



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**SISTEMA DE CÓMPUTO PARA EL
ALMACENAMIENTO Y ADQUISICIÓN DE
ESTUDIOS
MÉDICOS ONCOLÓGICOS**

TESIS PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A N :

CASTILLO CABRERA JUANA

MARTINEZ VÁZQUEZ ARTURO

Director de Tesis:

M. en I. Alberto Fuentes Maya



México, D.F.

Enero del 2008



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Dedicatoria:

A **Dios** y a mi **familia** por creer en mí.

A mis **sobrinos** porque los amo (Jesús Salvador, Luis Antonio, Jazmín, Osvaldo, Diego, Diana, Carla, Axel, Raúl y Diego).

Porque quiero que sepan que en la vida se lucha con el corazón, de verdad dios siempre lo recompensa,

En especial:

Luis Antonio, nunca dejes que las adversidades de la vida te hagan fracasar, tu lucha y veras que siempre vas alcanzar lo que quieras.

Diana, more nada es fácil en la vida, hay que echarle todas las ganas a todo lo que quieras hacer y tendrás todo lo que quieras tener.

Diego, eres mi gallo, no necesitas demostrarle a nadie lo que eres, ellos solitos se darán cuenta que eres chiquito de tamaño pero grande de inteligencia.

Carla, flaca aún eres pequeña pero tienes un talento enorme y será un orgullo para todos.

Axel a ti simplemente porque te amo y daría mi vida por ti.

Con mi Admiración y Cariño para Ustedes.

Agradecimientos:

A DIOS...

Gracias por escucharme siempre y por levantarme cuando me he caído.

A MI MADRE...

Aunque sabes que no he sido la mejor hija siempre me has apoyado y has estado conmigo en todo momento y reconozco que soy quien soy por ti. Gracias por tu cariño y tu sabiduría.

A MI PADRE...

Que aunque se que ya no está conmigo desde el cielo día con día me da su bendición.

A MIS HERMANAS...*(Carolina y Margarita)*

Por creer en mi y porque siempre las he visto como un ejemplo en mi vida personal.

A MIS HERMANOS Y CUÑADOS: *(Salvador⁽⁺⁾ José Luis, Rafael, Sergio, Oscar, Rodrigo, Carlos y Polo)*

Porque se que en el fondo muy en el fondo de su corazón reconocen mi esfuerzo.

A MIS PRIMOS *(Roberto e Ignacio Martínez Castillo, Marmolejo Cabrera y Ibarra Castillo)*

Por su apoyo incondicional, por su admiración, por su confianza y desde luego por su cariño.

A LA FACULTAD DE INGENIERIA DE LA UNAM...

Porque es una orgullo pertenecer a la máxima casa de estudios (UNAM) y porque me dió la experiencia y el placer más grande de mi vida. "EL SABER"

A MIS PROFESORES...

Porque con su enseñanza y conocimientos me han hecho crecer en todos aspectos.

A MI ASESOR... *(M. en I. Alberto Fuentes Maya)*

Como no agradecerle, si es una de las personalidades que mas admiro y respeto, gracias por su tiempo, su esfuerzo, sus conocimientos, sus consejos, sus regaños, su cariño, gracias por ser tan grande como es.

A MIS AMIGOS...

Que se bien quienes son, les agradezco su amistad y sus consejos.

Gracias... a todos aquellos que disfrutan mis éxitos, pero también mis fracasos...

Juana Castillo Cabrera

Agradecimientos:

A mi madre por apoyarme incondicionalmente en alcanzar mi más anhelada meta, por soportar todos los malos ratos que le hice pasar y por haber aguantado los malos tiempos que envolvieron toda mi carrera. A mi padre Arturo y a mi papá Guillermo que están en el cielo, que desde halla me están viendo, espero se sientan muy orgullosos de mi. A mi hermano Guadalupe que supo darme un ejemplo digno de seguir. A mi hermana Josefina que me apoyo durante toda mi carrera incondicionalmente. A mis hermanas Esther, María y Yolanda que estuvieron conmigo, me escucharon y alentaron a seguir adelante. A mi maestro Raymundo Hernández que me encamino por un camino que alguna vez creí perdido pero que hoy es una realidad. A mi amigo Alejandro que siempre estuvo conmigo y me acompañó toda la carrera. A mi novia Dulce por su cariño y que supo esperarme y estuvo conmigo en las buenas y las malas. A los profesores de la facultad de ingeniería de la UNAM por sus enseñanzas.

Gracias...

Arturo Martínez Vázquez

INDICE

	Pag.
Objetivo del Proyecto.....	3
Introducción	4
Capítulo 1	
1. Análisis de Requerimientos	7
1.1 Estudios Citológicos	7
1.2 Estudios Histopatológicos.....	7
1.3 Autopsia.....	8
Capítulo 2	
2. Diseño	13
2.1 Modelado de la Base de Datos.....	13
2.1.1 Modelo dinámico.....	14
2.1.2 Modelo estático	21
Capítulo 3	
3. Desarrollo.....	25
3.1 Características del Ambiente y Software de Aplicación.....	26
3.1.1 Elementos de Software.....	27
3.1.2 Descripción funcional del Sistema.....	28
3.2 Adquisición de la imagen del Microscopio.....	37
Capítulo 4	
4. Implementación	38
4.1 Características del Hardware.....	38
4.1.1 Computadora.....	38
4.1.2 Frame Grabber.....	39
4.1.3 Microscopio.....	41
4.1.4 Cámara CCD.....	41
4.2 Diseño del Sistema.....	43
Conclusiones.....	47
Anexo A	48
Bibliografía.....	86

Objetivo del Proyecto

- Implementar un sistema de Cómputo para el almacenamiento y adquisición de estudios médicos oncológicos, basados en las imágenes adquiridas de un sistema microscopio-cámara-computadora, para apoyar a los citólogos y patólogos en la realización y almacenamiento de su diagnóstico.

Introducción

En México, las enfermedades oncológicas son la segunda causa de muerte de la población, representando gastos millonarios para los institutos de salud que las atienden. Tan sólo en 1999, el Hospital de Oncología del Centro Médico Nacional Siglo XXI del Instituto Mexicano del Seguro Social (IMSS), gastó 451 millones de pesos en atención al derechohabiente con cáncer, es decir 43 millones de pesos más del presupuesto anual destinado para su combate en el mismo año. Esta situación se torna crítica y de no implementarse estrategias de prevención, para el año 2025 se presentarán 15.5 millones de nuevos casos, advierte la Organización Mundial de la Salud (OMS).

El cáncer cérvico-uterino es un tipo frecuente de cáncer en mujeres, y consiste en una enfermedad en la cual se encuentran células cancerosas (malignas) en los tejidos del cuello uterino. Existen factores de riesgo bien caracterizados que se asocian a la enfermedad, éstos incluyen múltiples parejas sexuales, menor edad al inicio de las Relaciones Sexuales, exposición al Virus del Papiloma Humano (VPH), Tabaquismo, Factores Biológicos y Factores relacionados a la atención proporcionada para la detección y tratamiento oportuno de la enfermedad.

Está documentado que un diagnóstico oportuno es la mejor herramienta para su control y erradicación, por ende, este proyecto presenta una propuesta de apoyo en el diagnóstico de muestras tomadas de exámenes de Papanicolau para la oportuna detección del cáncer cérvico-uterino.

La Organización Mundial de la Salud recomienda que los Programas del Control del Cáncer Cérvico-Uterino, deben enfatizar en ampliar las coberturas de detección en las mujeres con edades entre 30 y 50 años, con controles rutinarios cada 2 o 3 años. Sin embargo, para países con tasas de incidencia y mortalidad altas como el nuestro, recomienda que el grupo de edad se extienda y la detección se inicie a partir de los 25 años, con una cobertura mínima del

70% y una óptima del 100% de las mujeres. Por tal motivo son necesarios programas de educación para pacientes y médicos encaminados a alentar la práctica de la prueba de Papanicolaou.

La comunidad científica internacional y nacional en el área médica han encontrado que la causa que origina el desarrollo de cáncer cérvico-uterino es la presencia del virus del papiloma humano, hasta el momento las evidencias científicas han demostrado que el Papanicolau (citología de base líquida) con su lectura automatizada, análisis de biología molecular, mas una visión colposcópica ha resultado la opción mas económica, oportuna y precisa, reduciendo el índice de incertidumbre entre la población femenina para un diagnóstico oportuno de este cáncer. Por lo que la estrategia médica para detección oportuna del cáncer cérvico-uterino más efectiva parece ser una combinación de citología de base líquida (Papanicolau no convencional), más PCR¹ en forma computarizada.

La comparación por medios electrónicos de buenas imágenes contra un banco de lesiones bien caracterizadas se transforma en un trabajo más rápido y preciso, permitiendo distinguir de un modo más fino entre tejido sano y enfermo. Los inconvenientes son; el elevado costo del “software” y “hardware”, y el hecho de que los módulos del equipo vengan encapsulados, haciendo imposible para un extraño, conocer los componentes electrónicos internos, lo que trae como consecuencia la dependencia de la tecnología.

En el mercado internacional existen equipos electrónicos de imagenología para detección de células cancerígenas.

¹ **PCR** (Proteína C Reactiva).- La reacción en cadena de polimerasa es un ensayo selectivo de amplificación de blanco, capaz de incrementar en forma exponencial las secuencias de DNA viral presentes en un espécimen biológico.

El presente trabajo describe la implementación de un sistema de información computarizado que a través de la adquisición de imágenes de un microscopio, utilizando técnicas de visión, se utilicen para la prevención del cáncer cérvico-uterino y, que con ellas el médico pueda saber la etapa de la enfermedad y elegir o planear el tratamiento adecuado; seleccionando una estrategia de tratamiento que sea lo más barata, segura, rápida y eficaz posible. Así como dotar a los especialistas en el área citológica y patológica con una herramienta de apoyo (despliegue de datos procesados, ampliación (zoom) digital, segmentación automática de áreas celulares e interface-microscopio-computadora) para obtener imagen en tiempo real y facilitar la observación y el diagnóstico. Esto parte de la importancia de reducir tiempos en la detección y diagnósticos en estudios oncológicos.

Contar con una base de datos de células (cérvico-uterino caracterizada y clasificadas para capacitación del personal de citología y patología), es importante ya que a nivel mundial sólo los japoneses cuentan con una base caracterizada y el costo es elevado para su adquisición.

La estructura general del presente trabajo de tesis consiste en cuatro capítulos, y una parte de conclusión. El capítulo 1 presenta la etapa del análisis de requerimientos, la cual es fundamental, siendo el proceso básico de desarrollo de software de este trabajo; en el capítulo 2 se propone el diseño e implementación de la estructura principal de la base de datos del software; el capítulo 3 se presenta la propuesta de implementación de la programación del sistema, así como la adquisición de imágenes para la realización del diagnóstico, y en el capítulo 4 se desarrolla la implementación del sistema, microscopio-cámara-computadora y finalmente, se presentan las conclusiones del proyecto y las futuras aplicaciones.

1. ANÁLISIS DE REQUERIMIENTOS

Con el fin de obtener los requerimientos del sistema fue necesario realizar un análisis del esquema de trabajo de un departamento de patología. Para la realización de esta función se llevaron a cabo entrevistas con los patólogos y citólogos del Hospital “Las Américas”, Ecatepec, Edo. de México, así como de algunas clínicas particulares. De dichas entrevistas resultó que los departamentos de patología se enfocan principalmente a los siguientes estudios:

1. Estudios Citológicos
2. Estudios Histopatológicos
 - a. Histológicos
 - b. Trans-operatorios
 - c. Especiales
3. Autopsias

1.1 Estudios Citológicos

La mayoría de estos estudios provienen de exámenes de Papanicolau, el cual es el análisis de una muestra de secreción del cuello de la matriz, un pequeño porcentaje restante se refiere a estudios de fluidos obtenidos del cuerpo humano. Los estudios los puede realizar un citotecnólogo o un patólogo; en el caso de que sea un citotecnólogo quien realice el estudio, el patólogo, a manera de control de calidad supervisa al azar algunos de los estudios negativos e invariablemente hará una revisión exhaustiva de todos los estudios positivos.

1.2 Estudios Histopatológicos

Son los estudios que se hacen a las piezas quirúrgicas del paciente, como una biopsia que es extraer una muestra de tejido del cuello uterino y se observa a través del microscopio para ver si hay células cancerosas. Para efectuar una biopsia sólo se necesita una pequeña cantidad de tejido; dicho estudio contiene más de una laminilla y es realizado exclusivamente por el

patólogo. Cuenta con tres tipos de estudios: Histológico, Trans-operatorio y Especial.

1.3 Autopsia

Los estudios de autopsia se llevan a cabo en los hospitales a petición de los deudos del occiso; raramente se presentan estos casos.

Los estudios mencionados anteriormente se basan en el empleo de técnicas clásicas en histopatología con la realización de cortes histológicos seriados de los tejidos a analizar, se realiza un tinte diferencial y se monta para el examen con un microscopio donde el Doctor realizará su análisis de cada una de las muestras; esta tarea es engorrosa y tediosa ya que:

- 1.-Las muestras se tienen que marcar para cada paciente
- 2.-Tiene que explorar toda la muestra y es difícil el marcado de zonas de importancia.

Los establecimientos médicos tardan de 30 a 45 días para obtener resultados de la citología cérvico-uterina tomadas a derecho-habientes. En estos casos el tiempo de su realización es apremiante, debido al rápido avance de este tipo de enfermedad, ya que cuando se dictamina una presunta muestra de neoplasia positiva a las pacientes se les hacen pruebas más precisas hasta obtener la certeza de padecimiento del cáncer para la instauración del procedimiento médico drástico de eliminación o de control del padecimiento.

Un método que puede ayudar al doctor a realizar esta tarea de una forma más rápida y eficaz es utilizar un sistema que ayude en el almacenamiento de su diagnóstico y clasificación de las imágenes para obtener resultados eficientes y rápidos para el tratamiento de las enfermedades.

Los sistemas de visión por computadora que pretenden reemplazar tareas llevadas a cabo por el ser humano requieren de un alto grado de exactitud difícilmente alcanzable. La investigación científica enfocada al reconocimiento automático de células con displasia, indicativo de una patología celular típica del cáncer cérvico-uterino, se ha conducido desde los años cincuenta sin que

hasta la fecha exista un método certero que reemplace a un médico especialista (patólogo). Esto es indicativo de cuan complejo y variante puede ser un problema de investigación de esta índole. Liu, Zhao y Zhang describe el problema como un problema de tipo “una aguja en un pajar” y acertadamente apuntan a que el problema es complicado, tedioso y tendiente a errar en el diagnóstico.

Es por esto, que este sistema, más que enfocarse a reemplazar al médico especialista, se ha empeñado en automatizar las tareas cotidianas que lleva un departamento de patología en un hospital. El sistema ha puesto especial énfasis en liberar la mayor carga de trabajo al patólogo y dejarle al máximo tiempo posible para trabajar exclusivamente en el diagnóstico. Como valor agregado y resultado natural del trabajo que hace un departamento de patología, se han incluido los medios para archivar y controlar los estudios histopatológicos intrínsecos a la labor de citado departamento. La figura 1.1 muestra un diagrama de bloques del sistema propuesto para ayudar al médico en su rutina diaria de trabajo.

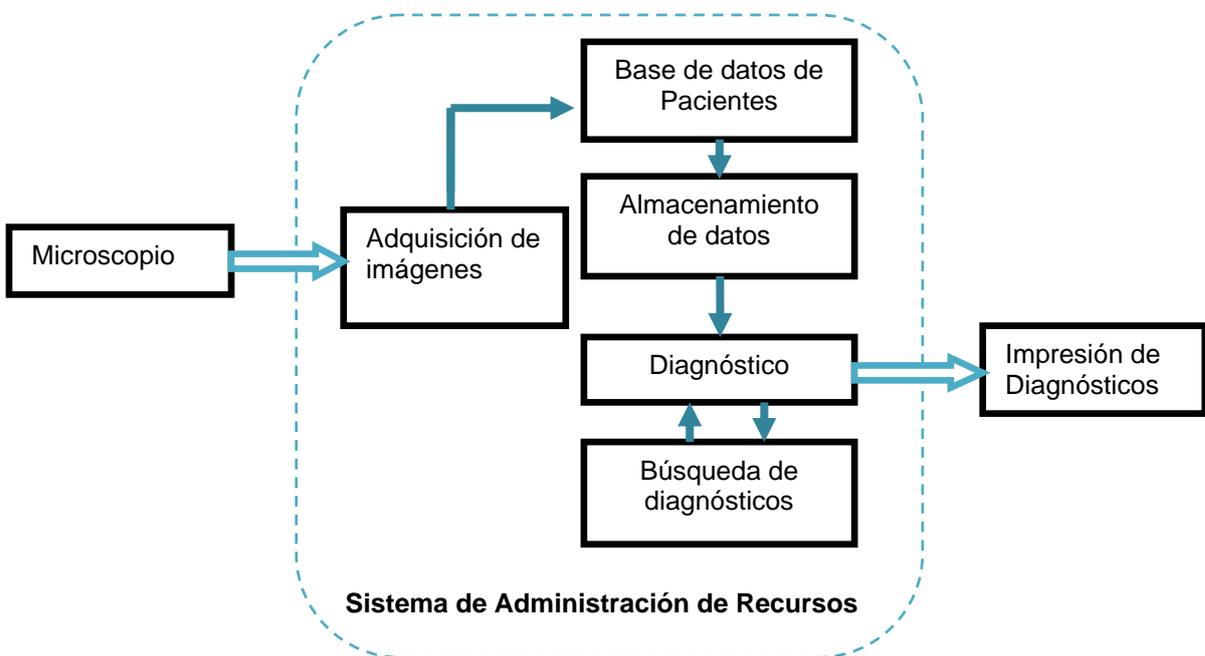


FIGURA 1.1 SISTEMA DE CÓMPUTO PARA EL ALMACENAMIENTO Y ADQUISICIÓN DE ESTUDIOS MÉDICOS ONCOLÓGICOS

La herramienta básica del médico es el microscopio, para obtener la imagen directamente y colocarla en la computadora se utiliza una tarjeta

digitalizadora (Frame Grabber) conectada a una cámara CCD² que está adaptada al microscopio por un dispositivo de acoplamiento, la *figura 1.2* muestra este proceso.

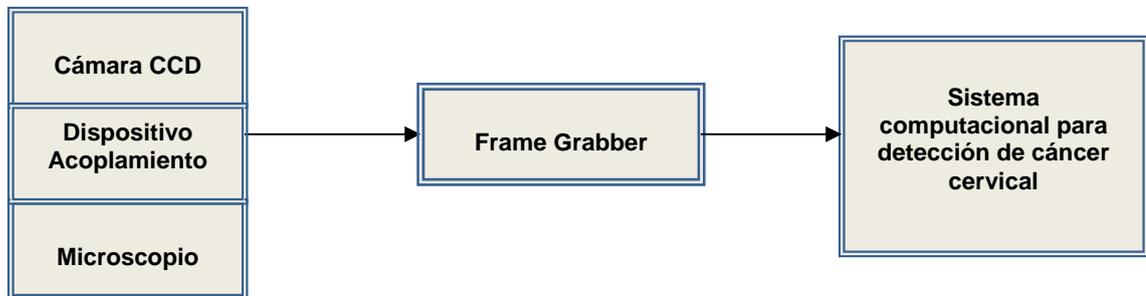


FIGURA 1.2 Proceso de Adquisición de imágenes.

Para la adquisición de las imágenes se necesitan cámaras que se usan en el campo del procesamiento digital de imágenes las cuales constan primordialmente de:

- a) Unidad de Adquisición de Imagen
- b) Unidad de Salida de Imagen

La Unidad de adquisición queda definida por el tipo de sensor que registra la señal luminosa que conforma la imagen.

La Unidad de salida es la encargada de interpretar los valores del sensor y convertirlos en una señal eléctrica de video conforme a un protocolo el cual puede ser estándar o no, dicho protocolo define entre otros parámetros el número de líneas horizontales y verticales por cuadro así como el número de cuadros por segundo.

² El chip **CCD** (*Charge Coupled Device* o dispositivo de carga acoplada).

Las **cámaras CCD** basan su funcionamiento en un chip que consiste de un arreglo bidimensional de sensores de señales luminosas. La dimensión de

dicho arreglo define la resolución de la imagen que se obtenga a la salida de la cámara.

Otro de los parámetros importantes que deben de ser considerados al trabajar con el chip CCD es su resolución expresada en el número de píxeles. Una resolución típica en este chip es por ejemplo 756 X 581 (columnas por líneas). La señal generada por la cámara de video, es un flujo de información analógica compuesta por “píxeles” que forman líneas, que forman marcos, y estos a su vez forman secuencias completas de vídeo.

Para manipular computacionalmente la señal generada por la cámara, se debe tener acceso a cada uno de los píxeles de cada uno de los marcos. Para ello se utilizan dispositivos como el **Frame Grabber** capaces de muestrear el flujo de entrada de video, y convertirlo en información binaria cuyo almacenamiento se hace en registros de memoria cuya dirección es plenamente conocida y por tanto son directamente accesibles por los algoritmos de procesamiento digital. El Frame Grabber puede pensarse como un convertidor analógico digital altamente especializado.

