ls_gm

//Salir de la ventana

closewithreturn(parent,1)

//Retornar valor

return 1

end event

VENTANA W_EDT_USUARIO

forward

global type w_edt_usuario from w_edt

end type

type p_1 from picture within w_edt_usuario

end type

type tab_1 from tab within w_edt_usuario

end type

type tabpage_1 from userobject within tab_1



```
end type
type dw_1 from u_dw within tabpage_1
end type
type tabpage_1 from userobject within tab_1
dw_1 dw_1
end type
type tab_1 from tab within w_edt_usuario
tabpage_1 tabpage_1
end type
type cb_1 from commandbutton within w_edt_usuario
end type
end forward

global type w_edt_usuario from w_edt
boolean visible = true
integer width = 3264
integer height = 1428
string title = "Edición de Usuarios"
p_1 p_1
tab_1 tab_1
cb_1 cb_1
end type
global w_edt_usuario w_edt_usuario

type variables
//Declaracion de variables
String is_password, is_idusuario
Boolean ib_generado = false
Integer ii_esinsercion
```



Boolean

lb_select_usuario, ib_select_usuario_io

//Declaracion de instancias

n_ds

ids_data

u_nv_cst_encryptor inv_encryptor

end variables

on w_edt_usuario.create

int iCurrent

call super::create

this.p_1=create p_1

this.tab_1=create tab_1

this.cb_1=create cb_1

iCurrent=UpperBound(this.Control)

this.Control[iCurrent+1]=this.p_1

this.Control[iCurrent+2]=this.tab_1

this.Control[iCurrent+3]=this.cb_1

end on

on w_edt_usuario.destroy

call super::destroy

destroy(this.p_1)

destroy(this.tab_1)

destroy(this.cb_1)

end on

event open;call super::open;//Verificar en que modalidad esta la ventana

If this.event ue_is_insertmode() = 1 Then cb_1.Enabled = False Else

cb_1.Enabled = True



//Inicializar variables

```
is_password = ""  
ii_esinsercion = this.event ue_is_insertmode()
```

//Creación de instancias

```
inv_encryptor = create u_nv_cst_encryptor  
end event
```

```
type p_rowfocus from w_edt`p_rowfocus within w_edt_usuario  
end type
```

```
type cb_cancel from w_edt`cb_cancel within w_edt_usuario  
integer x = 2853  
integer y = 1184  
integer height = 88  
end type
```

```
type cb_ok from w_edt`cb_ok within w_edt_usuario  
integer x = 2469  
integer y = 1184  
integer height = 88  
end type
```

```
type p_1 from picture within w_edt_usuario  
integer x = 37  
integer y = 24  
integer width = 539  
integer height = 1144
```



```
boolean bringtotop = true
string picturename = "GRAPHIC\wizard.bmp"
boolean border = true
borderstyle borderstyle = stylelowered!
boolean focusrectangle = false
end type
```

```
type tab_1 from tab within w_edt_usuario
event create ( )
event destroy ( )
integer x = 626
integer y = 20
integer width = 2592
integer height = 1148
integer taborder = 10
boolean bringtotop = true
integer textsize = -8
integer weight = 700
fontcharset fontcharset = ansi!
fontpitch fontpitch = variable!
fontfamily fontfamily = swiss!
string facename = "MS Sans Serif"
long backcolor = 12632256
boolean raggedright = true
boolean focusonbuttondown = true
integer selectedtab = 1
tabpage_1 tabpage_1
end type
```



```
on tab_1.create  
this.tabpage_1=create tabpage_1  
this.Control[]={this.tabpage_1}  
end on
```

```
on tab_1.destroy  
destroy(this.tabpage_1)  
end on
```

```
type tabpage_1 from userobject within tab_1  
event create ( )  
event destroy ( )  
integer x = 18  
integer y = 100  
integer width = 2555  
integer height = 1032  
long bgcolor = 12632256  
string text = "Datos de usuario"  
long tabtextcolor = 33554432  
long tabbgcolor = 12632256  
long picturemaskcolor = 536870912  
dw_1 dw_1  
end type
```

```
on tabpage_1.create  
this.dw_1=create dw_1  
this.Control[]={this.dw_1}  
end on
```



```
on tabPage_1.destroy
```

```
destroy(this.dw_1)
```

```
end on
```

```
type dw_1 from u_dw within tabPage_1
```

```
integer x = 23
```

```
integer y = 16
```

```
integer width = 2546
```

```
integer height = 1008
```

```
integer taborder = 20
```

```
string dataobject = "d_edt_usuario"
```

```
boolean ib_master = true
```

```
end type
```

```
event retrieveend;call super::retrieveend;
```

```
//Obtener password del usuario
```

```
is_password = this.GetItemString(this.GetRow(),'password')
```

```
is_idusuario = this.GetItemString(this.GetRow(), 'id_usuario')
```

```
end event
```

```
event ue_validaterow;call super::ue_validaterow;
```

```
//Declaracion de variables
```

```
integer rc, li_id_usr
```

```
String ls_nombre, ls_apellido, ls_login, ls_password, ls_calculo, ls_estado,
```

```
ls_grupo, ls_id_usr
```

```
//Cargar datos en las variables
```

```
ls_nombre = this.GetItemString(al_row,'nombre_usuario')
```

```
ls_apellido = this.GetItemString(al_row,'apellido_usuario')
```



```
ls_login          = this.GetItemString(al_row,'login')
ls_password       = this.GetItemString(al_row,'password')
ls_estado         = this.GetItemString(al_row,'estado_usuario')
ls_grupo          = this.GetItemString(al_row,'id_grupousuario')
```

```
//Genera codigo de usuario
```

```
//Verificar si se esta en modo de insercion de datos
```

```
IF ii_esinsercion = 1 THEN
```

```
    //Consultar el codigo maximo
```

```
    SELECT max(id_usuario) INTO :li_id_usr FROM USUARIO;
```

```
    IF IsNull(li_id_usr) THEN
```

```
        ls_id_usr = '01'
```

```
    ELSE
```

```
        li_id_usr++
```

```
        if li_id_usr <10 then
```

```
            ls_id_usr = '0'+string(li_id_usr)
```

```
        else
```

```
            ls_id_usr = string(li_id_usr)
```

```
        end if
```

```
    END IF
```

```
    //Asigna el codigo al campo
```

```
    THIS.SetItem(al_row,'id_usuario',ls_id_usr)
```

```
End if
```

```
//Validar si los campos no estan nulos
```

```
//Campo de estado de usuario
```

```
if isnull(ls_grupo) or trim(ls_grupo)="" then
```

```
    rc = this.uf_setmsgerror(al_row,"id_grupousuario","Debe seleccionar el  
grupo a que pertenece el usuario")
```



```
End if  
//Campo de nombre  
if isnull(ls_nombre) or trim(ls_nombre)="" then  
    rc = this.uf_setmsgerror(al_row,"nombre_usuario","Debe digitar el Nombre  
del Usuario")  
end if  
//Campo de apellido  
if isnull(ls_apellido) or trim(ls_apellido)="" then  
    rc = this.uf_setmsgerror(al_row,"apellido_usuario","Debe digitar el Apellido  
del Usuario")  
end if  
//Campo de login de sistema  
if isnull(ls_login) or trim(ls_login)="" then  
    rc = this.uf_setmsgerror(al_row,"login","Debe digitar el Login para entrar al  
Sistema")  
end if  
//Campo de password de usuario  
if isnull(ls_password) or trim(ls_password)="" then  
    rc = this.uf_setmsgerror(al_row,"password","Debe digitar el Password")  
End if  
//Campo de estado de usuario  
if isnull(ls_estado) or trim(ls_estado)="" then  
    rc = this.uf_setmsgerror(al_row,"estado_usuario","Debe seleccionar el estado  
del usuario")  
End if  
  
//Retornar valores  
return rc  
end event
```



```
event ue_preupdate;call super::ue_preupdate;
```

```
//Declaracion de variables
```

```
String ls_login, ls_password
```

```
//Declaracion de variables
```

```
Long ll_row
```

```
//Verificar si el registro se esta eliminando
```

```
If TRUE Then
```

```
    //Obtener indice de registro
```

```
    ll_row = this.GetRow()
```

```
    //Determinar si ha cambiado el campo de password
```

```
    //Cargar password
```

```
    ls_password = this.GetItemString(ll_row,'password')
```

```
    //Cargar el login del usuario
```

```
    ls_login = Trim(this.GetItemString(ll_row,'login'))
```

```
    If isnull(ls_login) Then ls_login = "
```

```
    //Verificar si el login esta nulo
```

```
    If ls_login = " then
```

```
        MessageBox('Error en el Login del Usuario','Debe digitar el Login  
antes de crear el '+&
```

```
        'Password',exclamation!)
```

```
    Else
```

```
        //Comparar si el password a cambiado
```

```
        if ls_password <> is_password then
```



```
//Encriptar el password  
inv_encryptor.of_set(ls_login, ls_password)  
ls_password = trim(inv_encryptor.of_getEncrypted())  
//almacenar el password  
tab_1.tabpage_1.dw_1.SetItem(ll_row,'password', ls_password)  
end if  
end if  
End if  
  
//Retonar valores  
Return 1  
end event  
  
type cb_1 from commandbutton within w_edt_usuario  
integer x = 2085  
integer y = 1184  
integer width = 361  
integer height = 88  
integer taborder = 30  
boolean bringtotop = true  
integer textsize = -8  
integer weight = 700  
fontcharset fontcharset = ansi!  
fontpitch fontpitch = variable!  
fontfamily fontfamily = swiss!  
string facename = "MS Sans Serif"  
string text = "&Opciones"  
end type
```



event clicked;

//Declaracion de variables

String ls_parametro

//Construir parametro

ls_parametro = '1'+ is_idusuario

//Abrir ventana de configuracion de opciones

OpenWithParm(w_asigna_opcion, ls_parametro)

end event

VENTANA W_EDT_GRUPO_USUARIO

forward

global type w_edt_grupousuario from w_edt

end type

type cb_1 from commandbutton within w_edt_grupousuario

end type

type dw_1 from u_dw within w_edt_grupousuario

end type

end forward

global type w_edt_grupousuario from w_edt

boolean visible = true

integer width = 2002

integer height = 496



```
string title = "Edición del Grupo de Usuarios"  
cb_1 cb_1  
dw_1 dw_1  
end type  
global w_edt_grupousuario w_edt_grupousuario  
  
type variables  
//Declaracion de variables  
String is_idgrupousuario  
end variables  
  
on w_edt_grupousuario.create  
int iCurrent  
call super::create  
this.cb_1=create cb_1  
this.dw_1=create dw_1  
iCurrent=UpperBound(this.Control)  
this.Control[iCurrent+1]=this.cb_1  
this.Control[iCurrent+2]=this.dw_1  
end on  
  
on w_edt_grupousuario.destroy  
call super::destroy  
destroy(this.cb_1)  
destroy(this.dw_1)  
end on  
  
event open;call super::open;//Verificar en que modalidad esta la ventana
```



```
If this.event ue_is_insertmode() = 1 Then cb_1.Enabled = False Else  
cb_1.Enabled = True
```

```
end event
```

```
type p_rowfocus from w_edt`p_rowfocus within w_edt_grupousuario  
end type
```

```
type cb_cancel from w_edt`cb_cancel within w_edt_grupousuario  
integer x = 1577  
integer y = 292  
integer height = 88  
end type
```

```
type cb_ok from w_edt`cb_ok within w_edt_grupousuario  
integer x = 1193  
integer y = 292  
integer height = 88  
end type
```

```
type cb_1 from commandbutton within w_edt_grupousuario  
integer x = 809  
integer y = 292  
integer width = 361  
integer height = 88  
integer taborder = 30  
boolean bringtotop = true  
integer textsize = -8  
integer weight = 700
```



```
fontcharset fontcharset = ansi!  
fontpitch fontpitch = variable!  
fontfamily fontfamily = swiss!  
string facename = "MS Sans Serif"  
string text = "&Opciones"  
end type  
  
event clicked;  
//Declaracion de variables  
String ls_parametro  
  
//Construir parametro  
ls_parametro = '2'+ is_idgrupousuario  
  
//Abrir ventana de configuracion de opciones  
OpenWithParm(w_asigna_opcion, ls_parametro)  
end event  
  
type dw_1 from u_dw within w_edt_grupousuario  
integer width = 1979  
integer height = 288  
integer taborder = 10  
boolean bringtotop = true  
string dataobject = "d_edt_grupousuario"  
boolean ib_master = true  
end type  
  
event retrieveend;call super::retrieveend;
```



```
//Obtener el grupo del usuario
is_idgrupousuario = this.GetItemString(this.GetRow(), 'id_grupousuario')
end event

event ue_validaterow;call super::ue_validaterow;

//Declaracion de variables
integer rc
String ls_grupo, ls_estado
//Declaracion de variables
integer li_id_usr
String ls_id_usr

//Cargar datos en las variables
ls_grupo = this.GetItemString(al_row,'id_grupousuario')
ls_estado = this.GetItemString(al_row,'estado')

//Validar si los campos no estan nulos
//Campo de estado de grupo de usuario
if isnull(ls_grupo) or trim(ls_grupo)="" then
    rc = this.uf_setmsgerror(al_row,"id_grupousuario","Debe digitar la
descripción del grupo")
End if
//Campo de estado de usuario
if isnull(ls_estado) or trim(ls_estado)="" then
    rc = this.uf_setmsgerror(al_row,"estado_usuario","Debe seleccionar el estado
del usuario")
End if
```



```
//Insertar el correlativo del registro
If Parent.event ue_is_insertmode() = 1 Then
    //Consultar el codigo maximo
    SELECT      max(id_grupousuario)      INTO      :li_id_usr      FROM
GRUPO_USUARIO;
    IF IsNull(li_id_usr) THEN
        ls_id_usr = '01'
    ELSE
        li_id_usr++
        if li_id_usr <10 then
            ls_id_usr = '0'+string(li_id_usr)
        else
            ls_id_usr = string(li_id_usr)
        end if
    END IF
    //Asigna el codigo al campo
    THIS.SetItem(THIS.GetRow(),'id_grupousuario',ls_id_usr)
End if

//Retornar valores
return rc
end event
```

OBJETO UNV_SEGURIDAD

```
forward
global type unv_seguridad from n_base
end type
end forward
```



```
global type unv_seguridad from n_base
```

```
end type
```

```
global unv_seguridad unv_seguridad
```

```
type variables
```

```
//Declaracion de instancias
```

```
n_ds          ids_opcion
```

```
end variables
```

```
forward prototypes
```

```
public subroutine uf_menu_items (menu am_menu, integer ai_nivel1, integer  
ai_nivel2, integer ai_nivel3, boolean ab_authorized)
```

```
public function integer uf_configura_entidad (string as_codigo_entidad, string  
as_mostrar_mensaje, string as_servidor, string as_codigo_sistema)
```

```
public function menu uf_menu_upperbound (menu am_menu)
```

```
public function integer uf_servicio_sql (string as_nombre_pc)
```

```
public function string uf_desencriptar (string as_password)
```

```
public function long uf_clave_numerica (string as_password)
```

```
public function integer uf_habilita_opciones_menu ()
```

```
end prototypes
```

```
public subroutine uf_menu_items (menu am_menu, integer ai_nivel1, integer  
ai_nivel2, integer ai_nivel3, boolean ab_authorized);
```

```
//Asignacion de datos
```

```
am_menu = uf_menu_upperbound(am_menu)
```

```
//Verificar el nivel
```



```
if ai_nivel2 = 0 then
    am_menu.Item[ai_nivel1].enabled=ab_autorizado
else
    if ai_nivel3 = 0 then
        am_menu = uf_menu_upperbound(am_menu.Item[ai_nivel1])
        am_menu.Item[ai_nivel2].enabled=ab_autorizado
    else
        am_menu = uf_menu_upperbound(am_menu.Item[ai_nivel1])
        am_menu = uf_menu_upperbound(am_menu.Item[ai_nivel2])
        am_menu.Item[ai_nivel3].enabled=ab_autorizado
    end if
end if

end subroutine

public function integer uf_configura_entidad (string as_codigo_entidad, string
as_mostrar_mensaje, string as_servidor, string as_codigo_sistema);

//Declaracion de variables
Integer          li_rc,li_file_num
String          ls_write_string

/* Open the INI file with Replace option */
li_file_num = FileOpen ( 'minisalud.ini', LineMode!, Write!, LockReadWrite!,
Replace! )

/* Write the appropriate entries to the file */
ls_write_string = "[Database]"
FileWrite ( li_file_num, ls_write_string )
```



```
ls_write_string = "ServerName=" + trim(as_servidor) + ""
```

```
FileWrite ( li_file_num, ls_write_string )
```

```
ls_write_string = "Database=" + "Minis_Salud"
```

```
FileWrite ( li_file_num, ls_write_string )
```

```
ls_write_string = "LogId=" + ""
```

```
FileWrite ( li_file_num, ls_write_string )
```

```
ls_write_string = "LogPass=" + ""
```

```
FileWrite ( li_file_num, ls_write_string )
```

```
ls_write_string = "DSN=" + "LOCAL"
```

```
FileWrite ( li_file_num, ls_write_string )
```

```
ls_write_string = "[Configuracion]"
```

```
FileWrite ( li_file_num, ls_write_string )
```

```
ls_write_string = "entidad=" + as_codigo_entidad
```

```
FileWrite ( li_file_num, ls_write_string )
```

```
ls_write_string = "servidor=" + as_servidor
```

```
FileWrite ( li_file_num, ls_write_string )
```

```
/*ls_write_string = "dir_backup=" + as_dir_backup
```

```
FileWrite ( li_file_num, ls_write_string ) */
```

```
ls_write_string = "codigo sistema=" + as_codigo_sistema
```

```
FileWrite ( li_file_num, ls_write_string )
```

```
ls_write_string = "mostrar mensaje=" + as_mostrar_mensaje
```

```
FileWrite ( li_file_num, ls_write_string )
```

```
ls_write_string = "[Preference]"
```



```
FileWrite ( li_file_num, ls_write_string )
ls_write_string = "Send on Enter=0"
FileWrite ( li_file_num, ls_write_string )
ls_write_string = "Show All Errors=1"
FileWrite ( li_file_num, ls_write_string )
ls_write_string = "Print Preview=1"
FileWrite ( li_file_num, ls_write_string )
ls_write_string = "Right Mouse Support=1"
FileWrite ( li_file_num, ls_write_string )
ls_write_string = "Confirm Delete=1"
FileWrite ( li_file_num, ls_write_string )
ls_write_string = "Save on Exit=1"
FileWrite ( li_file_num, ls_write_string )

ls_write_string = "[Toolbar]"
FileWrite ( li_file_num, ls_write_string )
ls_write_string = "Show=1"
FileWrite ( li_file_num, ls_write_string )
ls_write_string = "Count=2"
FileWrite ( li_file_num, ls_write_string )
ls_write_string = "Position=T"
FileWrite ( li_file_num, ls_write_string )
ls_write_string = "Sheet Position=T"
FileWrite ( li_file_num, ls_write_string )
ls_write_string = "Text=0"
FileWrite ( li_file_num, ls_write_string )

/* Close the INI file */
li_rc = FileClose ( li_file_num )
```



// Retornar valores

RETURN li_rc

end function

public function menu uf_menu_upperbound (menu am_menu);

//Declaracion de variables

integer li_rc

//Obtener arreglo

li_rc = UpperBound(am_menu.Item)

//Retornar valor

return am_menu

end function

public function integer uf_servicio_sql (string as_nombre_pc);

// Declaracion de variables

Integer li_rc=1

long i, li_id

String ls_ruta, ls_servername

integer result

// Configuracion de puntero

SetPointer(hourglass!)

