



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMA DE ADQUISICIÓN DE DIAGNÓSTICOS CLÍNICOS

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO ELÉCTRICO ELECTRÓNICO
P R E S E N T A

NAYELI GRISEL TORRES SÁNCHEZ

DIRECTOR DE TESIS: M. en I. MIGUEL ANGEL BAÑUELOS SAUCEDO

LABORATORIO DE ELECTRÓNICA
CENTRO DE CIENCIAS APLICADAS Y DESARROLLO TECNOLÓGICO



MÉXICO, D.F.

2005



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.

A mis Padres, Socorro y Luis.

A mis hermanos, Luis, Fabiola y Erik.

A mis sobrinos, Isela, Luis y Lorena.

A mis cuñados, Laura y Juan Carlos.

A todos ellos por ser una gran familia, por el apoyo que todos me han dado para salir adelante. Sinceramente Gracias.

A

Julio Ramírez

Cesar Rodríguez

Mario Rocha

Jaime Monroy

Roberto Molero

Arturo Nevárez

Por la confianza que tuvieron en mí. Por el apoyo desinteresado que me han dado desde el momento en que los conocí. Por ser esas grandes personas que son. Por todo lo que me enseñaron. Porque a pesar de que el tiempo ha pasado, seguimos juntos y por lo que han compartido conmigo.

Gracias de verdad

Este trabajo es de ustedes.

POR SU AYUDA TÉCNICA Y FINANCIERA AGRADEZCO A:

- **Centro de Ciencias Aplicadas y Desarrollo Tecnológico. (CCADET)**

En especial al Laboratorio de Electrónica y a todos sus integrantes.

- **Laboratorios Silanes S:A: de C.V.**

Finalmente agradezco a todas aquellas personas que han influido de manera positiva en mi vida y que han estado conmigo en las buenas y en las malas.

ÍNDICE

1. Introducción	
Antecedentes.....	1
Definición del problema	2
Justificación	3
Objetivo General	4
Objetivos específicos.....	5
Metodología.....	5
2. Análisis de los requerimientos del sistema	
2.1. Recopilación de información.....	8
a) Definición de sistema	8
a.1) Software.....	9
Clasificación del software	9
Sistema Operativo.....	10
Lenguaje de Programación	11
Software de uso general.....	13
Software de aplicación	14
a.2) Hardware	14
a.3) Gente	15
a.4) Bases de datos	15
Modelos de las bases de datos	16
Modelo relacional.....	16
a.5) Documentación.....	23
2.2. Identificación de las necesidades de los usuarios del sistema	24
Como trabaja el sistema de adquisición de diagnósticos clínicos	25
Especificaciones de las funciones del sistema.....	28
2.3. Análisis de los componentes del sistema	
Casos de uso	31
Representación de un modelo de casos de uso	31
Casos de uso del registro de los pacientes	32
Casos de uso de recepción de los diagnósticos.....	35
Casos de uso del registro presentación de los registros.....	37
2.4. Análisis de los elementos que constituirán el sistema	
Visual Basic.Net.....	42
Características de Visual Basic.Net.....	43
Microsoft Access	45
Ventajas	46
Desventajas.....	47
ADO.NET.....	47
Componentes de ADO.NET	48
El Dataset.....	48
El Proveedor de datos .NET Framework.....	49
Setup Factory 7.0.....	51

Características	51
2.5. Análisis de los requerimientos de hardware y software	
Hardware	53
Software.....	53
3. Diseño del sistema	
3.1. Diagrama de flujo de datos	54
Descripción de cada uno de los elementos de un diagrama de flujo de datos...	55
3.2. Diseño del diagrama Entidad-Relación	61
Diagramas de entidad relación primaria.....	62
Diagramas de entidad relación secundaria	62
4. Desarrollo del Sistema de Adquisición de datos	
Descripción del ambiente	63
Ventana Principal	64
Botones de minimizar, restaurar y cerrar.....	65
Barra de título	65
Barra de menú.....	65
Ventana de trabajo.....	67
Ventana de presentación.....	67
Ventanas secundarias.....	67
a) Registro de Pacientes.....	68
a.1) Registro de Pacientes.....	69
a.2) Barra de navegación	69
a.3) Edición de registros de pacientes	70
b) Recepción de Datos.....	75
b.1) Barra de botones de adquisición.....	76
b.2) Tabla de Datos.....	78
b.3) Barra datos de Estado.....	78
b.4) Puerto de Comunicación	79
b.5) Recepción de Datos	80
b.6) Funciones adicionales	82
c) Tabla de datos.....	82
c.1) Barra de botones de Tabla de Datos	84
c.2) Exportar tabla de datos a Excel	85
c.3) Exportar datos.....	86
c.4) Abrir Excel	87
c.5) Reporte de diagnósticos clínicos	88
c.6) Estadísticas	91
d) Historial.....	91
d.1) Barra de botones de Historial.....	92
e) Salir	92

5. Fases de prueba y corrección.....	93
Pruebas de caja negra	94
Limitaciones de las pruebas de caja negra.....	95
Pruebas de caja blanca	95
Limitaciones de las pruebas de caja blanca.....	95
Pruebas Unitarias	95
Pruebas de integración	96
Pruebas de regresión.....	96
Consideraciones para la ejecución de las pruebas.....	96
Pruebas aplicadas.....	97
6. Mantenimiento.....	106
Conclusiones.....	108
Bibliografía.....	111
Páginas Web	112

1. INTRODUCCIÓN

ANTECEDENTES

Muchos de los exámenes clínicos que son hechos hoy en día, todavía dependen de la mano del hombre. Un ejemplo de esto, son las pruebas instantáneas de embarazo, estas pruebas pueden causar conflictos a los usuarios ya que algunas veces el resultado no es claro, esto se debe a que las banditas no toman la suficiente tonalidad para que el ojo humano pueda percibirla.

En muchas ocasiones se necesita de una persona capacitada para detectar las banditas. Una persona capacitada puede percibir con mayor facilidad las tonalidades, pero a veces es difícil entrenar a las personas para hacer pruebas, además de que éste es un proceso que necesita demasiado tiempo.

Para evitar la necesidad de capacitar a las personas siempre que se requiera, se pretenden automatizar algunos diagnósticos clínicos de forma completa, esto es, tanto en la adquisición de diagnósticos como en la interpretación y manipulación de los mismos. Por este motivo se ha diseñado un instrumento para que, por medio de éste, las personas puedan tomar los diagnósticos clínicos. De igual manera disminuir la probabilidad de error, esto es muy importante ya que una persona aún especializada tiende a cometer errores, esta situación se presenta debido a que algunas veces la persona puede ver la presencia de tonalidad cuando no la hay o viceversa.

Hoy en día la comunicación de los instrumentos de cualquier clase (sobre todo biomédicos y de uso para fines educativos) y la computadora ha llegado a ser muy importante, ya que esto implica que la mano del hombre no sea necesaria y evita errores.

Es un hecho que al copiar los datos del instrumento a la hoja de papel, pueden generarse grandes errores y pérdida de tiempo. Conectar los instrumentos a la PC permite manejar grandes cantidades de datos en pequeños intervalos de tiempos y la disminución de errores es muy grande y permite una mayor precisión. Así también se puede monitorear el comportamiento de algunos instrumentos en tiempo real.

El uso de la PC hoy en día ha llegado a ser muy importante, minimiza el trabajo y permite realizar otras actividades. Así también puede crearse un respaldo de la información que los instrumentos generan para consultas posteriores, con esto se evita el uso de grandes cantidades de papel y permite que los datos estén contenidos en un mismo espacio.

El Laboratorio de Electrónica del Centro de Ciencias Aplicadas y Desarrollo Tecnológico (CCADET), diseñó un instrumento que permite generar diagnósticos clínicos instantáneos, pero estos diagnósticos deben ser dados con el menor error posible. Para lograrlo, el instrumento contará con un sistema de adquisición, el cual se encargará básicamente de adquirir los datos que el instrumento guarda en memoria, estos datos son el diagnóstico (el cual es puramente cualitativo es decir solamente dará los resultados de Positivo, Negativo o Inválido), la hora y fecha en que se realizó el diagnóstico, posteriormente, se les dará formato a dichos datos. Así también, el sistema complementará los datos que el instrumento no otorga, como por ejemplo el nombre y sexo del paciente al que se le realice la prueba y la fecha de nacimiento, para más adelante, poder realizar un informe como el que se elabora en los laboratorios clínicos.

DEFINICIÓN DEL PROBLEMA

El instrumento de diagnósticos clínicos desarrollado por el CCADET, tiene la opción de almacenar los diagnósticos realizados y además tiene la opción de enviarlos a la PC vía puerto serial. Por lógica este instrumento necesita una interfaz de adquisición de datos.

Hoy en día existe una inmensa variedad de interfaces de adquisición para los distintos instrumentos que se utilizan en muchos lugares como hospitales, laboratorios, institutos, escuelas, etc. Las interfaces generalmente son de uso específico para cada uno de los instrumentos y la mayor parte de ellas no son comerciales. Por lo tanto la interfaz que el instrumento necesita, debe ser desarrollada para los requerimientos que el instrumento demanda y para el uso que se le dará.