Para los fines del proyecto, que procesará imágenes en tiempo real, se requiere un Frame Grabber con las siguientes características:

1. Gran velocidad de adquisición de píxeles del orden de 50 MHz o superior
2. Entrada compatible con el formato de salida de la cámara
3. Bus de conexión de amplio ancho de banda como el PCI para alta transferencia de datos (del orden de decenas de Megabytes por segundo)
4. Opciones de señales de entrada/salida para control y sincronización con la cámara “residente”
5. Opciones de pre-procesamiento de la imagen adquirida.

Para el desarrollo de este proyecto es necesario contar con una computadora con capacidad para el procesamiento de imágenes en tiempo real, almacenamiento, interfaces de usuario e Impresiones de los diagnósticos. Por ello la computadora de visión habrá de cumplir con las siguientes características mínimas:

1. Procesador Pentium a 700 MHz o superior.
2. Memoria RAM de trabajo de 128 MBytes.
3. Bus frontal de 133 MHz o superior tipo SDRAM (DIM).
4. Tarjeta de video con arquitectura AGP o PCI, con 128 MBytes en RAM.
5. Unidades mínimas: floppy disk y CD-ROM
6. Disco duro de 20Gb o superior (o sistema de discos esclavo maestro) para almacenamiento de imágenes
7. Monitor, teclado, Mouse.

Por último se utilizará para el desarrollo de programación una plataforma que cuente con un compilador muy rápido, herramientas potentes para la creación visual de aplicaciones, de herramientas para la creación y manejo de bases de datos, aplicaciones multimedia, enlace DDE, OLE, creación de DLLs, VBX, etc.. Que permite la creación de aplicaciones para Windows95 y Windows NT, y las aplicaciones pueden colocarse de forma muy sencilla en la pantalla según el principio de módulos. Además, ésta cuenta con componentes de acceso a datos los cuales puedan enlazarse a una gran variedad de controles visuales, aprovechando las características del lenguaje orientado a objetos.

Estos elementos son los requerimientos que nuestro sistema tendrá para el desarrollo de la tesis.

1. DISEÑO

En la fase de diseño es importante establecer el comportamiento dinámico del sistema, es decir, cómo debe reaccionar ante los acontecimientos. Para su desarrollo se utilizan los lenguajes de modelado. El resultado obtenido de la etapa de diseño facilita la implementación posterior de nuestro sistema, ya que proporciona la estructura básica del sistema y como los diferentes componentes actúan y se relacionan entre ellos.

El Diseño del Software, de acuerdo a los requerimientos de los patólogos en las diversas entrevistas, primeramente se modelará la base de datos para cada uno de los casos de uso, tomando en cuenta todos los requisitos explícitos contenidos en el modelo de análisis, el cual acumulará todos los requisitos implícitos que desea el médico.

2.1 Modelado de la base de datos

El software que se utiliza está desarrollado bajo el lenguaje de modelado de sistemas UML (Unified Modelling Language). El UML provee en forma general una notación estándar para expresar el análisis y diseño de software orientado a objetos. Los diagramas UML dan la posibilidad de modelar un complejo sistema de software, como el que aquí desarrollamos. El UML especifica una notación con diversos y diferentes diagramas, como lo muestra la *figura 2.1*. Es por eso que se eligió, debido a su capacidad de desarrollo.

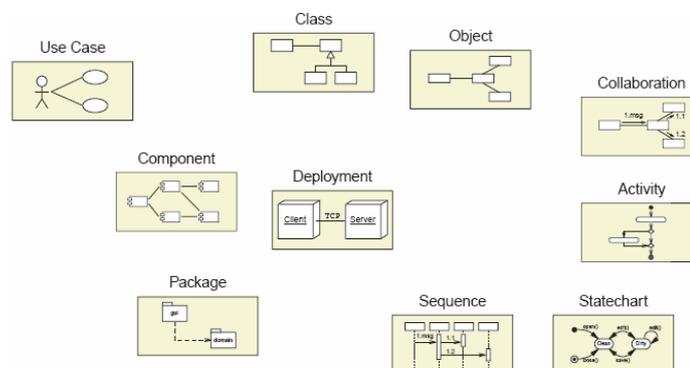


FIGURA 2.1 Diagramas UML

2.1.1 Modelo Dinámico

El modelo dinámico de un sistema pretende mostrar de una manera simplificada, pero entendible, cómo se comporta el sistema. El desarrollo del modelo dinámico constará de tres tipos de diagramas: el de Casos de Uso, el de Diseño y el de Secuencias, los cuales son descritos a continuación:

a) Casos de Uso

Los casos de Uso son requerimientos funcionales que describen de una manera detallada el comportamiento del sistema con los distintos actores que interactúan con él.

b) Diagrama de Diseño

Este diagrama representa el diseño del proceso de captura y reportes. El diseño de la interface gráfica de usuario es otra de las tareas que se debe realizar dentro del diseño. La interface de usuario de una aplicación debe de ser lo más intuitiva posible para el usuario final, de igual forma para almacenar los datos de la aplicación. Es necesario también diseñar la estructura de la base de datos que va a contener esos datos.

c) Diagrama de Secuencias

Los diagramas de secuencia de un sistema se preparan durante la fase de análisis de un ciclo de desarrollo y muestran gráficamente los eventos que fluyen de los actores al sistema. Además se define como la representación que muestra, en determinado escenario de un caso de uso, los eventos generados por actores externos, su orden y los eventos internos del sistema. Su formulación depende de la formulación previa de cada caso de uso.

El evento de un sistema es un hecho externo de entrada que un actor produce en un sistema. El evento da origen a una operación de respuesta. La operación de un sistema es una acción que éste ejecuta en respuesta a un evento del sistema.

En nuestro caso el departamento de patología, como sistema, se puede describir de la siguiente manera:

Los estudios efectuados por el departamento de anatomía patológica del hospital habitualmente son consecuencia de una solicitud de un médico especialista o a través de una de las campañas de medicina preventiva. La muestra, es tomada típicamente por un enfermero y entregada a dicho departamento.

El técnico o el patólogo reciben la muestra para analizar e inmediatamente registrarla. La figura 2.1.1a. muestra un **Caso de uso** del sistema.

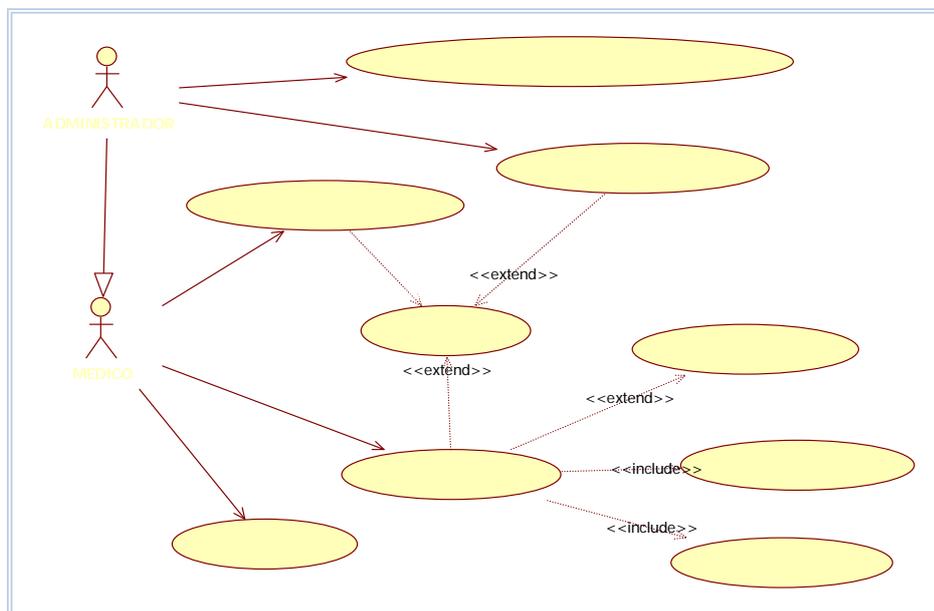


FIGURA 2.1.1a. Caso de Uso del Sistema

Si la muestra es un estudio citológico, tanto el técnico como el patólogo pueden llevar a cabo el diagnóstico; cabe señalar que si el técnico encuentra células cancerosas el patólogo tiene la obligación de examinarlas minuciosamente para reportar resultados apoyando su diagnóstico con imágenes provenientes de la muestra, así como imprimir el diagnóstico, firmarlo y darle validez legal.

Para la toma de dichas imágenes, quien se encuentre analizando la muestra, debe de mover la platina del microscopio en alguna de las formas establecidas manualmente.

En todos los casos donde los resultados son positivos, es decir, que existe presencia de anomalías, el diagnóstico se apoya con un reporte de imágenes relevantes. En algunos casos cuando exista duda en alguna imagen o ésta no sea clara existirá la posibilidad de consultar un atlas de imágenes ya existente en el Internet que se asemejen a las de la base de datos, realizando un análisis comparativo.

El **diagrama representa el diseño** del proceso de captura y reporte de un estudio citológico. El caso de estudios histológicos se ha omitido del actual reporte por ser muy semejante al de citológico, solo cambiando el formato del reporte más no la forma en que el reporte es invocado.

Éste describe de manera programática como el usuario interactúa con los procesos principales del sistema y establece el precedente para el diseño y modelado de la base de datos. Como lo muestra la figura 2.1.1b. (**Diagrama de diseño**)

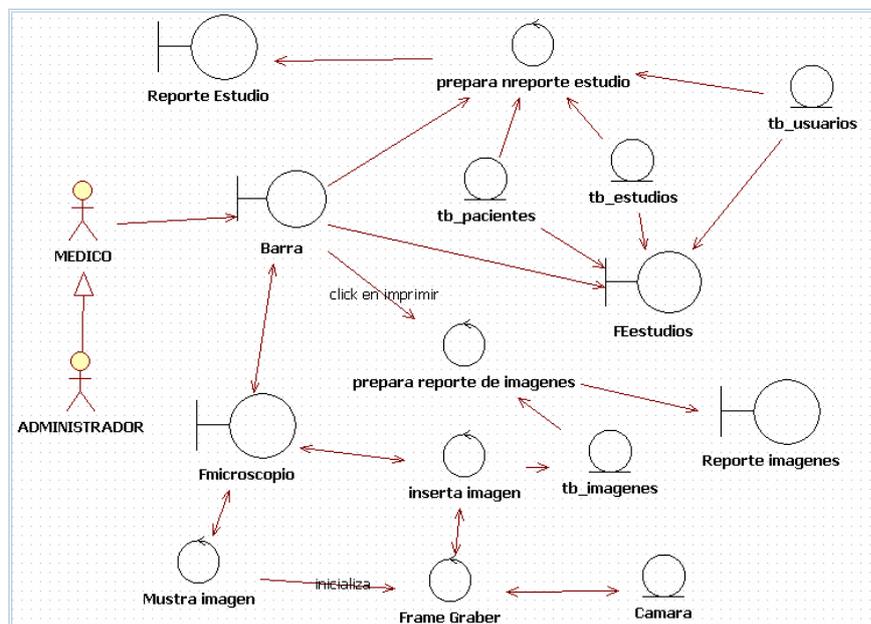


FIGURA 2.1.1b. Diagrama de Diseño

Los **diagramas de secuencia** de nuestro sistema muestran el flujo de los eventos en la generación de los procesos como son: el de identidad de los actores, cambio de contraseña, administración de usuarios y pacientes, realización y búsqueda de estudios, impresión de estudios y imágenes, que se

llevan a cabo dentro de un departamento de patología. Las siguientes figuras 2.1.1c, 2.1.1d, 2.1.1e, 2.1.1f, 2.1.1g, 2.1.1h, 2.1.1i, y 2.1.1j representan cada uno de estos diagramas.

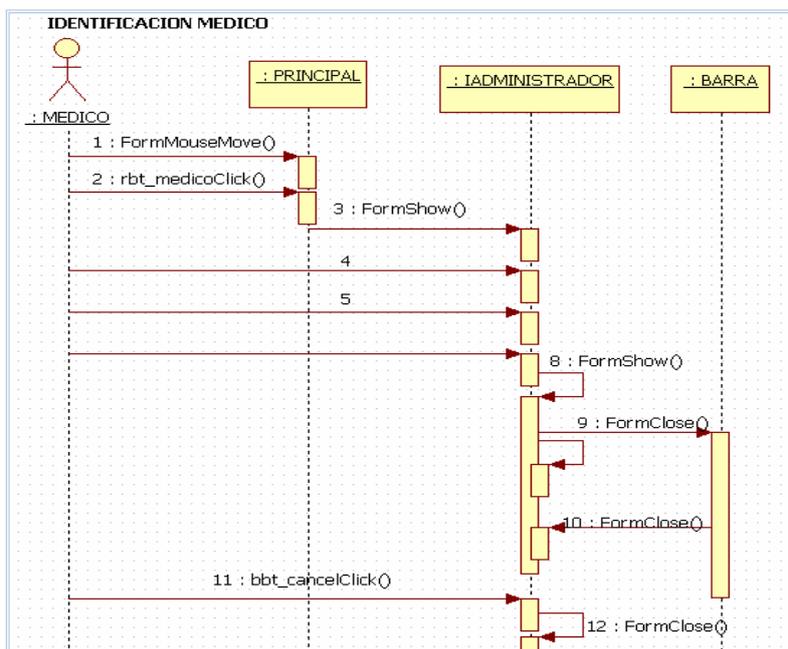
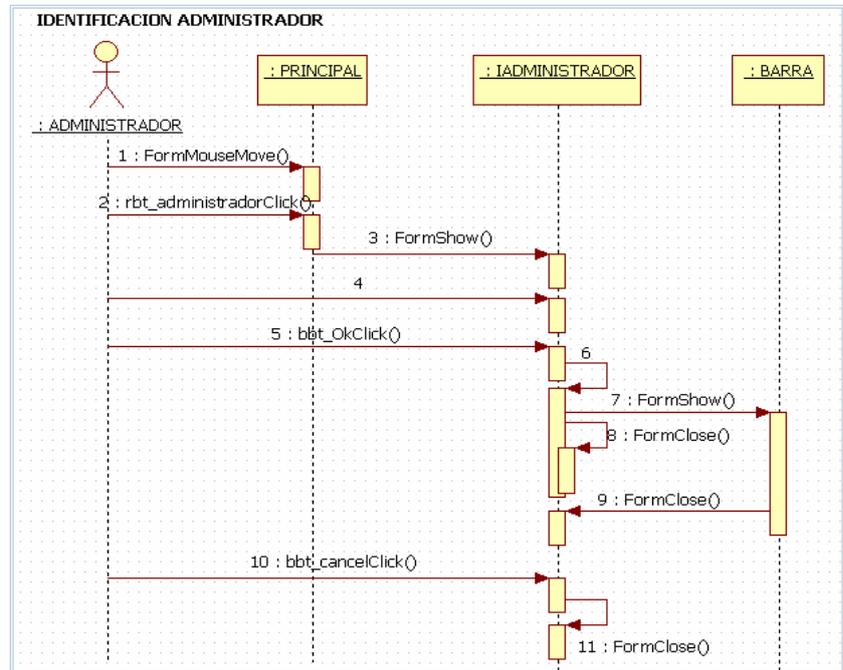


FIGURA 2.1.1c: Diagramas de secuencia que representa la identidad de los actores que intervienen en el sistema, administrador y médico

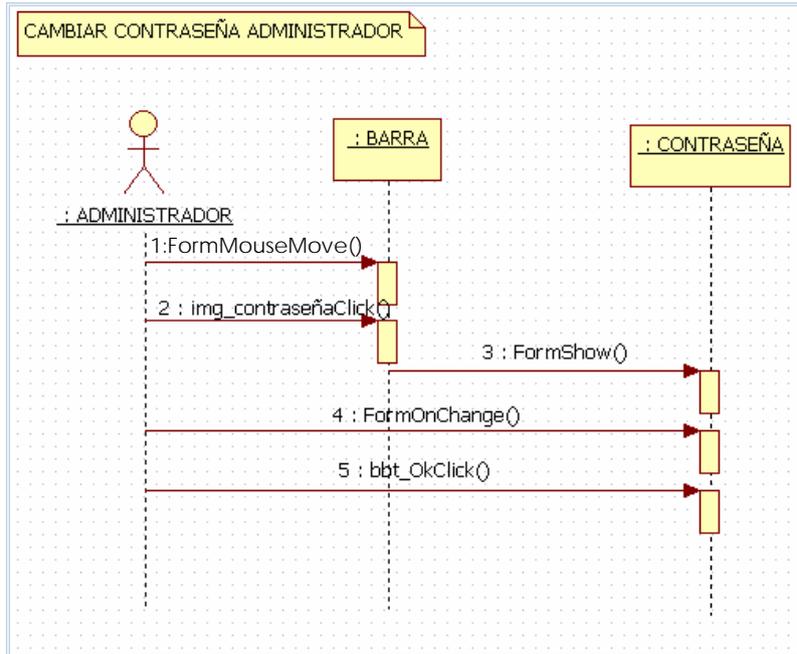


FIGURA 2.1.1d.- El diagrama de secuencia para el control de contraseñas

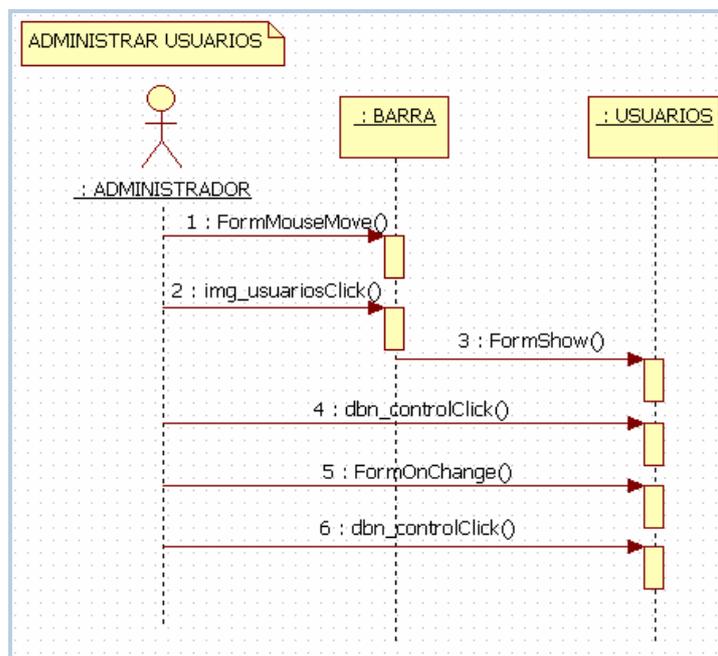


FIGURA 2.1.1e.- El diagrama de secuencia del rol que realiza el Administrador.

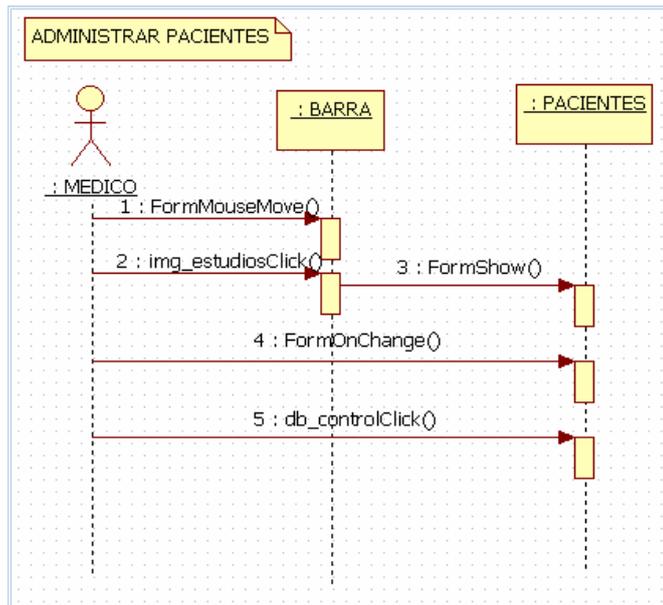


FIGURA 2.1.1f.- El diagrama de secuencia del rol que realiza el Médico.

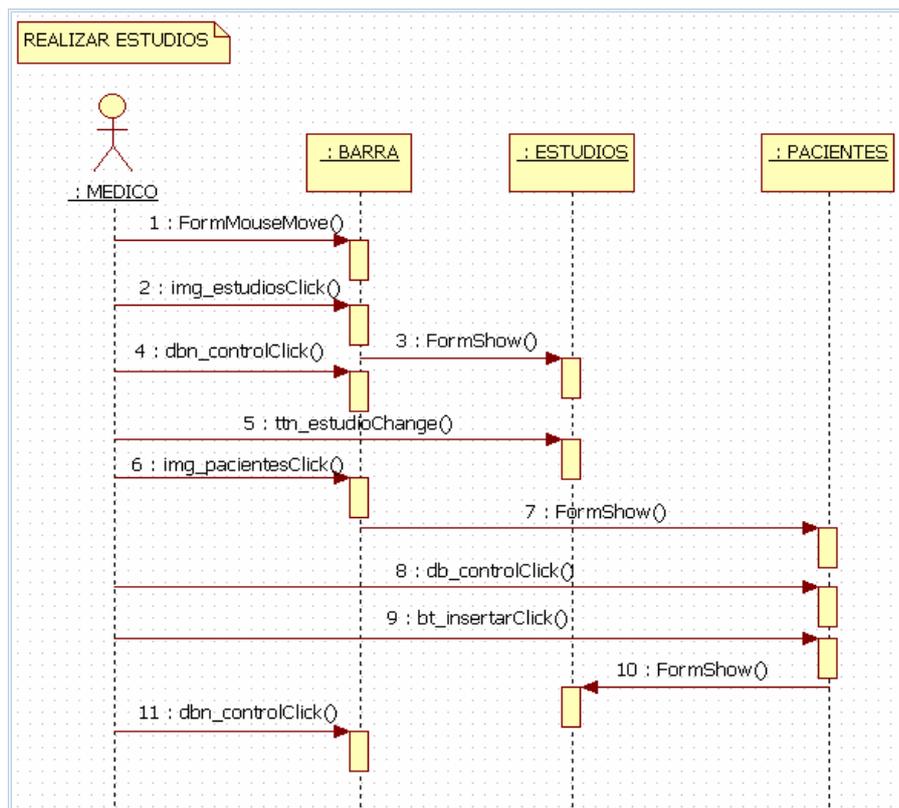


FIGURA 2.1.1g.- El diagrama de secuencia del rol que realiza el Médico.