// Inicializacion de variables



```
li_id          = 0
//ls_serverName = ProfileString("C:\SISTEMAS\SIGES\siges.ini",
"Configuracion","servidor","")
//ls_ruta      = ProfileString("C:\SISTEMAS\SIGES\siges.ini",
"Configuracion","dir backup","")

// Verificar si existe servidor configurado
if isnull(ls_servername) or trim(ls_servername)="" then
    ls_servername=as_nombre_pc
end if

// Declaracion de instancias
OLEObject BASE
BASE = CREATE OLEObject
result = BASE.ConnectToNewObject("SQLDMO.SQLSERVER")

// Verificar si se pudo crear el enlace
IF result <> 0 THEN
    MESSAGEBOX("Error","No se encontró ningún Componente del Servidor
de Base de Datos SQL Server "+string(result))
    return -1
END IF

// Configuracion de conexion
BASE.Name = ls_servername
//BASE.Login = "GESTION"
//BASE.Password = "MRADIAN7"
BASE.LoginSecure = False
//BASE.LoginTimeout = 100
```



// Verificar estatus de conexión

CHOOSE CASE BASE.STATUS

CASE 1

//Esta encendido, unicamente nos conectamos

BASE.Connect()

li_id = BASE.ConnectionID

CASE 2

MESSAGEBOX("ESTADO DE LA BASE","BASE EN PAUSA")

CASE 3

**MESSAGEBOX("ESTADO DE LA BASE","BASE DETENIDA ...
REINICIANDO")**

//Reiniciar servidor

BASE.Start(True, ls_servername, "GESTION","MRADIAN7")

CASE 4

**MESSAGEBOX("ESTADO DE LA BASE","BASE EN PROCESO
DE ENCENDIDO")**

CASE 5

**MESSAGEBOX("ESTADO DE LA BASE","BASE
DETENIENDOSE")**

CASE 6

**MESSAGEBOX("ESTADO DE LA BASE","BASE
REESTABLECIENDOSE")**

CASE 7

**MESSAGEBOX("ESTADO DE LA BASE","BASE A PUNTO DE
PAUSAR")**

CASE 0



```
MESSAGEBOX("ESTADO DE LA BASE","BASE EN ESTADO  
DESCONOCIDO")  
END CHOOSE
```

```
// Restaurar configuracion de puntero  
SetPointer(Arrow!)
```

```
//Retornar valores  
return li_rc  
end function
```

```
public function integer uf_habilita_opciones_menu ();
```

```
// Declaracion de variables
```

```
integer      li_rc, li_count, i, li_nivel1, li_nivel2, li_nivel3  
String ls_usuario, ls_opcion, ls_expresion, ls_acesa, ls_opcioncad  
long        ll_row, ll_count  
Boolean     lb_habilitar = true
```

```
// Declaracion de instancias
```

```
menu        lm_menu
```

```
//Contrucción de instancias
```

```
lm_menu = w_frame.menuid  
ids_opcion.SetTransObject(sqlca)
```

```
// Consulta de usuario
```

```
select id_usuario into :ls_usuario from usuario where login = :gs_login;
```



```
if sqlca.sqlcode<> 0 then
    MessageBox('Error','El usuario no existe en la base de datos')
    return -1
end if

// Habilitar opciones
ll_row = ids_opcion.Retrieve(gs_grupo, gs_codigo_usuario)

//Verificar si existen opciones asignadas
If ll_row <> 0 then
    For ll_count = 1 to ll_row
        //Recuperar datos de menu
        li_nivel1 = ids_opcion.GetItemNumber(ll_count,'nivel1')
        li_nivel2 = ids_opcion.GetItemNumber(ll_count,'nivel2')
        li_nivel3 = ids_opcion.GetItemNumber(ll_count,'nivel3')

        //ls_opcioncad = ids_opcion.GetItemString(ll_count,'opcion')
        //      messagebox("Mensaje      error",string(li_nivel1)      +'"-"+
string(li_nivel2)+"-"+string(li_nivel3) +"="+ls_opcioncad )
        //Habilitar el menu
        this.uf_menu_items(lm_menu,    li_nivel1,    li_nivel2,    li_nivel3,
lb_habilitar)
    next
End if

//Retornar datos
return li_rc
end function
on unv_seguridad.create
```



call super::create

end on

on unv_seguridad.destroy

call super::destroy

end on

event constructor;call super::constructor;//Declaracion de objeto

ids_opcion = create n_ds

//confgiuracion de objeto

ids_opcion.DataObject ='d_ds_opcion_ug'

end event

OBJETO U_NV_CST_ECRYPTOR

forward

global type u_nv_cst_encryptor from nonvisualobject

end type

end forward

global type u_nv_cst_encryptor from nonvisualobject

end type

global u_nv_cst_encryptor u_nv_cst_encryptor

type variables



```
string is_raw, is_encrypted, is_key="CGI"
```

```
end variables
```

```
forward prototypes
```

```
public function string of_set (string thestr)
```

```
public function STRING of_getraw ()
```

```
public function STRING of_getencrypted ()
```

```
public function string of_decrypt (string thestr)
```

```
public function string of_setkey (string thekey)
```

```
public function string of_set (string thetext, string thekey)
```

```
public function string of_decrypt (string thetext, string thekey)
```

```
end prototypes
```

```
public function string of_set (string thestr);string retVal, tempStr, tStr
```

```
int sourcePtr, keyPtr, keyLen, sourceLen, tempVal, tempKey
```

```
retVal = is_raw
```

```
is_raw = thestr
```

```
keyPtr = 1
```

```
keyLen = len(is_key)
```

```
sourceLen = len(is_raw)
```

```
is_encrypted = ""
```

```
for sourcePtr = 1 to sourceLen
```

```
    tempVal = asc(right(is_raw, sourceLen - sourcePtr + 1))
```

```
    tempKey = asc(right(is_key, keyLen - keyPtr + 1))
```

```
    tempVal += tempKey
```



```
// Added this section to ensure that ASCII Values stay within 0 to 255 range
DO WHILE tempVal > 255
    if tempVal > 255 then
        tempVal = tempVal - 255
    end if
LOOP
// End of Section
tStr = char(tempVal)
is_encrypted += tStr
keyPtr ++
if keyPtr > len(is_key) then keyPtr = 1
next

return is_encrypted
end function

public function STRING of_getraw ();return is_raw
end function

public function STRING of_getencrypted ();return is_encrypted
end function

public function string of_decrypt (string thestr);string retVal, tempStr, tStr
int sourcePtr, keyPtr, keyLen, sourceLen, tempVal, tempKey

is_encrypted = thestr

keyPtr = 1
keyLen = len(is_key)
```



```
// Fixed so that decryption is done on encrypted input string of proper length
//sourceLen = len(is_raw)
sourceLen = len(is_encrypted)
is_raw = ""
for sourcePtr = 1 to sourceLen
    tempVal = asc(right(is_encrypted, len(is_encrypted) - sourcePtr + 1))
    tempKey = asc(right(is_key, len(is_key) - keyPtr + 1))
    tempVal -= tempKey
    // Added this section to ensure that ASCII codes stay in 0 to 255 range
    DO WHILE tempVal < 0
        if tempVal < 0 then
            tempVal = tempVal + 255
        end if
    LOOP
    // end of section
    tStr = char(tempVal)
    is_raw += tStr
    keyPtr ++
    if keyPtr > len(is_key) then keyPtr = 1
next

retVal = is_raw

return retVal
end function

public function string of_setkey (string thekey);string retVal
retVal = is_key
is_key = theKey
```



```
return retVal
```

```
end function
```

```
public function string of_set (string thetext, string thekey);of_setKey(theKey)
```

```
return of_set(theText)
```

```
end function
```

```
public function string of_decrypt (string thetext, string thekey);// Chagned input  
variable order to match documentation
```

```
of_setKey(theKey)
```

```
return of_decrypt(theText)
```

```
end function
```

```
on u_nv_cst_encryptor.create
```

```
call super::create
```

```
TriggerEvent( this, "constructor" )
```

```
end on
```

```
on u_nv_cst_encryptor.destroy
```

```
TriggerEvent( this, "destructor" )
```

```
call super::destroy
```

```
end on
```

OBJETO E OPCIONES

```
global type e_opciones from structure
```

```
    string        as_idopcion
```

```
    integer       ai_seleccion
```



end type



LIBRERIA DE VENTANAS (RFYD_W.PBL)

VENTANA W_CITA_EXAMEN

forward

global type w_cita_examen from window

end type

type cb_3 from commandbutton within w_cita_examen

end type

type cb_2 from commandbutton within w_cita_examen

end type

type cb_1 from commandbutton within w_cita_examen

end type

type dw_1 from datawindow within w_cita_examen

end type

type gb_1 from groupbox within w_cita_examen

end type

end forward

global type w_cita_examen from window

integer width = 3195

integer height = 1768

boolean titlebar = true

string title = "Medicamentos de Farmacia"

boolean controlmenu = true

boolean minbox = true

boolean maxbox = true

boolean resizable = true

long bgcolor = 12632256



```
string icon = "AppIcon!"  
boolean center = true  
event filtro ()  
cb_3 cb_3  
cb_2 cb_2  
cb_1 cb_1  
dw_1 dw_1  
gb_1 gb_1  
end type  
global w_cita_examen w_cita_examen
```

```
on w_cita_examen.create  
this.cb_3=create cb_3  
this.cb_2=create cb_2  
this.cb_1=create cb_1  
this.dw_1=create dw_1  
this.gb_1=create gb_1  
this.Control[]={this.cb_3,&  
this.cb_2,&  
this.cb_1,&  
this.dw_1,&  
this.gb_1}  
end on
```

```
on w_cita_examen.destroy  
destroy(this.cb_3)  
destroy(this.cb_2)  
destroy(this.cb_1)  
destroy(this.dw_1)
```



```
destroy(this.gb_1)  
end on  
  
event open;  
  
// Inicializando variables  
integer ll_contador, ll_numero, ll_codigoexamen  
  
// Inicializando datawindows que contendra los datos de la cita  
dw_1.SetTransObject(SQLCA)  
dw_1.retrieve()  
gs_codigopaciente=""  
dw_1.event filtro()  
dw_1.retrieve()  
  
end event  
  
type cb_3 from commandbutton within w_cita_examen  
integer x = 2121  
integer y = 64  
integer width = 402  
integer height = 112  
integer taborder = 20  
boolean bringtotop = true  
integer textsize = -10  
integer weight = 700  
fontcharset fontcharset = ansi!  
fontpitch fontpitch = variable!
```



```
fontfamily fontfamily = swiss!  
string facename = "Arial"  
string text = "&Leer"  
end type
```

```
event clicked;  
  
// Inicializando variables  
integer li_FileNum, li_i, li_ind, li_j  
string ls_Emp_Input, ls_atributo  
long ll_FLength, li_atributo  
  
// Calculando la longitud del archivo de texto donde se guarda el codigo del paciente  
ll_FLength = FileLength("C:\codigo.TXT")  
// Abriendo el archivo de texto  
li_FileNum = FileOpen("C:\codigo.TXT", StreamMode!)  
// Asegurandonos que no se pase del máximo permitido  
IF ll_FLength < 32767 THEN  
    // Leer el contenido del archivo  
    FileRead(li_FileNum, ls_atributo)  
    // Asignando un valor a la variable con el contenido del archivo  
    li_atributo= long(ls_atributo)  
    ls_atributo = trim(ls_atributo)  
    // Buscando el codigo del paciente  
    gs_codigopaciente = "  
    Select codigo_paciente  
into :gs_codigopaciente  
from expediente  
where convert(dec(10),atributo) =:li_atributo;  
    // Actualizando el datawindows con el valor del código  
    dw_1.event filtro()
```



```
dw_1.retrieve()
// Cerrando el archivo
FileClose(li_FileNum)
// Abriendo el archivo en modo escritura/ reemplazo
li_FileNum = FileOpen("C:\codigo.TXT",
LineMode!,Write!,Shared!,Replace!)
// Escribiendo un caracter en blanco
FileWrite(li_FileNum, " ")
// Cerrando el archivo
FileClose(li_FileNum)
// Inicializando variables
gs_codigopaciente = ""
ls_atributo=""
li_atributo=0
END IF

end event

type cb_2 from commandbutton within w_cita_examen
integer x = 2560
integer y = 64
integer width = 402
integer height = 112
integer taborder = 10
integer textsize = -10
integer weight = 700
fontcharset fontcharset = ansi!
fontpitch fontpitch = variable!
fontfamily fontfamily = swiss!
```



```
string facename = "Arial"
```

```
string text = "&Salir"
```

```
end type
```

```
event clicked;
```

```
// Cerrar la ventana de trabajo
```

```
Close(w_cita_examen)
```

```
end event
```

```
type cb_1 from commandbutton within w_cita_examen
```

```
integer x = 1682
```

```
integer y = 64
```

```
integer width = 402
```

```
integer height = 112
```

```
integer taborder = 10
```

```
integer textsize = -10
```

```
integer weight = 700
```

```
fontcharset fontcharset = ansi!
```

```
fontpitch fontpitch = variable!
```

```
fontfamily fontfamily = swiss!
```

```
string facename = "Arial"
```

```
string text = "&Actualizar"
```

```
end type
```

```
event clicked;
```

```
// Inicializando variables locales
```

```
string ls_despachado, ls_codigocita, ls_grupo,ls_subgrupo,ls_medicamento
```

```
integer ll_contador
```



Commit;

// Cargando los datos en variables de los datawindos

FOR ll_contador = 1 TO dw_1.RowCount()

ls_despachado = dw_1.getitemString(ll_contador, "despachado")

ls_codigocita = dw_1.getitemString(ll_contador, "codigo_cita")

ls_grupo = dw_1.getitemString(ll_contador, "id_grupo")

ls_subgrupo = dw_1.getitemString(ll_contador, "id_subgrupo")

ls_medicamento = dw_1.getitemString(ll_contador, "id_medicamento")

if ls_despachado='1' then

// Actualizando la tabla paciente medicamento en el caso de que este

// se le ponga la viñeta de despachado

update paciente_medicamento

set despachado = '1'

where codigo_cita =: ls_codigocita and

id_grupo =: ls_grupo and

id_subgrupo =: ls_subgrupo and

id_medicamento =: ls_medicamento;

end if

next

**// Refrescando el data windows para que solo salgan los datos que no se han
despachado**

dw_1.retrieve()

// Mensajes de texto para usuarios

messagebox("Actualización de Datos", "Proceso terminado exitosamente!!!")