El instrumento está orientado para uso médico, por lo cual la interfaz se debe diseñar pensando en las necesidades de un laboratorista, un doctor o una enfermera. Estas necesidades pueden ser las de tener los diagnósticos del paciente con fecha, hora en la que se generó el diagnóstico, así también, los datos personales del paciente. Necesita generar un reporte con ciertos detalles sobre el resultado del diagnóstico y mantener esos datos almacenados para, en caso de ser necesario más adelante, poder consultarlos y así poder comparar los resultados obtenidos en determinado momento con los resultados anteriores de un mismo paciente. De esta manera se puede tener un historial clínico de un paciente.

Debido a que se orienta a personas que son ajenas al manejo cotidiano de la computadora personal, el sistema de Adquisición de Diagnósticos Clínicos, debe ser sencillo, debe ser gráfico y con entorno similar a los programas comerciales.

JUSTIFICACIÓN

Tomar datos por medio de instrumentos de medición es algo muy sencillo en nuestros días, pero poder manejar esos datos y capturarlos a una computadora a veces resulta ser un poco complicado y tardado, además es posible cometer errores en la adquisición de estos. Debido a esto se ha establecido la comunicación entre instrumentos de medición de cualquier tipo y la computadora. Esto se logra por medio de una interfaz, la cual consiste de una parte electrónica y una parte de software, de tal manera que los datos son adquiridos de manera directa y podrán ser manipulados disminuyendo errores.

La comunicación entre instrumentos de medición y una computadora personal, suele tener muchas ventajas ya que una vez contenidos los datos en la computadora, estos pueden ser manipulados, de manera que se pueden guardar en bases de datos, para que en un futuro esos datos puedan ser consultados. De igual manera pueden ser cambiados o bien editados, se pueden realizar gráficas sobre el comportamiento de un instrumento en específico, etc.

Para poder realizar la comunicación, es necesario desarrollar un sistema que reciba la información ya sea vía puerto serial o puerto USB, existen diversos paquetes que pueden ayudar a la recepción; algunos son más amigables, aunque algunos pueden ser más completos, la elección de ellos depende de las necesidades del usuario.

Hoy en día existe una gran cantidad de sistemas de adquisición de datos, cada uno posee características específicas dependiendo de los datos que se desea adquirir y del instrumento con el que se está trabajando, muchos sistemas son muy costosos y algunos no manejan interfaz gráfica para la adquisición de datos, lo cual hace más difícil su manejo e inclusive algunas veces hasta la interpretación de los datos adquiridos resulta ser complicada.

OBJETIVO GENERAL

Desarrollar un sistema de adquisición de datos para un instrumento de diagnósticos clínicos, el cual se encargará de almacenar y procesar los datos recibidos desde dicho instrumento. El sistema permitirá la consulta de los diagnósticos generados por el instrumento y a su vez la impresión de un reporte, el cual contendrá tanto los datos personales (nombre, fecha de nacimiento, sexo, número de paciente) de cada uno de los pacientes así como su respectivo diagnóstico.

OBJETIVOS ESPECÍFICOS

- Desarrollar un sistema de adquisición que se adapte a las necesidades de un usuario ajeno al uso cotidiano de la computadora como herramienta de trabajo.
- Personalizar los diagnósticos adquiridos por la computadora, es decir desplegar los datos tales como, el resultado del diagnóstico clínico (positivo, negativo o inválido), hora y fecha en la que se tomó el diagnóstico, estos datos son enviados por el microcontrolador, vía puerto serial, el sistema será capaz de desplegar el nombre de la persona a la que se le tomó dicho diagnóstico, su sexo y fecha de nacimiento, estos últimos datos serán tomados por medio de la PC.
- Elaborar la historia de los diagnósticos tomados por el instrumento, de manera que los diagnósticos puedan ser guardados en tablas de datos, existirá una tabla que no podrá ser borrada y esta contendrá cada uno de los diagnósticos que se tomaron desde el primer uso del instrumento.
- Imprimir un reporte con el resultado la hora y fecha en que se tomó el diagnóstico así como el nombre del paciente fecha de nacimiento y su sexo.

METODOLOGÍA

Para desarrollar sistemas de adquisición de datos como el sistema de diagnósticos clínicos que se pretende desarrollar, es necesario seguir una metodología, la cual debe ser útil para organizar las actividades necesarias para la construcción del sistema y para especificar la secuencia en que se deben tratar las actividades para su desarrollo.

El modelo que va a ser utilizado, es el modelo más conocido en la Ingeniería de Software llamado Modelo Lineal, Ciclo de Vida Clásico o Modelo de cascada.

Este modelo especifica qué pasos y en qué orden deben llevarse a cabo cada una de las actividades para que el sistema se desarrolle de manera adecuada y en un corto lapso de tiempo.

El modelo de ciclo de vida clásico se muestra en la siguiente figura:

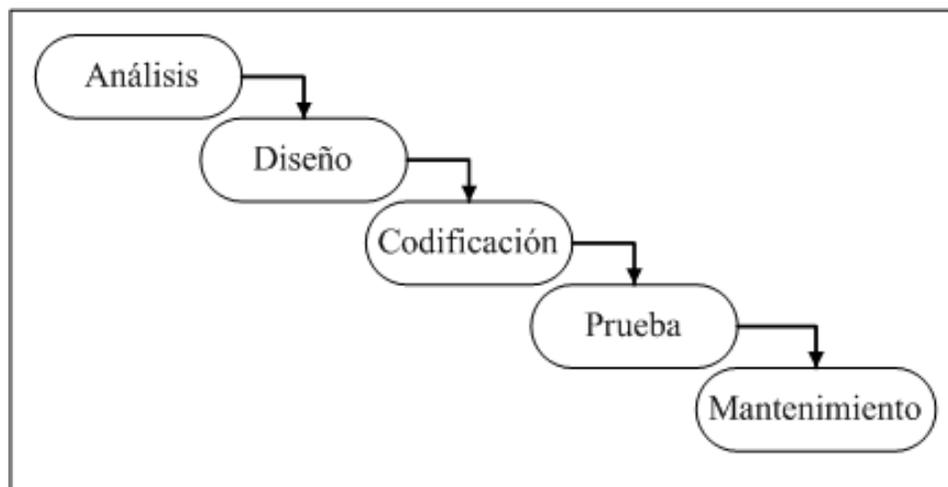


Figura 1.1. Sistema Lineal.

A continuación se describirán cada una de las partes que se seguirán para el desarrollo del Sistema de Adquisición de Diagnósticos Clínicos basadas en el Ciclo de Vida Clásico.

Análisis. El primer paso del ciclo de vida de un proyecto consiste en un análisis de las características y el comportamiento del sistema del cual el software va a formar parte.

Inicialmente se trabajará en la definición del problema a resolver, para después realizar una recopilación de la información que permita conocer a fondo las características y el funcionamiento que el sistema debe tener. Así también se debe comprender cuáles son los datos que se van a manejar, cuál va a ser la función que tiene que cumplir el software, cuáles son las interfaces requeridas y cuál es el rendimiento que se espera lograr.

Diseño. El diseño de un sistema de información es el paso en el que se establece la forma en la que el sistema cumplirá con los requerimientos identificados durante la etapa de análisis.

En el diseño, los requisitos del sistema se traducen a una serie de diagramas que representan la estructura de un sistema, de sus datos, de sus programas y de sus interfaces teniendo como resultado el modelo propuesto.

Codificación. Es la traducción del diseño a un lenguaje de programación. A veces, cuando se está en esta fase surgen problemas que obligan a volver al análisis o al diseño.

Pruebas. Es el proceso de evaluación del sistema que tiene como objetivo el verificar que se satisfagan los requerimientos, e identificar las diferencias entre los resultados esperados y los que produce el sistema.

Mantenimiento. En esta fase se realizarán cambios, bien por errores que no se hayan detectado antes, por cambios en el entorno (por ejemplo que el usuario cambie de computadora, impresora, etc.) o por ampliaciones a petición del cliente.

2. ANÁLISIS DE LOS REQUERIMIENTOS DEL SISTEMA

2.1. RECOPIACIÓN DE INFORMACIÓN

La tesis que a continuación se desarrolla trata de un sistema computacional, el cual es capaz de adquirir datos de un instrumento.

Para poder entender este trabajo, es necesario definir conceptos que serán muy utilizados durante el desarrollo del mismo.

Las definiciones son simples, ya que algunos de los temas pueden llevarse capítulos enteros, por ello, es que solamente se da una introducción del concepto y si se desea ampliar dichos conceptos se puede consultar la bibliografía que se encuentra la final de este documento.

a) DEFINICIÓN DE SISTEMA

La palabra “sistema” posiblemente es una de las palabras más utilizadas en nuestros días y puede tener distintos significados dependiendo del contexto de las frases. Por ejemplo, podemos hablar de sistemas de transporte, sistemas políticos, sistemas computacionales, sistemas bancarios, sistemas educativos, sistema nervioso, etc.

Un sistema computacional, se puede definir de la siguiente manera:

Conjunto de elementos organizados para llevar a cabo algún método, procedimiento o control mediante el proceso de información.