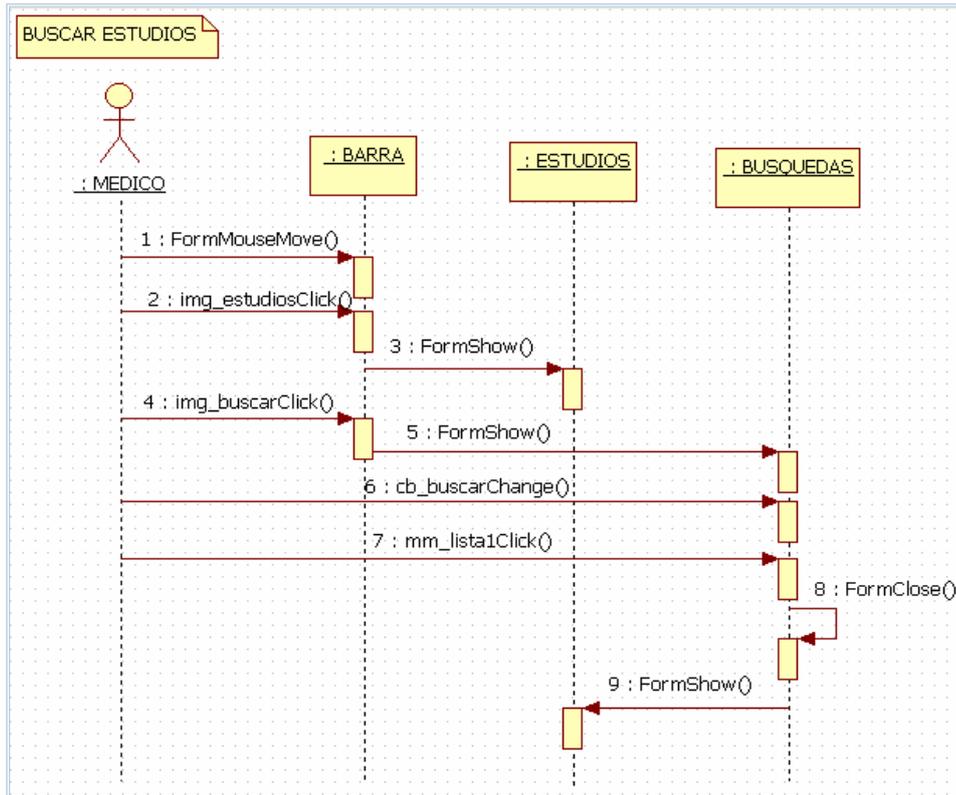


FIGURA 2.1.1h.- El diagrama de secuencia que realiza el usuario para la búsqueda de estudios.

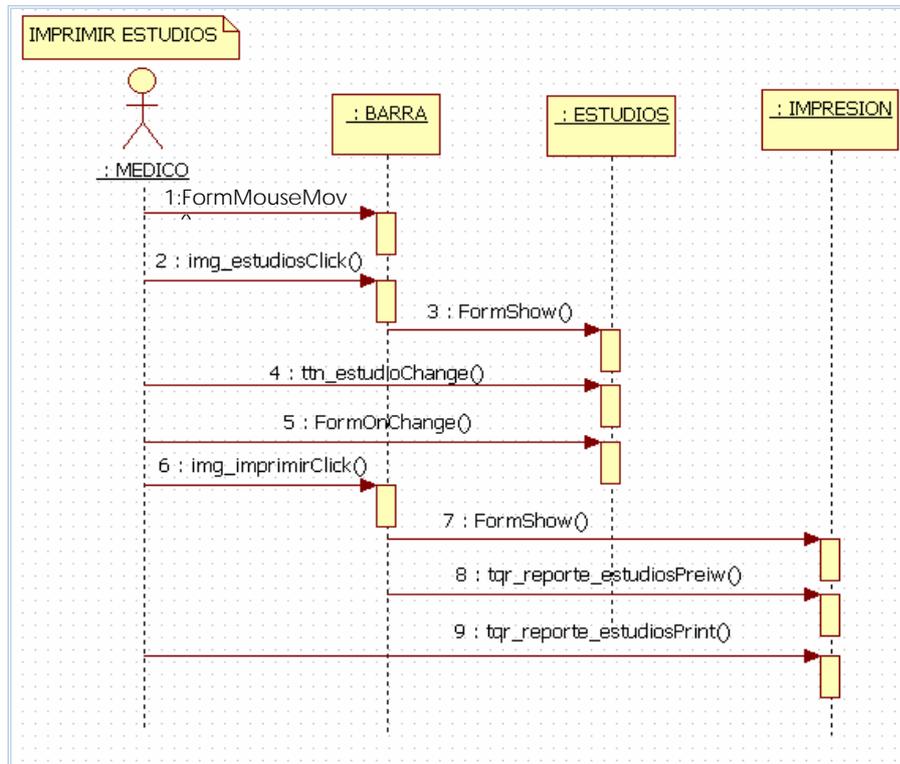


FIGURA 2.1.1i.- El diagrama de secuencia del rol que realiza el usuario para la impresión de estudios.

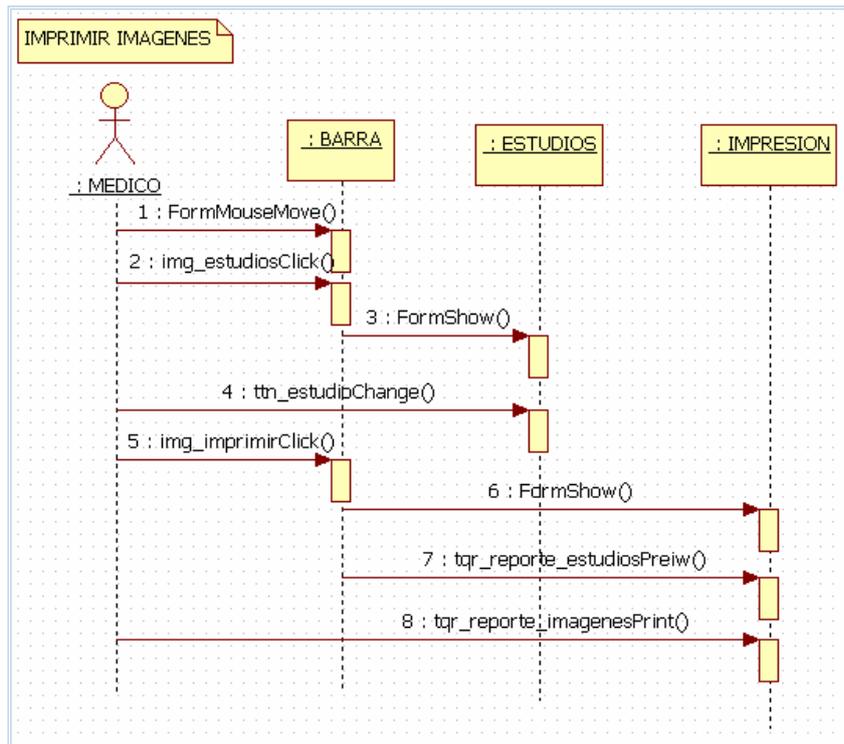


FIGURA 2.1.1j.- El diagrama de secuencia del rol que realiza el usuario para la impresión de imágenes.

2.1.2 Modelo Estático

En general, el modelo estático relaciona el comportamiento del sistema con su estructura. Éste es descrito principalmente por el diagrama de clases (figura 2.1.2a) y diagrama entidad-relación (figura 2.1.2b). Como se describen a continuación se presenta:

a) Diagrama de Clases

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. Éste es el diagrama principal para el análisis y diseño.

Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. La definición de clase incluye definiciones para atributos y operaciones.

b) Diagrama de Entidad-Relación

El diagrama Entidad / Relación (E/R) es la representación gráfica de las relaciones entre las entidades que conforman un sistema. Existen dos niveles en el diseño de un E/R, el nivel lógico; donde se definen los nombres de las entidades en singular y se muestran los atributos de cada entidad. El nivel físico del diagrama entidad-relación; se definen los nombres de las entidades en plural, así como también se define el tipo de dato que va a asociarse a cada atributo y el comportamiento que va a tener dicho atributo; por ejemplo: se define si el atributo va a ser obligatorio u opcional, si va a pertenecer a algún dominio, si se restringirá a que acepte solamente un conjunto de valores determinados, etc. El nivel físico del diagrama E/R será el que finalmente determinará la estructura de la base de datos del sistema.

El diagrama de clases de nuestro sistema representa el módulo de estudios citológicos, al igual que en el caso del diagrama de diseño. Los estudios histológicos han sido omitidos intencionalmente debido a que ésta es sólo una generalización del caso de estudios citológicos.

Debido a que el software será desarrollado en Delphi versión 7.0, todas las clases heredan atributos del objeto Object.

Object Pascal es totalmente compatible con el Borland Pascal 7.0, lo que permite que programas desarrollados con este último puedan ser convertidos a Delphi. Prácticamente todos los elementos del entorno Delphi son objetos, que unas veces se encuentran definidos de antemano (las plantillas (forms), los componentes de la paleta, objetos no visuales, etc).

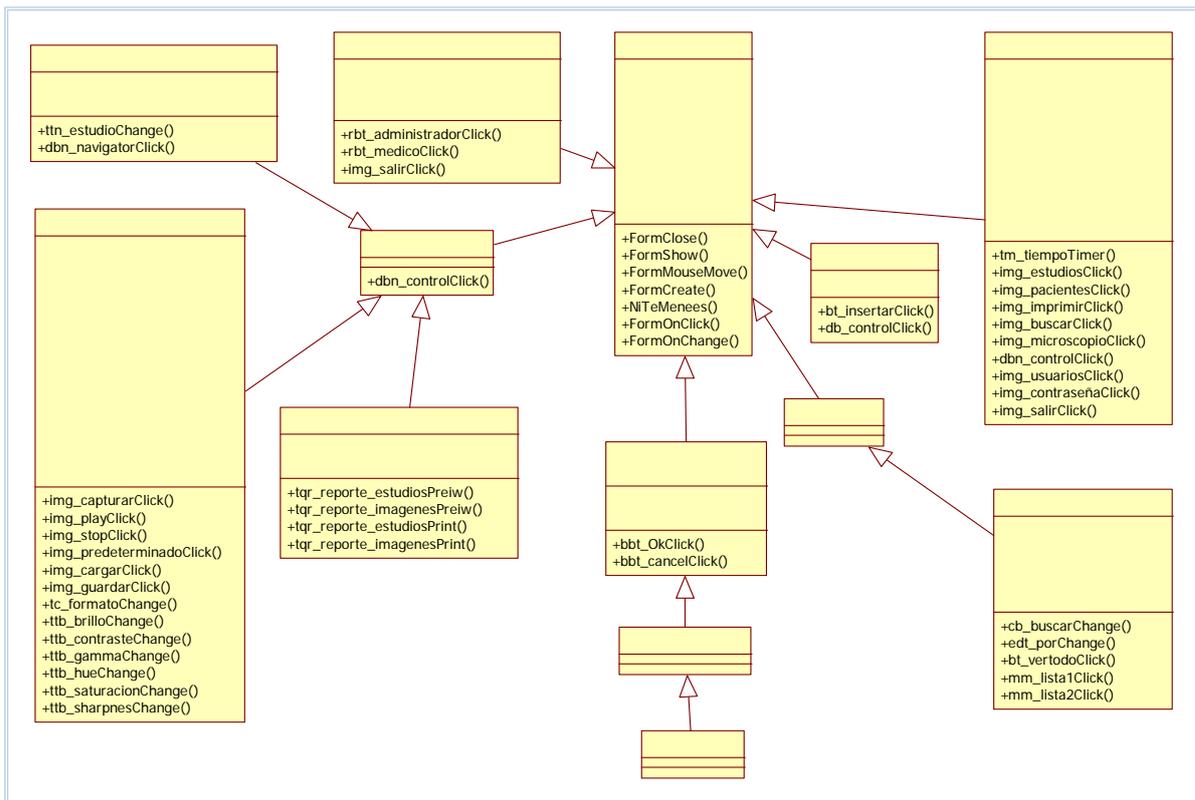


FIGURA 2.1.2a.- Diagrama de Clases

El modelado de la base de datos (**Diagrama de Entidad-Relación**) empleada en el desarrollo de nuestro sistema fue Paradox v. 7.0. Esto debido a que es el manejador de base de datos estándar de Delphi 7.0. El sistema cuenta con cinco entidades interrelacionadas como son: pacientes, imágenes, contraseña, estudios y usuarios.

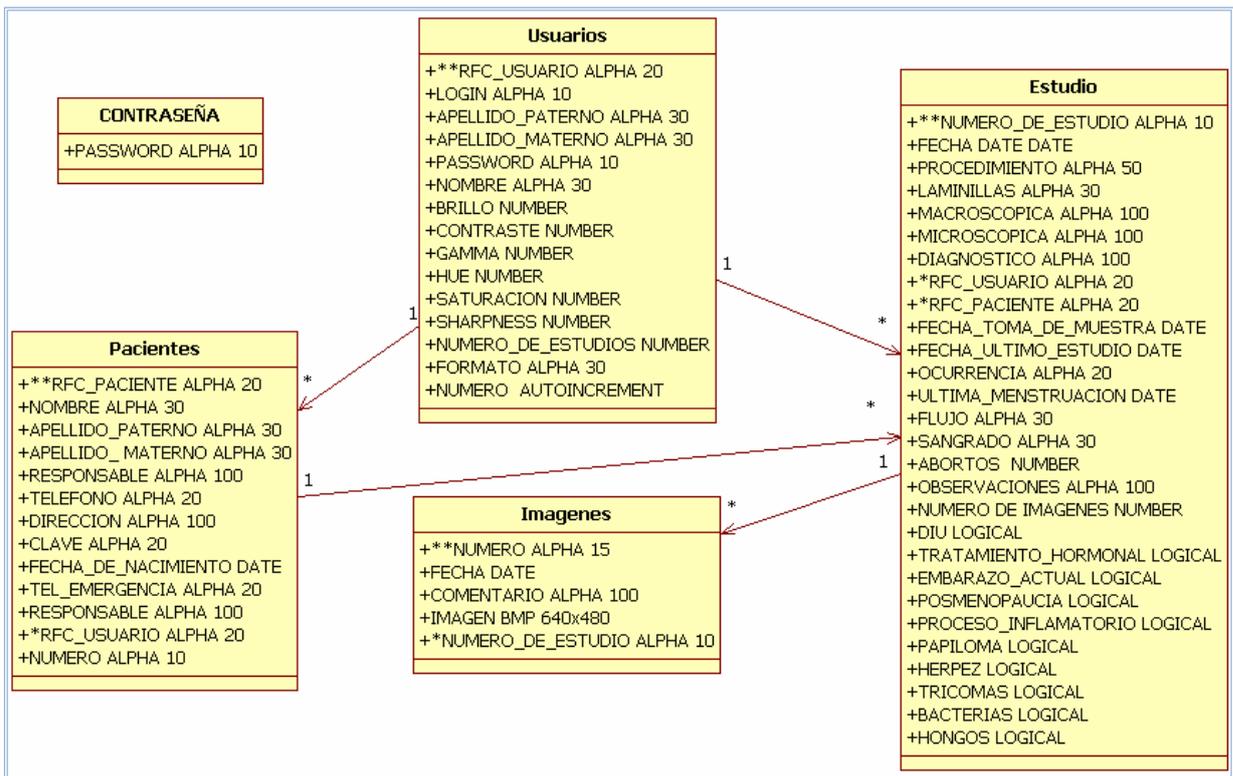


FIGURA 2.1.2b. Diagrama de entidad-relación

1 → * Relación de uno a muchos

** Llave primaria de cada tabla

* Relación que existe entre ambas tablas.

Con los diagramas obtenidos del diseño del sistema, se realizará la tarea de programación para el desarrollo e implementación de éste, como se verá en los siguientes capítulos.

1. DESARROLLO

En los capítulos anteriores se obtuvieron el análisis y diseño del sistema, mediante los diagramas UML los cuales muestran las abstracciones de éste; es decir, las representaciones gráficas que muestran la información relevante del sistema y definen la estructura de la base de datos relacional que deberá soportar. Con esto se cuenta con las condiciones necesarias para comenzar la programación, utilizado como plataforma de desarrollo en Delphi.

Delphi es algo más que el sucesor del famoso Pascal 7.0 de Borland. Dispone no sólo de un compilador muy rápido (más que la gran mayoría de los compiladores de C++, como ya era tradicional en Turbo Pascal), sino también de potentes herramientas para la creación visual de aplicaciones, de herramientas completas para la creación y manejo de bases de datos, aplicaciones multimedia, enlace DDE, OLE, creación de DLLs, VBX, etc, además dispone del Object Pascal, un lenguaje de programación muy poderoso. Este lenguaje surge a partir del desarrollo del Borland Pascal 7.0, un lenguaje que ocupa un lugar muy importante en la programación de computadoras personales. Es un lenguaje totalmente orientado a objetos. Prácticamente todos los elementos del entorno Delphi son objetos, que unas veces se encuentran definidos de antemano (los forms, los componentes de la paleta, objetos no visuales, etc).

Delphi permite la creación de aplicaciones para Windows 3.x, Windows95 y Windows NT, y las aplicaciones pueden colocarse de forma muy sencilla en la pantalla según el principio de módulos. Para ello se dispone de una paleta dotada de una gran variedad de componentes, algo así como los bloques de construcción de cada programa. Esta paleta es denominada por Borland VCL (Visual Component Library), o biblioteca de componentes visuales. Tiene un aspecto similar a Visual Basic, pero el aspecto externo indica la misma facilidad de uso que Visual Basic.

Una de las principales características y ventajas de Delphi es su capacidad para desarrollar aplicaciones con conectividad a bases de datos y se puede realizar una conexión a bases de datos usando diferentes capas o motores de conexión.

1.1 Características del Ambiente y Software de Aplicación

El término visión por computadora, dentro del campo de la inteligencia artificial es considerado como el conjunto de todas aquellas técnicas y modelos que permiten procesar, analizar y explicar cualquier tipo de información obtenida a través de imágenes digitales.

Debido a que la información visual es una de las principales fuentes del mundo real, resulta útil proveer a una computadora digital del sentido de la vista (a partir de imágenes tomadas con cámaras digitales o analógicas), que junto con otros mecanismos como el aprendizaje hagan de ésta una herramienta capaz de detectar y ubicar objetos en el mundo real, objetivo principal de la visión por computadora.

El sistema desarrollado en esta tesis para el almacenamiento, consulta y diagnóstico de estudios médicos oncológicos, se deberá ejecutar en computadoras que tengan capacidad de soportar: Un Frame Grabber, motor de base de datos de Borland (Borland Data Base Engine), tener memoria RAM suficiente y un sistema operativo que soporte la aplicación del desarrollo. Estos requerimientos se describirán en el capítulo de Implementación. Además deberá de tener en cuenta las siguientes características en el software de desarrollo:

1. Registro de pacientes en una base de datos
2. Registro y control de los estudios histológicos y citológicos que elabora el departamento de anatomía patológica
3. Capacidad de búsqueda de estudios por: apellidos, CURP del paciente, fecha de estudio, entre otros.
4. Despliegue en el monitor del equipo de cómputo, de la muestra del microscopio en una resolución máxima.

5. Capacidad de inclusión de imágenes significativas en los registros de los estudios histológicos y citológicos.
6. Elaboración de reportes de diagnóstico de estudios histológicos y citológicos.
7. Manual del usuario
8. Control de ingreso al sistema mediante un esquema nombre de usuario y clave de acceso.
9. Capacidad de impresión de reportes en formato para su manejo digital.

3.1.1 Elementos de Software

a) Sistema Operativo Windows

La computadora con que se cuenta para el desarrollo del proyecto es una PC con un sistema operativo Windows 2000. La Computadora tiene la capacidad de soportar la aplicación que se está desarrollando. El Sistema Operativo Win2000 tiene la facilidad para manejar distintos usuarios y cuenta con un ambiente gráfico para aplicaciones multiusuario, además Windows se ejecuta en un segundo plano y protege todos los archivos instalados por el programa de instalación.