Commit;
end event

type dw_1 from datawindow within w_cita_examen

event filtro ()

integer x = 46

integer y = 216

integer width = 2985

integer height = 1224

integer taborder = 10

string title = "none"

string dataobject = "d_edt_cita_examen"

boolean vscrollbar = true

boolean border = false

boolean livescroll = true

end type

event filtro();

// Inicializando variables

string ls_filter1

// Filtrando por código del paciente

ls_filter1="codigo_paciente='"+gs_codigopaciente+'"'

// Actualizando el data windows

dw_1.setFilter(ls_filter1);

//dw_1.filter();

end event

type gb_1 from groupbox within w_cita_examen



```
integer x = 1609  
integer width = 1390  
integer height = 192  
integer taborder = 20  
integer textsize = -10  
integer weight = 700  
fontcharset fontcharset = ansi!  
fontpitch fontpitch = variable!  
fontfamily fontfamily = swiss!  
string facename = "Arial"  
long textcolor = 16711680  
long backcolor = 12632256  
end type
```

VENTANA W_CITA_LABORATORIO

```
forward  
global type w_cita_laboratorio from window  
end type  
type cb_2 from commandbutton within w_cita_laboratorio  
end type  
type cb_1 from commandbutton within w_cita_laboratorio  
end type  
type dw_1 from datawindow within w_cita_laboratorio  
end type  
type gb_1 from groupbox within w_cita_laboratorio
```



end type

end forward

global type w_cita_laboratorio from window

integer width = 2683

integer height = 1804

boolean titlebar = true

string title = "Exámenes de Laboratorio"

boolean controlmenu = true

boolean minbox = true

boolean maxbox = true

boolean resizable = true

long backcolor = 12632256

string icon = "AppIcon!"

boolean center = true

cb_2 cb_2

cb_1 cb_1

dw_1 dw_1

gb_1 gb_1

end type

global w_cita_laboratorio w_cita_laboratorio

on w_cita_laboratorio.create

this.cb_2=create cb_2

this.cb_1=create cb_1

this.dw_1=create dw_1

this.gb_1=create gb_1

this.Control[]={this.cb_2,&

this.cb_1,&



```
this.dw_1,&  
this.gb_1}  
end on  
on w_cita_laboratorio.destroy  
destroy(this.cb_2)  
destroy(this.cb_1)  
destroy(this.dw_1)  
destroy(this.gb_1)  
end on  
event open;  
// Inicializando variables  
integer ll_contador, ll_numero, ll_codigoexamen  
  
// Inicializando data windows  
dw_1.SetTransObject(SQLCA)  
dw_1.retrieve()  
  
end event  
  
type cb_2 from commandbutton within w_cita_laboratorio  
integer x = 2117  
integer y = 60  
integer width = 402  
integer height = 112  
integer taborder = 10  
integer textsize = -10  
integer weight = 700
```



fontcharset fontcharset = ansi!

fontpitch fontpitch = variable!

fontfamily fontfamily = swiss!

string facename = "Arial"

string text = "&Salir"

end type

event clicked;

// Cerrando la ventana de laboratorios

Close(w_cita_laboratorio)

end event

type cb_1 from commandbutton within w_cita_laboratorio

integer x = 1687

integer y = 60

integer width = 402

integer height = 112

integer taborder = 10

integer textsize = -10

integer weight = 700

fontcharset fontcharset = ansi!

fontpitch fontpitch = variable!

fontfamily fontfamily = swiss!

string facename = "Arial"

string text = "&Actualizar"

end type

event clicked;

// Inicializando variables



string ls_despachado, ls_codigocita

integer ll_contador, li_idexamen

Commit;

// Cargando los datos del datawindows en variables

FOR ll_contador = 1 TO dw_1.RowCount()

ls_despachado = dw_1.getItemString(ll_contador, "despachado")

ls_codigocita = dw_1.getItemString(ll_contador, "codigo_cita")

li_idexamen = dw_1.getItemNumber(ll_contador, "id_examen")

if ls_despachado='1' then

// Actualizando los datos en la tabla paciente_examen en caso de ser

// despachado

update paciente_examen

set despachado = '1'

where codigo_cita =: ls_codigocita and

id_examen =: li_idexamen;

end if

next

// Actualizando el data windows para que solo salgan los exámenes que se han actualizado

dw_1.retrieve()

// mensaje de aviso para usuarios

messagebox("Actualización de Datos","Proceso terminado exitosamente!!!")

Commit;



end event

type dw_1 from datawindow within w_cita_laboratorio

integer x = 46

integer y = 216

integer width = 2505

integer height = 1224

integer taborder = 10

string title = "none"

string dataobject = "d_edt_cita_laboratorio"

boolean vscrollbar = true

boolean border = false

boolean livescroll = true

end type

type gb_1 from groupbox within w_cita_laboratorio

integer x = 1646

integer width = 905

integer height = 196

integer taborder = 20

integer textsize = -10

integer weight = 700

fontcharset fontcharset = ansi!

fontpitch fontpitch = variable!

fontfamily fontfamily = swiss!

string facename = "Arial"

long textcolor = 16711680

long bgcolor = 12632256

end type



VENTANA W_EDT_CITA

forward

global type w_edt_cita from w_edt

end type

type dw_1 from u_dw within w_edt_cita

end type

type cb_1 from commandbutton within w_edt_cita

end type

type cb_2 from commandbutton within w_edt_cita

end type

type dw_2 from datawindow within w_edt_cita

end type

type dw_3 from datawindow within w_edt_cita

end type

type dw_4 from datawindow within w_edt_cita

end type

type cb_6 from commandbutton within w_edt_cita

end type

type cb_3 from commandbutton within w_edt_cita

end type

end forward

global type w_edt_cita from w_edt

integer width = 3543

integer height = 2168

string title = "Diagnostico"



dw_1 dw_1

cb_1 cb_1

cb_2 cb_2

dw_2 dw_2

dw_3 dw_3

dw_4 dw_4

cb_6 cb_6

cb_3 cb_3

end type

global w_edt_cita w_edt_cita

type variables

end variables

forward prototypes

**public subroutine wf_habilitardeshabilitarcontroles (string a_tipoproveedor, integer
a_accion)**

end prototypes

**public subroutine wf_habilitardeshabilitarcontroles (string a_tipoproveedor, integer
a_accion);**

end subroutine

on w_edt_cita.create

int iCurrent

call super::create

this.dw_1=create dw_1

this.cb_1=create cb_1

this.cb_2=create cb_2



```
this.dw_2=create dw_2  
this.dw_3=create dw_3  
this.dw_4=create dw_4  
this.cb_6=create cb_6  
this.cb_3=create cb_3  
iCurrent=UpperBound(this.Control)  
this.Control[iCurrent+1]=this.dw_1  
this.Control[iCurrent+2]=this.cb_1  
this.Control[iCurrent+3]=this.cb_2  
this.Control[iCurrent+4]=this.dw_2  
this.Control[iCurrent+5]=this.dw_3  
this.Control[iCurrent+6]=this.dw_4  
this.Control[iCurrent+7]=this.cb_6  
this.Control[iCurrent+8]=this.cb_3  
end on  
  
on w_edt_cita.destroy  
call super::destroy  
destroy(this.dw_1)  
destroy(this.cb_1)  
destroy(this.cb_2)  
destroy(this.dw_2)  
destroy(this.dw_3)  
destroy(this.dw_4)  
destroy(this.cb_6)  
destroy(this.cb_3)  
end on  
  
event open;call super::open;
```



```
// Inicializando datawindows
dw_1.SetTransObject(SQLCA)
dw_2.SetTransObject(SQLCA)
dw_3.SetTransObject(SQLCA)
dw_4.SetTransObject(SQLCA)

end event

type p_rowfocus from w_edt`p_rowfocus within w_edt_cita
end type

type cb_cancel from w_edt`cb_cancel within w_edt_cita
integer x = 2880
integer y = 640
integer width = 462
string text = "&Salir"
end type

type cb_ok from w_edt`cb_ok within w_edt_cita
integer x = 2880
integer y = 40
integer width = 462
end type

type dw_1 from u_dw within w_edt_cita
integer x = 23
integer y = 28
integer width = 2587
integer height = 336
```



```
integer taborder = 40
boolean bringtotop = true
string dataobject = "d_edt_cita"
end type

event itemchanged;call super::itemchanged;
string ls_nombre,    ls_nombre2,    ls_nombre3,    ls_apellido1,    ls_apellido2,
ls_codigopaciente
string ls_nombret, ls_codigoconsulta, ls_idproveedor

// Despues de haber digitado el codigo de la consulta
if dwo.name='codigo_consulta' then
    // Asegurandonos que el usuario digito los digitos del código de la consulta
    ls_codigoconsulta = data
    ls_codigoconsulta = fill('0', 10 - len(ls_codigoconsulta)) + ls_codigoconsulta
    // Capturando el código del paciente
    Select codigo_paciente, id_proveedor into :ls_codigopaciente, :ls_idproveedor
from consulta where codigo_consulta =: ls_codigoconsulta;
    //      Haciendo una consulta para capturar el nombre del paciente
    Select primer_nombre, segundo_nombre, tercer_nombre, primer_apellido,
segundo_apellido
    into :ls_nombre, :ls_nombre2, :ls_nombre3, :ls_apellido1, :ls_apellido2
    from expediente
    where codigo_paciente =: ls_codigopaciente;
    // En la variable guardamos el nombre del paciente
    ls_nombret = trim(ls_nombre)+' '+trim(ls_nombre2)+' '+ trim(ls_nombre3)+'
'+trim(ls_apellido1)+' '+trim(ls_apellido2)
    // El nombre es actualizado en el datawindows
    this.Modify('t_1.text = ''' + trim(ls_nombret)+''''')
```



```
// Actualizando el datawindows con el código del paciente y la fecha de la cita
dw_1.SetItem(row, "cita_codigo_paciente", ls_codigopaciente)
dw_1.SetItem(row, "cita_id_proveedor", ls_idproveedor)
dw_1.SetItem(row, "codigo_consulta", ls_codigoconsulta)
dw_1.SetItem(row, "cita_fecha_cita", today())
dw_1.AcceptText()

// Filtrando los demas datawindows para que solo muestren los registros
// que pertenecen a este paciente en esta cita
dw_2.event filtro()
dw_2.retrieve()
dw_3.event filtro()
dw_3.retrieve()
dw_4.event filtro()
dw_4.retrieve()

end if
end event

event retrieveend;call super::retrieveend;

//Declaración de variables
string ls_nombre,    ls_nombre2,    ls_nombre3,    ls_apellido1,    ls_apellido2,
ls_codigopaciente
string ls_nombret
// Capturando las variables a utilizar para la actualización
ls_codigopaciente = dw_1.GetItemString(rowcount,"cita_codigo_paciente")
gs_codcita          = dw_1.GetItemString(rowcount,"cita_codigo_cita")
// Capturando el nombre del paciente mediante una instrucción sql
```



```
Select primer_nombre, segundo_nombre, tercer_nombre, primer_apellido,  
segundo_apellido  
into :ls_nombre, :ls_nombre2, :ls_nombre3, :ls_apellido1, :ls_apellido2  
from expediente  
where codigo_paciente =: ls_codigopaciente;  
// Guardando el nombre en una variable  
ls_nombret = trim(ls_nombre)+' '+trim(ls_nombre2)+' '+ trim(ls_nombre3)+'  
'+trim(ls_apellido1)+' '+trim(ls_apellido2)  
// Modificando el datawindows para que acepte el nombre  
this.Modify('t_1.text = '' + trim(ls_nombret)+''')  
gs_codcita = dw_1.GetItemString (rowcount,"cita_codigo_cita")  
// Actualizando los datawindows para que solo muestren los datos del paciente y cita  
dw_2.event filtro()  
dw_2.retrieve()  
dw_3.event filtro()  
dw_3.retrieve()  
dw_4.event filtro()  
dw_4.retrieve()  
  
return 1  
end event  
  
event ue_validaterow;call super::ue_validaterow;  
  
//Declaración de variables  
integer rc  
string ls_codigopaciente, ls_idproveedor  
datetime ldt_fecha
```



```
//Obtener los campos que se validaran
ls_codigopaciente = this.GetItemString(al_row, "cita_codigo_paciente")
ls_codigopaciente = Trim(ls_codigopaciente)
ls_codigopaciente = fill('0', 10 - len(ls_codigopaciente)) + ls_codigopaciente
ls_idproveedor      = this.GetItemString(al_row, "cita_id_proveedor")

dw_1.SetItem(al_row, "cita_codigo_paciente", ls_codigopaciente)
//*****
*****

//VALIDACION
//*****
*****

//Validar los campos que no deben quedar nulos
If isnull(ls_codigopaciente) or trim(ls_codigopaciente) = "" then
    rc = this.uf_setmsgerror(al_row,"cita_codigo_paciente","Debe digitarse el
código del paciente")
End if
If isnull(ls_idproveedor) or trim(ls_idproveedor) = "" then
    rc = this.uf_setmsgerror(al_row,"cita_id_proveedor","Debe digitarse médico
a consultar")
End if

// Agregando datos de auditoria de usuarios
If parent.event ue_is_insertmode() = 1 then
    dw_1.SetItem(al_row, "cita_usuario_d", gs_login)
    dw_1.SetItem(al_row, "cita_fecha_d", datetime(today()))
Else
    dw_1.SetItem(al_row, "cita_usuario_m", gs_login)
```



```
dw_1.SetItem(al_row, "cita_fecha_m", datetime(today()))
```

```
End if
```

```
//Retornar valores
```

```
Return 1
```

```
end event
```

```
event ue_insertkeyprimer;call super::ue_insertkeyprimer;
```

```
// Inicializando variables
```

```
string ls_codigo
```

```
// Buscando el mayor código registrado en la tabla cita
```

```
Select max(codigo_cita) into :ls_codigo from cita;
```

```
// Si el resultado es nulo, debemos inicializar la tabla con el primer código
```

```
// aceptable
```

```
if isnull(ls_codigo) or trim(ls_codigo) ='' then
```

```
    ls_codigo = '0000000001'
```

```
else
```

```
    //Calcular el correlativo
```

```
    ls_codigo = string(long(ls_codigo) + 1)
```

```
    //Verificar que el codigo sea de 10 caracteres
```

```
    if len(ls_codigo) <> 10 then
```

```
        ls_codigo = fill('0', 10 - len(ls_codigo)) + ls_codigo
```

```
    end if
```

```
end if
```

```
// Actualizando el datawindows con el código proporcionado para esta cita
```



```
dw_1.SetItem(al_row, "cita_codigo_cita", ls_codigo)
// Actualizando la variable global
gs_codcita= ls_codigo

return 1
end event

event ue_postupdate;call super::ue_postupdate;

// Inicializando las variables locales
long ll_cuenta, ll_contador, ll_numero, ll_suma, ll_num2, ll_cuenta2
string ls_idgrupo, ls_idsubgrupo, ls_idmedicamento
// Buscando si existe una vez el código de la cita
Select count(*) into :ll_cuenta from cita where codigo_cita =: gs_codcita;
commit;
// Si el código resulta ser nulo o cero, es decir que se borro la cita
if isnull(ll_cuenta) or ll_cuenta=0 then
    // Contador que tiene por finalidad actualizar el inventario de la base de
datos
    // cuando se borra una consulta
    FOR ll_contador = 1 TO dw_2.RowCount()
        // Guardando en variables los parametros para hacer la comparación y
actualizar el inventario
        ll_cuenta2=dw_2.rowcount()
        ls_idgrupo= dw_2.getitemstring(ll_contador,"id_grupo")
        ls_idsubgrupo = dw_2.GetItemString(ll_contador,"id_subgrupo")
        ls_idmedicamento =
dw_2.GetItemString(ll_contador,"id_medicamento")
        ll_numero = dw_2.GetItemNumber(ll_contador,"cantidad")
```



// Haciendo una consulta para saber la cantidad guardada en medicamentos

```
Select cantidad into :ll_num2 from medicamento
where id_grupo =: ls_idgrupo and
      id_subgrupo =: ls_idsubgrupo and
      id_medicamento =: ls_idmedicamento;
```

// En la variable suma se guarda la cantidad real que debe ser reflejada en el inventario

```
ll_suma = ll_numero + ll_num2
```

// Actualizando la tabla medicamento despues de hacer el nuevo calculo

```
update medicamento
set cantidad =: ll_suma
where id_grupo =: ls_idgrupo and
      id_subgrupo =: ls_idsubgrupo and
      id_medicamento =: ls_idmedicamento;
```

Next

// Borrando las tablas relacionadas con la cita

```
delete from paciente_emfermedad where codigo_cita =: gs_codcita;
delete from paciente_medicamento where codigo_cita =: gs_codcita;
delete from paciente_examen where codigo_cita =: gs_codcita;
```

end if

commit;



return 1

end event

type cb_1 from commandbutton within w_edt_cita

integer x = 2880

integer y = 404

integer width = 462

integer height = 108

integer taborder = 20

boolean bringtotop = true

integer textsize = -8

integer weight = 700

fontcharset fontcharset = ansi!

fontpitch fontpitch = variable!

fontfamily fontfamily = swiss!

string facename = "MS Sans Serif"

string text = "&Medicamentos"

end type

event clicked;

// Abre la ventana para agregar medicamentos

Open(w_edt_cita_medicamento)

end event

event losefocus;dw_2.event filtro()



dw_2.retrieve()

end event

type cb_2 from commandbutton within w_edt_cita

integer x = 2880

integer y = 160

integer width = 462

integer height = 108

integer taborder = 30

boolean bringtotop = true

integer textsize = -8

integer weight = 700

fontcharset fontcharset = ansi!