Los elementos que conforman un sistema son:

- a.1) Software
- a.2) Hardware
- a.3) Usuarios
- a.4) Bases de datos
- a.5) Documentación

A continuación se definirá cada uno de los elementos:

a.1) SOFTWARE

El Software es un conjunto de programas, documentos, procedimientos, y rutinas que las computadoras emplean para manipular datos. Si la computadora no tuviera ninguna clase de software ésta simplemente no podría trabajar.

Clasificaciones del Software

El software se clasifica en cuatro diferentes categorías que son

- Sistemas Operativos
- Lenguajes de Programación
- Software de uso general
- Software de Aplicación

SISTEMA OPERATIVO

Un sistema operativo es un programa que se carga en la memoria de la computadora por una rutina de inicio, cuyo objetivo es el de gestionar y administrar eficientemente el hardware, permitiendo que se ejecuten al mismo tiempo varios programas, sin que haya conflictos en el acceso de cada uno de ellos a cada uno de los recursos que necesite.

Además de que permite el uso de los recursos que se solicitan sin que ningún programa los monopolice.

Un sistema operativo puede efectuar las siguientes funciones:

- Facilita el uso de la computadora, en general, la comunicación computadora usuario.
- Gestiona y asigna recursos de hardware (procesador, memoria y periféricos) a los distintos programas o tareas.
- Gestiona y mantiene los archivos en dispositivos de memoria masiva.
- Protege los datos y los programas, cuestión especialmente compleja en sistemas multiusuarios.
- Identifica y autentifica a los usuarios que hacen uso de la computadora.
- Contabiliza la utilización de los recursos realizados por los distintos usuarios.
- Si es un sistema multitareas: asigna y controla los recursos del sistema, define que aplicación y en que orden deben ejecutarse.
- Maneja la memoria del sistema que comparten las múltiples aplicaciones.
- Maneja los sistemas de entrada y salida, incluidos discos duros, impresoras y todo tipo de puertos.
- Envío de mensajes de estado a las aplicaciones, al administrador de sistemas o al propio usuario, sobre cualquier error o información necesaria para el trabajo estable y uniforme del sistema.

- Administra el procesamiento en paralelo.

Algunos ejemplos de sistemas operativos son UNIX, Linux, MS-DOS y todas las versiones conocidas de Windows.

LENGUAJE DE PROGRAMACIÓN

Los lenguajes de programación son las herramientas que se emplean para desarrollar aplicaciones.

Existe una gran variedad de lenguajes de programación, algunos son desarrollados para cumplir objetivos específicos y algunos otros son de uso generalizado.

Al desarrollar aplicaciones, en ocasiones es necesario utilizar más de un lenguaje de programación para su desarrollo.

La decisión de utilizar un determinado lenguaje de programación es difícil ya que éste debe adecuarse a las necesidades del sistema, del hardware y sobre todo a las necesidades del cliente. Elegir una buena herramienta puede reducir dificultades de codificación, así también puede reducir la fase de pruebas y correcciones.

Un lenguaje de programación debe tener facilidades de estructuración que permitan producir un código legible y lo más simplificado posible, tales como procedimientos y funciones globales, debe tener las opciones de detectar errores en la compilación debido a malas asignaciones de valores etc.

Los lenguajes de programación se dividen en:

- Lenguajes de bajo nivel

Utilizan códigos muy cercanos a los de la máquina, lo que hace posible la elaboración de programas muy potentes y rápidos.

- Lenguajes de alto nivel

Son una serie de comandos o secuencia de instrucción que el programador escribe y que a su vez estas equivalen a cientos de instrucciones de código de máquina.

En este tipo de lenguajes, antes de ejecutar el programa, la computadora traduce el código que se va a ejecutar en código de máquina (compila) o bien lo interpreta línea por línea (interprete).

Los archivos de programas ejecutables contienen el código máquina, que la unidad central de procesamiento (CPU) identifica como sus instrucciones. Son lo que conocemos como programas objeto.

El proceso de conversión de programas fuente a programas objeto se realiza mediante un programa denominado compilador. El compilador toma un programa fuente y lo traduce a programa objeto y almacena este último en otro archivo.

Existe una gran variedad de lenguajes de programación algunos ejemplos de ellos son: Pascal, Cobol, Lenguaje ensamblador, C, C++, Java, Visual Basic, etc.

Existen diferentes tipos de programación que son la programación estructurada y la programación orientada a objetos.

La programación estructurada se basa en el diseño de la aplicación de lo general a lo particular y en el teorema de la estructura, el cual se refiere a que todo programa puede ser diseñado utilizando únicamente las estructuras básicas: secuencial alternativa y repetitiva.

A partir de la programación estructurada, los esfuerzos se centraron a la modularidad y la reutilización del código lo que condujo a la programación orientada a objetos.

La programación orientada a objetos es una forma de programación que emplea objetos (entidades del software), definidos por una clase, que poseen estado y comportamiento y que están ligados, mediante mensajes, para la solución de problemas.

SOFTWARE DE USO GENERAL

El software para uso general ofrece la estructura para un gran número de aplicaciones empresariales, científicas y personales. El software de hoja de cálculo, de diseño asistido por computadora (CAD), de procesamiento de texto, de manejo de Bases de Datos, pertenece a esta categoría. La mayoría de software para uso general se vende como paquete; es decir, con software y documentación orientada al usuario (manual de referencia, plantillas de teclado y demás).

Algunos ejemplos de software de uso general son: Excel, Access, Word, Autocad, P-CAD Matlab, etc.

SOFTWARE DE APLICACIÓN

Son aplicaciones diseñadas para realizar tareas específicas que el usuario pretenda utilizar. Son aplicaciones en las cuales se procesan datos de recepción y generan cierta información en el sistema operativo que es de utilidad para un usuario en específico.

El ejemplo de software de aplicación es cualquier programa elaborado para un uso específico.

a.2) HARDWARE

Son los dispositivos electrónicos que proporcionan la capacidad de la computadora y los dispositivos electromecánicos que proporcionan las funciones del mundo exterior.

Para el sistema de Adquisición de Diagnósticos Clínicos se utilizará como dispositivo para conectar los componentes externos a la computadora, el puerto serial RS232 del cual, a continuación se dará una breve explicación.

El puerto serie es aquel que permite la comunicación de la computadora personal con otros dispositivos como son impresoras, instrumentos de medición, módems, conmutadores y otras computadoras. Tiene sus inicios en los 60's por la EIA (Electronics Industries Association de los EE.UU), y fue creado para ofrecer una interfaz entre aparatos que requieren comunicación de datos.

El tipo de comunicación que utiliza el puerto serie es por medio de la comunicación asíncrona, es decir, cada vez que se inicia la comunicación llegan paquetes de 8 bites o bien 1 byte (en código ASCII), pero cada bit dentro del byte debe llegar uno a uno por ello es que el puerto serie es asíncrono por carácter y síncrono por bit.

El puerto serie tiene parámetros importantes como son la velocidad de comunicación, ésta puede ser desde los 300 Bauds (1 Baud = 1bit/seg.) hasta 115 200 Bauds, pero generalmente se trabaja con 9600 Bauds.

El protocolo de comunicación puede variar ya que algunos datos pueden ser enviado uno a uno y otras veces los datos son almacenados en memoria para que sean enviados todos en una sola transmisión.

a.3) USUARIOS

Son las personas que utilizarán el sistema, es decir, son los operadores del software y hardware.

a.4) BASES DE DATOS

Una base de datos puede definirse como una colección de datos interrelacionados entre sí y almacenados en conjunto, la cual tiene como finalidad servir a una o más aplicaciones de la mejor manera.

En su manera más simple, una base de datos es una lista con una estructura específica, definida por sus campos. La estructura de la clase de datos la proporcionan tablas diseñadas para almacenar datos fácilmente, con precisión y sin redundancias. Las tablas están formadas por campos o categorías de información, y registros o filas de elementos individuales.

Considerando su función primordial, las bases de datos pueden clasificarse en dos grupos y son los siguientes:

- Bases de datos analíticas

Éstas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

- Bases de datos dinámicas

Éstas son bases de datos más versátiles, orientadas a almacenar información que es modificada con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta.

MODELOS DE LAS BASES DE DATOS

Los modelos de datos son básicamente "descripciones" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas; son abstracciones que permiten la implementación de un sistema eficiente de base de datos.

Uno de los modelos más utilizados en nuestros días, debido a la simplicidad en su estructura de datos, a la facilidad y manejo de los mismos, es el modelo relacional.

En este tipo de modelo, la información puede ser recuperada o almacenada por medio de "consultas" que ofrecen una amplia flexibilidad y administración de la información.

El modelo relacional se describirá a continuación.

MODELO RELACIONAL

Éste es el modelo más utilizado en la actualidad para diseñar e implementar problemas reales, así como para administrar datos dinámicamente.

La base de datos relacional es un conjunto de tablas que se relacionan entre ellas para crear nuevas estructuras denominadas relaciones o tablas. Las relaciones no modifican la estructura de las tablas originales simplemente permiten que los usuarios vean dicha información de manera ordenada.