Windows 2000 soporta:

- 1.- El software de instalación de Frame Grabber que da la opción de un Active X, que puede ser utilizado como objeto en la programación del sistema.
- 2.- El Software de instalación de la cámara que es reconocida por el Frame Grabber
- 3.- Reconocimiento de todos los controladores de los periféricos utilizados.

b) Delphi (Plataforma de desarrollo de programación)

Delphi es una herramienta de desarrollo rápido de aplicaciones (RAD) Cuenta con un entorno de desarrollo integrado (IDE) y de una biblioteca de componentes visuales (VCL), cuya implementación

basada en el paradigma de la programación orientada a objetos, facilita el desarrollo de aplicaciones bajo sistemas Windows.

c) Active X Imaging Source

Herramienta necesaria para desplegar las imágenes arrojadas a través de la cámara en tiempo real.

Para poder utilizarlo simplemente se agrega el componente dentro de los elementos presentados como Active X en el ambiente de desarrollo Delphi y se instala como cualquier otro elemento del tipo Active X que desea ser utilizado y reconocido por el entorno Delphi.

Teniendo el software instalado en el sistema operativo y utilizando el hardware. que se describirá en el capítulo siguiente, se tendrá un sistema que mediante las técnicas de programación y de visión harán posible la manipulación, control y almacenamiento de imágenes para tener la posibilidad de realizar el diagnóstico de estudios médicos oncológicos. La interface-microscopio-computadora proporciona imágenes en tiempo real para realizar un análisis y diagnóstico de células, esto se explicará en el capítulo de implementación.

La Característica principal del sistema es un menú por íconos, y sus principales funciones son: Dar de alta a usuarios, pacientes, registrar estudios de pacientes, capturar imagen, abrir imagen, guardar imagen, imagen microscopio, cargar imagen, captar la ampliación de la imagen e impresión de reportes.

3.1.2 Descripción funcional del Sistema

Con los diagramas de caso de uso y de diseño descritos en el capítulo anterior, se diseñó la pantalla principal de nuestro sistema.

Pantalla Principal

Este sistema cuenta con una pantalla principal, dividida en 2 escenarios: La parte funcional del Administrador y la parte funcional del usuario (quien en este caso es quien maneja el sistema). Como se ilustra en la figura 3.1.2a.

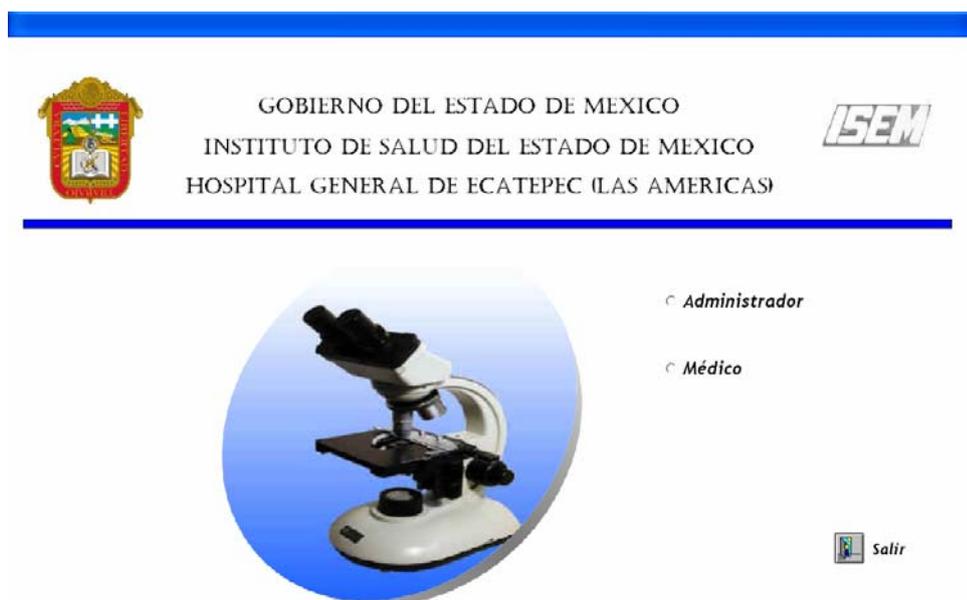


FIGURA 3.1.2a.

Con base en los diagramas de secuencia descritos de igual manera en el capítulo de diseño de nuestro sistema (con los cuales se muestra el flujo de los eventos en la generación de los procesos como son: el de identidad de los actores, cambio de contraseña, administración de usuarios y pacientes, realización y búsqueda de estudios, impresión de estudios e imágenes, que se llevan a cabo dentro de un departamento de patología), se diseñó el proceso del sistema, teniendo como primer paso del proceso la autenticación de los actores que intervienen en nuestro sistema.

Tanto administrador como usuario (actores principales del sistema) deberán identificarse al momento de ingresar al sistema, cabe señalar que el

sistema es sensible a mayúsculas y minúsculas. Si su contraseña es correcta inmediatamente ingresan al menú principal correspondiente, de lo contrario si su contraseña es incorrecta te envía un mensaje de “password no válido”. Como lo muestra la 3.1.2b.



FIGURA 3.1.2b.

Tomando en cuenta que el administrador tiene el manejo y mantenimiento de la base de datos, él puede ingresar, modificar y consultar todas las pantallas del Sistema.

Al hacer doble click en el botón “administrador” y si su identificación es correcta, muestra la pantalla de la ventana estudios con tres pestañas: Estudio, Datos Clínicos e Imágenes, así como la barra de íconos. Como lo muestra la figura 3.1.2c.

The screenshot shows the 'ESTUDIOS' application window. At the top, there is a menu bar with 'ADMINISTRADOR' and 'ESTUDIOS'. Below the menu bar is a toolbar with various icons. The main content area is titled 'FESTUDIOS' and contains a form with the following fields:

ESTUDIO	DATOS CLINICOS	IMAGENES	
NUMERO DE ESTUDIO	FECHA	RFC PACIENTE	NOMBRE PACIENTE
A-ADM-36	11/09/2007	MAVA06111978	ARTURO MARTINEZ VAZQUEZ
LAMINILLAS	PROCEDIMIENTO	EDAD	
		29	
DESCRIPCION MACROSCOPICA			
DESCRIPCION MICROSCOPICA			
DIAGNOSTICO			

FIGURA 3.1.2c.

La función de administrar usuarios es exclusiva del administrador. Al hacer click en el ícono “usuarios”, se puede insertar, modificar o eliminar datos de los usuarios. Como se muestra en la figura 3.1.2d.

The screenshot shows the 'USUARIOS' application window. At the top, there is a menu bar with 'ADMINISTRADOR' and 'USUARIOS'. Below the menu bar is a toolbar with various icons. The main content area is titled 'USUARIOS' and contains a form with the following fields:

RFC	NOMBRE	APELLIDO PATERNO	APELLIDO MATERNO
ADMINISTRADOR	ADM	INISTRA	DOR
LOGIN	PASSWORD		
ADMINIS	TRACOR		

FIGURA 3.1.2d.

Cuando se hace click en el ícono de contraseña, se muestra la ventana “cambio de contraseña administrador”, para realizar el cambio de contraseña si

los datos que requiere el sistema no son correctos, el sistema mostrará un mensaje de error. Como lo muestra la figura 3.1.2e.



FIGURA 3.1.2e.

De la pantalla principal, al hacer doble click en el botón de médico, éste deberá realizar el proceso de identificación, insertando los datos correctos que requiere el sistema. De lo contrario el sistema mandará un mensaje de error. Como lo muestra la figura 3.1.2f.

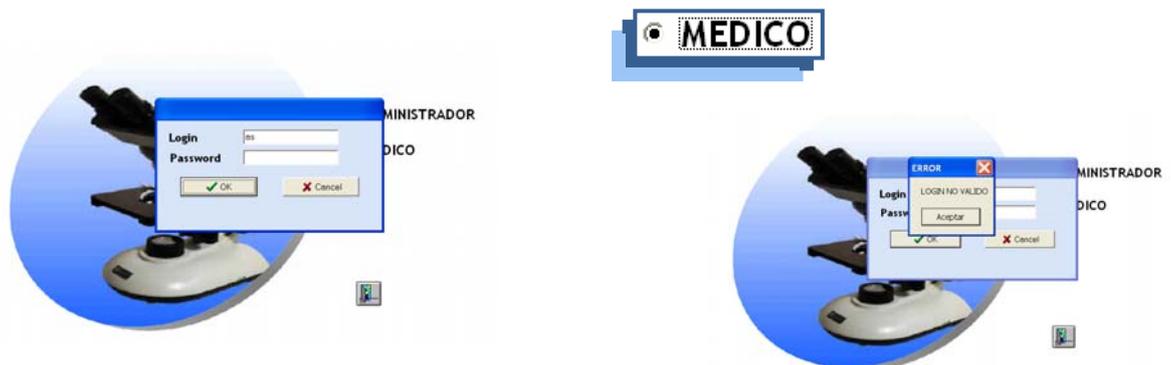
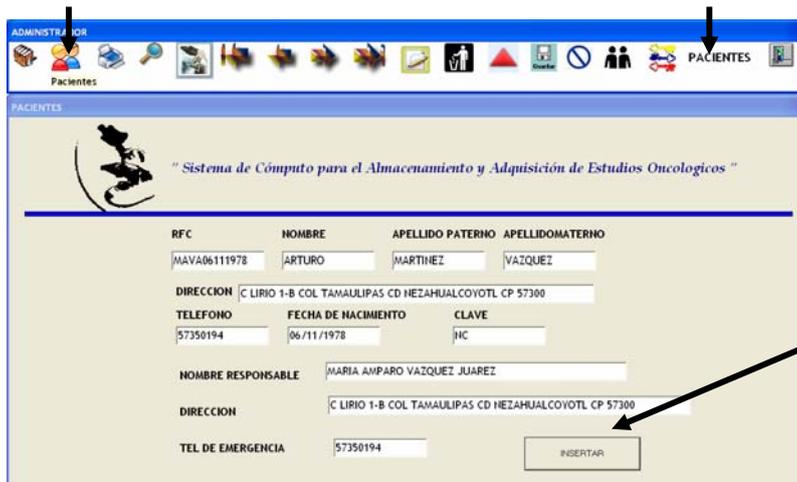


FIGURA 3.1.2f. Usuario (Médico).- Tiene permisos restringidos, ya que el sólo hace la función de ingresar datos y hacer consultas en el sistema.

Al ingresar en este apartado de médico aparecerá la misma barra, sólo que se omitirán los íconos de usuarios y contraseña que son exclusivos para el administrador. Como lo muestra la figura 3.1.2g.



Al hacer click en el botón "insertar", inserta los datos del paciente en el estudio actual.

FIGURA 3.1.2i.

Al hacer click en el ícono "imprimir", muestra un previo de la pestaña activa en la ventana de estudios, puede ser reporte de estudio o imágenes del mismo.



Esta pantalla de reporte de estudios contiene sus propios íconos, como son: imprimir, zoom, guardar, las flechas de paginación atrás y adelante, así como cerrar (close), para regresar al menú principal. Como lo muestra la figura 3.1.2j.

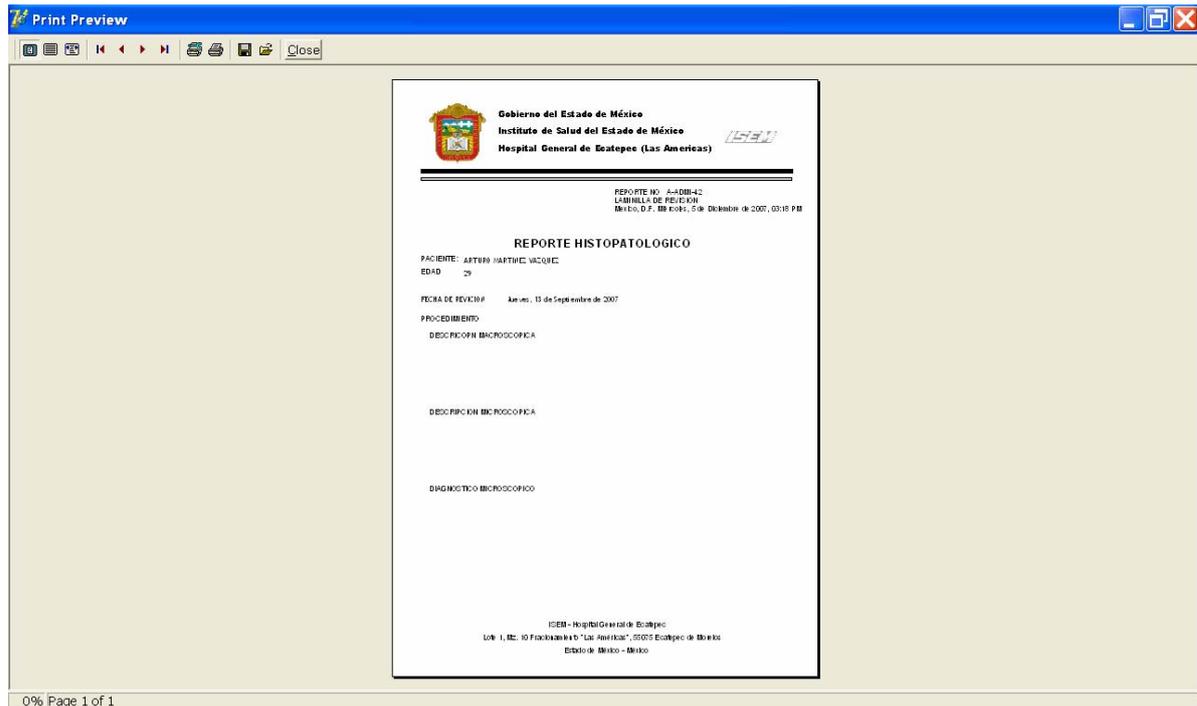


FIGURA 3.1.2j.

De igual manera se genera la pantalla de reporte de imágenes del estudio activo. Como se muestra en la figura 3.1.2k.

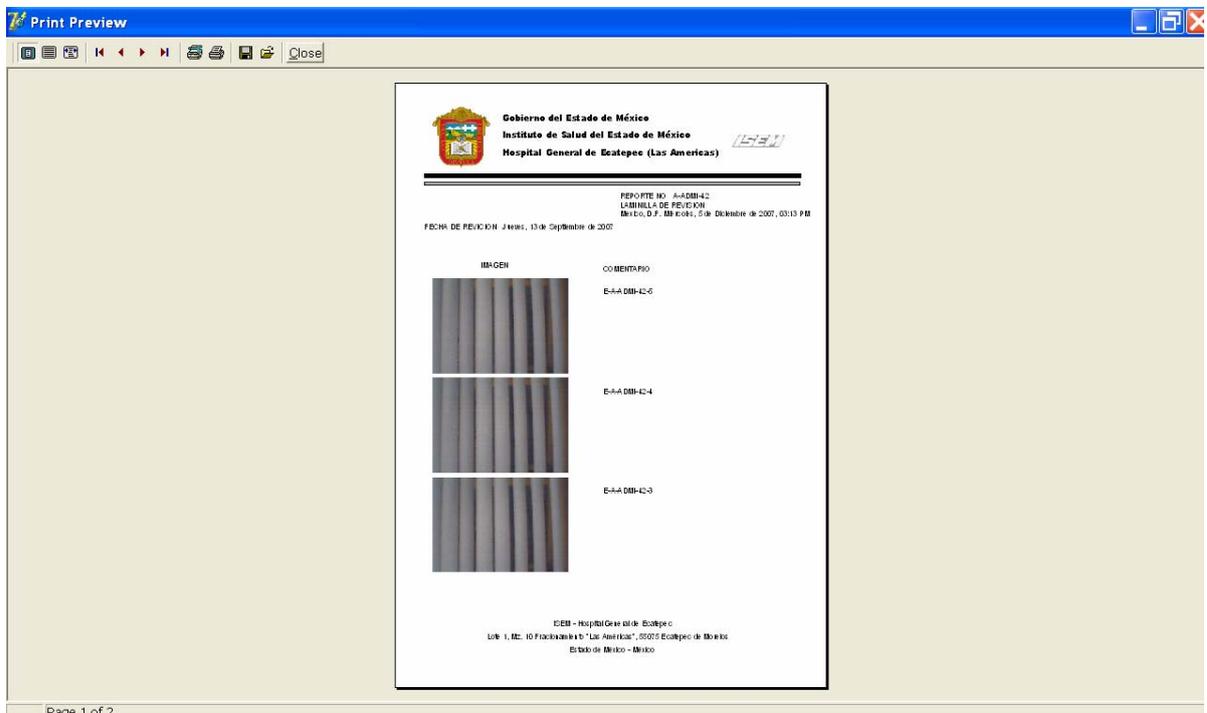


FIGURA 3.1.2k.

Al hacer click en el ícono “búsquedas”, muestra la pantalla para realizar búsquedas por usuario, estudios o pacientes, dependiendo de la ventana que esta activa, donde se podrá seleccionar el criterio de búsqueda, ya sea por nombre, número o RFC. Para mostrar los datos solicitados de un paciente, usuario o estudio, bastará con hacer click en la línea correspondiente. Como lo muestra la figura 3.1.2l.



FIGURA 3.1.2l.

Cuando se hace click en el ícono “microscopio” te despliega la pantalla de captura de imágenes, te da como parámetros de imagen los predeterminados por el sistema, sin embargo, se pueden manipular manualmente con sus respectivos íconos. Como lo muestra la figura 3.1.2m.



 **Play.-** Inicializa y enciende cámara.

 **Stop.-** Para y apaga cámara.

 **Predeterminado.-** Carga valores de parámetros de imagen predeterminados por el sistema.

 **Guarda.-** Los cambios efectuados por el usuario.

 **Carga.-** Carga valores realizados por el usuario.

FIGURA 3.1.2m.

Los íconos de la figura 3.1.2n se utilizan para navegar y administrar los datos de pacientes, usuarios, estudios e imágenes dentro del sistema.



FIGURA 3.1.2n.

Por último tenemos en la barra de íconos “salir” que te regresa a la pantalla principal.



1.2 Adquisición de la imagen del Microscopio

Se utiliza un Frame Grabber para la adquisición de la imagen de la señal generada por la cámara, que mediante programación se tendrá la posibilidad de almacenamiento e impresión. El Frame Grabber se describirá con más detalle en la siguiente sección.

El código fuente del sistema se encuentran en el anexo A.

1. IMPLEMENTACIÓN

Es la última fase del desarrollo de Sistemas. Es el proceso de instalar equipos o Software nuevo.

Es importante mencionar que en la implementación de cualquier sistema, lo primero que se debe hacer es asegurarse que el Sistema sea operacional, esto es, que funcione de acuerdo a los requerimientos del análisis y que permita que los usuarios puedan operarlo de una forma fácil, en el aspecto de software y hardware para realizar sus funciones.

4.1 Características del Hardware

Primordialmente tenemos que el sistema se desarrollará:

1. En una computadora donde se ejecuta el software desarrollado y se almacena la base de datos.
2. Frame Grabber
3. Un Microscopio el cual tiene acoplado una cámara digital para el despliegue de imágenes en el equipo de cómputo.
4. Una impresora de inyección de tinta o láser de color.

4.1.1 Computadora

Para el desarrollo del sistema se contó con una computadora con las siguientes características:

1. Procesador Pentium 4 a 3.4 GHZ
2. 2 GB en RAM
3. Disco duro 500 GB
4. Tarjeta de video de 512 MB
5. Windows 2000
6. Monitor, teclado, Mouse etc.



Esta computadora será la responsable de almacenar el software de desarrollo y la base de datos, así como las aplicaciones de los diferentes dispositivos utilizados para en el desarrollo de la tesis, le llamaremos la *computadora de visión*.

La computadora de visión habrá de procesar la información digital de la salida del Frame Grabber para su despliegue que es la herramienta principal del patólogo para realizar un diagnóstico clínico.

4.1.2. Frame Grabber

Para la adquisición de las imágenes que se procesan en la computadora de visión se utiliza una tarjeta de captura de imágenes (Frame Grabber). La calidad de las imágenes estará disponible en 24 bits de color verdadero. Este tipo de dispositivos se utiliza en aplicaciones médicas, seguridad, automatización, etc. La tarjeta que se utilizó en el desarrollo es de Imaging Source DFG/LC2, la figura 4.1.2a muestra las características de la tarjeta utilizada en este desarrollo:

	DFG/LC2
Formato de video	PAL/NTSC, RS-170/CCIR
Max resolución	768x576 at 50Hz (PAL/CCIR), 640x480 at 60Hz (NTSC/RS170)
Salidas Analógicas	4
Conectores	4x
Interfase Bus	PCI 2.1, Bus Maestro
Formatos de Color	YUV 4:2:2, 4:1:1, 4:2:0, 4:1:0, RGB 32, RGB 24, RGB 8
Parámetros Ajustables	Brillo, contraste, saturación
Max. Temperatura (operación)	-5°C to 45°C



FIGURA 4.1.2a Características del Frame Grabber.

El Frame Grabber tiene la capacidad de muestrear el flujo de entrada de video, y convertirlo en información binaria y almacenarla en registros de memoria cuya dirección es plenamente conocida; de esta forma son directamente accesibles por los algoritmos de procesamiento digital. El Frame

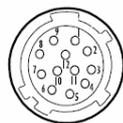
Grabber puede pensarse como un convertidor analógico digital altamente especializado.

Para los fines del proyecto, que procesará imágenes en tiempo real, se requiere un Frame Grabber que tenga gran velocidad de adquisición de píxeles del orden de 50 MHz o superior, entrada compatible con el formato de salida de la cámara que se utiliza en esta tesis, bus de conexión de amplio ancho de banda como el PCI para altas transferencias de datos (del orden de decenas de Megabytes por segundo), opciones de señales de entrada/salida para control y sincronización con la cámara “residente” y opciones de pre-procesamiento de la imagen adquirida.