fontpitch fontpitch = variable!

fontfamily fontfamily = swiss!

string facename = "MS Sans Serif"

string text = "&Enfermedades"

end type

event clicked;

// Abre la ventana para agregar las enfermedades

Open(w_edt_cita_enfermedad)

end event

type dw_2 from datawindow within w_edt_cita



```
event filtro ( )
integer x = 37
integer y = 1464
integer width = 3200
integer height = 532
integer taborder = 30
boolean bringtotop = true
string title = "none"
string dataobject = "d_edt_cita2"
boolean vscrollbar = true
boolean border = false
boolean livescroll = true
end type

event filtro();
String ls_filter1

// Filtro según número de cita
ls_filter1="codigo_cita="+gs_codcita+""
dw_2.setFilter(ls_filter1);
dw_2.filter();

end event

type dw_3 from datawindow within w_edt_cita
event filtro ( )
integer x = 37
integer y = 388
```



```
integer width = 2779
integer height = 528
integer taborder = 40
boolean bringtotop = true
string title = "none"
string dataobject = "d_edt_cita33"
boolean vscrollbar = true
boolean border = false
boolean livescroll = true
end type

event filtro();
String ls_filter1

// Filtro según número de cita
ls_filter1="codigo_cita='"+gs_codcita+'"'
dw_3.setFilter(ls_filter1);
dw_3.filter();

end event

type dw_4 from datawindow within w_edt_cita
event filtro ( )
integer x = 37
integer y = 928
integer width = 2917
integer height = 528
integer taborder = 50
```



```
boolean bringtotop = true
string title = "none"
string dataobject = "d_edt_cita6"
boolean vscrollbar = true
boolean border = false
boolean livescroll = true
end type

event filtro();
String ls_filter1

// Filtro según número de cita
ls_filter1='codigo_cita="'+gs_codcita+"'"
dw_4.setFilter(ls_filter1);
dw_4.filter();

end event

type cb_6 from commandbutton within w_edt_cita
integer x = 2880
integer y = 280
integer width = 462
integer height = 108
integer taborder = 40
boolean bringtotop = true
integer textsize = -8
integer weight = 700
fontcharset fontcharset = ansi!
```



```
fontpitch fontpitch = variable!  
fontfamily fontfamily = swiss!  
string facename = "MS Sans Serif"  
string text = "E&xámenes"  
end type
```

```
event clicked;
```

```
// Abre la ventana para agregar exámenes de laboratorio  
Open(w_edt_cita_examen)
```

```
end event
```

```
type cb_3 from commandbutton within w_edt_cita  
integer x = 2880  
integer y = 512  
integer width = 462  
integer height = 108  
integer taborder = 40  
boolean bringtotop = true  
integer textsize = -8  
integer weight = 700  
fontcharset fontcharset = ansi!  
fontpitch fontpitch = variable!  
fontfamily fontfamily = swiss!  
string facename = "MS Sans Serif"  
string text = "&Reestablecer"  
end type
```



```
event clicked;dw_2.event filtro()
dw_2.retrieve()
dw_3.event filtro()
dw_3.retrieve()
dw_4.event filtro()
dw_4.retrieve()

end event
```

VENTANA W_EDT_CITA_ENFERMEDAD

```
forward
global type w_edt_cita_enfermedad from window
end type
type sle_1 from singlelineedit within w_edt_cita_enfermedad
end type
type cb_3 from commandbutton within w_edt_cita_enfermedad
end type
type cb_2 from commandbutton within w_edt_cita_enfermedad
end type
type cb_1 from commandbutton within w_edt_cita_enfermedad
end type
type st_2 from statictext within w_edt_cita_enfermedad
end type
type st_1 from statictext within w_edt_cita_enfermedad
end type
```



```
type dw_1 from datawindow within w_edt_cita_enfermedad  
end type  
type gb_1 from groupbox within w_edt_cita_enfermedad  
end type  
end forward
```

```
global type w_edt_cita_enfermedad from window  
integer width = 2853  
integer height = 1860  
boolean titlebar = true  
string title = "Catálogo de Enfermedades"  
boolean controlmenu = true  
boolean minbox = true  
boolean maxbox = true  
boolean resizable = true  
long bgcolor = 12632256  
string icon = "AppIcon!"  
boolean center = true  
sle_1 sle_1  
cb_3 cb_3  
cb_2 cb_2  
cb_1 cb_1  
st_2 st_2  
st_1 st_1  
dw_1 dw_1  
gb_1 gb_1  
end type  
global w_edt_cita_enfermedad w_edt_cita_enfermedad
```



on w_edt_cita_enfermedad.create

this.sle_1=create sle_1

this.cb_3=create cb_3

this.cb_2=create cb_2

this.cb_1=create cb_1

this.st_2=create st_2

this.st_1=create st_1

this.dw_1=create dw_1

this.gb_1=create gb_1

this.Control[]={this.sle_1,&

this.cb_3,&

this.cb_2,&

this.cb_1,&

this.st_2,&

this.st_1,&

this.dw_1,&

this.gb_1}

end on

on w_edt_cita_enfermedad.destroy

destroy(this.sle_1)

destroy(this.cb_3)

destroy(this.cb_2)

destroy(this.cb_1)

destroy(this.st_2)

destroy(this.st_1)

destroy(this.dw_1)

destroy(this.gb_1)

end on



```
event open;  
// Inicializando variables  
integer ll_contador, ll_numero, ll_codigoenfermedad  
  
// Inicializando datawindows  
dw_1.SetTransObject(SQLCA)  
dw_1.retrieve()  
// Actualizando la ventana con el número de la cita  
st_2.text=gs_codcita  
end event  
type sle_1 from singlelineedit within w_edt_cita_enfermedad  
integer x = 37  
integer y = 256  
integer width = 2597  
integer height = 96  
integer taborder = 60  
integer textsize = -10  
integer weight = 700  
fontcharset fontcharset = ansi!  
fontpitch fontpitch = variable!  
fontfamily fontfamily = swiss!  
string facename = "Arial"  
long textcolor = 255  
textcase textcase = upper!  
borderstyle borderstyle = stylelowered!  
end type  
  
type cb_3 from commandbutton within w_edt_cita_enfermedad
```



```
integer x = 1792  
integer y = 60  
integer width = 402  
integer height = 112  
integer taborder = 50  
integer textsize = -10  
integer weight = 700  
fontcharset fontcharset = ansi!  
fontpitch fontpitch = variable!  
fontfamily fontfamily = swiss!  
string facename = "Arial"  
string text = "&Buscar"  
end type  
  
event clicked;  
  
// Inicialización de variables  
// Asignando a la variable el código digitado  
gs_enfermedad = trim(sle_1.text)  
// Filtrando de acuerdo a la variable asignada  
dw_1.event filtro()  
dw_1.retrieve()  
  
end event  
  
type cb_2 from commandbutton within w_edt_cita_enfermedad  
integer x = 2235  
integer y = 60
```



```
integer width = 402  
integer height = 112  
integer taborder = 10  
integer textsize = -10  
integer weight = 700  
fontcharset fontcharset = ansi!  
fontpitch fontpitch = variable!  
fontfamily fontfamily = swiss!  
string facename = "Arial"  
string text = "&Salir"  
end type  
  
event clicked;  
// Cerrando la ventana de la enfermedad  
Close(w_edt_cita_enfermedad)  
end event  
  
type cb_1 from commandbutton within w_edt_cita_enfermedad  
integer x = 1367  
integer y = 60  
integer width = 402  
integer height = 112  
integer taborder = 20  
integer textsize = -10  
integer weight = 700  
fontcharset fontcharset = ansi!  
fontpitch fontpitch = variable!  
fontfamily fontfamily = swiss!  
string facename = "Arial"
```



```
string text = "&Agregar"  
end type  
  
event clicked;  
// Inicialización de variables  
string ls_agregar  
integer ll_contador, ll_numero, ll_cuenta, ll_codigoenfermedad  
datetime fecha  
  
// Inicializando la variable fecha  
fecha=datetime(today())  
  
Commit;  
// Primero se borran las enfermedades asignadas a un número de cita  
delete from paciente_emfermedad  
where codigo_cita =: gs_codcita;  
  
// Se agrega un contador para distinguir aquellas enfermedades que estan marcadas  
o no  
FOR ll_contador = 1 TO dw_1.RowCount()  
    ll_codigoenfermedad=  
dw_1.getitemnumber(ll_contador,'codigo_enfermedad')  
    ls_agregar = dw_1.getitemString(ll_contador, "agregar")  
  
    // Si estan marcadas se deben agregar a la base de datos  
    if ls_agregar='1' then  
        insert into paciente_emfermedad (codigo_cita, codigo_enfermedad,  
usuario_d,fecha_d)  
        values (:gs_codcita, :ll_codigoenfermedad,:gs_login,:fecha);
```



end if

next

// Mensaje de consulta a los usuarios

messagebox("Actualización de Datos","Proceso terminado exitosamente!!!")

Commit;

end event

type st_2 from statictext within w_edt_cita_enfermedad

integer x = 398

integer y = 60

integer width = 498

integer height = 80

integer textsize = -10

integer weight = 700

fontcharset fontcharset = ansi!

fontpitch fontpitch = variable!

fontfamily fontfamily = swiss!

string facename = "Arial"

long textcolor = 16711680

long backcolor = 12632256

boolean border = true

borderstyle borderstyle = stylelowered!

boolean focusrectangle = false

end type

type st_1 from statictext within w_edt_cita_enfermedad



```
integer x = 82  
integer y = 60  
integer width = 320  
integer height = 80  
integer textsize = -10  
integer weight = 700  
fontcharset fontcharset = ansi!  
fontpitch fontpitch = variable!  
fontfamily fontfamily = swiss!  
string facename = "Arial"  
long textcolor = 33554432  
long backcolor = 12632256  
string text = "Cod. Cita: "  
boolean focusrectangle = false  
end type  
  
type dw_1 from datawindow within w_edt_cita_enfermedad  
event filtro ( )  
integer x = 37  
integer y = 384  
integer width = 2743  
integer height = 1280  
integer taborder = 30  
string title = "none"  
string dataobject = "d_edt_cita5"  
boolean vscrollbar = true  
boolean border = false  
boolean livescroll = true  
end type
```



```
event filtro();  
// Inicializando variables  
string ls_filter1  
// Filtrando por código de enfermedad  
ls_filter1='enfermedad like'%'"+gs_enfermedad+'%''  
// Actualizando el data windows  
dw_1.setFilter(ls_filter1);  
dw_1.filter();  
  
end event  
  
event retrieveend;  
// Inicializando variables  
long ll_contador, ll_codigoenfermedad, ll_numero  
Commit;  
// Actualizando el datawindows con los datos que pertenecen al número de la cita  
FOR ll_contador = 1 TO dw_1.RowCount()  
    ll_codigoenfermedad=  
dw_1.getItemNumber(ll_contador,'codigo_enfermedad')  
    // Consultando si existen datos agregados en este número de cita  
    Select count(codigo_enfermedad)  
    into :ll_numero  
    from paciente_emfermedad  
    where codigo_cita =: gs_codcita and  
           codigo_enfermedad =: ll_codigoenfermedad;  
    // Si existen se debe actualizar el data windows marcando con una flecha  
  
    if ll_numero>0 then
```



```
dw_1.SetItem(ll_contador,"agregar",'1')
```

```
dw_1.AcceptText()
```

```
end if
```

```
next
```

```
Commit;
```

```
end event
```

```
type gb_1 from groupbox within w_edt_cita_enfermedad
```

```
integer x = 1317
```

```
integer width = 1353
```

```
integer height = 192
```

```
integer taborder = 40
```

```
integer textsize = -10
```

```
integer weight = 700
```

```
fontcharset fontcharset = ansi!
```

```
fontpitch fontpitch = variable!
```

```
fontfamily fontfamily = swiss!
```

```
string facename = "Arial"
```

```
long textcolor = 16711680
```

```
long backcolor = 12632256
```

```
end type
```

VENTANA W_EDT_CITA_EXAMEN

```
forward
```

```
global type w_edt_cita_examen from window
```

```
end type
```

```
type sle_1 from singlelineedit within w_edt_cita_examen
```



```
end type
type cb_3 from commandbutton within w_edt_cita_examen
end type
type cb_2 from commandbutton within w_edt_cita_examen
end type
type cb_1 from commandbutton within w_edt_cita_examen
end type
type st_2 from statictext within w_edt_cita_examen
end type
type st_1 from statictext within w_edt_cita_examen
end type
type dw_1 from datawindow within w_edt_cita_examen
end type
type gb_1 from groupbox within w_edt_cita_examen
end type
end forward

global type w_edt_cita_examen from window
integer width = 3109
integer height = 2004
boolean titlebar = true
string title = "Catálogo de Exámenes"
boolean controlmenu = true
boolean maxbox = true
boolean resizable = true
long bgcolor = 12632256
string icon = "AppIcon!"
boolean center = true
sle_1 sle_1
```



```
cb_3 cb_3
cb_2 cb_2
cb_1 cb_1
st_2 st_2
st_1 st_1
dw_1 dw_1
gb_1 gb_1
end type
global w_edt_cita_examen w_edt_cita_examen
```

```
on w_edt_cita_examen.create
this.sle_1=create sle_1
this.cb_3=create cb_3
this.cb_2=create cb_2
this.cb_1=create cb_1
this.st_2=create st_2
this.st_1=create st_1
this.dw_1=create dw_1
this.gb_1=create gb_1
this.Control[]={this.sle_1,&
this.cb_3,&
this.cb_2,&
this.cb_1,&
this.st_2,&
this.st_1,&
this.dw_1,&
this.gb_1}
end on
```



```
on w_edt_cita_examen.destroy
destroy(this.sle_1)
destroy(this.cb_3)
destroy(this.cb_2)
destroy(this.cb_1)
destroy(this.st_2)
destroy(this.st_1)
destroy(this.dw_1)
destroy(this.gb_1)
end on

event open;
// Inicializando data windows
dw_1.SetTransObject(SQLCA)
dw_1.retrieve()
// Asignado el número de cita en el cuadro de texto
st_2.text=gs_codcita

end event

type sle_1 from singlelineedit within w_edt_cita_examen
integer x = 73
integer y = 256
integer width = 2779
integer height = 96
integer taborder = 60
integer textsize = -10
integer weight = 700
```



```
fontcharset fontcharset = ansi!  
fontpitch fontpitch = variable!  
fontfamily fontfamily = swiss!  
string facename = "Arial"  
long textcolor = 255  
textcase textcase = upper!  
borderstyle borderstyle = stylelowered!  
end type
```

```
type cb_3 from commandbutton within w_edt_cita_examen  
integer x = 1975  
integer y = 60  
integer width = 402  
integer height = 112  
integer taborder = 40  
integer textsize = -10  
integer weight = 700  
fontcharset fontcharset = ansi!  
fontpitch fontpitch = variable!  
fontfamily fontfamily = swiss!  
string facename = "Arial"  
string text = "&Buscar"  
end type
```

```
event clicked;  
// Inicialización de variables  
// Asignando a la variable el código digitado  
gs_examen = trim(sle_1.text)  
// Filtrando de acuerdo a la variable asignada
```



dw_1.event filtro()

dw_1.retrieve()

end event

type cb_2 from commandbutton within w_edt_cita_examen

integer x = 2418

integer y = 60

integer width = 402

integer height = 112

integer taborder = 10

integer textsize = -10

integer weight = 700

fontcharset fontcharset = ansi!

fontpitch fontpitch = variable!

fontfamily fontfamily = swiss!

string facename = "Arial"

string text = "&Salir"

end type

event clicked;