En este tipo de modelo no existen jerarquías entre los campos de los registros, por lo que cada campo puede ser usado como un identificador o llave. Los datos son almacenados como una colección de valores en forma de simples registros llamados tuplas o duplas.

Una tupla equivale a un renglón dentro de una tabla y un atributo a una columna. Cada dupla representa en realidad un conjunto de valores relacionados permanentemente. Estas tuplas son agrupadas en tablas bidimensionales, donde cada tabla generalmente es almacenada como un archivo separado, incluso los resultados de cualquier consulta se presentan en otra tabla.

En una base de datos relacional, las tablas están conectadas una a otra por medio de campos únicos los cuales son campos con valores que solo tienen lugar una vez en una tabla. Los

campos o combinación de campos que solo identifica a cada registro de la tabla se le denomina clave primaria.

La clave primaria en una tabla es el campo o campos utilizados para identificar de manera unívoca cada registro de la tabla. Cuando se incluye el campo clave en otra tabla y este es utilizado para relacionar ambas tablas, el campo de la tabla relacionada se denomina clave secundaria.

Toda relación o tabla debe cumplir las siguientes reglas:

- Toda relación o tabla debe tener un nombre único y distinto de otros dentro de la misma base de datos.
- Cada atributo o columna debe tener un nombre propio, único y distinto de otros, dentro de la misma tabla.
- Cada celda o intersección renglón-columna debe aceptar exactamente uno y solo un valor y nunca un arreglo o conjunto de datos.
- Los valores en una misma columna deben ser del mismo tipo.
- No debe haber renglones idénticos. Sobre todo si se trata de los campos con llave primaria
- El orden de los renglones no es significativo.

Las principales ventajas del modelo relacional son:

- Simplicidad: es decir, la facilidad de comprensión por parte del usuario. El usuario final se enfrenta con un modelo de datos sencillo. Un modelo relacional es prácticamente lo que el usuario ve, y no necesariamente lo que se implantará.
- Independencia de los datos: esto debe constituir uno de los principales objetivos de cualquier modelo de datos. El modelo relacional elimina los detalles relativos a la estructura del almacenamiento.
- Búsqueda de datos: ésta puede ser mucho más rápida que en los sistemas que deben de seguir una cadena de apuntadores.
- Las estructuras relacionales, son mucho más fáciles de modificar que las estructuras jerárquicas o de red. En ambientes donde la flexibilidad es importante, esto es primordial.
- Globalización de la información. Permite a los diferentes usuarios considerar la información como un recurso corporativo que carece de dueños específicos.

Existen tres tipos de relaciones y son los siguientes:

- Uno a uno. Cuando las tablas están relacionadas en un modelo de uno a uno, cada registro de la tabla A no tiene más de un registro en la tabla B. Este tipo de relaciones permiten seleccionar un único registro en una única tabla y ver todos los registros relacionados en una tabla diferente.
- Uno a varios. En el modelo más común de uno a varios, un registro de la tabla A puede relacionarse a muchos registros de la tabla B.
- Varios a varios. Una relación varios a varios significa que un registro de la tabla A puede relacionarse con muchos registros de la tabla B y que un registro de la tabla B puede estar, al mismo tiempo, relacionado con muchos registros de la tabla A.

Para poder trabajar con las bases de datos necesitamos utilizar un sistema de gestión de datos, el cual es un conjunto de programas, procedimientos y lenguajes que nos proporcionan las herramientas necesarias para trabajar con una base de datos. Incorpora una serie de funciones que nos permita definir los registros, campos, relaciones, insertar, suprimir, modificar y consultar datos. Algunos gestores son Access, Oracle, Microsoft SQL Server, Informix, etc.

Casi todos los gestores de datos tienen un lenguaje especializado llamado lenguaje de manipulación de datos, el cual es utilizado con algunos lenguajes de programación convencionales para manipular las bases de datos.

El lenguaje de manipulación, contiene comandos que permiten a los usuarios finales y a los especialistas en programación extraer datos de las bases de datos para satisfacer las solicitudes de información y creación de aplicaciones. El lenguaje de manipulación de datos más destacado en la actualidad es SQL (Structured Query Language o Lenguaje de Consultas Estructurado), el cual es un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales tal como Microsoft Access.

El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos.

- Lenguaje de definición de datos (DDL). Proporciona órdenes para crear y definir nuevas bases de datos, campos, índices, esquemas de relación. De la misma manera proporciona órdenes para eliminar relaciones y modificar esquemas de relación. Así también incluye órdenes para definir vistas.
- Lenguaje de manipulación de datos interactivos (DML). Permite generar consultas para ordenar, filtrar y extraer datos de la base de datos.

Algunos tipos de bases de datos son SQL, Oracle, DBASE, Foxpro, Foxbase, Access, Approach.

a.5) DOCUMENTACIÓN

Son los manuales y otra información descriptiva del sistema.

2.2. IDENTIFICACIÓN DE LAS NECESIDADES DE LOS USUARIOS DEL SISTEMA

El sistema de adquisición de datos, es un sistema completamente nuevo, es decir, es un sistema cuyo instrumento aún es un prototipo. Por lo tanto, éste se considera como un prototipo también, el cual está abierto a mejoras.

El sistema está totalmente dirigido a laboratorios clínicos u hospitales, tanto el instrumento como el sistema han sido desarrollados de manera paralela, por esta razón no se tiene clientes específicos para que aporten ideas. Por lo que el análisis se hizo con base a las necesidades perceptibles de un laboratorio clínico.

La finalidad de un diagnóstico clínico de cualquier tipo es la de informar tanto al paciente como al doctor que lo atiende, el estado en el que se encuentran los niveles de ciertas sustancias en el organismo del paciente.

Los pasos que comúnmente se siguen para la realización de análisis clínicos son los siguientes:

1. Tomar una muestra de sangre.
2. Realizar las actividades correspondientes para la toma del diagnóstico clínico. Estas actividades pueden variar dependiendo de la prueba que se requiere.
En el caso del instrumento para la adquisición de diagnósticos, simplemente se coloca una muestra de sangre en la tira y se espera 30 minutos, se inserta la tira en el instrumento y se toma el diagnóstico.
3. Se realiza un reporte con los datos personales del paciente.
Estas tareas pretenden ser automatizadas por el Sistema de Adquisición de Diagnósticos Clínicos.

COMO TRABAJA EL INSTRUMENTO DE DIAGNÓSTICOS CLÍNICOS

Existen diferentes tipos de diagnósticos clínicos:

- Cualitativos. Los que solamente dicen si existe o no la sustancia, como por ejemplo la prueba de embarazo.
- Cuantitativos. Los que dicen la cantidad de sustancias que se tienen en el organismo tales como los niveles de glucosa, colesterol, etc.

El instrumento para la toma de diagnósticos clínicos presentará un resultado cualitativo, de manera que solamente dará los resultados de Positivo, Negativo o Inválido. Así mismo, los datos que el instrumento enviará, serán fecha y hora en que se tomó la prueba, el diagnóstico (positivo, negativo o inválido) y el número de diagnóstico.

Para poder recibir los datos en el sistema es necesario saber como trabaja el microprocesador por lo que se hará un breve resumen de cómo se transmiten los datos desde el microprocesador.

Cuando el instrumento toma los diagnósticos, este presenta la opción de salvar los datos; en el momento en que los datos son salvados, estos se guardan en pequeños paquetes de números binarios, en el siguiente orden:

1. Hora
2. Diagnóstico
3. Minutos
4. Año
5. Estado
6. Mes
7. Día

Para poder entender un poco mejor el proceso se da el siguiente ejemplo, que es la pantalla del instrumento después de la toma de un diagnóstico.



Figura 2.2.1. Display del instrumento.

El instrumento preguntará después de la toma si desea salvar datos, se presionará el botón de aceptar para guardar los datos de manera que para el ejemplo de la figura 2.2.1. lo que se guardará será lo siguiente:

1. 17
2. El número que le corresponde a cada diagnóstico.
3. 16
4. 01
5. Número utilizado para fines del algoritmo en el microcontrolador.
6. 01
7. 10

El instrumento enviará un arreglo de una determinada dimensión, el cual contendrá los “paquetes” de siete datos de cada uno de los registros que son salvados.

Se requiere que el instrumento reciba estos datos, los cuales deberán ser presentados de manera que el usuario pueda identificarlos a simple vista.

Estos datos recibidos se colocarán en una tabla de datos y los cuatro campos formarán un solo registro. Esto permitirá una mejor interpretación y manejo de los registros. De igual manera, se podrán revisar los datos del instrumento con los de la PC, y así verificar si es que existió un error al recibirlo

Los datos recibidos se mostrarán con el siguiente formato:

Datos que se mostrarán**Formato en que se mostrará**

Número de diagnóstico

Se mostrará con números ordinarios (1, 2, 3, 4, 5, 6, etc.), se pretende que se reciban un máximo de 100 diagnósticos (debido a las especificaciones del instrumento) y un mínimo de 1 diagnóstico.

Los diagnósticos pueden ser llamados en cualquier momento sin importar la cantidad de ellos que estén almacenados. Así, si se desea enviar 6 diagnósticos estos podrán ser recibidos de igual manera si se desea enviar 50 ó 70.