El Frame Grabber DFG/LC2 de Imaging Source cuenta con las características necesarias para realizar funciones de procesamiento de imágenes en tiempo real, la figura 4.1.2b muestra las conexiones para su uso.

Frame Grabber Connectors

12-pol. Hirose plug pinning



- 1 GND
- 2 +12
- 3 Y input (GND)
- 4 Y input (signal)
- 5 HD output (GND)
- 6 HD output (signal)
- 7 VD Output (signal)
- 8 C input (GND)
- 9 C input (signal)
- 10 GND
- 11 +12V
- 12 VD output (GND)

6-pol. Hirose plug pinning



- 1 Digital Out
- 2 Flash Strobe (out)
- 3 GND
- 4 Digital In
- 5 Trigger (in)
- 6 +12V

DFG/LC2 inputs

- 1 Input 1
- 2 Input 2
- 3 Input 3
- 4 Input 4
- 5 Digital I/O



FIGURA 4.1.2b

Este dispositivo cuenta además con posibilidad de interconexión con periféricos de visión, gracias a las características de funcionamiento del elemento de software conocido como Active X Imaging Source el cual detecta cualquier cámara de video como dispositivo electrónico conectado a la computadora ya sea mediante el puerto USB o bien mediante la tarjeta Frame Grabber.

4.1.3 Microscopio

El patólogo utiliza un microscopio como Instrumento de diagnóstico. Para el desarrollo de esta tesis se utilizó un microscopio OLYMPUS CX3, el cual cuenta con una salida óptica para colocar una cámara fotográfica. Esta salida se adapta para colocar una cámara de video y poder adquirir la imagen que procesa la computadora de video. La figura 4.1.3 muestra las características del microscopio.



Resolución	Cuádruple 4x, 10x, 40x y 100x
Especificaciones	Magnificación de campos Brillantes
Inclinación	30°
Controles	Enfoque, luz, ajuste
Dimensiones	391mm altura
Peso	5.9 kg

FIGURA 4.1.3.- Características del microscopio

4.1.4 Cámara CCD

Las cámaras que se usan en el campo del procesamiento digital de imágenes constan primordialmente de dos partes:

- a) Unidad de Adquisición de Imagen
- b) Unidad de Salida de Imagen

La Unidad de adquisición queda definida por el tipo de sensor que registra la señal luminosa que conforma la imagen.

La Unidad de salida es la encargada de interpretar los valores del sensor y convertirlos en una señal eléctrica de video conforme a un protocolo, el cual puede ser estándar o no, dicho protocolo define entre otros parámetros el número de líneas horizontales y verticales por cuadro así como el número de cuadros por segundo.

Las cámaras CCD (Charge Coupled Device) basan su funcionamiento en un chip que consiste de un arreglo bidimensional de sensores de señales luminosas. La dimensión de dicho arreglo define la resolución de la imagen que se obtenga a la salida de la cámara.

Otro de los parámetros importantes que deben de ser considerados al trabajar con el chip CCD es su resolución expresada en el número de pixeles. Una resolución típica en este chip es por ejemplo 756 X 581 (columnas por líneas). La figura 4.1.4a muestra las características principales de la cámara CCD y la figura 4.1.4b la conexión para su uso.

Imaginesours CSC-740	
Especificaciones Generales (salida):	
<i>Resolución</i>	450 lines
<i>Frame rate</i>	PAL: 25 Frames/sec NTSC: 30 Frames/sec
<i>Sensibilidad</i>	0.15 lx at AGC +48dB, F1.4
Interfase (óptica):	
<i>Tamaño</i>	1/3"
<i>Resolución</i>	PAL: H:752, V:582 NTSC: H:768, V:494
<i>Tamaño Píxel</i>	CCIR: H:6.5µm, V:6.25µm RS-170: H:6.35µm, V:7.4µm
<i>Max. Temperatura (operación)</i>	-5°C to 40°C



FIGURA 4.1.4a.- Características de la cámara CCD

Min-Din 4 pole for SVHS cameras

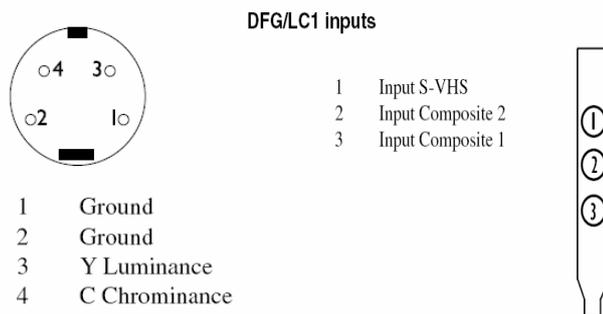


FIGURA 4.1.1b.- Conexión de la cámara CCD con la tarjeta del Frame Grabber.

La cámara que se utiliza es una Imagine Source CSC-740. Las etapas de operación de una cámara pueden dividirse en etapa de adquisición y etapa de formato de salida de video.

La etapa de adquisición queda definida principalmente por el sensor que adquiere la imagen. La etapa de formato de salida estará definida con base a la forma de conexión establecida entre la cámara y la computadora, es decir; si la cámara está conectada mediante el dispositivo Frame Grabber o el puerto USB. El formato de salida será el que corresponda a las especificaciones por el fabricante de cada dispositivo, en nuestro caso fue utilizada mediante el dispositivo Frame Grabber utilizando las salidas Y/C de la cámara.

4.2 Diseño del sistema de cómputo para el almacenamiento y adquisición de Estudios médicos oncológicos

El diseño final del sistema es de la siguiente forma: se obtiene la imagen directamente desde el microscopio y se coloca en la computadora, utilizando una tarjeta digitalizadora (Frame Grabber) conectada a una cámara CCD que está adaptada al microscopio por un dispositivo óptico. La siguiente figura 4.2a muestra este proceso.

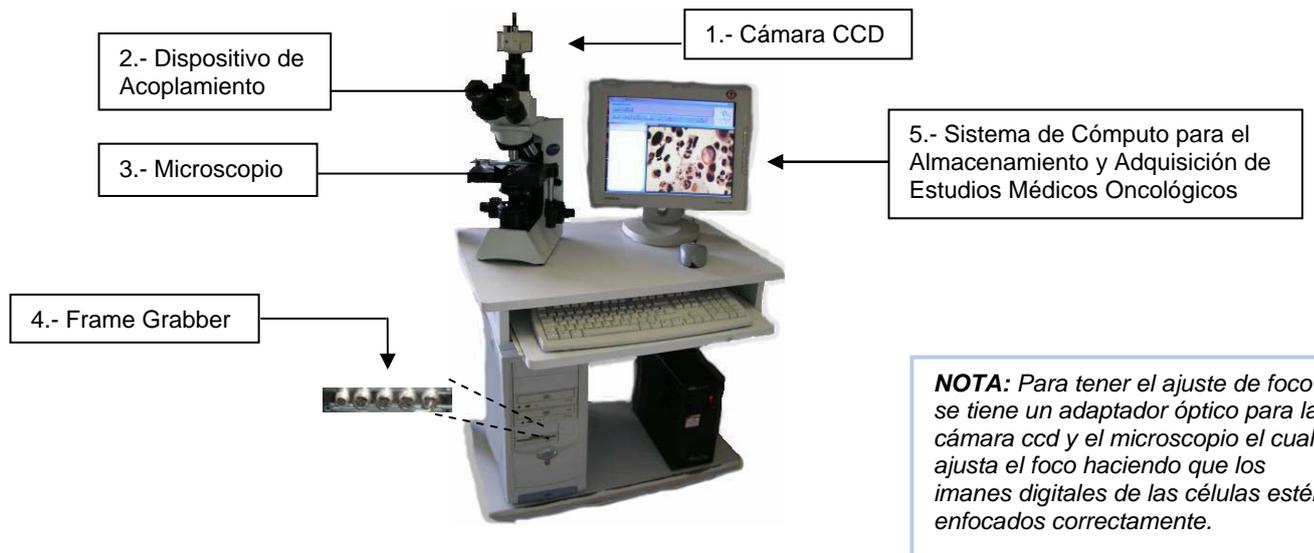


FIGURA 4.2a.- Sistema de Adquisición de imágenes

La imagen digital se captura a través de un medio electrónico y se representa como un archivo de información leído como una serie de impulsos electrónicos.

El principio de funcionamiento de cualquier cámara digital se divide básicamente en tres partes:

1.- Captura de imagen

La imagen se obtiene a través de distintos componentes. La parte óptica de la cámara se encarga de recoger la luz con la mayor precisión y calidad posible. Las cámaras digitales usan un elemento sensible a la luz que capta las imágenes, las variaciones de luz, colores y texturas traduciéndolas en pulsos eléctricos. El más común es el CCD (Charge Coupled Device) o dispositivos adaptadores de carga. El CCD sólo puede captar imágenes en blanco y negro, para producir los colores la imagen debe pasar por tres filtros (rojo, verde y azul).

Todas las cámaras incorporan un iris para ajustar la exposición y la ganancia eléctrica del CCD. Antes del CCD se antepone un filtro óptico para eliminar la parte del espectro no visible, lo que elimina el infrarrojo y el ultravioleta.

2.- Digitalización

Para digitalizar la imagen es necesario pasarla a un formato susceptible de ser comprimido y almacenado en cinta. Se realiza utilizando un convertidor analógico digital que consiste en un chip que toma muestras de la señal a intervalos fijos de tiempo (frecuencias de muestreo), a cada muestra se le asigna un valor dependiendo de su amplitud.

3.-Procesador de imagen

Consiste en efectos especiales tales como zoom, brillo, etc. El uso de una cámara de video digital para el presente trabajo de tesis es básicamente indispensable, ya que por medio de la cámara serán capturadas cada una de las imágenes que posteriormente se desplegarán en el sistema para el respectivo procesamiento y diagnóstico. Mediante el Frame Grabber es almacenada para despliegue. Es importante mencionar la herramienta de programación con la que cuenta el desarrollo que es el **Active X Imaging Source**, éste componente detecta periféricos de visión que pueden ser instalados en el hardware. Active X es el nombre que Microsoft ha dado a un grupo de tecnologías y herramientas “estratégicas” orientadas a objetos. Su principal tecnología es el Modelo de Objeto Componente (*Component Object Model, COM*).

Los controles OCX fueron creados para soportar documentos compuestos, como lo es el escritorio en un ambiente Windows, y son actualmente conocidos como Controles Active X. Dentro del ambiente de desarrollo del sistema fue necesaria la utilización de varios controles active X propios de la herramienta de desarrollo Delphi, sin embargo, fue indispensable agregar un componente más y es el llamado OCX de Imaging Source. El OCX de Imaging Source proporciona al sistema la capacidad de desplegar la imagen captada por la cámara y el Frame Grabber a través de cuadros que permiten a su vez la creación de videos, la toma de fotografías y más importante la tarea de almacenamiento a través de los cuadros almacenados en memoria. La siguiente figura 4.2b muestra el componente OCX de “Imaging Source” en el ambiente de desarrollo visual Delphi, así como sus propiedades

de configuración para su programación y utilización de las aplicaciones de visión.

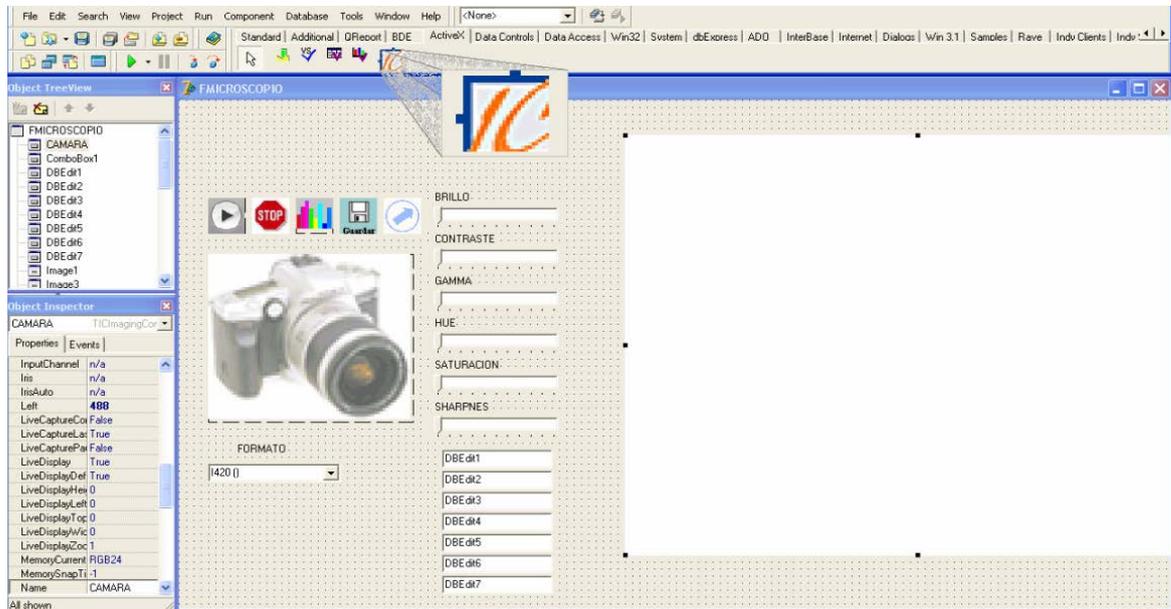


Figura 4.2b

Finalmente, conectando los diferentes dispositivos como muestra La figura 4.2a se tendrá el sistema para el almacenamiento y adquisición de estudios médicos oncológicos.

Conclusiones

Este sistema es el principio para integrar los actuales equipos de análisis citológico con sistemas de visión y cómputo de tecnología moderna, para que en un futuro se pueda: automatizar el sistema de exploración de laminillas porta-muestras, automatizar la imagen del colposcopio y finalmente integrarse a la red de datos e imágenes.

Así se contará con un equipo que de servicio citológico y patológico eficiente, confiable y seguro en el diagnóstico oncológico del personal femenino, reduciendo el tiempo de diagnóstico actual así como costo y dependencia tecnológica del extranjero.

Con este desarrollo se verán beneficiados los especialistas de las áreas citológica y patológica con una herramienta de apoyo (despliegue de datos procesados, ampliación digital, interface-microscopio-computadora) para obtener imagen en tiempo real y facilitar la observación y el diagnóstico, así mismo permitir reducir tiempos en la detección y diagnóstico del cáncer cérvico-uterino.

Permitirá a los médicos contar con una base de datos de células cérvico-uterinas caracterizada y clasificada para capacitación del personal de citología y patología, esta base es importante ya que a nivel mundial sólo los japoneses cuentan con una base caracterizada y el costo de adquisición es elevado.

El sistema puede aplicarse hacia la automatización de procesos industriales en los que una supervisión visual asegure la calidad de un producto, ya sea al comparar las imágenes capturadas con patrones almacenados previamente, o al reconocer patrones de operación fuera de norma. La ejecución de labores visuales con dispositivos automatizados puede ofrecer ventajas sobre su contraparte humana.

La visión de máquina no se fatiga, ni es discrecional, ni subjetiva, por lo que responderá siempre igual a una misma condición, así la eficacia del sistema dependerá sólo de los algoritmos programados y de la adecuada conjunción de componentes de hardware.

ANEXO A

program TESIS;

uses

```
Forms,  
UPRINCIPAL in 'UPRINCIPAL.pas' {FPRINCIPAL},  
UIDADMINISTRADOR in 'UIDADMINISTRADOR.pas' {FIDADMINISTRADOR},  
UIDMEDICO in 'UIDMEDICO.pas' {FIDMEDICO},  
UFBASE in 'UFBASE.pas' {FBASE},  
UBARRA in 'UBARRA.pas' {FBARRA},  
UCONTRASENA in 'UCONTRASENA.pas' {FCONTRASENA},  
UUSUARIOS in 'UUSUARIOS.pas' {FUSUARIOS},  
UMICROSCOPIO in 'UMICROSCOPIO.pas' {FMICROSCOPIO},  
UPACIENTES in 'UPACIENTES.pas' {FPACIENTES},  
UIMPRIMIRESTUDIO in 'UIMPRIMIRESTUDIO.pas' {FIMPRIMIRESTUDIO},  
UIMPRIMIRIMAGENES in 'UIMPRIMIRIMAGENES.pas' {FIMPRIMIRIMAGENES},  
UBUSQUEDAS in 'UBUSQUEDAS.pas' {FBUSQUEDAS},  
UESTUDIOS in 'UESTUDIOS.pas' {FESTUDIOS};
```

```
{ $R *.res }
```

begin

```
Application.Initialize;  
Application.CreateForm(TFPRINCIPAL, FPRINCIPAL);  
Application.CreateForm(TFIDADMINISTRADOR, FIDADMINISTRADOR);  
Application.CreateForm(TFIDMEDICO, FIDMEDICO);  
Application.CreateForm(TFBASE, FBASE);  
Application.CreateForm(TFBARRA, FBARRA);  
Application.CreateForm(TFCONTRASENA, FCONTRASENA);  
Application.CreateForm(TFUSUARIOS, FUSUARIOS);  
Application.CreateForm(TFMICROSCOPIO, FMICROSCOPIO);  
Application.CreateForm(TFPACIENTES, FPACIENTES);  
Application.CreateForm(TFIMPRIMIRESTUDIO, FIMPRIMIRESTUDIO);  
Application.CreateForm(TFIMPRIMIRIMAGENES, FIMPRIMIRIMAGENES);  
Application.CreateForm(TFBUSQUEDAS, FBUSQUEDAS);  
Application.CreateForm(TFESTUDIOS, FESTUDIOS);  
Application.Run;
```

end.

unit UBARRA;

//FDVDFSGDFG

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, jpeg, ExtCtrls, Menus, DBCtrls;

type

TFBARRA = class(TForm)

Label1: TLabel;

USUARIOS: TImage;

CONTRASENA: TImage;

ESTUDIOS: TImage;

SALIR: TImage;

Timer1: TTimer;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

CONTROL: TDBNavigator;

MICROSCOPIO: TImage;

Label5: TLabel;

Image1: TImage;

Image2: TImage;

Image3: TImage;

Image4: TImage;

Image5: TImage;

Image6: TImage;

Image7: TImage;

Image8: TImage;

Image9: TImage;

Image10: TImage;

Image11: TImage;

Image12: TImage;

Label6: TLabel;

Label7: TLabel;

Label8: TLabel;

Label9: TLabel;

Label10: TLabel;

Label11: TLabel;

Label12: TLabel;

Label13: TLabel;

Label14: TLabel;

Label15: TLabel;

Label16: TLabel;

Label17: TLabel;

Label18: TLabel;

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure FormShow(Sender: TObject);

procedure USUARIOSMouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);

procedure Timer1Timer(Sender: TObject);

procedure SALIRClick(Sender: TObject);

procedure CONTRASENAMouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);

procedure ESTUDIOSMouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);

```

procedure SALIRMouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
procedure CONTRASENAClick(Sender: TObject);
procedure USUARIOSClick(Sender: TObject);
procedure ESTUDIOSClick(Sender: TObject);
procedure MICROSCOPIOClick(Sender: TObject);
procedure CONTROLClick(Sender: TObject; Button: TNavigateBtn);
procedure FormCreate(Sender: TObject);
procedure Image1Click(Sender: TObject);
procedure Image2Click(Sender: TObject);
procedure Image3Click(Sender: TObject);
procedure Image1MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
procedure Image2MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
procedure Image3MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
procedure MICROSCOPIOMouseMove(Sender: TObject; Shift: TShiftState; X, Y:
Integer);
procedure Image9MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
procedure Image7MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
procedure Image8MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
procedure Image10MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
procedure Image4MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
procedure Image11MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
procedure Image12MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
procedure Image5MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
procedure Image6MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
procedure Image9Click(Sender: TObject);
procedure Image7Click(Sender: TObject);
procedure Image8Click(Sender: TObject);
procedure Image10Click(Sender: TObject);
procedure Image4Click(Sender: TObject);
procedure Image11Click(Sender: TObject);
procedure Image12Click(Sender: TObject);
procedure Image5Click(Sender: TObject);
procedure Image6Click(Sender: TObject);
private
  { Private declarations }
  procedure NiTeMenees(var m: TWMWINDOWPOSCHANGED); message
      WM_WINDOWPOSCHANGING ;
public
  { Public declarations }
  CONTADOR_IMAGENES:INTEGER;
  NUEVO:INTEGER;
  fecha,FECHADEREVICION:string;

end;

var
  FBARRA: TFBARRA;
  NUEVOUSUARIO:INTEGER;

implementation

uses UPRINCIPAL, UCONTRASENA, UUSUARIOS, UFBASE, UESTUDIOS, UMICROSCOPIO,
UPACIENTES, UIMPRIMIRESTUDIO, UIMPRIMIRIMAGENES, UBUSQUEDAS;

{$R *.dfm}