// Cerrar la ventana del examen

Close(w_edt_cita_examen)

end event

type cb_1 from commandbutton within w_edt_cita_examen

integer x = 1550

integer y = 60



```
integer width = 402
integer height = 112
integer taborder = 20
integer textsize = -10
integer weight = 700
fontcharset fontcharset = ansi!
fontpitch fontpitch = variable!
fontfamily fontfamily = swiss!
string facename = "Arial"
string text = "&Agregar"
end type

event clicked;
// Inicializando variables a utilizar
string ls_agregar, ls_resultado
integer ll_contador, ll_numero, ll_cuenta, ll_codigoexamen, ll_cont
datetime fecha

fecha=datetime(today())

Commit;
// Iniciando contador para hacer una de tres operaciones: eliminar, agregar o
actualizar
FOR ll_contador = 1 TO dw_1.RowCount()
    ll_codigoexamen= dw_1.getitemnumber(ll_contador,'id_examen')
    ls_agregar = dw_1.getitemString(ll_contador, "agregar")
    ls_resultado = dw_1.getitemString(ll_contador, "resultado")
```



```
// Si la variable agregar la cual contiene si un examen esta chequeado o no es
igual a cero
// se borra el registro
if ls_agregar='0' then
    delete from paciente_examen where codigo_cita =: gs_codcita and
    id_examen =:ll_codigoexamen;
else
    // si la variable es igual a se verifica si ya se encuentra agregada en la
base de datos
    Select count(codigo_cita)
    into      :ll_cont
    from      paciente_examen
    where codigo_cita =: gs_codcita and
           id_examen =:ll_codigoexamen;
    // si ya se encuentra agregada se actualizan los datos necesarios

    if ll_cont>0 then
        updatepaciente_examen
        set      resultado =: ls_resultado,
                usuario_m =: gs_login,
                fecha_m   =: fecha
        where codigo_cita =: gs_codcita and
               id_examen =:ll_codigoexamen;
    // si no se encuentra agregada se inserta un nuevo registro

    else
        insert into paciente_examen (codigo_cita, id_examen,
usuario_d,fecha_d, despachado, resultado)
```



```
values (:gs_codcita, :ll_codigoexamen,:gs_login,:fecha,'0',
:ls_resultado);
end if
end if

next

// mensaje que se muestra cuando se finaliza el proceso
messagebox("Actualización de Datos","Proceso terminado exitosamente!!!")

Commit;

end event

type st_2 from statictext within w_edt_cita_examen
integer x = 398
integer y = 60
integer width = 498
integer height = 80
integer textsize = -10
integer weight = 700
fontcharset fontcharset = ansi!
fontpitch fontpitch = variable!
fontfamily fontfamily = swiss!
string facename = "Arial"
long textcolor = 16711680
long bgcolor = 12632256
boolean border = true
borderstyle borderstyle = stylelowered!
boolean focusrectangle = false
```



end type

type st_1 from statictext within w_edt_cita_examen

integer x = 82

integer y = 60

integer width = 320

integer height = 80

integer textsize = -10

integer weight = 700

fontcharset fontcharset = ansi!

fontpitch fontpitch = variable!

fontfamily fontfamily = swiss!

string facename = "Arial"

long textcolor = 33554432

long backcolor = 12632256

string text = "Cod. Cita: "

boolean focusrectangle = false

end type

type dw_1 from datawindow within w_edt_cita_examen

event filtro ()

integer x = 46

integer y = 408

integer width = 2853

integer height = 1224

integer taborder = 30

string title = "none"

string dataobject = "d_edt_cita7"

boolean vscrollbar = true



boolean border = false

boolean livescroll = true

end type

event filtro();

// Inicializando variables

string ls_filter1

// Filtrando por código de enfermedad

ls_filter1="exámenes_examen like '%"+gs_examen+"%'"

// Actualizando el data windows

dw_1.setFilter(ls_filter1);

dw_1.filter();

end event

event retrieveend;

// Inicializando variables

string ls_resultado

long ll_contador, ll_codigoexamen, ll_numero

Commit;

// Iniciando contador para reflejar los exámenes adicionales a cada cita

FOR ll_contador = 1 TO dw_1.RowCount()

ll_codigoexamen= dw_1.getItemNumber(ll_contador,'id_examen')

Select count(id_examen)

into :ll_numero

from paciente_examen

where codigo_cita =: gs_codcita and

id_examen =: ll_codigoexamen;



```
// Si existe entonces se muestra en el datawindows
if ll_numero>0 then
    Select resultado
    into      :ls_resultado
    from      paciente_examen
    where     codigo_cita =: gs_codcita and
              id_examen =: ll_codigoexamen;
    // Actualizando resultados en el data windows
    dw_1.SetItem(ll_contador,"agregar",'1')
    dw_1.SetItem(ll_contador,"resultado",ls_resultado)
    dw_1.AcceptText()
end if

next
Commit;
end event

type gb_1 from groupbox within w_edt_cita_examen
integer x = 1463
integer width = 1390
integer height = 192
integer taborder = 50
integer textsize = -10
integer weight = 700
fontcharset fontcharset = ansi!
fontpitch fontpitch = variable!
fontfamily fontfamily = swiss!
string facename = "Arial"
long textcolor = 16711680
```



long bgcolor = 12632256

end type

VENTANA W_EDT_CITA_MEDICAMENTO

forward

global type w_edt_cita_medicamento from window

end type

type cb_3 from commandbutton within w_edt_cita_medicamento

end type

type sle_1 from singlelineedit within w_edt_cita_medicamento

end type

type cb_2 from commandbutton within w_edt_cita_medicamento

end type

type cb_1 from commandbutton within w_edt_cita_medicamento

end type

type st_2 from statictext within w_edt_cita_medicamento

end type

type st_1 from statictext within w_edt_cita_medicamento

end type

type dw_1 from datawindow within w_edt_cita_medicamento

end type

type gb_1 from groupbox within w_edt_cita_medicamento

end type

end forward



```
global type w_edt_cita_medicamento from window
integer width = 3744
integer height = 1816
boolean titlebar = true
string title = "Catalogo de Medicamentos"
boolean controlmenu = true
boolean minbox = true
boolean maxbox = true
boolean resizable = true
long bgcolor = 12632256
string icon = "AppIcon!"
boolean center = true
cb_3 cb_3
sle_1 sle_1
cb_2 cb_2
cb_1 cb_1
st_2 st_2
st_1 st_1
dw_1 dw_1
gb_1 gb_1
end type
global w_edt_cita_medicamento w_edt_cita_medicamento

on w_edt_cita_medicamento.create
this.cb_3=create cb_3
this.sle_1=create sle_1
this.cb_2=create cb_2
this.cb_1=create cb_1
this.st_2=create st_2
```



```
this.st_1=create st_1
this.dw_1=create dw_1
this.gb_1=create gb_1
this.Control[]={this.cb_3,&
this.sle_1,&
this.cb_2,&
this.cb_1,&
this.st_2,&
this.st_1,&
this.dw_1,&
this.gb_1}
end on

on w_edt_cita_medicamento.destroy
destroy(this.cb_3)
destroy(this.sle_1)
destroy(this.cb_2)
destroy(this.cb_1)
destroy(this.st_2)
destroy(this.st_1)
destroy(this.dw_1)
destroy(this.gb_1)
end on

event open;
// Inicializando data windows
dw_1.SetTransObject(SQLCA)
dw_1.retrieve()
// Asignando valor de variable del codigo de la cita a un cuadro de texto
```



st_2.text=gs_codcita

end event

type cb_3 from commandbutton within w_edt_cita_medicamento

integer x = 2743

integer y = 60

integer width = 402

integer height = 112

integer taborder = 60

integer textsize = -10

integer weight = 700

fontcharset fontcharset = ansi!

fontpitch fontpitch = variable!

fontfamily fontfamily = swiss!

string facename = "Arial"

string text = "&Buscar"

end type

event clicked;

// Inicialización de variables

// Asignando a la variable el código digitado

gs_medicamento = trim(sle_1.text)

// Filtrando de acuerdo a la variable asignada

dw_1.event filtro()

dw_1.retrieve()



end event

type sle_1 from singlelineedit within w_edt_cita_medicamento

integer x = 37

integer y = 224

integer width = 3584

integer height = 96

integer taborder = 40

integer textsize = -10

integer weight = 700

fontcharset fontcharset = ansi!

fontpitch fontpitch = variable!

fontfamily fontfamily = swiss!

string facename = "Arial"

long textcolor = 255

textcase textcase = upper!

borderstyle borderstyle = stylelowered!

end type

type cb_2 from commandbutton within w_edt_cita_medicamento

integer x = 3182

integer y = 60

integer width = 402

integer height = 112

integer taborder = 10

integer textsize = -10

integer weight = 700

fontcharset fontcharset = ansi!

fontpitch fontpitch = variable!



```
fontfamily fontfamily = swiss!
```

```
string facename = "Arial"
```

```
string text = "&Salir"
```

```
end type
```

```
event clicked;
```

```
// Cerrando la ventana
```

```
Close(w_edt_cita_medicamento)
```

```
end event
```

```
type cb_1 from commandbutton within w_edt_cita_medicamento
```

```
integer x = 2313
```

```
integer y = 60
```

```
integer width = 402
```

```
integer height = 112
```

```
integer taborder = 20
```

```
integer textsize = -10
```

```
integer weight = 700
```

```
fontcharset fontcharset = ansi!
```

```
fontpitch fontpitch = variable!
```

```
fontfamily fontfamily = swiss!
```

```
string facename = "Arial"
```

```
string text = "&Agregar"
```

```
end type
```

```
event clicked;
```

```
// Inicializando variables
```

```
string ls_idgrupo, ls_idsubgrupo, ls_idmedicamento, ls_dosis
```



```
integer      ll_contador, ll_numero, ll_cuenta, ll_cantidad, ll_cant2, ll_suma,  
ll_resta  
datetime    fecha
```

```
fecha= datetime(today())
```

```
commit;
```

```
// Iniciando lazo for para hacer una de tres operaciones: eliminar, agregar o  
actualizar
```

```
FOR ll_contador = 1 TO dw_1.RowCount()
```

```
    ls_idgrupo= dw_1.getitemstring(ll_contador,'id_grupo')
```

```
    ls_idsubgrupo= dw_1.getitemstring(ll_contador,'id_subgrupo')
```

```
    ls_idmedicamento= dw_1.getitemstring(ll_contador,'id_medicamento')
```

```
    ls_dosis= dw_1.getitemstring(ll_contador,'dosis')
```

```
    ll_numero = dw_1.getitemNumber(ll_contador, "cant")
```

```
    ll_cantidad = dw_1.getitemNumber(ll_contador, "cantidad")
```

```
    ll_cant2 = dw_1.getitemNumber(ll_contador, "cant1")
```

```
// Si la cantidad es cero significa que debe eliminarse el medicamento
```

```
if ll_numero = 0 then
```

```
    // Primero se debe hacer el calculo para actualizar el stock del
```

```
medicamento
```

```
    if ll_cant2 >0 then
```

```
        ll_suma = ll_cantidad + ll_cant2
```

```
        update medicamento set cantidad =: ll_suma
```

```
        where id_grupo =: ls_idgrupo and
```

```
        id_subgrupo =: ls_idsubgrupo and
```

```
        id_medicamento =: ls_idmedicamento;
```

```
    end if
```



```
// Segundo se debe borrar el medicamento
delete from paciente_medicamento
where id_grupo =: ls_idgrupo and
      id_subgrupo =: ls_idsubgrupo and
      id_medicamento =: ls_idmedicamento and
      codigo_cita =: gs_codcita;

end if

// Si es mayor que cero, entonces primero se busca si esta registrado
if ll_numero>0      then

    if ll_numero <= ll_cantidad then
        // se hace la consulta para ver si ya existe
        Select count(codigo_cita)
        into      :ll_cuenta
        from      paciente_medicamento
        where     id_grupo =: ls_idgrupo and
                  id_subgrupo =: ls_idsubgrupo and
                  id_medicamento =: ls_idmedicamento and
                  codigo_cita =: gs_codcita;

        if ll_cuenta>0 then
            // Si ya existe se actualizan las cantidades del
medicamento

            ll_suma = ll_cantidad + ll_cant2
            ll_resto = ll_suma - ll_numero

            Update      paciente_medicamento
            set         cantidad =: ll_numero,
```



```

                                dosis =: ls_dosis,
                                usuario_m =: gs_login,
                                fecha_m = :fecha
                                where      id_grupo =: ls_idgrupo and
                                           id_subgrupo =: ls_idsubgrupo and
                                           id_medimento =:
ls_idmedimento and
                                           codigo_cita =: gs_codcita;

                                end if

                                // si no existe se agrega a la tabla de paciente_medimento
                                if ll_cuenta =0 or isnull(ll_cuenta) then
                                    ll_resta = ll_cantidad - ll_numero
                                    insert into paciente_medimento (codigo_cita,
id_grupo, id_subgrupo, id_medimento, cantidad, usuario_d, fecha_d, despachado,
dosis)
                                        values (:gs_codcita, :ls_idgrupo, :ls_idsubgrupo,
:ls_idmedimento, :ll_numero, :gs_login, :fecha, '0', :ls_dosis);
                                end if

                                // Se actualiza el stock del medicamento
                                update medicamento set cantidad =: ll_resta
                                where id_grupo =: ls_idgrupo and
                                id_subgrupo =: ls_idsubgrupo and
                                id_medimento =: ls_idmedimento;

                                else

                                // Mensaje para indicar que no se puede hacer una descarga de
medicamentos
```



Messagebox ("Verificación de medicamentos","No se puede despachar ya que no hay suficiente cantidad en stock")

end if

end if

Next

commit;

// Mensaje que se muestra cuando se termina el bucle for, indicando su finalización

Messagebox("Actualización de datos","Proceso terminado exitosamente!!!")

end event

type st_2 from statictext within w_edt_cita_medicamento

integer x = 398

integer y = 60

integer width = 498

integer height = 80

integer textsize = -10

integer weight = 700

fontcharset fontcharset = ansi!

fontpitch fontpitch = variable!

fontfamily fontfamily = swiss!

string facename = "Arial"

long textcolor = 16711680

long backcolor = 12632256

boolean border = true

borderstyle borderstyle = stylelowered!

boolean focusrectangle = false

end type



type st_1 from statictext within w_edt_cita_medicamento

integer x = 82

integer y = 60

integer width = 320

integer height = 80

integer textsize = -10

integer weight = 700

fontcharset fontcharset = ansi!

fontpitch fontpitch = variable!

fontfamily fontfamily = swiss!

string facename = "Arial"

long textcolor = 33554432

long backcolor = 12632256

string text = "Cod. Cita: "

boolean focusrectangle = false

end type

type dw_1 from datawindow within w_edt_cita_medicamento

event filtro ()

integer x = 27

integer y = 376

integer width = 3648

integer height = 1228

integer taborder = 30

string title = "none"

string dataobject = "d_edt_cita4"

boolean vscrollbar = true

boolean border = false



boolean livescroll = true

end type

event filtro();

// Inicializando variables

string ls_filter1

// Filtrando por código de enfermedad

ls_filter1='nombre_medicamento like'%'+gs_medicamento+'%''

// Actualizando el data windows

dw_1.setFilter(ls_filter1);

dw_1.filter();

end event

event retrieveend;

// Inicializando variables

string ls_idgrupo, ls_idsubgrupo, ls_idmedicamento, ls_dosis

long ll_contador, ll_numero

Commit;

// Iniciando contador para verificar si esta agregado en la base de datos

FOR ll_contador = 1 TO dw_1.RowCount()

ll_numero=0

ls_idgrupo= dw_1.getitemstring(ll_contador,'id_grupo')

ls_idsubgrupo= dw_1.getitemstring(ll_contador,'id_subgrupo')

ls_idmedicamento= dw_1.getitemstring(ll_contador,'id_medicamento')

// Consultando los valores en la tabla de paciente_medicamentos

Select cantidad, dosis

into ll_numero, :ls_dosis



```
From      paciente_medicamento
where     id_grupo =: ls_idgrupo and
            id_subgrupo =: ls_idsubgrupo and
            id_medicamento =: ls_idmedicamento and
            codigo_cita =: gs_codcita;

// Si se encuentra en la base de datos asigna el valor del campo en el
datawindows
if ll_numero>0 then
    dw_1.SetItem (ll_contador,"dosis",ls_dosis)
else
    ll_numero = 0
end if

// Asignando valores en el datawindows
dw_1.SetItem (ll_contador,"cant",ll_numero)
dw_1.SetItem (ll_contador,"cant1",ll_numero)

dw_1.AcceptText()

Next

commit;

end event

event rowfocuschanged;dw_1.AcceptText()

end event
```



```
type gb_1 from groupbox within w_edt_cita_medicamento
integer x = 2267
integer width = 1353
integer height = 192
integer taborder = 50
integer textsize = -10
integer weight = 700
fontcharset fontcharset = ansi!
fontpitch fontpitch = variable!
fontfamily fontfamily = swiss!
string facename = "Arial"
long textcolor = 16711680
long backcolor = 12632256
end type
```