Diagnóstico

Los datos serán enviados como números binarios, pero para la fácil comprensión del usuario, los datos se tomarán en la PC y los presentará como diagnósticos Positivos, Negativos o Inválidos

Fecha

La fecha será presentada en formato corto de fecha que es el siguiente DD/MM/AAAA

Donde

DD.- será el día en que se tomó el diagnóstico y estará representado por dos dígitos como por ejemplo el día 10 de algún mes.

MM.- Representará los meses del año en dos dígitos por ejemplo el mes de abril, el cual se representará como el mes 04.

AAAA.- Representará el año en el que se tomó el diagnóstico son cuatro dígitos porque se representa el año completo como por ejemplo 2005.

De tal manera que para representar la fecha que corresponde al 10 de abril del 2005, ésta se mostrará como: 10/04/2005.

Hora

La hora se representará de la siguiente manera:
hh: mm.

Donde:

hh. - Muestra la hora de 0 a 24 horas.

mm. - Representa los minutos.

Así el formato de fecha será el siguiente para representar la 1 de la tarde con 15 minutos.

13:15

ESPECIFICACIONES DE LAS FUNCIONES DEL SISTEMA

Debido a las condiciones en que los datos son adquiridos y la manera en que se propone trabajar con los datos de cada paciente, la opción más adecuada para el almacenamiento de registros es una base de datos, esto se debe a que es necesario trabajar con dos tipos de archivos, un archivo para los diagnósticos y un archivo para los pacientes. Por esta condición no pueden utilizarse hojas de datos de Excel o Archivos de tipo texto (.txt .doc), para los propósitos del proyecto debido a que solamente pueden trabajar como archivos planos, en cambio las bases de datos pueden tener dos tablas con grandes cantidades de campos y estos pueden ser relacionados por medio de campos llamados llaves primarias.

De esta manera en una tabla de datos se guardarán los diagnósticos y en otra tabla se almacenarán los datos del paciente y estos podrán ser relacionados para formar una sola tabla de consulta de registros.

El programa estará dividido en pantallas, se pretende dividirlo de esta manera para que exista una sola pantalla para cada una de las actividades que debe realizar el usuario, de manera que la recepción de diagnósticos podrán ser recibidos en una pantalla, para escribir datos personalizados del diagnóstico, es decir, el nombre del paciente al que se le tomó el diagnóstico, la fecha de nacimiento, etc. Por otra parte, debe aparecer en otra pantalla del programa la recepción de datos del instrumento y por último deberá asignarse una pantalla para el respaldo de los registros.

Una opción para la presentación del programa es la de utilizar un ambiente basado en ventanas (Windows) de tal manera que cada pantalla corresponderá a una ventana.

La información puede ser dividida en cuatro partes y se proponen las siguientes:

- Registro de los pacientes.
- Recepción de los diagnósticos del instrumento
- Presentación de registros completos, es decir, paciente y diagnósticos en un solo registro de pacientes.
- Una ventana donde aparezca la información respaldada.

Las primeras dos partes son capaces de trabajar de forma independiente, pero la tercera y la cuarta dependerán de si los datos fueron o no adquiridos en las tablas de datos.

El usuario únicamente tendrá como entrada los datos del paciente y los datos necesarios para generar el reporte del diagnóstico.

Los datos se transmiten del procesador (instrumento) a la PC con tan solo oprimir un botón, se propone así, con la idea de que el usuario no tenga que poner demasiada atención en esa tarea.

La recepción de diagnósticos, tendrá la opción de recibir los datos por medio de distintos puertos RS232. Se ha pensado así debido a que cada máquina tiene características diferentes, el sistema deberá adecuarse a esas características. Se ha propuesto la posibilidad de diez puertos debido a que algunas máquinas han podido extenderse hasta dicha cantidad.

Todas las ventanas trabajarán con botones, la mayoría de la información debe presentarse de manera que el usuario pueda entenderla fácilmente.

Los datos enviados por el instrumento, una vez contenidos en la PC, no podrán ser modificados por ninguna razón para no alterar diagnóstico alguno.

Algunos datos podrán ser de prueba, o simplemente puede ser que no se requiera de un paciente; por tal motivo, la tabla permite el acceso a cualquier número de paciente sin importar si lleva un orden ascendente o descendente, el número de paciente se conservará en la tabla de datos pero no en la de respaldos ya que ésta llevará su propia numeración.

2.3. ANÁLISIS DE LOS COMPONENTES DEL SISTEMA

El sistema de control de Adquisición de Diagnósticos Clínicos consta de una Base de datos y una interfaz gráfica, las cuales permiten que la información sea un conjunto consistente para poder visualizar modificar y realizar búsquedas. Debido a esto fue conveniente hacer un modelo de casos de uso, este modelo describe lo que hace el sistema para el usuario, contiene actores y sus relaciones.

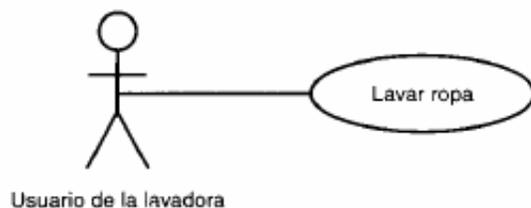
CASOS DE USO

Un Caso de Uso es un documento narrativo que describe a los actores utilizando un sistema para satisfacer un objetivo. Es una historia o una forma particular de usar un sistema. Los casos de uso son requisitos, en particular requisitos funcionales. Para los desarrolladores del sistema, ésta es una herramienta valiosa, ya que es una técnica de aciertos y errores para obtener los requerimientos del sistema desde el punto de vista del usuario. Esto es importante si la finalidad es crear un sistema que pueda ser utilizado por la gente en general.

Es una colección de escenarios iniciados por una entidad llamada actor (una persona, un componente de hardware, un lapso u otro sistema).

REPRESENTACIÓN DE UN MODELO DE CASO DE USO

La representación gráfica es directa. Una elipse representa a un caso de uso, una figura agregada representa a un actor. El actor que inicia se encuentra a la izquierda del caso de uso, y el que recibe a la derecha. El nombre del actor aparece justo debajo de él, y el nombre del caso de uso aparece sea dentro de la elipse o justo debajo de ella. Una línea asociativa conecta a un actor con el caso de uso, y representa la comunicación entre el actor y el caso de uso. La línea es asociativa y sólida. A continuación se muestra un ejemplo del diagrama de caso de uso.



2.3.1. Diagrama de casos de uso.

El diagrama de caso de uso se emplea para visualizar el comportamiento del sistema, una parte de él, de manera que se pueda conocer cómo responde esa parte del sistema, ya que sólo especifica como deben comportarse y no cómo están implementadas las partes que define.

Cada una de las ventanas tendrá funciones propias para el buen manejo de los datos.

La ventana de recepción de los diagnósticos deberá contener funciones que son importantes al momento de recibir los diagnósticos; tales como la de activar el puerto de comunicación, inicio de recepción, entre otras. Estas funciones se describirán mediante casos de uso.

CASO DE USO DEL REGISTRO DE LOS PACIENTES

Este caso debe dividirse en dos partes, una donde se pueda dar entrada a los datos para el registro de pacientes, y la otra donde contengan las funciones necesarias para la edición de un paciente. Las funciones de edición son Agregar, Borrar y Modificar.

Caso de uso de Registro

Usuario del sistema	Respuesta del sistema
1.- Un paciente pide el análisis clínico.	
2.- El encargado de realizar el análisis toma la muestra de sangre e introduce los datos del paciente en el sistema.	1.-El sistema realiza la validación de los campos del paciente e indica si se ha cometido un error en el momento de llenar los datos, permitiendo que se realicen todas las correcciones.

3.- El usuario guarda los datos del paciente dentro de sistema.	2.- El sistema guardará los registros en la tabla correspondiente al registro de pacientes. Y notifica si el registro es agregado.
4.- El usuario requiere un cambio ya que al momento de agregar el diagnóstico no se dió cuenta del error.	3.- El sistema debe hacer los cambios necesarios y notificar si estos se llevaron a cabo.
5.-El usuario desea eliminar un Registro.	4.- El sistema borra el registro de la tabla correspondiente al registro de pacientes.
6.- El usuario desea borrar la tabla de datos actual para recibir nuevos datos. El usuario se encuentra con dos casos. a.- <i>Guardar registros en la tabla de respaldos.</i> b.- <i>Borrar tabla de datos</i>	
7.- El usuario <i>navega</i> por la tabla de datos y se presentan los siguientes casos. a) <i>Siguiente</i> b) <i>Anterior</i> c) <i>Regresa 10 registros</i> d) <i>Avanza 10 registros</i> e) <i>Primer Registro</i> f) <i>Último Registro</i>	

En la figura 2.3.2 podemos observar cómo se realiza el caso de uso de Registro de pacientes

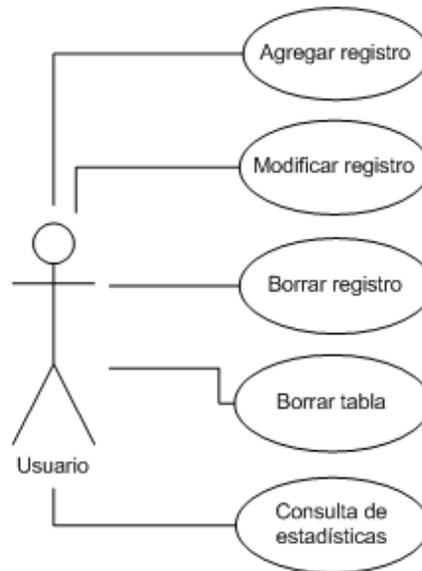


Figura 2.3.2. Caso de uso de Registro de Paciente.