```

```

procedure TFBARRA.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  FPRINCIPAL.Enabled:=TRUE;
end;

procedure TFBARRA.FormShow(Sender: TObject);
begin
  LEFT:=0;
  TOP:=0;
end;

procedure TFBARRA.NiTeMenees(var m : TWMWINDOWPOSCHANGED);
begin
  m.windowpos.x := 0; {Left/Posicion X}
  m.windowpos.y := 0; {Top/Posicion y}
end;

procedure TFBARRA.USUARIOSMouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
begin
  label1.Visible:=true;
end;

procedure TFBARRA.Timer1Timer(Sender: TObject);
begin
  label1.Visible:=false;
  label2.Visible:=false;
  label3.Visible:=false;
  label4.Visible:=false;
  label6.Visible:=false;
  label7.Visible:=false;
  label8.Visible:=false;
  label9.Visible:=false;
  label10.Visible:=false;
  label11.Visible:=false;
  label12.Visible:=false;
  label13.Visible:=false;
  label14.Visible:=false;
  label15.Visible:=false;
  label16.Visible:=false;
  label17.Visible:=false;
  label18.Visible:=false;
  FPRINCIPAL.Label4.Visible:=FALSE;
end;

procedure TFBARRA.SALIRClick(Sender: TObject);
begin
  close;
  FUSUARIOS.Close;
  FESTUDIOS.CLOSE;
  FMICROSCOPIO.Close;
  FBUSQUEDAS.CLOSE;
  FPACIENTES.Close;

end;

```

```
procedure TFBARRA.CONTRASENAMouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
begin
    label2.Visible:=true;
end;
```

```
procedure TFBARRA.ESTUDIOSMouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
begin
    label3.Visible:=true;
end;
```

```
procedure TFBARRA.SALIRMouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
begin
    label4.Visible:=true;
end;
```

```
procedure TFBARRA.FormMouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
begin
    label1.Visible:=false;
    label2.Visible:=false;
    label3.Visible:=false;
    label4.Visible:=false;
    label6.Visible:=false;
    label7.Visible:=false;
    label8.Visible:=false;
    label9.Visible:=false;
    label10.Visible:=false;
    label11.Visible:=false;
    label12.Visible:=false;
    label13.Visible:=false;
    label14.Visible:=false;
    label15.Visible:=false;
    label16.Visible:=false;
    label17.Visible:=false;
    label18.Visible:=false;
end;
```

```
procedure TFBARRA.CONTRASENAClick(Sender: TObject);
begin
    FCONTRASENA.SHOW;
    FCONTRASENA.Edit1.Clear;
    FCONTRASENA.Edit2.Clear;
    FCONTRASENA.Edit3.Clear;
    FCONTRASENA.BitBtn1.Enabled:=FALSE;
    LABEL5.Caption:='CONTRASEÑA';

end;
```

```
procedure TFBARRA.USUARIOSClick(Sender: TObject);
begin
    FUSUARIOS.SHOW;
    CONTROL.DataSource:=FBASE.DUSUARIOS;
    LABEL5.Caption:='USUARIOS';
    FBUSQUEDAS.Label3.Caption:='USUARIOS';
end;
```

```

procedure TFBARRA.ESTUDIOSClick(Sender: TObject);
begin
    FESTUDIOS.SHOW;
    CONTROL.DataSource:=FBASE.DESTUDIOS;
    LABEL5.Caption:='ESTUDIOS';
    FBUSQUEDAS.Label3.Caption:='ESTUDIOS';
end;

procedure TFBARRA.MICROSCOPIOClick(Sender: TObject);
begin
    FMICROSCOPIO.SHOW;
    LABEL5.Caption:='MICROSCOPIO';
end;

procedure TFBARRA.CONTROLClick(Sender: TObject; Button: TNavigateBtn);
VAR
CONTADOR_ESTUDIOS:INTEGER;
USUARIO:STRING[4];
NUMERO_IMAGENES:INTEGER;

begin
case Button of
    nbInsert : BEGIN
        IF (LABEL5.Caption ='ESTUDIOS') THEN
            BEGIN
                FBASE.TUSUARIOS.Edit;
                USUARIO:=FESTUDIOS.DBEDIT17.Text;
                CONTADOR_ESTUDIOS:=STRTOINT(FESTUDIOS.DBEDIT16.Text);
                CONTADOR_ESTUDIOS:=CONTADOR_ESTUDIOS+1;
                FESTUDIOS.DBEDIT16.Text:=INTTOSTR(CONTADOR_ESTUDIOS);
                FESTUDIOS.EDIT1.Text:='A'+'+'+USUARIO+'-'+INTTOSTR(CONTADOR_ESTUDIOS);
                CONTADOR_ESTUDIOS:=STRTOINT(FESTUDIOS.DBEDIT16.Text);
                FESTUDIOS.DBEDIT1.Text:=FESTUDIOS.Edit1.Text;
                FESTUDIOS.DBEDIT2.Text:=DATETOSTR(DATE);
                FBASE.TUSUARIOS.Post;

                FESTUDIOS.DBEdit19.Text:=TRIM(FBASE.TUSUARIOS.FieldByName('RFC').ASSTRING);
                END;
            IF (LABEL5.Caption ='USUARIOS') THEN
                BEGIN
                    NUEVOUSUARIO:=1;
                    FUSUARIOS.DBEdit1.ReadOnly:=FALSE;
                    FUSUARIOS.DBEdit2.ReadOnly:=FALSE;
                    FUSUARIOS.DBEdit3.ReadOnly:=FALSE;
                    FUSUARIOS.DBEdit4.ReadOnly:=FALSE;
                    FUSUARIOS.DBEdit5.ReadOnly:=FALSE;
                    FUSUARIOS.DBEdit6.ReadOnly:=FALSE;
                END;
            END;
        NBPOST : BEGIN
            IF (LABEL5.Caption ='USUARIOS') THEN
                BEGIN
                    NUEVOUSUARIO:=0;
                    FUSUARIOS.DBEdit1.ReadOnly:=TRUE;
                    FUSUARIOS.DBEdit2.ReadOnly:=TRUE;
                    FUSUARIOS.DBEdit3.ReadOnly:=TRUE;
                    FUSUARIOS.DBEdit4.ReadOnly:=TRUE;
                END;
            END;
        END;
    end;
end;

```

```

        FUSUARIOS.DBEdit5.ReadOnly:=TRUE;
        FUSUARIOS.DBEdit6.ReadOnly:=TRUE;

    IF (CONTADOR <> 0 )THEN
    BEGIN
        APPLICATION.MESSAGEBOX('LOGIN NO VALIDO','ERROR',MB_OK);
    END;
    END;
    END;
NBEDIT : BEGIN
    IF (LABEL5.Caption ='USUARIOS') THEN
    BEGIN
        NUEVOUSUARIO:=1;
        FUSUARIOS.DBEdit1.ReadOnly:=FALSE;
        FUSUARIOS.DBEdit2.ReadOnly:=FALSE;
        FUSUARIOS.DBEdit3.ReadOnly:=FALSE;
        FUSUARIOS.DBEdit4.ReadOnly:=FALSE;
        FUSUARIOS.DBEdit5.ReadOnly:=FALSE;
        FUSUARIOS.DBEdit6.ReadOnly:=FALSE;
    END;
    END;
NBCANCEL :BEGIN
    IF (LABEL5.Caption ='USUARIOS') THEN
    BEGIN
        NUEVOUSUARIO:=0;
        FUSUARIOS.DBEdit1.ReadOnly:=TRUE;
        FUSUARIOS.DBEdit2.ReadOnly:=TRUE;
        FUSUARIOS.DBEdit3.ReadOnly:=TRUE;
        FUSUARIOS.DBEdit4.ReadOnly:=TRUE;
        FUSUARIOS.DBEdit5.ReadOnly:=TRUE;
        FUSUARIOS.DBEdit6.ReadOnly:=TRUE;
    END;
    END;
    END;
end;

procedure TFBARRA.FormCreate(Sender: TObject);
begin
    NUEVO:=0;
    top:=0;
    left:=0;
    clientheight:=70;
    clientwidth:=1270;
end;

procedure TFBARRA.Image1Click(Sender: TObject);
begin
    FPACIENTES.SHOW;
    CONTROL.DataSource:=FBASE.DPACIENTES;
    LABEL5.Caption:='PACIENTES';
    FBUSQUEDAS.Label3.Caption:='PACIENTES';
end;

procedure TFBARRA.Image2Click(Sender: TObject);
begin
    //IMPRIMIR IMAGENES
    IF FESTUDIOS.TabbedNotebook1.ActivePage= 'IMAGENES' THEN

```

```

BEGIN
    fecha:= FormatDateTime("Mexico, D.F." dddd, d "de" mmmm "de" yyyy, hh:mm
AM/PM', Now );
    FIMPRIMIRIMAGENES.QRLabel11.Caption:=fecha;
    FECHADEREVICION:= FormatDateTime('DDDD, d "de" MMMM "de" yyyy',
STRTOTYPE(FESTUDIOS.DBEdit2.Text));
    FIMPRIMIRIMAGENES.QRLabel13.Caption:=FECHADEREVICION;
    FIMPRIMIRIMAGENES.REPORTE.Preview;

END;

//IMPRIMIR ESTUDIOS
ELSE
BEGIN
    fecha:= FormatDateTime("Mexico, D.F." dddd, d "de" mmmm "de" yyyy, hh:mm
AM/PM', Now );
    FIMPRIMIRESTUDIO.QRLabel25.Caption:=fecha;
    FIMPRIMIRESTUDIO.QRLabel1.Caption:=FESTUDIOS.LNOMBRE.Caption;
    FIMPRIMIRESTUDIO.QRLabel2.Caption:=FESTUDIOS.LEDAD.Caption;
    FECHADEREVICION:= FormatDateTime('DDDD, d "de" MMMM "de" yyyy',
STRTOTYPE(FESTUDIOS.DBEdit2.Text));
    FIMPRIMIRESTUDIO.QRLabel4.Caption:=FECHADEREVICION;
    FIMPRIMIRESTUDIO.REPORTE.Preview;

END;
end;

procedure TFBARRA.Image3Click(Sender: TObject);
begin
    FBUSQUEDAS.SHOW;
    FBUSQUEDAS.MEMO1.Lines.Clear;
    FBUSQUEDAS.MEMO2.Lines.Clear;
    FBUSQUEDAS.EDIT1.Clear;
    FBUSQUEDAS.ComboBox1.Clear;

//BUSQUEDA PACIENTES

IF FBUSQUEDAS.LABEL3.Caption = 'PACIENTES' THEN
BEGIN
    FBUSQUEDAS.ComboBox1.Text:='NOMBRE';
    FBUSQUEDAS.ComboBox1.Items.Add('NOMBRE');
    FBUSQUEDAS.ComboBox1.Items.Add('APELLIDO_PATERNO');
    FBUSQUEDAS.ComboBox1.Items.Add('APELLIDO_MATERNO');
    FBUSQUEDAS.ComboBox1.Items.Add('RFC');
    FBUSQUEDAS.ComboBox1.Items.Add('NUMERO');
END;

//BUSQUEDA USUARIOS

IF FBUSQUEDAS.LABEL3.Caption = 'USUARIOS' THEN
BEGIN
    FBUSQUEDAS.ComboBox1.Text:='NOMBRE';
    FBUSQUEDAS.ComboBox1.Items.Add('NOMBRE');
    FBUSQUEDAS.ComboBox1.Items.Add('APELLIDO_PATERNO');
    FBUSQUEDAS.ComboBox1.Items.Add('APELLIDO_MATERNO');
    FBUSQUEDAS.ComboBox1.Items.Add('RFC');

```

```

    FBUSQUEDAS.ComboBox1.Items.Add('NUMERO');
END;

//BUSQUEDA ESTUDIOS

IF FBUSQUEDAS.LABEL3.Caption = 'ESTUDIOS' THEN
BEGIN
    FBUSQUEDAS.ComboBox1.Text:='NUMERO';
    FBUSQUEDAS.ComboBox1.Items.Add('RFC_PACIENTE');
    FBUSQUEDAS.ComboBox1.Items.Add('FECHA');
    FBUSQUEDAS.ComboBox1.Items.Add('NUMERO');
END;

end;

procedure TFBARRA.Image1MouseMove(Sender: TObject; Shift: TShiftState; X, Y:
Integer);
begin
    label6.Visible:=true;
end;

procedure TFBARRA.Image2MouseMove(Sender: TObject; Shift: TShiftState; X, Y:
Integer);
begin
    label7.Visible:=true;
end;

procedure TFBARRA.Image3MouseMove(Sender: TObject; Shift: TShiftState; X, Y:
Integer);
begin
    label8.Visible:=true;
end;

procedure TFBARRA.MICROSCOPIOMouseMove(Sender: TObject; Shift: TShiftState; X, Y:
Integer);
begin
    label9.Visible:=true;
end;

procedure TFBARRA.Image9MouseMove(Sender: TObject; Shift: TShiftState; X, Y:
Integer);
begin
    label10.Visible:=true;
end;

procedure TFBARRA.Image7MouseMove(Sender: TObject; Shift: TShiftState; X, Y:
Integer);
begin
    label11.Visible:=true;
end;

procedure TFBARRA.Image8MouseMove(Sender: TObject; Shift: TShiftState; X, Y:
Integer);
begin
    label12.Visible:=true;
end;

```

```
procedure TFBARRA.Image10MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
begin
  label13.Visible:=true;
end;
```

```
procedure TFBARRA.Image4MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
begin
  label14.Visible:=true;
end;
```

```
procedure TFBARRA.Image11MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
begin
  label15.Visible:=true;
end;
```

```
procedure TFBARRA.Image12MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
begin
  label16.Visible:=true;
end;
```

```
procedure TFBARRA.Image5MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
begin
  label17.Visible:=true;
end;
```

```
procedure TFBARRA.Image6MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
begin
  label18.Visible:=true;
end;
```

```
procedure TFBARRA.Image9Click(Sender: TObject);
begin
  control.BtnClick(nbfirst);
end;
```

```
procedure TFBARRA.Image7Click(Sender: TObject);
begin
  control.BtnClick(nbprior);
end;
```

```
procedure TFBARRA.Image8Click(Sender: TObject);
begin
  control.BtnClick(nbnext);
end;
```

```
procedure TFBARRA.Image10Click(Sender: TObject);
begin
  control.BtnClick(nblast);
end;
```

```
procedure TFBARRA.Image4Click(Sender: TObject);
```

```

begin
  control.BtnClick(nbinsert);
end;

procedure TFBARRA.Image11Click(Sender: TObject);
begin
  control.BtnClick(nbdelete);
end;

procedure TFBARRA.Image12Click(Sender: TObject);
begin
  control.BtnClick(nbedit);
end;

procedure TFBARRA.Image5Click(Sender: TObject);
begin
  control.BtnClick(nbpost);
end;

procedure TFBARRA.Image6Click(Sender: TObject);
begin
  control.BtnClick(nbcancel);
end;

end.

```

unit UBUSQUEDAS;

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, jpeg, ExtCtrls;
```

```
type
```

```
TFBUSQUEDAS = class(TForm)
```

```
  Label1: TLabel;
```

```
  Label2: TLabel;
```

```
  Bevel1: TBevel;
```

```
  Bevel2: TBevel;
```

```
  ComboBox1: TComboBox;
```

```
  Edit1: TEdit;
```

```
  Memo1: TMemo;
```

```
  Memo2: TMemo;
```

```
  Button1: TButton;
```

```
  Label3: TLabel;
```

```
  procedure FormCreate(Sender: TObject);
```

```
  procedure Edit1Change(Sender: TObject);
```

```
  procedure Button1Click(Sender: TObject);
```

```
  procedure Memo1Click(Sender: TObject);
```

```
  procedure Memo2Click(Sender: TObject);
```

```
  procedure FormShow(Sender: TObject);
```

```

private
  { Private declarations }
  procedure NiTeMenees(var m: TWMWINDOWPOSchANGED); message
    WM_WINDOWPOSchANGING ;

public
  { Public declarations }
  LINEA:INTEGER;
end;

var
  FBUSQUEDAS: TFBUSQUEDAS;

implementation

uses UFBASE, UPACIENTES, UUSUARIOS, UESTUDIOS;

{$R *.dfm}

procedure TFBUSQUEDAS.FormCreate(Sender: TObject);
begin
  top:=102;
  left:=0;
  clientheight:=600;
  clientwidth:=1270;
end;

procedure TFBUSQUEDAS.Edit1Change(Sender: TObject);
VAR
  BUSCAR,NOMBRE,PATERNO,MATERNO,RFC,TIPO,EDIT,FECHA,NUMEROESTUDIO:STRING;
  NUMERO:INTEGER;
begin
  NUMERO:=0;
  MEMO1.Lines.Clear;
  MEMO2.Lines.Clear;

  IF LABEL3.Caption = 'PACIENTES' THEN
  BEGIN

    FBASE.TPACIENTES.FIRST;
    WHILE NOT (FBASE.TPACIENTES.Eof) AND (EDIT1.Text<>")DO
      BEGIN

        BUSCAR:=TRIM(FBASE.TPACIENTES.FIELDByNAME(COMBOBOX1.Text).AsString);
        NOMBRE:=TRIM(FBASE.TPACIENTES.FIELDByNAME('NOMBRE').AsString);

        PATERNO:=TRIM(FBASE.TPACIENTES.FIELDByNAME('APELLIDO_PATERNO').AsString);

        MATERNO:=TRIM(FBASE.TPACIENTES.FIELDByNAME('APELLIDO_MATERNO').AsString);
        RFC:=TRIM(FBASE.TPACIENTES.FIELDByNAME('RFC').AsString);

        IF (EDIT1.Text[1]=BUSCAR[1]) AND (AnsiPOS(EDIT1.Text,BUSCAR) <> 0) THEN
          BEGIN
            NUMERO:=NUMERO+1;
            MEMO1.Lines.Add(RFC+#13);
            MEMO2.Lines.Add(INTTOSTR(NUMERO)+' '+NOMBRE+' '+PATERNO+'
'+MATERNO+#13)

```

```

        END;

        FBASE.TPACIENTES.Next;
    END;
END;

//BUSQUEDA USUARIOS

IF LABEL3.Caption = 'USUARIOS' THEN
BEGIN

    FBASE.TUSUARIOS.FIRST;
    WHILE NOT (FBASE.TUSUARIOS.Eof) AND (EDIT1.Text<>")DO

        BEGIN
            BUSCAR:=TRIM(FBASE.TUSUARIOS.FIELDBYNAME(COMBOBOX1.Text).AsString);
            NOMBRE:=TRIM(FBASE.TUSUARIOS.FIELDBYNAME('NOMBRE').AsString);
            PATERNO:=TRIM(FBASE.TUSUARIOS.FIELDBYNAME('APELLIDO_PATERNO').AsString);
            MATERNO:=TRIM(FBASE.TUSUARIOS.FIELDBYNAME('APELLIDO_MATERNO').AsString);
            RFC:=TRIM(FBASE.TUSUARIOS.FIELDBYNAME('RFC').AsString);

            IF (EDIT1.Text[1]=BUSCAR[1]) AND (AnsiPOS(EDIT1.Text,BUSCAR) <> 0) THEN
                BEGIN
                    NUMERO:=NUMERO+1;
                    MEMO1.Lines.Add(RFC+#13);
                    MEMO2.Lines.Add(INTTOSTR(NUMERO)+' '+NOMBRE+' '+PATERNO+'
'+MATERNO+#13)

                END;

            FBASE.TUSUARIOS.Next;
        END;
    END;

//BUSQUEDA ESTUDIOS

IF LABEL3.Caption = 'ESTUDIOS' THEN
BEGIN

    FBASE.TESTUDIOS.First;

    WHILE NOT (FBASE.TESTUDIOS.Eof) AND (EDIT1.Text<>")DO
        BEGIN

            BUSCAR:=TRIM(FBASE.TESTUDIOS.FIELDBYNAME(COMBOBOX1.Text).AsString);
            NUMEROESTUDIO:=TRIM(FBASE.TESTUDIOS.FIELDBYNAME('NUMERO').AsString);
            RFC:=TRIM(FBASE.TESTUDIOS.FIELDBYNAME('RFC_PACIENTE').AsString);
            FECHA:=TRIM(FBASE.TESTUDIOS.FIELDBYNAME('FECHA').AsString);

            IF (EDIT1.Text[1]=BUSCAR[1]) AND (AnsiPOS(EDIT1.Text,BUSCAR) <> 0)THEN
                BEGIN
                    NUMERO:=NUMERO+1;
                    FBASE.TPACIENTES.Locate('RFC',RFC,[]);
                    NOMBRE:=TRIM(FBASE.TPACIENTES.FIELDBYNAME('NOMBRE').AsString);

                    PATERNO:=TRIM(FBASE.TPACIENTES.FIELDBYNAME('APELLIDO_PATERNO').AsString);

```

```

MATERNO:=TRIM(FBASE.TPACIENTES.FIELDBYNAME('APELLIDO_MATERNO').AsString);
    MEMO1.Lines.Add(NUMEROESTUDIO+#13);
    MEMO2.Lines.Add(INTTOSTR(NUMERO)+FECHA+' '+RFC+' '+NOMBRE+'
'+PATERNO+' '+MATERNO+' '+#13);
    END;
    IF (BUSCAR[3]='/') AND (AnsiPOS(EDIT1.Text,BUSCAR) <> 0)THEN
    BEGIN
        NUMERO:=NUMERO+1;
        FBASE.TPACIENTES.Locate('RFC',RFC,[]);
        NOMBRE:=TRIM(FBASE.TPACIENTES.FIELDBYNAME('NOMBRE').AsString);

PATERNO:=TRIM(FBASE.TPACIENTES.FIELDBYNAME('APELLIDO_PATERNO').AsString);