VENTANA W_EDT_CONSULTA

```
forward
global type w_edt_consulta from w_edt
end type
type dw_1 from u_dw within w_edt_consulta
end type
end forward

global type w_edt_consulta from w_edt
integer width = 3045
integer height = 980
string title = "Consulta"
```



dw_1 dw_1

end type

global w_edt_consulta w_edt_consulta

type variables

end variables

forward prototypes

**public subroutine wf_habilitardeshabilitarcontroles (string a_tipoproveedor, integer
a_accion)**

end prototypes

**public subroutine wf_habilitardeshabilitarcontroles (string a_tipoproveedor, integer
a_accion);**

end subroutine

on w_edt_consulta.create

int iCurrent

call super::create

this.dw_1=create dw_1

iCurrent=UpperBound(this.Control)

this.Control[iCurrent+1]=this.dw_1

end on

on w_edt_consulta.destroy

call super::destroy

destroy(this.dw_1)

end on



```
type p_rowfocus from w_edt`p_rowfocus within w_edt_consulta  
end type
```

```
type cb_cancel from w_edt`cb_cancel within w_edt_consulta  
integer x = 2633  
integer y = 184  
integer width = 375  
string text = "&Salir"  
end type
```

```
type cb_ok from w_edt`cb_ok within w_edt_consulta  
integer x = 2633  
integer y = 40  
integer width = 375  
end type
```

```
type dw_1 from u_dw within w_edt_consulta  
integer x = 27  
integer y = 44  
integer width = 2587  
integer height = 808  
integer taborder = 40  
boolean bringtotop = true  
string dataobject = "d_edt_consulta"  
end type
```

```
event itemchanged;call super::itemchanged;  
// Inicialización de variables
```



```
string ls_nombre,    ls_nombre2,    ls_nombre3,    ls_apellido1,    ls_apellido2,
ls_codigopaciente
string ls_nombret
// Despues que el campo del codigo_pacien de pierde el foco
if dwo.name='codigo_paciente' then
    // capturando el valor en la variable ls_codigopaciente
    ls_codigopaciente = data
    // rellenando con ceros a la izquierda
    ls_codigopaciente = fill('0', 10 - len(ls_codigopaciente)) + ls_codigopaciente
    // Consultando tabla expediente con ese código del paciente
    Select primer_nombre, segundo_nombre, tercer_nombre, primer_apellido,
segundo_apellido
    into :ls_nombre, :ls_nombre2, :ls_nombre3, :ls_apellido1, :ls_apellido2
    from expediente
    where codigo_paciente =: ls_codigopaciente;
    // Asignandole el nombre a una variable
    ls_nombret = trim(ls_nombre)+' '+trim(ls_nombre2)+' '+ trim(ls_nombre3)+'
'+trim(ls_apellido1)+' '+trim(ls_apellido2)
    // Asignando las variables en el datawindows
    dw_1.SetItem(row, "texto", ls_nombret)
    dw_1.SetItem(row, "codigo_paciente", ls_codigopaciente)
    dw_1.AcceptText()

end if
end event

event retrieveend;call super::retrieveend;
//Declaración de variables
```



```
string ls_nombre, ls_nombre2, ls_nombre3, ls_apellido1, ls_apellido2,  
ls_codigopaciente  
string ls_nombret
```

```
// Asignado el valor a la variable ls_codigopaciente
```

```
ls_codigopaciente = dw_1.GetItemString(rowcount,"codigo_paciente")
```

```
// Haciendo la consulta en la tabla expediente para hallar el nombre del paciente
```

```
Select primer_nombre, segundo_nombre, tercer_nombre, primer_apellido,  
segundo_apellido
```

```
into :ls_nombre, :ls_nombre2, :ls_nombre3, :ls_apellido1, :ls_apellido2
```

```
from expediente
```

```
where codigo_paciente =: ls_codigopaciente;
```

```
// Asignando el valor encontrado en una variable
```

```
ls_nombret = trim(ls_nombre)+' '+trim(ls_nombre2)+' '+ trim(ls_nombre3)+'  
'+trim(ls_apellido1)+' '+trim(ls_apellido2)
```

```
// Actualizando el datawindows con la variable
```

```
dw_1.SetItem(rowcount, "texto", ls_nombret)
```

```
return 1
```

```
end event
```

```
event ue_validaterow;call super::ue_validaterow;
```

```
//Declaración de variables
```

```
integer rc
```

```
string ls_codigopaciente, ls_idproveedor, ls_motivo, ls_cita, ls_vez
```

```
datetime ldt_fecha
```

```
//Obtener los campos que se validaran
```



```
ls_codigopaciente = this.GetItemString (al_row, "codigo_paciente")
ls_codigopaciente = Trim(ls_codigopaciente)
ls_codigopaciente = fill('0', 10 - len(ls_codigopaciente)) + ls_codigopaciente
ls_idproveedor      = this.GetItemString (al_row, "id_proveedor")
ls_motivo           = this.GetItemString (al_row, "motivo")
ls_cita             = this.GetItemString (al_row, "cita")
ls_vez             = this.GetItemString (al_row, "vez")
ldt_fecha          = this.GetItemDateTime (al_row, "fecha_consulta")

dw_1.SetItem(al_row, "codigo_paciente", ls_codigopaciente)
/*****
*****
//VALIDACION
/*****
*****
///Validar los campos a insertar en la tabla
If isnull(ls_codigopaciente) or trim(ls_codigopaciente) = "" then
    rc = this.uf_setmsgerror(al_row,"codigo_paciente","Debe digitarse el código
del paciente")
End if
If isnull(ls_idproveedor) or trim(ls_idproveedor) = "" then
    rc = this.uf_setmsgerror(al_row,"id_proveedor","Debe digitarse médico a
consultar")
End if
If isnull(ls_cita) or trim(ls_cita) = "" then
    rc = this.uf_setmsgerror(al_row,"cita","Debe digitarse si es primera vez o
subsecuente")
End if
If isnull(ls_vez) or trim(ls_Vez) = "" then
```



```
rc = this.uf_setmsgerror(al_row,"vez","Debe digitarse si tiene cita o no")
End if
If isnull(ldt_fecha) then
    ldt_fecha = datetime(today())
    dw_1.SetItem (al_row,"fecha_consulta",ldt_fecha)
End if

// Insertar las huellas de auditoría
If parent.event ue_is_insertmode() = 1 then
    dw_1.SetItem(al_row, "usuario_d", gs_login)
    dw_1.SetItem(al_row, "fecha_d", today())
Else
    dw_1.SetItem(al_row, "usuario_m", gs_login)
    dw_1.SetItem(al_row, "fecha_m", today())
End if

//Retornar valores

Return 1
end event

event ue_insertkeyprimer;call super::ue_insertkeyprimer;

// Inicializando variable
string ls_codigo
// Consultando el maximo código agregado en la tabla consulta
Select max(codigo_consulta) into :ls_codigo from consulta;
// Verificando si es el primer registro
```



```
if isnull(ls_codigo) or trim(ls_codigo) ='' then
    // Si no hay ningún registro se le asigna el primero
    ls_codigo = '0000000001'
else
    // Si hay registros, entonces
    //Calcular el correlativo
    ls_codigo = string(long(ls_codigo) + 1)

    //Verificar que el codigo sea de 10 caracteres
    if len(ls_codigo) <> 10 then
        ls_codigo = fill('0', 10 - len(ls_codigo)) + ls_codigo
    end if

end if

// Insertar el código en el datawindows
dw_1.SetItem(al_row, "codigo_consulta", ls_codigo)

return 1
end event
```

VENTANA W_EDT_CONSULTA_REPORTE1

```
forward
global type w_edt_consulta_reporte1 from window
end type
type dw_4 from datawindow within w_edt_consulta_reporte1
end type
type dw_3 from datawindow within w_edt_consulta_reporte1
```



```
end type
type dw_2 from datawindow within w_edt_consulta_reporte1
end type
type dw_1 from datawindow within w_edt_consulta_reporte1
end type
type cb_2 from commandbutton within w_edt_consulta_reporte1
end type
type gb_1 from groupbox within w_edt_consulta_reporte1
end type
end forward
```

```
global type w_edt_consulta_reporte1 from window
integer width = 3419
integer height = 2052
boolean titlebar = true
string title = "Historial Clínico"
boolean controlmenu = true
boolean minbox = true
boolean maxbox = true
boolean resizable = true
long backcolor = 12632256
string icon = "AppIcon!"
boolean center = true
dw_4 dw_4
dw_3 dw_3
dw_2 dw_2
dw_1 dw_1
cb_2 cb_2
gb_1 gb_1
```



end type

global w_edt_consulta_reporte1 w_edt_consulta_reporte1

on w_edt_consulta_reporte1.create

this.dw_4=create dw_4

this.dw_3=create dw_3

this.dw_2=create dw_2

this.dw_1=create dw_1

this.cb_2=create cb_2

this.gb_1=create gb_1

this.Control[]={this.dw_4,&

this.dw_3,&

this.dw_2,&

this.dw_1,&

this.cb_2,&

this.gb_1}

end on

on w_edt_consulta_reporte1.destroy

destroy(this.dw_4)

destroy(this.dw_3)

destroy(this.dw_2)

destroy(this.dw_1)

destroy(this.cb_2)

destroy(this.gb_1)

end on

event open;

// Inicializando variables



integer ll_contador, ll_numero, ll_codigoenfermedad

// Inicializando datawindows

dw_1.SetTransObject(SQLCA)

dw_1.retrieve()

// Inicializando el campo del código de paciente

dw_1.SetItem(1, "cita_codigo_paciente", "")

dw_1.AcceptText()

// Inicializando los otros datawindows

dw_2.SetTransObject(SQLCA)

dw_3.SetTransObject(SQLCA)

dw_4.SetTransObject(SQLCA)

end event

type dw_4 from datawindow within w_edt_consulta_reporte1

event filtro ()

integer x = 46

integer y = 1280

integer width = 3291

integer height = 600

integer taborder = 40

string title = "none"

string dataobject = "d_edt_con_3"

boolean vscrollbar = true

boolean border = false

boolean livescroll = true



end type

event filtro();

String ls_filter1

// Filtro según el código del paciente

ls_filter1="codigo_paciente="+gs_codpac+""

dw_4.setFilter(ls_filter1);

dw_4.filter();

end event

type dw_3 from datawindow within w_edt_consulta_reporte1

event filtro ()

integer x = 46

integer y = 768

integer width = 3291

integer height = 488

integer taborder = 30

string title = "none"

string dataobject = "d_edt_con_2"

boolean vscrollbar = true

boolean border = false

boolean livescroll = true

end type

event filtro();

String ls_filter1



```
// Filtro según el código del paciente
ls_filter1="v_cita_codigo_paciente="+gs_codpac+""
dw_3.setFilter(ls_filter1);
dw_3.filter();

end event

type dw_2 from datawindow within w_edt_consulta_reporte1
event filtro ( )
integer x = 37
integer y = 256
integer width = 3291
integer height = 480
integer taborder = 20
string title = "none"
string dataobject = "d_edt_con_1"
boolean vscrollbar = true
boolean border = false
boolean livescroll = true
end type

event filtro();
String ls_filter1

// Filtro según el código del paciente
ls_filter1="codigo="+gs_codpac+""
dw_2.setFilter(ls_filter1);
```



```
dw_2.filter();
```

```
end event
```

```
type dw_1 from datawindow within w_edt_consulta_reporte1
```

```
integer x = 46
```

```
integer y = 40
```

```
integer width = 2583
```

```
integer height = 172
```

```
integer taborder = 20
```

```
string title = "none"
```

```
string dataobject = "d_edt_con_4"
```

```
boolean border = false
```

```
boolean livescroll = true
```

```
end type
```

```
event itemchanged;
```

```
string ls_nombre, ls_nombre2, ls_nombre3, ls_apellido1, ls_apellido2,
```

```
ls_codigopaciente
```

```
string ls_nombret
```

```
// Despues de haber digitado el codigo del paciente
```

```
if dwo.name='cita_codigo_paciente' then
```

```
    // Asegurandonos que el usuario digito los digitos del código del paciente
```

```
    ls_codigopaciente = data
```

```
    ls_codigopaciente = fill('0', 10 - len(ls_codigopaciente)) + ls_codigopaciente
```

```
    gs_codpac = ls_codigopaciente
```

```
    // Haciendo una consulta para capturar el nombre del paciente
```



```
Select primer_nombre, segundo_nombre, tercer_nombre, primer_apellido,  
segundo_apellido  
into :ls_nombre, :ls_nombre2, :ls_nombre3, :ls_apellido1, :ls_apellido2  
from expediente  
where codigo_paciente =: ls_codigopaciente;  
// En la variable guardamos el nombre del paciente  
ls_nombret = trim(ls_nombre)+' '+trim(ls_nombre2)+' '+ trim(ls_nombre3)+'  
'+trim(ls_apellido1)+' '+trim(ls_apellido2)  
// El nombre es actualizado en el datawindows  
this.Modify("t_1.text = '" + trim(ls_nombret)+"'")  
// Actualizando el datawindows con el código del paciente y la fecha de la cita  
dw_1.SetItem(row, "cita_codigo_paciente", gs_codpac)  
// dw_1.SetItem(row, "cita_fecha_cita", today())  
dw_1.AcceptText()  
// Filtrando los demas datawindows para que solo muestren los registros  
// que pertenecen a este paciente en esta cita  
dw_2.event filtro()  
dw_2.retrieve()  
dw_3.event filtro()  
dw_3.retrieve()  
dw_4.event filtro()  
dw_4.retrieve()  
  
end if  
end event  
  
type cb_2 from commandbutton within w_edt_consulta_reporte1  
integer x = 2917  
integer y = 60
```



```
integer width = 402  
integer height = 112  
integer taborder = 10  
integer textsize = -10  
integer weight = 700  
fontcharset fontcharset = ansi!  
fontpitch fontpitch = variable!  
fontfamily fontfamily = swiss!  
string facename = "Arial"  
string text = "&Salir"  
end type  
  
event clicked;  
// Cerrando la ventana  
Close(w_edt_consulta_reporte1)  
end event  
  
type gb_1 from groupbox within w_edt_consulta_reporte1  
integer x = 2885  
integer width = 453  
integer height = 196  
integer taborder = 20  
integer textsize = -10  
integer weight = 700  
fontcharset fontcharset = ansi!  
fontpitch fontpitch = variable!  
fontfamily fontfamily = swiss!  
string facename = "Arial"  
long textcolor = 16711680
```



long bgcolor = 12632256

end type

VENTANA W_EDT_EXPEDIENTE

forward

global type w_edt_expediente from w_edt

end type

type dw_1 from u_dw within w_edt_expediente

end type

end forward

global type w_edt_expediente from w_edt

integer width = 4672

integer height = 2460

string title = "Paciente"

dw_1 dw_1

end type

global w_edt_expediente w_edt_expediente

type variables

end variables

forward prototypes

**public subroutine wf_habilitardeshabilitarcontroles (string a_tipoproveedor, integer
a_accion)**

end prototypes



```
public subroutine wf_habilitardeshabilitarcontroles (string a_tipoproveedor, integer  
a_accion);  
end subroutine
```

```
on w_edt_expediente.create  
int iCurrent  
call super::create  
this.dw_1=create dw_1  
iCurrent=UpperBound(this.Control)  
this.Control[iCurrent+1]=this.dw_1  
end on
```

```
on w_edt_expediente.destroy  
call super::destroy  
destroy(this.dw_1)  
end on
```

```
type p_rowfocus from w_edt`p_rowfocus within w_edt_expediente  
end type
```

```
type cb_cancel from w_edt`cb_cancel within w_edt_expediente  
integer x = 4247  
integer y = 176  
integer width = 375  
string text = "&Salir"  
end type
```

```
type cb_ok from w_edt`cb_ok within w_edt_expediente
```



integer x = 4247

integer y = 32

integer width = 375

end type

type dw_1 from u_dw within w_edt_expediente

integer x = 27

integer y = 44

integer width = 4192

integer height = 2244

integer taborder = 40

boolean bringtotop = true

string dataobject = "d_edt_expediente"

end type

event itemchanged;call super::itemchanged;