Caso de uso de Guardar registros en la tabla de respaldos

Usuario del sistema	Respuesta del sistema
1.-El usuario requiere guardar los registros en la tabla de respaldos.	1.- El sistema se encarga de enviar los registros a la tabla que contendrá la información respaldada. Se pretende que la mayoría de los datos sean respaldados pero no de forma automática, así el usuario podrá elegir los datos que crea necesario guardar.

Caso de uso de borrar tabla

Usuario del sistema	Respuesta del sistema
1.- El usuario borra la tabla de datos para recibir datos nuevos.	1.- El sistema borra la tabla de datos que será dinámica.

Entre los campos que deberán ser agregados se deberá incluir el número de paciente, el sexo para fines estadísticos, así como la fecha de nacimiento. En realidad el campo debería ser edad del paciente, pero una de las finalidades que tiene el instrumento de diagnósticos es la de tomar el diagnóstico a recién nacidos para verificar los niveles de hormona estimulante de la tiroides, es muy difícil colocar la edad de los recién nacidos. Por este motivo se ha decidido colocar únicamente fecha de nacimiento

CASO DE USO DE LA RECEPCIÓN DE LOS DIAGNÓSTICOS

Usuario del sistema	Respuesta del sistema
1.- El usuario elige el puerto de comunicación con el que se va a trabajar.	1.- El sistema busca y reconoce el puerto para inicializar la comunicación. Si se eligió un puerto con el que no trabaja la máquina el sistema es capaz de indicarlo para así cambiarlo.
2.-El usuario realiza la petición de recepción de datos. Se presentan los siguientes casos: a) <i>Error al elegir el puerto.</i> b) <i>El puerto no se ha detectado.</i> c) <i>No se ha podido establecer comunicación con el instrumento.</i> d) <i>Iniciar comunicación.</i>	
3.- El usuario verifica el número de campos positivos, negativos e inválidos.	3.- El sistema indica cuántos de los diagnósticos entrantes son positivos, cuántos son negativos y cuántos inválidos.

En la figura 2.3.3 podemos observar como se realiza el caso de uso de Recepción de los diagnósticos.

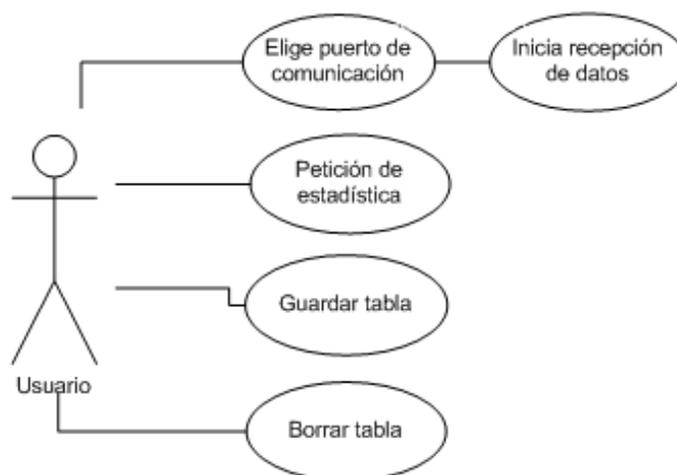


Figura 2.3.3. Caso de uso de Recepción de datos.

Caso de uso con error al elegir el puerto

Usuario del sistema	Respuesta del sistema
1.- El usuario elige un puerto de comunicación que no existe en la PC.	1.- El sistema verifica el puerto, al no existir éste, no se abre el puerto y envía un mensaje de error para que el usuario pueda elegir otro puerto.

Caso de uso de puerto que no se ha detectado

Usuario del sistema	Respuesta del sistema
1.- El usuario trata de establecer comunicación con el puerto correcto, pero el instrumento no está conectado a la corriente o bien el cable de transmisión no está conectado a alguna de sus dos entradas (PC, instrumento).	1.- El sistema verifica el puerto, pero al no detectarlo, envía un mensaje de error para que el usuario pueda corregir el error

Caso de uso en el que no se ha podido establecer comunicación con el instrumento

Usuario del sistema	Respuesta del sistema
1.- El usuario trata de establecer comunicación con el puerto correcto, pero el instrumento no está conectado a la corriente o bien el cable de transmisión esta fallando.	1.- El sistema verifica el puerto, trata de establecer comunicación, envía inicio de interrupción y espera la transmisión al no recibir caracteres, envía un mensaje de error para que el usuario pueda corregir el error.

Caso de uso con inicio de comunicación

Usuario del sistema	Respuesta del sistema
1.- El usuario trata de establecer comunicación con el puerto correcto, el instrumento está conectado y el cable de transmisión no falla.	1.- El sistema envía un carácter e inicia la interrupción de transmisión de datos.

CASO DE USO DE PRESENTACIÓN DE REGISTROS

Esta ventana deberá contener la información ya relacionada de cada uno de los registros que se han agregado en las dos partes anteriores, en esta parte los registros iguales de las ventanas anteriores se mostrarán como uno sólo.

Usuario del sistema	Respuesta del sistema
1.- El usuario exporta la tabla de datos a un archivo de Excel.	1.-El sistema envía la tabla a un archivo de tipo Excel.
2.- El usuario hace la petición de exportar sólo unos cuantos registros al archivo de tipo Excel. Indicando el intervalo de registros que desea enviar.	2.- El sistema envía el intervalo de archivos a la tabla de datos al archivo de tipo Excel.

<p>3.- El usuario abre archivos ya guardados en la hoja de datos de Excel.</p>	<p>3.- El sistema presenta el cuadro de abrir, mostrando los archivos con extensión .xls para elegir cualquiera de ellos. El archivo se abrirá en una ventana de tipo Excel.</p>
<p>4.- El usuario navega por la tabla de datos y se presentan los siguientes casos.</p> <p><i>a) Siguiete</i></p> <p><i>b) Anterior</i></p> <p><i>c) Primer registro</i></p> <p><i>d) Ultimo registro</i></p> <p><i>f) Ir a un registro</i></p>	
<p>5.-El usuario ejecuta la orden de impresión de reporte, eligiendo con el mouse el registro que desea imprimir.</p>	<p>4.- El sistema indica que se debe elegir un registro de la tabla de datos, para imprimir los datos. Permitiendo elegir el registro si este no ha sido elegido.</p>
<p>6.- El usuario introduce los datos necesarios para la generación del reporte.</p>	<p>5.-El sistema guarda los datos introducidos y espera la indicación de imprimir. Así presentará una vista previa para la verificación y permitir una corrección en caso de que ésta fuera necesaria.</p> <p>El cuerpo del reporte simplemente contendrá fechas tanto de nacimiento del paciente, como en la que se tomó el diagnóstico, el nombre del paciente y sus datos personales que se adquirieron cuando se registró el paciente y el diagnóstico.</p>

7.- El usuario hace petición de estadísticas	6.-El sistema despliega una lista especificando la cantidad total de diagnósticos, así también muestra la cantidad de diagnósticos positivos, negativos y cuántos de estos fueron de sexo masculino y cuántos de sexo femenino.
----------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

En la figura 2.3.4 podemos observar como se realiza el caso de uso de Presentación de Registros.

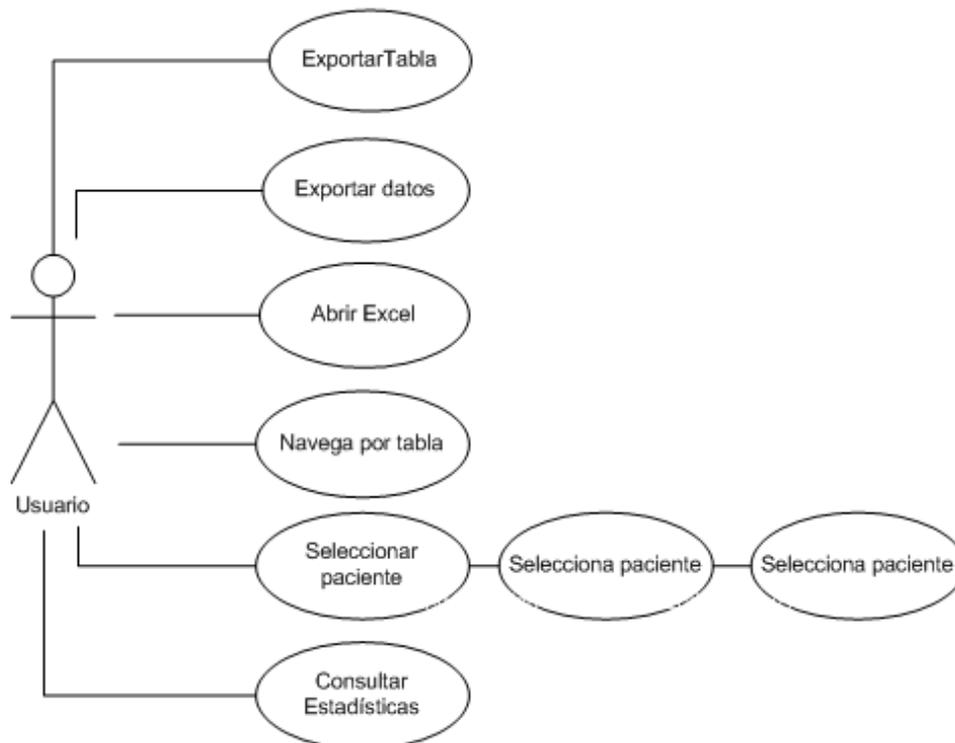


Figura 2.3.4. Caso de uso de Presentación de los datos.