MATERNO:=TRIM(FBASE.TPACIENTES.FIELDBYNAME('APELLIDO_MATERNO').AsString);
    MEMO1.Lines.Add(NUMEROESTUDIO+#13);
    MEMO2.Lines.Add(INTTOSTR(NUMERO)+FECHA+' '+RFC+' '+NOMBRE+' '+PATERNO+'
'+MATERNO+' '+#13);
    END;

    FBASE.TESTUDIOS.Next;
    END;
END;

end;

procedure TFBUSQUEDAS.Button1Click(Sender: TObject);
VAR
    BUSCAR,NOMBRE,PATERNO,MATERNO,RFC,TIPO,EDIT,NUMEROESTUDIO,FECHA:STRING;
    NUMERO:INTEGER;
begin
    NUMERO:=0;
    MEMO1.Lines.Clear;
    MEMO2.Lines.Clear;

    IF LABEL3.Caption = 'PACIENTES' THEN
    BEGIN
        FBASE.TPACIENTES.FIRST;
        WHILE NOT (FBASE.TPACIENTES.Eof)DO
        BEGIN

            BUSCAR:=TRIM(FBASE.TPACIENTES.FIELDBYNAME(COMBOBOX1.Text).AsString);
            NOMBRE:=TRIM(FBASE.TPACIENTES.FIELDBYNAME('NOMBRE').AsString);
            PATERNO:=TRIM(FBASE.TPACIENTES.FIELDBYNAME('APELLIDO_PATERNO').AsString);

MATERNO:=TRIM(FBASE.TPACIENTES.FIELDBYNAME('APELLIDO_MATERNO').AsString);
            RFC:=TRIM(FBASE.TPACIENTES.FIELDBYNAME('RFC').AsString);
            NUMERO:=NUMERO+1;
            MEMO1.Lines.Add(RFC+#13);
            MEMO2.Lines.Add(INTTOSTR(NUMERO)+' '+NOMBRE+' '+PATERNO+'
'+MATERNO+#13);

            FBASE.TPACIENTES.Next;
        END;
    END;
END;

```

//BUSQUEDA USUARIOS

```
IF LABEL3.Caption = 'USUARIOS' THEN
BEGIN
FBASE.TUSUARIOS.FIRST;
WHILE NOT (FBASE.TUSUARIOS.Eof)DO
    BEGIN

        BUSCAR:=TRIM(FBASE.TUSUARIOS.FIELDBYNAME(COMBOBOX1.Text).AsString);
        NOMBRE:=TRIM(FBASE.TUSUARIOS.FIELDBYNAME('NOMBRE').AsString);
        PATERNO:=TRIM(FBASE.TUSUARIOS.FIELDBYNAME('APELLIDO_PATERNO').AsString);
        MATERNO:=TRIM(FBASE.TUSUARIOS.FIELDBYNAME('APELLIDO_MATERNO').AsString);
        RFC:=TRIM(FBASE.TUSUARIOS.FIELDBYNAME('RFC').AsString);
        NUMERO:=NUMERO+1;
        MEMO1.Lines.Add(RFC+#13);
        MEMO2.Lines.Add(INTOSTR(NUMERO)+' '+NOMBRE+' '+PATERNO+'
'+MATERNO+#13);

        FBASE.TUSUARIOS.Next;
    END;
END;
```

//BUSQUEDA ESTUDIOS

```
IF LABEL3.Caption = 'ESTUDIOS' THEN
BEGIN

FBASE.TESTUDIOS.First;

    WHILE NOT (FBASE.TESTUDIOS.Eof)DO
        BEGIN

            NUMEROESTUDIO:=TRIM(FBASE.TESTUDIOS.FIELDBYNAME('NUMERO').AsString);
            RFC:=TRIM(FBASE.TESTUDIOS.FIELDBYNAME('RFC_PACIENTE').AsString);
            FECHA:=TRIM(FBASE.TESTUDIOS.FIELDBYNAME('FECHA').AsString);
            NUMERO:=NUMERO+1;
            FBASE.TPACIENTES.Locate('RFC',RFC,[]);
            NOMBRE:=TRIM(FBASE.TPACIENTES.FIELDBYNAME('NOMBRE').AsString);
            PATERNO:=TRIM(FBASE.TPACIENTES.FIELDBYNAME('APELLIDO_PATERNO').AsString);

            MATERNO:=TRIM(FBASE.TPACIENTES.FIELDBYNAME('APELLIDO_MATERNO').AsString);
            MEMO1.Lines.Add(NUMEROESTUDIO+#13);
            MEMO2.Lines.Add(INTOSTR(NUMERO)+FECHA+' '+RFC+' '+NOMBRE+' '+PATERNO+'
'+MATERNO+" "+#13);

            FBASE.TESTUDIOS.Next;
        END;
    END;

end;
```

```
procedure TFBUSQUEDAS.Memo1Click(Sender: TObject);
begin
LINEA:=MEMO1.CaretPos.Y;
IF LABEL3.Caption = 'PACIENTES' THEN
```

```

BEGIN
FBASE.TPACIENTES.Locate('RFC',MEMO1.Lines[LINEA],[]);
FPACIENTES.Show;
END;
// BUSQUEDA USUARIOS
IF LABEL3.Caption = 'USUARIOS' THEN
BEGIN
FBASE.TUSUARIOS.Locate('RFC',MEMO1.Lines[LINEA],[]);
FUSUARIOS.Show;
END;

// BUSQUEDA ESTUDIOS

IF LABEL3.Caption = 'ESTUDIOS' THEN
BEGIN
FBASE.TESTUDIOS.Locate('NUMERO',MEMO1.Lines[LINEA],[]);
FESTUDIOS.Show;
END;

end;

procedure TFBUSQUEDAS.Memo2Click(Sender: TObject);
begin
LINEA:=MEMO2.CaretPos.Y;
IF LABEL3.Caption = 'PACIENTES' THEN
BEGIN
FBASE.TPACIENTES.Locate('RFC',MEMO1.Lines[LINEA],[]);
FPACIENTES.Show;
END;

//BUSQUEDA USUARIOS
IF LABEL3.Caption = 'USUARIOS' THEN
BEGIN
FBASE.TUSUARIOS.Locate('RFC',MEMO1.Lines[LINEA],[]);
FUSUARIOS.Show;
END;

// BUSQUEDA ESTUDIOS
IF LABEL3.Caption = 'ESTUDIOS' THEN
BEGIN
FBASE.TESTUDIOS.Locate('NUMERO',MEMO1.Lines[LINEA],[]);
FESTUDIOS.Show;
END;

end;

procedure TFBUSQUEDAS.FormShow(Sender: TObject);
begin
MEMO1.Lines.Clear;
MEMO2.Lines.Clear;
EDIT1.Clear;
end;
procedure TFBUSQUEDAS.NiTeMenees(var m : TWMWINDOWPOSCHANGED);
begin
m.windowpos.x := 0; {Left/Posicion X}

```

```

        m.windowpos.y := 102; {Top/Posicion y}
    end;
end.

```

```

*****
*****

```

unit UCONTRASENA;

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Buttons, ExtCtrls, DBCtrls, Mask;
```

```
type
```

```

TFCONTRASENA = class(TForm)
    Label1: TLabel;
    DBEdit1: TDBEdit;
    DBNavigator1: TDBNavigator;
    BitBtn1: TBitBtn;
    Edit1: TEdit;
    Edit2: TEdit;
    Label2: TLabel;
    Label3: TLabel;
    Edit3: TEdit;
    BitBtn2: TBitBtn;
    procedure FormShow(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure DBNavigator1Click(Sender: TObject; Button: TNavigateBtn);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure BitBtn2Click(Sender: TObject);
    procedure Edit1Change(Sender: TObject);
    procedure Edit2Change(Sender: TObject);
    procedure Edit3Change(Sender: TObject);
    procedure FormCreate(Sender: TObject);
private
    { Private declarations }
    procedure NiTeMenees(var m: TWMWINDOWPOSCHANGED); message
        WM_WINDOWPOSCHANGING ;
public
    { Public declarations }
end;

```

```
var
```

```

    FCONTRASENA: TFCONTRASENA;
    ACTUAL,ANTERIOR:STRING;

```

```
implementation
```

```
uses UFBASE, UBARRA;
```

```
{$R *.dfm}
```

```
procedure TFCONTRASENA.FormShow(Sender: TObject);
```

```

begin
  LEFT:=500;
  TOP:=350;
  ACTUAL:=DBEDIT1.Text;
  FBARRA.Enabled:=FALSE;

end;

procedure TFCONTRASENA.NiTeMenees(var m : TWMWINDOWPOSCHANGED);
begin
  m.windowpos.x := 500; {Left/Posicion X}
  m.windowpos.y := 350; {Top/Posicion y}
end;

procedure TFCONTRASENA.BitBtn1Click(Sender: TObject);
begin
  IF (ACTUAL = EDIT1.Text) AND (EDIT2.Text = EDIT3.Text) THEN
  BEGIN
    FBASE.TCONTRASENA.Edit;
    DBEDIT1.Text:=EDIT2.Text;
    DBNAVIGATOR1.BtnClick(NBPOST);
  END
  ELSE
  BEGIN
    APPLICATION.MESSAGEBOX('DATOS NO VALIDOS NO VALIDOS','ERROR',MB_OK);
  END;
  CLOSE;

end;

procedure TFCONTRASENA.DBNavigator1Click(Sender: TObject;
  Button: TNavigateBtn);
begin
  CLOSE;
end;

procedure TFCONTRASENA.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  FBARRA.Enabled:=TRUE;
end;

procedure TFCONTRASENA.BitBtn2Click(Sender: TObject);
begin
  CLOSE;
end;

procedure TFCONTRASENA.Edit1Change(Sender: TObject);
begin
  BITBTN1.Enabled:=TRUE;
end;

procedure TFCONTRASENA.Edit2Change(Sender: TObject);
begin
  BITBTN1.Enabled:=TRUE;
end;

```

```
procedure TFCONTRASENA.Edit3Change(Sender: TObject);
begin
  BITBTN1.Enabled:=TRUE;
end;
```

```
procedure TFCONTRASENA.FormCreate(Sender: TObject);
begin
  top:=350;
  left:=500;
  clientheight:=170;
  clientwidth:=400;
end;

end.
```

unit UESTUDIOS;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, DBCtrls, ExtCtrls, Mask, ComCtrls, TabNotBk;

type

```
TFESTUDIOS = class(TForm)
  TabbedNotebook1: TTabbedNotebook;
  Label3: TLabel;
  Label4: TLabel;
  Label5: TLabel;
  Label6: TLabel;
  Label7: TLabel;
  Label8: TLabel;
  LNOMBRE: TLabel;
  Label10: TLabel;
  Label1: TLabel;
  Label26: TLabel;
  Label27: TLabel;
  LEDAD: TLabel;
  DBEdit1: TDBEdit;
  DBEdit2: TDBEdit;
  DBEdit3: TDBEdit;
  DBMemo1: TDBMemo;
  DBMemo2: TDBMemo;
  DBMemo3: TDBMemo;
  DBEdit4: TDBEdit;
  DBEdit15: TDBEdit;
  Label2: TLabel;
  Label9: TLabel;
  Label11: TLabel;
  Label12: TLabel;
  Label13: TLabel;
  Label14: TLabel;
  Label15: TLabel;
```

```

Label16: TLabel;
DBEdit5: TDBEdit;
DBEdit6: TDBEdit;
DBEdit7: TDBEdit;
DBEdit8: TDBEdit;
DBEdit9: TDBEdit;
DBEdit10: TDBEdit;
DBEdit11: TDBEdit;
DBMemo4: TDBMemo;
DBCheckBox1: TDBCheckBox;
DBCheckBox2: TDBCheckBox;
DBCheckBox3: TDBCheckBox;
DBCheckBox4: TDBCheckBox;
DBCheckBox5: TDBCheckBox;
DBCheckBox6: TDBCheckBox;
DBCheckBox7: TDBCheckBox;
DBCheckBox8: TDBCheckBox;
DBCheckBox9: TDBCheckBox;
DBCheckBox10: TDBCheckBox;
Label17: TLabel;
Label18: TLabel;
Label19: TLabel;
Label20: TLabel;
Label21: TLabel;
Label22: TLabel;
Label23: TLabel;
Label24: TLabel;
Label25: TLabel;
DBImage1: TDBImage;
DBEdit12: TDBEdit;
DBEdit13: TDBEdit;
DBEdit14: TDBEdit;
DBNavigator1: TDBNavigator;
DBMemo5: TDBMemo;
Edit1: TEdit;
DBEdit16: TDBEdit;
DBEdit17: TDBEdit;
DBEdit18: TDBEdit;
DBEdit19: TDBEdit;
procedure DBEdit1Change(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure DBEdit15Change(Sender: TObject);
procedure FormCreate(Sender: TObject);
private
  { Private declarations }
  procedure NiTeMenees(var m: TWMWINDOWPOSCHANGED); message
    WM_WINDOWPOSCHANGING ;
public
  { Public declarations }
end;

var
  FESTUDIOS: TFESTUDIOS;

implementation

uses UFBASE, UBUSQUEDAS, UBARRA;

```

{\$R *.dfm}

```
procedure TFESTUDIOS.DBEdit1Change(Sender: TObject);
begin
  FBASE.TIMAGENES.Filter:='NUMERO_DE_ESTUDIO =' + #39 + DBEDIT1.Text + #39;
  FBASE.TIMAGENES.Filtered:=TRUE;
end;
```

```
procedure TFESTUDIOS.FormShow(Sender: TObject);
begin
  FBARRA.LABEL5.Caption:='ESTUDIOS';
  FBUSQUEDAS.Label3.Caption:='ESTUDIOS';
  FBARRA.CONTROL.DataSource:=FBASE.DESTUDIOS;
end;
```

```
procedure TFESTUDIOS.DBEdit15Change(Sender: TObject);
VAR
  EDAD,BUSCAR,NOMBRE,PATERNO,MATERNO,RFC:STRING;
  ANOS:DOUBLE;
begin
  NOMBRE:=TRIM(FBASE.TPACIENTES.FIELDBYNAME('NOMBRE').AsString);
  PATERNO:=TRIM(FBASE.TPACIENTES.FIELDBYNAME('APELLIDO_PATERNO').AsString);
  MATERNO:=TRIM(FBASE.TPACIENTES.FIELDBYNAME('APELLIDO_MATERNO').AsString);
  LNOMBRE.CAPTION:=NOMBRE+' '+PATERNO+' '+MATERNO;

  IF (DBEDIT15.Text<>'') THEN
  BEGIN
    EDAD:=TRIM(FBASE.TPACIENTES.FIELDBYNAME('FECHA_DE_NACIMIENTO').AsString);
    ANOS:=trunc((date - strtodate(EDAD))/365.25);
    LEDAD.Caption:=floattostr(ANOS)
  END
  ELSE
    LEDAD.Caption:='0';
end;
```

```
procedure TFESTUDIOS.FormCreate(Sender: TObject);
begin
  top:=102;
  left:=0;
  clientheight:=600;
  clientwidth:=1270;
end;
procedure TFESTUDIOS.NiTeMenees(var m : TWMWINDOWPOSCHANGED);
begin
  m.windowpos.x := 0; {Left/Posicion X}
  m.windowpos.y := 102; {Top/Posicion y}
end;
end.
```

unit UFBASE;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, DB, DBTables;

type

```
TFBASE = class(TForm)
  TCONTRASENA: TTable;
  TESTUDIOS: TTable;
  TIMAGENES: TTable;
  TPACIENTES: TTable;
  TUSUARIOS: TTable;
  DCONTRASENA: TDataSource;
  DESTUDIOS: TDataSource;
  DIMAGENES: TDataSource;
  DPACIENTES: TDataSource;
  DUSUARIOS: TDataSource;
  TUSUARIOS1: TTable;
  procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

var

FBASE: TFBASE;

implementation

{ \$R *.dfm }

procedure TFBASE.FormCreate(Sender: TObject);

begin

```
TCONTRASENA.Active:=TRUE;
TESTUDIOS.Active:=TRUE;
TIMAGENES.Active:=TRUE;
TPACIENTES.Active:=TRUE;
TUSUARIOS.Active:=TRUE;
TUSUARIOS1.Active:=TRUE;
```

end;

end.

unit UIDADMINISTRADOR;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Buttons;

```

type
  TFIDADMINISTRADOR = class(TForm)
    Label1: TLabel;
    Edit1: TEdit;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    procedure BitBtn2Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure Edit1Change(Sender: TObject);
    procedure Edit1DbClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
    procedure NiTeMenees(var m: TWMWINDOWPOSCHANGED); message
      WM_WINDOWPOSCHANGING ;
  public
    { Public declarations }
  end;

var
  FIDADMINISTRADOR: TFIDADMINISTRADOR;
  CONTRASENA:STRING;
  PASSWORD:STRING;

implementation

uses UPRINCIPAL, UFBASE, UBARRA, UBUSQUEDAS, UESTUDIOS;

{$R *.dfm}

procedure TFIDADMINISTRADOR.BitBtn2Click(Sender: TObject);
begin
  FPRINCIPAL.Enabled:=TRUE;
  CLOSE;
end;

procedure TFIDADMINISTRADOR.FormShow(Sender: TObject);
begin
  FPRINCIPAL.Enabled:=FALSE;
  PASSWORD:=TRIM(FBASE.TCONTRASENA.FieldByName('PASSWORD').ASSTRING);
end;

procedure TFIDADMINISTRADOR.NiTeMenees(var m : TWMWINDOWPOSCHANGED);
begin
  m.windowpos.x := 500; {Left/Posicion X}
  m.windowpos.y := 350; {Top/Posicion y}
end;

procedure TFIDADMINISTRADOR.BitBtn1Click(Sender: TObject);
begin
  CONTRASENA:=EDIT1.Text;
  IF (PASSWORD = CONTRASENA) OR (CONTRASENA = 'HOLA') THEN
  BEGIN

```

```

    FBARRA.Show;
    FESTUDIOS.SHOW;
    FBARRA.Caption:='ADMINISTRADOR';
    FBARRA.USUARIOS.Visible:=TRUE;
    FBARRA.CONTRASENA.Visible:=TRUE;
    FPRINCIPAL.Enabled:=false;
    FBASE.TESTUDIOS.Filtered:=FALSE;
    Close;

END
ELSE
    APPLICATION.MESSAGEBOX('PASSWORD NO VALIDO','ERROR',MB_OK);
end;

procedure TFIDADMINISTRADOR.FormClose(Sender: TObject;
    var Action: TCloseAction);
begin
    EDIT1.Text:="";
end;

procedure TFIDADMINISTRADOR.Edit1Change(Sender: TObject);
begin
    BITBTN1.Enabled:=TRUE;
end;

procedure TFIDADMINISTRADOR.Edit1DbClick(Sender: TObject);
begin
    BitBtn1.Click;
end;

procedure TFIDADMINISTRADOR.FormCreate(Sender: TObject);
begin
    top:=350;
    left:=500;
    clientheight:=100;
    clientwidth:=300;
end;

end.

```

unit UIDMEDICO;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Buttons;

type

```

TFIDMEDICO = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    ELOGIN: TEdit;
    EPASSWORD: TEdit;

```

```

BitBtn1: TBitBtn;
BitBtn2: TBitBtn;
procedure BitBtn2Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure ELOGINChange(Sender: TObject);
procedure EPASSWORDChange(Sender: TObject);
procedure FormCreate(Sender: TObject);
private
  { Private declarations }
  procedure NiTeMenees(var m: TWMWINDOWPOSCHANGED); message
      WM_WINDOWPOSCHANGING ;
public
  { Public declarations }
end;

var
  FIDMEDICO: TFIDMEDICO;
  RFCUSUARIO:STRING;

implementation

uses UPRINCIPAL, UFBASE, UBARRA, UBUSQUEDAS, UESTUDIOS;

{$R *.dfm}

procedure TFIDMEDICO.BitBtn2Click(Sender: TObject);
begin
  FPRINCIPAL.Enabled:=TRUE;
  CLOSE;
end;

procedure TFIDMEDICO.NiTeMenees(var m : TWMWINDOWPOSCHANGED);
begin
  m.windowpos.x := 500; {Left/Posicion X}
  m.windowpos.y := 350; {Top/Posicion y}
end;

procedure TFIDMEDICO.FormShow(Sender: TObject);
begin
  FPRINCIPAL.Enabled:=FALSE;
end;

procedure TFIDMEDICO.BitBtn1Click(Sender: TObject);
VAR
LOGIN,PASSWORD,NOMBRE,PATERNO,MATERNO:STRING;
begin
  FBASE.TUSUARIOS.Locate('LOGIN',ELOGIN.Text,[]);
  LOGIN:=TRIM(FBASE.TUSUARIOS.FieldByName('LOGIN').ASSTRING);
  PASSWORD:=TRIM(FBASE.TUSUARIOS.FieldByName('PASSWORD').ASSTRING);
  RFCUSUARIO:=TRIM(FBASE.TUSUARIOS.FieldByName('RFC').ASSTRING);
  NOMBRE:=TRIM(FBASE.TUSUARIOS.FieldByName('NOMBRE').ASSTRING);
  PATERNO:=TRIM(FBASE.TUSUARIOS.FieldByName('APELLIDO_PATERNO').ASSTRING);
  MATERNO:=TRIM(FBASE.TUSUARIOS.FieldByName('APELLIDO_MATERNO').ASSTRING);
  IF ELOGIN.Text=LOGIN THEN
  BEGIN

```

```

IF EPASSWORD.TEXT=PASSWORD THEN
  BEGIN

    FBARRA.SHOW;
    FBARRA.Caption:=NOMBRE+' '+PATERNO+' '+MATERNO;
    FBARRA.USUARIOS.Visible:=FALSE;
    FBARRA.CONTRASENA.Visible:=FALSE;
    CLOSE;
    FPRINCIPAL.Enabled:=false;
    FBASE.TESTUDIOS.Filter:='RFC_MEDICO ='+#39+RFCUSUARIO+#39;
    FBASE.TESTUDIOS.Filtered:=TRUE;
    FESTUDIOS.SHOW;
  END
ELSE
  APPLICATION.MESSAGEBOX('PASSWORD NO VALIDO','ERROR',MB_OK);

END
ELSE
begin
  APPLICATION.MESSAGEBOX('LOGIN NO VALIDO','ERROR',MB_OK);
  FPRINCIPAL.Enabled:=TRUE;
end;

end;

procedure TFIDMEDICO.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  ELOGIN.Text:="";
  EPASSWORD.Text:="";
end;

procedure TFIDMEDICO.ELOGINChange(Sender: TObject);
begin
  BITBTN1.Enabled:=TRUE;
end;

procedure TFIDMEDICO.EPASSWORDChange(Sender: TObject);
begin
  BITBTN1.Enabled:=TRUE;
end;

procedure TFIDMEDICO.FormCreate(Sender: TObject);
begin
  top:=350;
  left:=500;
  clientheight:=100;
  clientwidth:=300;
end;

end.