//Declaración de variables

DataWindowChild ldwc_municipio

String ls_filtro, ls_dato

Int li_error, li_fechaactual, li_fechanacimiento,

li_edad

datetime ldt_fechanac

//Filtro MUNICIPIO

//Verificar si ha cambiado de valor el campo de codigo_departamento

if dwo.name='codigo_departamento' then

//Obtener el datawindows original

if this.getchild('codigo_municipio',ldwc_municipio)= -1 then

MessageBox('Error','Error en funcion getchild(codigo_municipio)')



```
else
    //Reconectar a la base de datos
    ldwc_municipio.SetTransObject(sqlca)
    //Cargar filtro
    ls_filtro      = "codigo_departamento = '"+data+"'"
    li_error      = ldwc_municipio.SetFilter(ls_filtro)
    //Ejecutar el filtro
    li_error      = ldwc_municipio.Filter()
    if li_error = -1 then MessageBox('Error','No se consiguio filtrar los
municipios')
    end if

    // Asignar zona geográfica
    if data ='01' or data ='02' or data ='03' then
        dw_1.setitem(row, "area_geograica",'1')
    end if
    if data ='04' or data ='05' or data ='06' then
        dw_1.setitem(row, "area_geograica",'2')
    end if
    if data ='07' or data ='08' or data ='09' or data='10' then
        dw_1.setitem(row, "area_geograica",'3')
    end if
    if data ='11' or data ='12' or data ='13' or data='14' then
        dw_1.setitem(row, "area_geograica",'4')
    end if

end if

// Cuando el campo cambia el foco
```



```
if dwo.name='codigo_municipio' then
    // Asignando la fecha de nacimiento en una variable
    ldt_fechanac = dw_1.GetItemDatetime(row,"fecha_nacimiento")
    li_fechaactual = year(date(today()))
    li_fechanacimiento = year(date(ldt_fechanac))
    // Calculando la edad del paciente
    li_edad = li_fechaactual - li_fechanacimiento
    // Asignando la edad al datawindows
    dw_1.setitem(row, "edad",li_edad)

end if
end event

event retrieveend;call super::retrieveend;
//Declaración de variables
DataWindowChild ldwc_municipio
String ls_filtro, ls_dato, ls_codigo_depto, ls_codigo_muni
Int li_error

//Obtner datos de departamento y municipio
ls_codigo_depto = this.GetItemString(rowcount,'codigo_departamento')
ls_codigo_muni = this.GetItemString(rowcount,'codigo_municipio')

//Filtrar el municipio segun el departamento
If this.getchild('codigo_municipio',ldwc_municipio)=-1 then
    MessageBox('Error','Error en funcion getchild(codigo_municipio)')
Else
    ldwc_municipio.SetTransObject(sqlca)
```



```
ls_filtro      = "codigo_departamento =" + ls_codigo_depto + ""
li_error       = ldwc_municipio.SetFilter(ls_filtro)
li_error       = ldwc_municipio.Filter()
if li_error = -1 then MessageBox('Error','No se consiguio filtrar los
municipios')
End if

return 1
end event

event ue_validaterow;call super::ue_validaterow;
//Declaración de variables
integer        rc
string         ls_primernombre, ls_primerapellido, lssexo, ls_codigoocupacion,
ls_documentoidentidad
string         ls_ndocumento, ls_estadocivil, ls_lugartrabajo
datetime       ldt_fechanacimiento

//Obtener los campos que se validaran
ls_primernombre = this.GetItemString(al_row, "primer_nombre")
ls_primerapellido = this.GetItemString(al_row, "primer_apellido")
lssexo           = this.GetItemString(al_row, "sexo")
ls_codigoocupacion = this.GetItemString(al_row, "codigo_ocupacion")
ls_documentoidentidad=this.GetItemString(al_row, "documento_identidad")
ls_ndocumento   = this.GetItemString(al_row, "n_documento")
ls_estadocivil   = this.GetItemString(al_row, "estado_civil")
ls_lugartrabajo  = this.GetItemString(al_row, "lugar_trabajo")
ldt_fechanacimiento=this.GetItemDateTime(al_row, "fecha_nacimiento")
```



```
//*****  
*****  
//VALIDACION  
//*****  
*****  
///Validar los campos  
If isnull(ls_primernombre) or trim(ls_primernombre) = "" then  
    rc = this.uf_setmsgerror(al_row,"primer_nombre","Debe digitarse el primer  
nombre")  
End if  
If isnull(ls_primerapellido) or trim(ls_primerapellido) = "" then  
    rc = this.uf_setmsgerror(al_row,"primer_apellido","Debe digitarse el primer  
apellido")  
End if  
If isnull(ls_sexo) or trim(ls_sexo) = "" then  
    rc = this.uf_setmsgerror(al_row,"sexo","Debe digitarse el sexo de la  
persona")  
End if  
If isnull(ls_codigoocupacion) or trim(ls_codigoocupacion) = "" then  
    rc = this.uf_setmsgerror(al_row,"codigo_ocupacion","Debe digitarse la  
ocupacion")  
End if  
If isnull(ls_documentoidentidad) or trim(ls_documentoidentidad) = "" then  
    rc = this.uf_setmsgerror(al_row,"documentoidentidad","Debe digitarse el  
documento de identidad")  
End if  
If isnull(ls_ndocumento) or trim(ls_ndocumento) = "" then
```



```
rc = this.uf_setmsgerror(al_row,"n_documento","Debe digitarse el número  
de documento")
```

```
End if
```

```
If isnull(ls_estadocivil) or trim(ls_estadocivil) = "" then
```

```
rc = this.uf_setmsgerror(al_row,"estado_civil","Debe digitarse el estado  
civil")
```

```
End if
```

```
If isnull(ls_lugartrabajo) or trim(ls_lugartrabajo) = "" then
```

```
rc = this.uf_setmsgerror(al_row,"lugar_trabajo","Debe digitarse el lugar de  
trabajo")
```

```
End if
```

```
If isnull(ldt_fechanacimiento) then
```

```
rc = this.uf_setmsgerror(al_row,"fecha_nacimiento","Debe digitarse la fecha  
de nacimiento")
```

```
End if
```

```
//Retornar valores
```

```
Return 1
```

```
end event
```

```
event ue_insertkeyprimer;call super::ue_insertkeyprimer;
```

```
// Inicializando variable
```

```
string ls_codigopaciente
```

```
// Seleccionando el máximo código del paciente en la tabla expediente
```

```
Select max(codigo_paciente) into :ls_codigopaciente from expediente;
```

```
// en el caso de que no existiera ningún registro
```

```
if isnull(ls_codigopaciente) or trim(ls_codigopaciente) = '' then
```



```
// Se asigna el primer valor
ls_codigopaciente = '0000000001'
else
//Si ya existen registros, Calcular el correlativo
ls_codigopaciente = string(long(ls_codigopaciente) + 1)
//Verificar que el codigo sea de 10 caracteres
if len(ls_codigopaciente) <> 10 then
    ls_codigopaciente = fill('0', 10 - len(ls_codigopaciente)) +
ls_codigopaciente
    end if
end if

// Asignando el código del paciente al datawindows
dw_1.SetItem(al_row, "codigo_paciente", ls_codigopaciente)

return 1
end event
```

VENTANA W_LST_CONSULTA

```
forward
global type w_lst_consulta from w_lst
end type
type uo_1 from u_prmgr_consulta within w_lst_consulta
end type
end forward

global type w_lst_consulta from w_lst
```



```
string title = "Consulta del Paciente"  
uo_1 uo_1  
end type  
global w_lst_consulta w_lst_consulta  
  
on w_lst_consulta.create  
int iCurrent  
call super::create  
this.uo_1=create uo_1  
iCurrent=UpperBound(this.Control)  
this.Control[iCurrent+1]=this.uo_1  
end on  
  
on w_lst_consulta.destroy  
call super::destroy  
if IsValid(MenuID) then destroy(MenuID)  
destroy(this.uo_1)  
end on  
  
type p_rowpoint from w_lst`p_rowpoint within w_lst_consulta  
end type  
  
type p_linkpoint from w_lst`p_linkpoint within w_lst_consulta  
end type  
  
type st_popup from w_lst`st_popup within w_lst_consulta  
end type  
  
type dw_1 from w_lst`dw_1 within w_lst_consulta
```



integer x = 50

integer y = 316

integer width = 3355

integer height = 1308

string dataobject = "d_lst_consulta"

end type

event dw_1::constructor; call super::constructor;

// Llamando la pantalla de edición de consulta

this.is_w_edt_classname = "w_edt_consulta"

this.ib_rpt_allrecords = false

this.ii_edtlaunchmethod = 1 // 1. recupera un sólo reg. 2. los recupera todos

end event

type uo_1 from u_prmgr_consulta within w_lst_consulta

integer x = 37

integer y = 20

integer taborder = 30

boolean bringtotop = true

end type

on uo_1.destroy

call u_prmgr_consulta::destroy

end on

VENTANA W_LST_EXPEDIENTE

forward



```
global type w_lst_expediente from w_lst
end type
type uo_1 from u_prmgr_expediente within w_lst_expediente
end type
type cb_1 from commandbutton within w_lst_expediente
end type
end forward
```

```
global type w_lst_expediente from w_lst
integer x = 1
integer y = 1
integer width = 3498
integer height = 2065
string title = "Expediente del Paciente"
uo_1 uo_1
cb_1 cb_1
end type
global w_lst_expediente w_lst_expediente
```

```
on w_lst_expediente.create
int iCurrent
call super::create
this.uo_1=create uo_1
this.cb_1=create cb_1
iCurrent=UpperBound(this.Control)
this.Control[iCurrent+1]=this.uo_1
this.Control[iCurrent+2]=this.cb_1
end on
```



```
on w_lst_expediente.destroy
call super::destroy
if IsValid(MenuID) then destroy(MenuID)
destroy(this.uo_1)
destroy(this.cb_1)
end on

type p_rowpoint from w_lst`p_rowpoint within w_lst_expediente
end type

type p_linkpoint from w_lst`p_linkpoint within w_lst_expediente
end type

type st_popup from w_lst`st_popup within w_lst_expediente
end type

type dw_1 from w_lst`dw_1 within w_lst_expediente
event filtro ( )
integer x = 50
integer y = 260
integer width = 3365
integer height = 1308
string dataobject = "d_lst_expediente"
end type

event dw_1::filtro();
// Inicializando variables
string ls_filter1
// Filtrando por código del paciente
```



```
ls_filter1="codigo_paciente="+gs_codigopaciente+""
// Actualizando el data windows
dw_1.setFilter(ls_filter1);
dw_1.filter();

end event

event dw_1::constructor;call super::constructor;
// Llamando la pantalla de edición de expediente
this.is_w_edt_classname = "w_edt_expediente"
this.ib_rpt_allrecords = false
this.ii_edtlaunchmethod = 1 // 1. recupera un sólo reg. 2. los recupera todos
end event

type uo_1 from u_prmgr_expediente within w_lst_expediente
integer x = 41
integer y = 32
integer width = 3328
integer taborder = 20
boolean bringtotop = true
end type

on uo_1.destroy
call u_prmgr_expediente::destroy
end on

type cb_1 from commandbutton within w_lst_expediente
integer x = 2779
integer y = 64
```



```
integer width = 402
integer height = 96
integer taborder = 20
boolean bringtotop = true
integer textsize = -10
integer weight = 700
fontcharset fontcharset = ansi!
fontpitch fontpitch = variable!
fontfamily fontfamily = swiss!
string facename = "Arial"
string text = "&Leer"
end type

event clicked;
// Inicializando variables
integer li_FileNum, li_i, li_ind, li_j
string ls_Emp_Input, ls_atributo
long ll_FLength, li_atributo
// Calculando la longitud del archivo de texto donde se guarda el codigo del paciente
ll_FLength = FileLength("C:\codigo.TXT")
// Abriendo el archivo de texto
li_FileNum = FileOpen("C:\codigo.TXT", StreamMode!)
// Asegurandonos que no se pase del máximo permitido
IF ll_FLength < 32767 THEN
    // Leer el contenido del archivo
FileRead(li_FileNum, ls_atributo)
    // Asignando un valor a la variable con el contenido del archivo
li_atributo= long(ls_atributo)
ls_atributo = trim(ls_atributo)
```



```
// Buscando el codigo del paciente
gs_codigopaciente = ''
Select codigo_paciente
into :gs_codigopaciente
from expediente
where convert(dec(10),atributo) =:li_atributo;
// Actualizando el datawindows con el valor del código
dw_1.event filtro()
dw_1.retrieve()
// Cerrando el archivo
FileClose(li_FileNum)
// Abriendo el archivo en modo escritura/ reemplazo
li_FileNum = FileOpen('C:\codigo.TXT',
LineMode!,Write!,Shared!,Replace!)
// Escribiendo un caracter en blanco
FileWrite(li_FileNum, " ")
// Cerrando el archivo
FileClose(li_FileNum)
// Inicializando variables
gs_codigopaciente = ''
ls_atributo=""
li_atributo=0
END IF

end event
```

VENTANA W_LST_FARMACIA

forward



```
global type w_lst_farmacia from w_lst
end type
type uo_1 from u_prmgr_cita within w_lst_farmacia
end type
end forward

global type w_lst_farmacia from w_lst
string title = "Recetas medicadas al paciente"
uo_1 uo_1
end type
global w_lst_farmacia w_lst_farmacia

on w_lst_farmacia.create
int iCurrent
call super::create
this.uo_1=create uo_1
iCurrent=UpperBound(this.Control)
this.Control[iCurrent+1]=this.uo_1
end on

on w_lst_farmacia.destroy
call super::destroy
if IsValid(MenuID) then destroy(MenuID)
destroy(this.uo_1)
end on

type p_rowpoint from w_lst`p_rowpoint within w_lst_farmacia
end type
```



```
type p_linkpoint from w_lst`p_linkpoint within w_lst_farmacia  
end type
```

```
type st_popup from w_lst`st_popup within w_lst_farmacia  
end type
```

```
type dw_1 from w_lst`dw_1 within w_lst_farmacia  
integer x = 50  
integer y = 316  
integer width = 3355  
integer height = 1308  
string dataobject = "d_lst_cita"  
end type
```

```
event dw_1::constructor;call super::constructor;  
// Llamando la pantalla de edición de cita  
this.is_w_edt_classname = "w_edt_cita"  
this.ib_rpt_allrecords = true  
this.ii_edtlaunchmethod = 1 // 1. recupera un sólo reg. 2. los recupera todos  
end event
```

```
event dw_1::doubleclicked;call super::doubleclicked;//gs_codcita =  
this.GetItemString(row,"cita_codigo_cita")  
//open(w_edt_cita)  
end event
```

```
type uo_1 from u_prmgr_cita within w_lst_farmacia  
integer x = 41  
integer y = 20
```



```
integer taborder = 30
boolean bringtotop = true
end type
```

```
on uo_1.destroy
call u_prmgr_cita::destroy
end on
```

LIBRERIA DE REPORTE (RFYD_R.PBL)

VENTANA W_WPP_RECETA

```
forward
global type w_ppv_receta from w_showdata
end type
type dw_1 from u_ppv within w_ppv_receta
end type
type uo_1 from u_prmgr_rpt_recetas within w_ppv_receta
end type
end forward
```

```
global type w_ppv_receta from w_showdata
integer width = 3621
integer height = 2276
string title = "Receta de medicamentos"
string menuname = "m_ppvrun"
dw_1 dw_1
uo_1 uo_1
```



end type

global w_ppv_receta w_ppv_receta

on w_ppv_receta.create

int iCurrent

call super::create

if this.MenuName = "m_ppvrun" then this.MenuID = create m_ppvrun

this.dw_1=create dw_1

this.uo_1=create uo_1

iCurrent=UpperBound(this.Control)

this.Control[iCurrent+1]=this.dw_1

this.Control[iCurrent+2]=this.uo_1

end on

on w_ppv_receta.destroy

call super::destroy

if IsValid(MenuID) then destroy(MenuID)

destroy(this.dw_1)

destroy(this.uo_1)

end on

type p_rowpoint from w_showdata`p_rowpoint within w_ppv_receta

end type

type p_linkpoint from w_showdata`p_linkpoint within w_ppv_receta

end type

type st_popup from w_showdata`st_popup within w_ppv_receta

end type



type dw_1 from u_ppv within w_ppv_receta

integer x = 9

integer y = 328

integer width = 3511

integer height = 1652

integer taborder = 10

boolean bringtotop = true

string dataobject = "d_rpt_receta"

end type

type uo_1 from u_prmgr_rpt_recetas within w_ppv_receta

integer x = 18

integer y = 32

integer taborder = 10

boolean bringtotop = true

end type

on uo_1.destroy

call u_prmgr_rpt_recetas::destroy

end on

LIBRERIA DE FILTROS (RFYD_U.PBL)

OBJETO U_PRMGR_CITA

forward



```
global type u_prmgr_cita from u_prmgrp
end type
type dw_1 from u_prm within u_prmgr_cita
end type
type dw_2 from u_prm within u_prmgr_cita
end type
type dw_4 from u_prm within u_prmgr_cita
end type
type dw_5 from u_prm within u_prmgr_cita
end type
end forward
```

```
global type u_prmgr_cita from u_prmgrp
integer width = 3305
integer height = 284
dw_1 dw_1
dw_2 dw_2
dw_4 dw_4
dw_5 dw_5
end type
global u_prmgr_cita u_prmgr_cita
```

```
on u_prmgr_cita.create
int iCurrent
call super::create
this.dw_1=create dw_1
this.dw_2=create dw_2
this.dw_4=create dw_4
this.dw_5=create dw_5
```



```
iCurrent=UpperBound(this.Control)  
this.Control[iCurrent+1]=this.dw_1  
this.Control[iCurrent+2]=this.dw_2  
this.Control[iCurrent+3]=this.dw_4  
this.Control[iCurrent+4]=this.dw_5  
end on
```

```
on u_prmgr_cita.destroy  
call super::destroy  
destroy(this.dw_1)  
destroy(this.dw_2)  
destroy(this.dw_4)  
destroy(this.dw_5)  
end on
```

```
type gb_header from u_prmgrp`gb_header within u_prmgr_cita  
integer x = 0  
integer width = 3296  
integer height = 268  
string text = "Buscar por"  
end type
```

```
type dw_1 from u_prm within u_prmgr_cita  
integer x = 41  
integer y = 60  
integer width = 649  
integer height = 96  
integer taborder = 20  
boolean bringtotop = true
```



```
string dataobject = "d_prm_cita_codigo"  
end type  
  
event losefocus;call super::losefocus;  
//Declaracion de variables  
String ls_idcod  
  
//ObtenerCodigo del proyecto  
ls_idcod      =      this.GetItemString(this.GetRow(), 'codigo_paciente')  
  
//Rellenar de Ceros  
ls_idcod = fill('0', 10 - len(ls_idcod)) + ls_idcod  
  
//Almacenar el registro  
this.setitem(This.GetRow(), "codigo_paciente", ls_idcod)  
end event  
  
type dw_2 from u_prm within u_prmgr_cita  
integer x = 686  
integer y = 60  
integer width = 1481  
integer height = 96  
integer taborder = 10  
boolean bringtotop = true  
string dataobject = "d_prm_cita_nombrepac"  
end type  
  
event constructor;call super::constructor;//Realizar el filtro por nombre  
this.uf_setoperator("v_expediente.nombre_exp","contains")
```



end event

type dw_4 from u_prm within u_prmgr_cita

integer x = 2149

integer y = 60

integer width = 773

integer height = 96

integer taborder = 20

boolean bringtotop = true

string dataobject = "d_prm_cita_fecha"

end type

event constructor;call super::constructor;//Realizar el filtro por nombre

this.uf_setoperator("v_cita.f_cita","contains")