Caso de uso de navegar

Usuario del sistema	Respuesta del sistema
1.- El usuario verifica el siguiente registro.	1.- El sistema recorre un registro de la base de datos y muestra sus campos.
2.- El usuario regresa un registro.	2.- El sistema regresa un registro en la base de datos y muestra sus campos.
3.- El usuario avanza 10 registros.	3.- El sistema recorre diez registros de la base de datos y muestra sus campos.
4.- El usuario regresa 10 registros.	4.- El sistema regresa diez registros en la base de datos y muestra sus campos.
5.- El usuario regresa a registro inicial	5.- El sistema regresa al registro inicial en la base de datos y muestra sus campos.
6.- El usuario avanza a registro final	6.- El sistema se posiciona en el registro final de la base de datos y muestra sus campos.
7.- El usuario escribe el número de registro al que desea ir.	7.- El sistema busca el archivo indicado y los despliega.

En la figura 2.3.5 podemos observar como se realiza el caso de uso de Navegar.

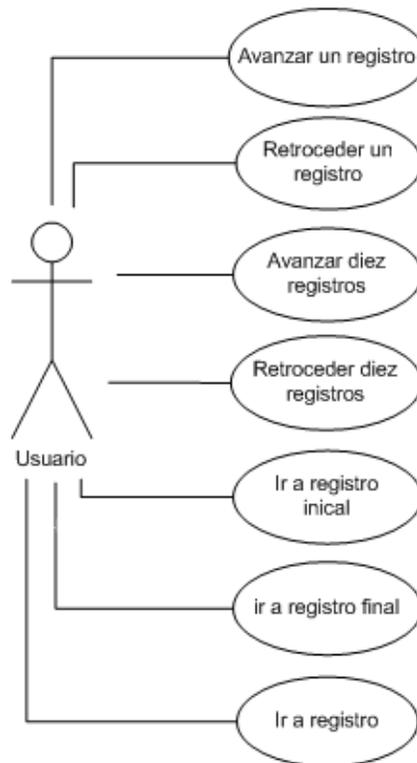


Figura 2.3.5. Caso de uso de Presentación de los datos.

DEFINICIÓN DE LAS TABLAS DE DATOS

Como ya se ha explicado, debido a las especificaciones del sistema, se ha optado por una Base de Datos, para el almacenamiento de los diagnósticos clínicos.

Ésta Base de Datos será creada al vuelo, es decir, cada vez que se instala el Sistema de Adquisición de Diagnósticos Clínicos y éste es ejecutado por primera, automáticamente será creada la Base de Datos en la que estarán contenidos los datos necesarios para el sistema.

La razón por la cual esta Base de Datos es creada de esta manera, es simplemente para evitar problemas de compatibilidad con las distintas versiones de Microsoft Access así también el programa de instalación del Sistema de Adquisición de Diagnósticos Clínicos será más pequeño y por lo tanto su instalación será más rápida. Otra razón es por que de esta manera, la Base de datos es creada, en una carpeta específica, la cual queda

relativamente oculta para el usuario de esta manera serán evitados problemas de direccionamiento y de violación de los campo de la Base de Datos.

Las tablas y registros de dicha tabla son descritos a continuación.

Se ha considerado la opción de dos tablas dentro de una base de datos, para guardar los datos que permitirán la generación del reporte de los diagnósticos clínicos. Esto se debe a que una parte de los datos proviene del instrumento y la otra deberá ser recolectada por la PC. Estas tablas contendrán los datos adquiridos, se relacionará entre ellas y formarán una consulta, la cual presentará los registros completos de un paciente, es decir, presentará el nombre del paciente, fecha de nacimiento, sexo, el diagnóstico adquirido, la hora y fecha en que éste diagnóstico fue realizado.

A continuación se describe la función de cada una de las tablas:

La primera tabla es Clínica, ésta deberá adquirir los datos que el microprocesador envía por “paquetes” de manera que, en una sola recepción serán enviados hasta un total de 100 diagnósticos, cada uno con su respectiva hora, fecha y número de diagnóstico. Y por seguridad ninguno de los registros de esta tabla podrá ser modificado

La segunda tabla se denomina tabla de General Paciente y será una tabla que podrá ser manipulada por la PC. En esta tabla se agregarán los datos del paciente como son nombre, sexo y fecha de nacimiento. Esta tabla permitirá hacer correcciones en cada uno de los campos guardados (nombre del paciente, sexo, fecha de nacimiento) en un registro, así como también podrán borrarse registros no deseados

En la siguiente imagen podemos ver como son creadas las tablas en Access:

The screenshot shows two tables in Microsoft Access. The top table, 'GeneralPaciente', has columns: NumPaciente, Nombre, FechaNac, and Sexo. The bottom table, 'Clínica', has columns: NumPaciente, Diagnostico, Hora, Fecha, and Indice. Both tables show three records.

NumPaciente	Nombre	FechaNac	Sexo
1	Juan Pérez	10/05/2005	Masculino
2	Ana Maria Juárez	12/05/2005	Femenino
3	Raúl Alcantar	27/05/2005	Masculino

NumPaciente	Diagnostico	Hora	Fecha	Indice
1	Positivo	10:35	01/06/2005	
2	Negativo	12:37	01/06/2005	
3	Invalido	11:18	01/06/2004	

Figura 2.3.6 Tablas Clínica y General Paciente.

La posibilidad de error entre estas tablas es nula ya que se relacionan por medio del número de paciente, de tal manera que, una vez agregado el número de paciente en la tabla General Paciente, este no podrá ser modificado al menos que el registro sea borrado, así como en la tabla de Clínica los números se generan automáticamente y por ninguna razón se modificarán.

La generación del número de paciente en la tabla de General Paciente es manual debido a que se ha pensado en que no todos los registros adquiridos necesitarán de un nombre del paciente ya que estos pueden ser simplemente pruebas para verificar el comportamiento del instrumento, pero aún así los diagnósticos se enviaron a la PC. Este registro se quedará contenido en la tabla de Clínica, pero si no se desea, no tendrá su respectivo registro en la tabla de General Paciente. Este registro al no presentar un paciente, no será mostrado en la consulta generada, es decir, cada uno de los registros en la tabla de Clínica deberá tener un registro en la tabla de General Paciente para que este pueda mostrarse en la tabla de consulta y pueda generarse un reporte de diagnósticos clínicos.

Estas tablas, tienen la virtud de poder ser borradas o bien respaldadas. Pueden borrarse si no es importante guardar los datos. De esta manera la base de datos no será saturada rápidamente. Se respaldarán cuando los registros sean de gran interés y se deseen consultar en un futuro.

Las tablas de respaldo funcionarán de igual manera que las tablas Clínica y General Paciente y los datos serán mostrados de la misma manera.

La tabla que guardará los registros de Clínica será nombrada Instrumento y la que guardará los datos de General Paciente se llamará Paciente.

Cada vez que un dato sea agregado a las tablas de respaldo, éste será colocado en el último registro de las tablas, de tal manera que los datos quedarán en cola. No obstante, la tabla de datos mostrada tendrá la opción de ordenar los datos de distintas maneras para que estos puedan ser visualizados fácilmente visualizados. Esta es una herramienta muy utilizada en Excel y será aplicada en las ventanas de visualización de datos del Sistema de Adquisición de Diagnósticos Clínicos.

2.4. ANÁLISIS DE LOS ELEMENTOS QUE CONSTITUIRÁN EL SISTEMA

VISUAL BASIC

Hoy en día existen millones de personas que utilizan el Sistema Operativo Microsoft Windows, por lo que se considera que Visual Basic.NET es una herramienta sencilla y potente para el desarrollo de aplicaciones para el Sistema Operativo Windows basándose en el lenguaje Basic (Beginners All-Purpose Symbolic Instruction Code) y la programación orientada a objetos.

Visual Basic.NET tiene la posibilidad de crear auténticos programas ejecutables e incluso la posibilidad de poder crear sus propios controles. Es además una versión muy simple y potente.

Visual Basic.NET permite la incorporación de menús, cuadro de texto, botones de orden, botones de opción, casillas de verificación, cuadro de lista, barras de desplazamiento y cuadros de archivos y directorios para ventanas en blanco. Se pueden utilizar cuadrículas para manejar datos tabulados para comunicarse con otras aplicaciones del Sistema Operativo Windows. A todos estos componentes se les denomina controles.