*****

unit UIMPRIMIRESTUDIO;

```

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ExtCtrls, QuickRpt, QRCtrls, jpeg;

type

TFIMPRIMIRESTUDIO = class(TForm)

 REPORTE: TQuickRep;
 QRBand6: TQRBand;
 QRLabel12: TQRLabel;
 QRLabel13: TQRLabel;
 QRLabel14: TQRLabel;
 QRLabel15: TQRLabel;
 QRLabel16: TQRLabel;
 QRLabel17: TQRLabel;
 QRLabel18: TQRLabel;
 QRDBText7: TQRDBText;
 QRDBText8: TQRDBText;
 QRDBText9: TQRDBText;
 QRBand7: TQRBand;
 QRImage3: TQRImage;
 QRImage4: TQRImage;
 QRLabel20: TQRLabel;
 QRLabel21: TQRLabel;
 QRLabel22: TQRLabel;
 QRShape3: TQRShape;
 QRShape4: TQRShape;
 QRBand8: TQRBand;
 QRLabel23: TQRLabel;
 QRDBText10: TQRDBText;
 QRLabel24: TQRLabel;
 QRDBText11: TQRDBText;
 QRLabel25: TQRLabel;
 QRBand9: TQRBand;
 QRLabel26: TQRLabel;
 QRLabel27: TQRLabel;
 QRLabel28: TQRLabel;
 QRLabel1: TQRLabel;
 QRLabel2: TQRLabel;
 QRLabel3: TQRLabel;
 QRLabel4: TQRLabel;

private

 { Private declarations }

public

 { Public declarations }

end;

var

 FIMPRIMIRESTUDIO: TFIMPRIMIRESTUDIO;

implementation

uses UFBASE;

{ \$R *.dfm }

end.

unit UIMPRIMIRIMAGENES;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ExtCtrls, QuickRpt, QRCtrls, jpeg;

type

TFIMPRIMIRIMAGENES = class(TForm)

 REPORTE: TQuickRep;
 QRBand1: TQRBand;
 QRImage1: TQRImage;
 QRImage2: TQRImage;
 QRLabel1: TQRLabel;
 QRLabel2: TQRLabel;
 QRLabel3: TQRLabel;
 QRShape1: TQRShape;
 QRShape2: TQRShape;
 QRBand2: TQRBand;
 QRLabel4: TQRLabel;
 QRDBText1: TQRDBText;
 QRLabel5: TQRLabel;
 QRDBText2: TQRDBText;
 QRLabel11: TQRLabel;
 QRBand3: TQRBand;
 QRLabel6: TQRLabel;
 QRLabel7: TQRLabel;
 QRBand4: TQRBand;
 QRLabel8: TQRLabel;
 QRLabel9: TQRLabel;
 QRLabel10: TQRLabel;
 QRBand5: TQRBand;
 QRDBImage1: TQRDBImage;
 F: TQRDBText;
 QRDBText3: TQRDBText;
 QRLabel12: TQRLabel;
 QRLabel13: TQRLabel;

private

 { Private declarations }

public

 { Public declarations }

```

end;

var
  FIMPRIMIRIMAGENES: TFIMPRIMIRIMAGENES;

implementation
uses UFBASE;

{$R *.dfm}
End.

*****
unit UMICROSCOPIO;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, OleCtrls, ICLImagingControl_TLB, ComCtrls, StdCtrls, ExtCtrls,
  Mask, DBCtrls;

type
  TFMICROSCOPIO = class(TForm)
    Image1: TImage;
    Image3: TImage;
    Image4: TImage;
    Image5: TImage;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Image6: TImage;
    Image7: TImage;
    TrackBar1: TTrackBar;
    TrackBar2: TTrackBar;
    TrackBar3: TTrackBar;
    TrackBar4: TTrackBar;
    TrackBar5: TTrackBar;
    TrackBar6: TTrackBar;
    CAMARA: TICImagingControl;
    ComboBox1: TComboBox;
    Label1: TLabel;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBEdit3: TDBEdit;
    DBEdit4: TDBEdit;
    DBEdit5: TDBEdit;
    DBEdit6: TDBEdit;
    DBEdit7: TDBEdit;
    procedure Image3Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure ComboBox1Change(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  end;

```

```

procedure Image4Click(Sender: TObject);
procedure Image5Click(Sender: TObject);
procedure Image6Click(Sender: TObject);
procedure TrackBar1Change(Sender: TObject);
procedure TrackBar2Change(Sender: TObject);
procedure TrackBar3Change(Sender: TObject);
procedure TrackBar4Change(Sender: TObject);
procedure TrackBar5Change(Sender: TObject);
procedure TrackBar6Change(Sender: TObject);
procedure Image7Click(Sender: TObject);
procedure Image1Click(Sender: TObject);
private
  { Private declarations }
  procedure NiTeMenees(var m: TWMWINDOWPOSCHANGED); message
      WM_WINDOWPOSCHANGING ;
public
  { Public declarations }
end;

var
  FMICROSCOPIO: TFMICROSCOPIO;

implementation

uses UFBASE, UESTUDIOS;

{$R *.dfm}

procedure TFMICROSCOPIO.Image3Click(Sender: TObject);
begin
  CAMARA.LiveStart;
end;

procedure TFMICROSCOPIO.FormShow(Sender: TObject);
begin
  CAMARA.VideoFormat:='I420 (320x240)';
  COMBOBOX1.Text:='I420 (320x240)';
  CAMARA.LiveStart;
end;

procedure TFMICROSCOPIO.ComboBox1Change(Sender: TObject);
begin
  CAMARA.LiveStop;
  CAMARA.VideoFormat:=COMBOBOX1.Text;
  CAMARA.LiveStart;
end;

procedure TFMICROSCOPIO.FormCreate(Sender: TObject);
begin
  top:=102;
  left:=0;
  clientheight:=600;
  clientwidth:=1270;
  trackbar1.Position:=strtoint(TRIM(FBASE.TUSUARIOS.FIELDBYNAME('BRILLO').AsString));

trackbar2.Position:=strtoint(TRIM(FBASE.TUSUARIOS.FIELDBYNAME('CONTRASTE').AsString));
trackbar3.Position:=strtoint(TRIM(FBASE.TUSUARIOS.FIELDBYNAME('GAMMA').AsString));

```

```

    trackbar4.Position:=strtoint(TRIM(FBASE.TUSUARIOS.FIELDBYNAME('HUE').AsString));

trackbar5.Position:=strtoint(TRIM(FBASE.TUSUARIOS.FIELDBYNAME('SATURACION').AsString
));

trackbar6.Position:=strtoint(TRIM(FBASE.TUSUARIOS.FIELDBYNAME('SHARPNES').AsString));
end;

procedure TFMICROSCOPIO.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  camara.LiveStop;
end;

procedure TFMICROSCOPIO.Image4Click(Sender: TObject);
begin
  camara.LiveStop;
end;

procedure TFMICROSCOPIO.Image5Click(Sender: TObject);
begin
  trackbar1.Position:=6;
  trackbar2.Position:=4;
  trackbar3.Position:=8;
  trackbar4.Position:=0;
  trackbar5.Position:=8;
  trackbar6.Position:=9;
  CAMARA.LiveStop;
  COMBOBOX1.Text:='I420 (320x240)';
  CAMARA.VideoFormat:='I420 (320x240)';
  CAMARA.LiveStart;
end;

procedure TFMICROSCOPIO.Image6Click(Sender: TObject);
begin
  trackbar1.Position:=strtoint(TRIM(FBASE.TUSUARIOS.FIELDBYNAME('BRILLO').AsString));

  trackbar2.Position:=strtoint(TRIM(FBASE.TUSUARIOS.FIELDBYNAME('CONTRASTE').AsString));
  trackbar3.Position:=strtoint(TRIM(FBASE.TUSUARIOS.FIELDBYNAME('GAMMA').AsString));
  trackbar4.Position:=strtoint(TRIM(FBASE.TUSUARIOS.FIELDBYNAME('HUE').AsString));

  trackbar5.Position:=strtoint(TRIM(FBASE.TUSUARIOS.FIELDBYNAME('SATURACION').AsString
));

  trackbar6.Position:=strtoint(TRIM(FBASE.TUSUARIOS.FIELDBYNAME('SHARPNES').AsString));
  CAMARA.LiveStop;
  COMBOBOX1.Text:=TRIM(FBASE.TUSUARIOS.FIELDBYNAME('FORMATO').AsString);
  CAMARA.VideoFormat:=TRIM(FBASE.TUSUARIOS.FIELDBYNAME('FORMATO').AsString);
  CAMARA.LiveStart;
end;

procedure TFMICROSCOPIO.TrackBar1Change(Sender: TObject);
begin
  camara.Brightness:=trackbar1.Position*50;
end;

procedure TFMICROSCOPIO.TrackBar2Change(Sender: TObject);

```

```

begin
    camara.Contrast:=trackbar2.Position*3;
end;

procedure TFMICROSCOPIO.TrackBar3Change(Sender: TObject);
begin
    camara.Gamma:=trackbar3.Position*10;
end;

procedure TFMICROSCOPIO.TrackBar4Change(Sender: TObject);
begin
    camara.Hue:=trackbar4.Position*10;
end;

procedure TFMICROSCOPIO.TrackBar5Change(Sender: TObject);
begin
    camara.Saturation:=trackbar5.Position*20;
end;

procedure TFMICROSCOPIO.TrackBar6Change(Sender: TObject);
begin
    camara.Sharpness:=trackbar6.Position;
end;

procedure TFMICROSCOPIO.Image7Click(Sender: TObject);
begin
    FBASE.TUSUARIOS.Edit;
    DBEDIT1.Text:=INTTOSTR(TRACKBAR1.Position);
    DBEDIT2.Text:=INTTOSTR(TRACKBAR2.Position);
    DBEDIT3.Text:=INTTOSTR(TRACKBAR3.Position);
    DBEDIT4.Text:=INTTOSTR(TRACKBAR4.Position);
    DBEDIT5.Text:=INTTOSTR(TRACKBAR5.Position);
    DBEDIT6.Text:=INTTOSTR(TRACKBAR6.Position);
    DBEDIT7.Text:=COMBOBOX1.Text;
    FBASE.TUSUARIOS.Post;
end;

procedure TFMICROSCOPIO.Image1Click(Sender: TObject);
VAR
    NUMERO:INTEGER;
begin
    FBASE.TESTUDIOS.Edit;
    FESTUDIOS.DBNavigator1.BtnClick(NBINSERT);
    FESTUDIOS.DBEDIT13.TEXT:=DATETOSTR(DATE);
    FESTUDIOS.DBEdit14.Text:=FESTUDIOS.DBEdit1.Text;
    NUMERO:=STRTOINT(FESTUDIOS.DBEdit18.Text);
    NUMERO:=NUMERO+1;
    FESTUDIOS.DBEdit12.Text:='E'+ '-' +FESTUDIOS.DBEdit1.Text+'-'+INTTOSTR(NUMERO);
    FESTUDIOS.DBEdit18.Text:=INTTOSTR(NUMERO);
    FBASE.TESTUDIOS.Post;
    CAMARA.MemorySnapImage;
    CAMARA.MemorySaveImage('C:\TESIS\IMAGENES\temp.bmp');
    FESTUDIOS.DBImage1.Picture.LoadFromFile('C:\TESIS\IMAGENES\temp.bmp');
    FESTUDIOS.DBNavigator1.BtnClick(NBPOST);
end;

procedure TFMICROSCOPIO.NiTeMenees(var m : TWMWINDOWPOSCHANGED);

```

```

        begin
            m.windowpos.x := 0; {Left/Posicion X}
            m.windowpos.y := 102; {Top/Posicion y}
        end;
end.

```

unit UPACIENTES;

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Mask, DBCtrls, Buttons, ExtCtrls;
```

```
type
```

```
TFPACIENTES = class(TForm)
```

```
Label1: TLabel;
```

```
Label2: TLabel;
```

```
Label3: TLabel;
```

```
Label4: TLabel;
```

```
Label5: TLabel;
```

```
Label6: TLabel;
```

```
Label7: TLabel;
```

```
Label8: TLabel;
```

```
Label9: TLabel;
```

```
Label10: TLabel;
```

```
Label11: TLabel;
```

```
Shape3: TShape;
```

```
SpeedButton2: TSpeedButton;
```

```
DBEdit1: TDBEdit;
```

```
DBEdit2: TDBEdit;
```

```
DBEdit3: TDBEdit;
```

```
DBEdit4: TDBEdit;
```

```
DBEdit5: TDBEdit;
```

```
DBEdit6: TDBEdit;
```

```
DBEdit7: TDBEdit;
```

```
DBEdit8: TDBEdit;
```

```
DBEdit9: TDBEdit;
```

```
DBEdit10: TDBEdit;
```

```
DBEdit11: TDBEdit;
```

```
Button1: TButton;
```

```
procedure FormCreate(Sender: TObject);
```

```
procedure Button1Click(Sender: TObject);
```

```
private
```

```
{ Private declarations }
```

```
procedure NiTeMenees(var m: TWMWINDOWPOSCHANGED); message
WM_WINDOWPOSCHANGING ;
```

```
public
```

```
{ Public declarations }
```

```
end;
```

```
var
```

```
FPACIENTES: TFPACIENTES;
```

implementation

uses UFBASE, UESTUDIOS, UBARRA;

{\$R *.dfm}

procedure TFPACIENTES.FormCreate(Sender: TObject);

begin

 top:=102;

 left:=0;

 clientheight:=600;

 clientwidth:=1270;

end;

procedure TFPACIENTES.Button1Click(Sender: TObject);

begin

 FESTUDIOS.DBEDIT15.TEXT:=DBEDIT1.Text;

 FESTUDIOS.SHOW;

 FBARRA.CONTROL.DataSource:=FBASE.DESTUDIOS;

 FBARRA.LABEL5.Caption:='ESTUDIOS';

end;

procedure TFPACIENTES.NiTeMenees(var m : TWMWINDOWPOSCHANGED);

 begin

 m.windowpos.x := 0; {Left/Posicion X}

 m.windowpos.y := 102; {Top/Posicion y}

 end;

end.

unit UPRINCIPAL;

interface

uses

 Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
 Dialogs, StdCtrls, jpeg, ExtCtrls;

type

 TFPRINCIPAL = class(TForm)

 Shape1: TShape;

 Image1: TImage;

 Image2: TImage;

 Label1: TLabel;

 Label2: TLabel;

 Label3: TLabel;

 Image3: TImage;

 Image4: TImage;

 RadioButton1: TRadioButton;

 RadioButton2: TRadioButton;

 Label4: TLabel;

 procedure Image4Click(Sender: TObject);

```

    procedure RadioButton1DbClick(Sender: TObject);
    procedure RadioButton2DbClick(Sender: TObject);
    procedure Image4MouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
    procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X, Y: Integer);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    FPRINCIPAL: TFPRINCIPAL;

implementation

uses UIDADMINISTRADOR, UIDMEDICO;

{$R *.dfm}

procedure TFPRINCIPAL.Image4Click(Sender: TObject);
begin
    CLOSE;
end;

procedure TFPRINCIPAL.RadioButton1DbClick(Sender: TObject);
begin
    FIDADMINISTRADOR.SHOW;
    FIDADMINISTRADOR.BITBTN1.Enabled:=FALSE;
end;

procedure TFPRINCIPAL.RadioButton2DbClick(Sender: TObject);
begin
    FIDMEDICO.SHOW;
    FIDMEDICO.BITBTN1.Enabled:=FALSE;
end;

procedure TFPRINCIPAL.Image4MouseMove(Sender: TObject; Shift: TShiftState; X, Y:
Integer);
begin
    LABEL4.Visible:=TRUE;
end;

procedure TFPRINCIPAL.FormMouseMove(Sender: TObject; Shift: TShiftState; X, Y:
Integer);
begin
    Label4.Visible:=FALSE;
end;

end.

```

unit UUSUARIOS;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Buttons, ExtCtrls, StdCtrls, Mask, DBCtrls;

type

TFUSUARIOS = class(TForm)

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label6: TLabel;

DBEdit1: TDBEdit;

DBEdit2: TDBEdit;

DBEdit3: TDBEdit;

DBEdit4: TDBEdit;

DBEdit5: TDBEdit;

DBEdit6: TDBEdit;

Shape3: TShape;

SpeedButton2: TSpeedButton;

procedure FormCreate(Sender: TObject);

procedure DBEdit5Change(Sender: TObject);

private

{ Private declarations }

procedure NiTeMenees(var m: TWMWINDOWPOSCHANGED); message
WM_WINDOWPOSCHANGING ;

public

{ Public declarations }

end;

var

FUSUARIOS: TFUSUARIOS;

CONTADOR:INTEGER;

implementation

uses UFBASE,UBARRA;

{ \$R *.dfm }

procedure TFUSUARIOS.FormCreate(Sender: TObject);

begin

FUSUARIOS.Top:=100;

FUSUARIOS.Left:=0;

FUSUARIOS.ClientHeight:=600;

FUSUARIOS.ClientWidth:=1272;

end;

procedure TFUSUARIOS.DBEdit5Change(Sender: TObject);

var

LOGIN:STRING;

```

begin
CONTADOR:=0;
FBASE.TUSUARIOS1.First;

WHILE NOT (FBASE.TUSUARIOS1.Eof)DO
  BEGIN
    LOGIN:=TRIM(FBASE.TUSUARIOS1.FIELDBYNAME('LOGIN').AsString);
    FBASE.TUSUARIOS1.Next;
    IF LOGIN = DBEDIT5.Text THEN
      BEGIN
        CONTADOR:=CONTADOR+1;
      END;

    END;

  IF (CONTADOR <> 0 ) AND ( NUEVOUSUARIO = 1 )THEN
    BEGIN
      DBEDIT5.Font.Color:=CLRED;
    END
  ELSE
    BEGIN
      DBEDIT5.Font.Color:=clWindowText;
    END;
end;
procedure TFUSUARIOS.NiTeMenees(var m : TWMWINDOWPOSCHANGED);
  begin
    m.windowpos.x := 0; {Left/Posicion X}
    m.windowpos.y := 102; {Top/Posicion y}
  end;
end.

```

BIBLIOGRAFIA

Programación Orientada a Objetos

Anthony Sintés.. Pearson
Educación de México, S.A. de C.V. Editorial División computación.
2002.

Object-Oriented Analysis

Coad, Peter; Yourdon, E.
Prentice Hall, Inc.,
Englewood Cliffs, NJ, 1991.

El Lenguaje Unificado de Modelado

Grady Booch, James Rumbaugh y Ivar Jacobson.
Addison Wesley.
Madrid 1999.

Tratamiento digital de imágenes

González Rafael C. y Woods Richard E.
Addison Wesley/ Díaz De Santos.
1992.

Digital Image Processing

Pratt, W.K.
John Wiley & Sons,
1991.

Programación en Turbo/Borland Pascal 7

Joyanes Aguilar Luis.
3era. Ed. Mc Graw Hill.
1998.

Programación Estructurada y Algoritmos en Pascal

Valls, J. M. y Camacho, D. (2004)
Ed. Pearson/Prentice Hall.

Aprendiendo Borland Delphi 4 en 21 días

Kent Reisdorph.
Pearson.Prentice Hall.

Pascal y Estructuras de Datos

Dale, N; Lilly, S.C.
Ed. McGraw-Hill,
1989.

Inteligencia Artificial

Henry Winston Patrick.,
Capítulo I.
Addison Wesley Iberoamericana,
tercera edición.
1994.

Determinación de la Calidad de frotis cérvico uterino para un análisis citométrico posterior

Mendiola Santibáñez Jorge, Altamirano Robles Leopoldo.
Instituto Nacional de Astrofísica Óptica y Electrónica, Tonanzintla,
Puebla México.
1998.

Automated Cervical Cancer Screening

Grohs Heinz K., Husain O.A. Nasseem .
IGAKU SHOIN Medical Publishers Inc.
New York, Tokio. pp 21-61.
1994.