end event

type dw_5 from u_prm within u_prmgr_cita

integer x = 50

integer y = 160

integer width = 1335

integer height = 96

integer taborder = 30

boolean bringtotop = true

string dataobject = "d_prm_cita_nombreprov"

end type

event constructor;call super::constructor;//Realizar el filtro por nombre

this.uf_setoperator("v_proveedor_nombreprov","contains")

end event



OBJETO U_PRMGR_CONSULTA

forward

global type u_prmgr_consulta from u_prmgrp

end type

type dw_1 from u_prm within u_prmgr_consulta

end type

type dw_2 from u_prm within u_prmgr_consulta

end type

type dw_4 from u_prm within u_prmgr_consulta

end type

type dw_5 from u_prm within u_prmgr_consulta

end type

end forward

global type u_prmgr_consulta from u_prmgrp

integer width = 3305

integer height = 284

dw_1 dw_1

dw_2 dw_2

dw_4 dw_4

dw_5 dw_5

end type

global u_prmgr_consulta u_prmgr_consulta

on u_prmgr_consulta.create

int iCurrent



```
call super::create  
this.dw_1=create dw_1  
this.dw_2=create dw_2  
this.dw_4=create dw_4  
this.dw_5=create dw_5  
iCurrent=UpperBound(this.Control)  
this.Control[iCurrent+1]=this.dw_1  
this.Control[iCurrent+2]=this.dw_2  
this.Control[iCurrent+3]=this.dw_4  
this.Control[iCurrent+4]=this.dw_5  
end on  
on u_prmgr_consulta.destroy  
call super::destroy  
destroy(this.dw_1)  
destroy(this.dw_2)  
destroy(this.dw_4)  
destroy(this.dw_5)  
end on  
  
type gb_header from u_prmgrp`gb_header within u_prmgr_consulta  
integer x = 0  
integer width = 3296  
integer height = 268  
string text = "Buscar por"  
end type  
  
type dw_1 from u_prm within u_prmgr_consulta  
integer x = 41  
integer y = 60
```



```
integer width = 649  
integer height = 96  
integer taborder = 20  
boolean bringtotop = true  
string dataobject = "d_prm_consulta_codigo"  
end type  
  
event losefocus;call super::losefocus;  
//Declaracion de variables  
String ls_idcod  
  
//ObtenerCodigo del proyecto  
ls_idcod = this.GetItemString(this.GetRow(), 'codigo_paciente')  
  
//Rellenar de Ceros  
ls_idcod = fill('0', 10 - len(ls_idcod)) + ls_idcod  
  
//Almacenar el registro  
this.setitem(This.GetRow(), "codigo_paciente", ls_idcod)  
end event  
  
type dw_2 from u_prm within u_prmgr_consulta  
integer x = 658  
integer y = 52  
integer width = 1769  
integer height = 96  
integer taborder = 10  
boolean bringtotop = true  
string dataobject = "d_prm_expediente_nombre"
```



end type

```
event constructor;call super::constructor;//Realizar el filtro por nombre  
this.uf_setoperator("v_expediente.nombreexp","contains")  
end event
```

```
type dw_4 from u_prm within u_prmgr_consulta  
integer x = 2400  
integer y = 52  
integer width = 773  
integer height = 96  
integer taborder = 20  
boolean bringtotop = true  
string dataobject = "d_prm_consulta_fecha"  
end type
```

```
event constructor;call super::constructor;//Realizar el filtro por nombre  
this.uf_setoperator("v_consulta.f_consulta","contains")  
end event
```

```
type dw_5 from u_prm within u_prmgr_consulta  
integer x = 50  
integer y = 160  
integer width = 1874  
integer height = 96  
integer taborder = 30  
boolean bringtotop = true  
string dataobject = "d_prm_consulta_nombreprov"  
end type
```



```
event constructor;call super::constructor;//Realizar el filtro por nombre
this.uf_setoperator("V_proveedor_nombreprov","contains")
end event
```

OBJETO U_PRMGR_EXPEDIENTE

```
forward
global type u_prmgr_expediente from u_prmgrp
end type
type dw_1 from u_prm within u_prmgr_expediente
end type
type dw_2 from u_prm within u_prmgr_expediente
end type
end forward
```

```
global type u_prmgr_expediente from u_prmgrp
integer width = 2683
integer height = 208
dw_1 dw_1
dw_2 dw_2
end type
global u_prmgr_expediente u_prmgr_expediente
```

```
on u_prmgr_expediente.create
int iCurrent
call super::create
this.dw_1=create dw_1
```



```
this.dw_2=create dw_2  
iCurrent=UpperBound(this.Control)  
this.Control[iCurrent+1]=this.dw_1  
this.Control[iCurrent+2]=this.dw_2  
end on
```

```
on u_prmgr_expediente.destroy  
call super::destroy  
destroy(this.dw_1)  
destroy(this.dw_2)  
end on
```

```
type gb_header from u_prmgrp`gb_header within u_prmgr_expediente  
integer x = 0  
integer width = 2647  
integer height = 192  
string text = "Buscar por"  
end type
```

```
type dw_1 from u_prm within u_prmgr_expediente  
integer x = 41  
integer y = 60  
integer width = 649  
integer height = 96  
integer taborder = 20  
boolean bringtotop = true  
string dataobject = "d_prm_expediente_codigo"  
end type
```



```
event losefocus;call super::losefocus;
```

```
//Declaracion de variables
```

```
String ls_idcod
```

```
//Obtener Codigo del proyecto
```

```
ls_idcod      =      this.GetItemString(this.GetRow(), 'codigo_paciente')
```

```
//Rellenar de Ceros
```

```
ls_idcod = fill('0', 10 - len(ls_idcod)) + ls_idcod
```

```
//Almacenar el registro
```

```
this.setitem(This.GetRow(), "codigo_paciente", ls_idcod)
```

```
end event
```

```
type dw_2 from u_prm within u_prmgr_expediente
```

```
integer x = 686
```

```
integer y = 60
```

```
integer width = 1833
```

```
integer height = 96
```

```
integer taborder = 10
```

```
boolean bringtotop = true
```

```
string dataobject = "d_prm_expediente_nombre"
```

```
end type
```

```
event constructor;call super::constructor;//Realizar el filtro por nombre
```

```
this.uf_setoperator("v_expediente.nombreexp","contains")
```

```
end event
```



OBJETO U_PRMGR_RECETAS

forward

global type u_prmgr_rpt_recetas from u_prmgrp

end type

type dw_1 from u_prm within u_prmgr_rpt_recetas

end type

type dw_2 from u_prm within u_prmgr_rpt_recetas

end type

type dw_5 from u_prm within u_prmgr_rpt_recetas

end type

end forward

global type u_prmgr_rpt_recetas from u_prmgrp

integer width = 2725

integer height = 284

dw_1 dw_1

dw_2 dw_2

dw_5 dw_5

end type

global u_prmgr_rpt_recetas u_prmgr_rpt_recetas

on u_prmgr_rpt_recetas.create

int iCurrent

call super::create

this.dw_1=create dw_1

this.dw_2=create dw_2

this.dw_5=create dw_5

iCurrent=UpperBound(this.Control)



```
this.Control[iCurrent+1]=this.dw_1  
this.Control[iCurrent+2]=this.dw_2  
this.Control[iCurrent+3]=this.dw_5  
end on
```

```
on u_prmgr_rpt_recetas.destroy  
call super::destroy  
destroy(this.dw_1)  
destroy(this.dw_2)  
destroy(this.dw_5)  
end on
```

```
type gb_header from u_prmgrp`gb_header within u_prmgr_rpt_recetas  
integer x = 0  
integer width = 2674  
integer height = 268  
string text = "Buscar por"  
end type
```

```
type dw_1 from u_prm within u_prmgr_rpt_recetas  
integer x = 41  
integer y = 60  
integer width = 713  
integer height = 96  
integer taborder = 20  
boolean bringtotop = true  
string dataobject = "d_prm_consulta_codigo_rec"  
end type
```



```
event losefocus;call super::losefocus;
//Declaracion de variables
String ls_idcod

//Obtener Codigo del proyecto
ls_idcod      =      this.GetItemString(this.GetRow(), 'codigo_paciente')

//Rellenar de Ceros
ls_idcod = fill('0', 10 - len(ls_idcod)) + ls_idcod

//Almacenar el registro
this.setitem(This.GetRow(), "codigo_paciente", ls_idcod)
end event

type dw_2 from u_prm within u_prmgr_rpt_recetas
integer x = 722
integer y = 48
integer width = 1920
integer height = 96
integer taborder = 10
boolean bringtotop = true
string dataobject = "d_prm_expediente_nombre_rec"
end type

event constructor;call super::constructor;//Realizar el filtro por nombre
this.uf_setoperator("v_paciente_med.nombreexp","contains")
end event

type dw_5 from u_prm within u_prmgr_rpt_recetas
```



```
integer x = 50  
integer y = 160  
integer width = 2395  
integer height = 96  
integer taborder = 30  
boolean bringtop = true  
string dataobject = "d_prm_consulta_nombreprov_rec"  
end type  
  
event constructor;call super::constructor;//Realizar el filtro por nombre  
this.uf_setoperator("V_paciente_med.nombreprov","contains")  
end event
```



10 REFERENCIAS

- [1] <http://www.monografias.com/trabajos42/codigo-de-barras/codigo-de-barras.shtml>
- [2] <http://www.monografias.com/trabajos42/codigo-de-barras/codigo-de-barras3.shtml>
- [3] http://www.ucsg.edu.ec/catolica/secundarias_ucsg/html/publicaciones/ingenieria/ultima%20edicion/ingenieria.htm
- [4] http://es.wikipedia.org/wiki/Juan_Vucetich
- [5] L. Hong and A. Jain, "Integrating Faces and Fingerprints for Personal Identification", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 12, pp. 1295-1307, 1998.
- [6] <http://www.homini.com/index.htm>
- [7] <http://www2.ing.puc.cl/~iing/ed429/>
- [8] <http://porsche.ls.fi.upm.es/>
- [9] http://es.wikipedia.org/wiki/Banda_magn%C3%A9tica
- [10] <http://www.iso.org>
- [11] <http://www.cyberd.co.uk/support/technotes/isocards.htm>
- [12] <http://www.it.uc3m.es/jmb/RFID/rfid.pdf>
- [13] <http://www.channelplanet.com/index.php?idcategoria=17371>



- [14] www.etsit.uma.es/descargas/2006/ActasAFCAbril2006.pdf
- [15] Capacidad de memoria de tarjetas inteligentes.
http://www.c3po.es/tarjetas_chip.html
- [16] Capacidad de memoria de tarjetas de banda magnética.
<http://www.anu.edu.au/people/Roger.Clarke/DV/SCTISK1.html>
- [17] Capacidad de memoria de tarjetas RFID.
http://www.atmel.com/dyn/resources/Prod_documents/secref_largemem_3_04.pdf
- [18] Datos obtenidos de los indicadores demográficos de la Dirección General de Estadística y Censos (DIGESTYC). <http://www.digestyc.gob.sv/>
- [19] Comisión Económica para América Latina y el Caribe (CEPAL).
<http://www.eclac.cl/deype/>
- [20] Ranking HDI (Human Development Index).
<http://hdr.undp.org/hdr2006/statistics/>
- [21] Cuentas Nacionales en Salud, 2006, Dirección de Planificación, MSPAS
- [22] <http://www.mh.gob.sv>
- [23] <http://fuma.org.sv/documentos/SW2006.pdf>
- [24] http://www.mspas.gob.sv/depto_elsalvador2007.asp
- [25] http://www.mspas.gob.sv/pobl_estab2005/mapa_san_salvador2005.pdf



AGRADECIMIENTOS

Son tantas personas a las cuales debo parte de este triunfo, de lograr alcanzar mi culminación académica, la cual es el anhelo de todos los que así lo deseamos. A todas ellas quiero expresar mi más sincero agradecimiento.

Definitivamente, a ti Dios Todopoderoso, mi Proveedor, mi Guía que me diste la oportunidad de vivir, por darme fe, fortaleza, bienestar y esperanza en el transcurso de mi carrera; y que hoy que termino y he alcanzado mi propósito, sé que estuviste en cada momento iluminando mi mente y corazón, eternamente agradecido Dios Padre.

Con mucho cariño a mis padres Edgar y Marina, gracias papá y mamá por el apoyo que me brindaron, por fomentar en mi el deseo de saber y superarme, y que a pesar de todas las dificultades ha sido todo su amor y el anhelo de verme desarrollado, lo que les dio fuerza para impulsarme y salir adelante. Este triunfo no es solo mío, es de ustedes, ¡los quiero!

A mis tíos Rafael y Estelita, por todo su afecto y ayuda incondicional, por ser quienes se convirtieron en mis segundos padres y darme un techo en donde vivir, comer y dormir, de no ser por ustedes no hubiese alcanzado mi sueño. Por estar ahí cuando mas los necesitaba, ¡muchas gracias!.

A mama Gloria por ser la abuelita con quien todo nieto sueña, por su inmenso cariño y amor que nunca permitió que nada nos faltase.

A mis hermanos Gustavo, José y Rafita por su compañía, aliento y comprensión, por ser mi familia, gracias, los quiero mucho.



A mi querida Cecy, mi gemelita, por enseñarme que sin la ayuda de Dios nada es posible, que cada meta que se logra es porque Dios así lo permite, gracias por tu amor y tus palabras.

A mi equipo de tesis; mis compañeros y amigos Rodrigo y Alex, gracias a mi trío por ser el último escalón para poder alcanzar este triunfo, este mi sueño, que ahora es una realidad. Y gracias también a tu familia Rodrigo, por apoyarnos siempre.

Al Ing. Juan Carlos Castro por su asesoría y orientación durante todo nuestro trabajo.

Y con especial dedicatoria a vos Jorge (QEPD), mi inolvidable amigo, que desde allá en el cielo sé has de estar feliz con mi logro, no llegue solo hasta el final, también llegaste vos, por lo que alguna vez luchaste conmigo hoy lo has logrado conmigo, ¡Va para ti mi amigo!.

Y a todos aquellos, que han quedado en los recintos más escondidos de mi memoria, pero que fueron participes en cincelar mi persona. ¡GRACIAS!.

Edgard Adonay Aguirre.



AGRADECIMIENTOS

Primero y antes que nada, dar gracias a Dios, por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio, además de brindarme, la fortaleza, la salud y la esperanza para terminar este proyecto.

Agradecer a mis padres Dina y Rodrigo por su cariño, comprensión y apoyo sin condición ni medida. Gracias porque desde pequeño me guiaron sobre el camino de la educación y me enseñaron a luchar por alcanzar mis metas, porque si no fuese por el esfuerzo realizado por ellos, la culminación de este trabajo no hubiese sido posible.

Gracias a mi hermanos Nelson y Marielos por su apoyo y compañía incondicional a lo largo de mi carrera.

A Mariana por haber estado siempre a mi lado, por brindarme consuelo cada vez que me sentía desanimado, gracias por el apoyo en los buenos y malos momentos.

A mis compañeros de tesis Alexander y Edgard, por cada una de sus críticas y consejos, por el tiempo y el esfuerzo que cada uno invirtió para culminar esta tarea que un día nos propusimos.

Al Ing. Juan Carlos Castro por su asesoría y dirección durante todo el trabajo.

Gracias a todas las personas que de una u otra forma se vieron involucradas en este proyecto y las que participaron en mi desarrollo profesional durante mi carera, pues sin su colaboración esto no hubiese sido posible.

Rodrigo Arnaldo Najarro.



AGRADECIMIENTOS

Muchas han sido las personas que de manera directa o indirecta me han ayudado en la realización de esta tesis. Quiero dejar constancia de todas ellas y agradecerles con sinceridad su participación.

Es de gran importancia para mí mencionar en primer lugar la inmensa gratitud que debo a toda mi familia en general pero especialmente a mis padres Salvador Castillo Zelaya y Magdalena del Carmen Lucero Figueroa por apoyarme no solamente en lo económico sino también en todo lo que me hizo falta para que yo me sintiera tranquilo y con ánimos para seguir adelante. Muchas gracias padre y madre por inyectarme su fuerza y su valor para conseguir este objetivo. Muchas gracias por darme todo su cariño, su optimismo y su comprensión. El apoyo, ha sido fundamental para que yo pudiera llegar hasta aquí. Millones de gracias amados padres por creer en mí.

A mi hermana Karla Lizeth Castillo Lucero, por toda su ayuda, su ejemplo de lucha, honestidad, superación, paciencia, inteligencia y generosidad.

A mis compañeros y amigos de tesis Edgard Adonay Aguirre López y Rodrigo Arnoldo Najarro Alvarez por su paciencia y su constante apoyo durante el desarrollo de este trabajo de graduación, por compartir las angustias y gratificaciones a lo largo de la toda la carrera, gracias por su amistad y ayuda brindadas en todo momento.

Al Ingeniero Juan Carlos Castro por su asesoría y dirección en nuestro trabajo de graduación.

A todos mis amigos quienes siempre creyeron en mis capacidades para alcanzar mis metas, gracias por brindarme su confianza y colaboración en cada momento de mi vida.



Y a todas aquellas personas que de una u otra forma, colaboraron o participaron en la realización de este proyecto de tesis, hago extensivo mi más sincero agradecimiento.

Alexander Salvador Castillo.