Visual Basic.NET permite crear programas para uso personal, para un grupo de trabajo, para una empresa, aplicaciones distribuidas a través de Internet, aplicaciones de bases de datos y otras aplicaciones más.

Visual Basic.NET proporciona herramientas que permiten crear ventanas y controles sin escribir código. También incluye un entorno de desarrollo que permite ejecutar todas las tareas de edición, ejecución y mantenimiento de programas de forma fácil y completa, lo que permitirá solucionar cualquier duda que surja mientras se crea una aplicación. Todo esto hace posible que en muy poco tiempo se puedan escribir programas simples

La mayoría del código de programación que indica al programa cómo responder a eventos tales como pulsaciones del ratón, suceden en lo que en Visual Basic.NET se denominan procedimientos de eventos, que no es más que el código necesario para indicar a Visual Basic.NET cómo responder a un evento. Los programas dirigidos por eventos son más reactivos que activos, y eso los hace más amigables para el usuario.

Para desarrollar una aplicación, se crean ventanas y sobre ellas se dibujan controles (etiquetas, botones, cajas de texto, etc.). Cuando se haya terminado de dibujar la interfaz, los botones, cuadros de texto y otros controles que se hayan colocado en la pantalla reconocerán automáticamente las acciones del usuario, como movimientos del ratón y pulsaciones de los botones. Y a continuación se escribe el código fuente relacionado con cada objeto. Esto es, cada objeto está ligado a un código que permanece inactivo hasta que se dé el evento que lo activa (por ejemplo, presionando el botón del ratón)

Una vez diseñada la interfaz, es posible trabajar como en la programación tradicional. Los objetos en Visual Basic.NET reconocen eventos tales como pulsaciones de ratón, la forma en que los eventos respondan a ellos depende del código que se escriba. Un programa Visual Basic.NET es un conjunto de fragmentos independientes de código que son activados mediante eventos y solamente responden a los eventos que se han indicado que reconocerán.

Los siguientes pasos son la búsqueda de errores en el código y su posterior corrección.

CARACTERÍSTICAS DE VISUAL BASIC.NET

Visual Basic.NET incluye como características más sobresalientes las siguientes:

- Una biblioteca de clases que da soporte a los objetos Windows tales como ventanas, cuadros de diálogo, etc. controles (por ejemplo, etiquetas, cajas de texto, botones de pulsación, etc.)

- Asistentes de desarrollo de aplicaciones, barras de herramientas, formularios de empaquetado y distribución, creación de la interfaz pública de controles ActiveX, páginas de propiedades, objetos de datos y generador de clases.
- Una interfaz para múltiples documentos MDI (Multiple Document Interface) que permite crear una aplicación con una ventana principal y múltiples ventanas de documento. Un ejemplo de este tipo de aplicaciones es Microsoft Word.
- Editar y continuar. Durante una sesión de depuración, se pueden realizar modificaciones en el código de la aplicación sin tener que salir de dicha sesión.
- Creación y utilización de bibliotecas dinámicas DLL (Dynamic Link Libraries).
- Acceso a bases de datos a través del control de datos ADO (Active Data Object), utilizando el motor de Access o controladores ODBC (Open Database Connectivity).
- Acceso a bases de datos utilizando OLE DB (Object Linking and embedding for databases) como un proveedor de datos objetos ADO, como tecnología de acceso a datos.
- Biblioteca SQL (Structured Query Language) que permite manipular bases de datos relacionales, tales como Microsoft Access.
- Un administrador visual de datos para manipular datos.

Cuando se combinan estas características, se dispone de un sistema de desarrollo que permite diseñar rápidamente aplicaciones sofisticadas.

MICROSOFT ACCESS

Microsoft Access es un sistema de gestión de bases de datos relacional. Los programas como Access, que es un programa de bases de datos relacionales, permiten separar los datos en tablas diferentes y almacenar los distintos elementos una única vez. Para después relacionar tablas ente sí. El almacenamiento de datos puede ser más flexible y los datos no tienen que estar duplicados.

Su funcionamiento se basa en un motor llamado Microsoft Jet, y permite el desarrollo de pequeñas aplicaciones formadas por formularios Windows y código Visual Basic.

Tiene un límite de 1.2 gigabytes de datos por base de datos. Tratándose de una base de datos de escritorio. Tiene un límite de 255 usuarios totales.

Access es algo más que una base de datos relacionales. También es un ambiente completo de desarrollo de aplicaciones. Teniendo como herramientas las siguientes:

Tablas	Una tabla es un conjunto de registros que se pueden dividir en campos. Cada campo conserva una sola parte de la información en relación con el registro en el cual se encuentra.
Consultas	Se utilizan para buscar, ver y modificar los datos incluidos en las tablas. También se pueden usar las consultas para modificar la estructura de las tablas.
Formularios	Estos pueden utilizarse para escribir, editar y buscar sus datos. Una vez creado un formulario, tendrá la apariencia y funcionará como una aplicación de Windows.

Informes	Permiten enviar datos a cualquier tipo de sitio en un formato fácil de leer. Los informes utilizan una fuente de datos subyacente, ya sea una tabla o una consulta.
Macro y módulos	Son un conjunto de acciones en el cual cada una ejecuta una tarea específica. Los módulos son conjuntos de procedimientos de Visual Basic for Applications. Access incluye un lenguaje de programación incorporado, esencialmente idéntico a las aplicaciones de Visual Basic.NET.

VENTAJAS

- Tiene gran facilidad para definir datos, tales como texto, números, fechas, horas, monedas, imágenes, sonidos, documentos, hojas electrónicas. Así también permite definir la forma en que se almacenarán los datos (longitud de cadena, precisión de los números, precisión de la fecha/hora), y para definir la apariencia de los datos cuando se visualicen o se impriman.
- Por ser una aplicación para Microsoft Windows podemos utilizar todas las posibilidades de Intercambio Dinámico de Datos (DDE) y de la incrustación y vinculación de objetos OLE. DDE permite ejecutar funciones y enviar datos entre Access y cualquier otra aplicación basada en Windows que soporte DDE.
- OLE permite incluir tablas, gráficos, o imágenes desde otras aplicaciones.
- Puede acceder y actualizar directamente archivos de Paradox, DBASE III, Fox Pro etc.
- Puede trabajar con la mayoría de las bases de datos más populares que soportan la normalización (ODBC), incluyendo el Servidor SQL de Microsoft, Oracle; DB2 y RDB.
- Utiliza el lenguaje de bases de datos SQL para procesar datos de las tablas.

- Access está diseñado para su utilización como un sistema de gestión de bases de datos independiente en una única estación de trabajo o en un modo cliente servidor compartido a través de una red.
- Posee un excelente sistema de seguridad de los datos y de integridad de los mismos.

DESVENTAJAS

Entre sus mayores inconvenientes figuran que no es multiplataforma, pues sólo está disponible para sistemas operativos de Microsoft, y que no permite transacciones. Su uso es inadecuado para grandes proyectos de software que requieren tiempos de respuesta críticos o muchos accesos simultáneos a la base de datos.

ADO.NET

Es un conjunto de interfaces, clases, estructuras y enumeraciones que permiten el acceso a datos en la plataforma Microsoft .NET

ADO.NET incluye proveedores de datos de .NET Framework para conectarse a una base de datos, ejecutar comandos de recuperación de datos, los procesa directamente o los coloca en un objeto DataSet con el fin de exponerlos al usuario o para un propósito específico.

Permite acceso coherente a bases de datos tales como SQL Server y Oracle así como a los orígenes de datos mediante proveedores como lo son OLE DB y ODBC.

COMPONENTES DE ADO.NET

ADO.NET tiene dos componentes principales que son: el DataSet y el proveedor de datos de .NET Framework, los cuales están diseñados para separar el acceso a datos de la manipulación de datos.

EL DATASET

El DataSet es una memoria de registros recuperados de una base de datos que actúa como un sistema de almacenamiento virtual lógico, y que contiene una o más tablas basadas en las tablas reales de la base de datos. Adicionalmente, almacena las relaciones y reglas de integridad existentes entre ellas para garantizar la estabilidad e integridad de la información de la base de datos.

El DataSet se compone de clases de soporte, que representan cada una, los elementos de la base de datos: tablas, columnas, filas, sus reglas de chequeo, sus relaciones, las vistas asociadas a la tabla, etc.

Cabe mencionar, que los DataSets son almacenes pasivos de datos, esto es, no se ven alterados ante cambios que se realizan después de llamar a la base de datos. Es necesario recargarlos siempre que queramos estar *al día*, en cuanto a datos se refiere.

Aquí se tiene el mejor concepto de datos desconectados: una copia en el cliente de la arquitectura de la base de datos, basada en un esquema XML (**eXtensible Markup Language**) que la independiza del fabricante, proporcionando al desarrollador la libertad de trabajo independiente de la plataforma.

Una consecuencia lógica de este tipo de arquitecturas, es la de conseguir que los DataSets sean independientes de los orígenes de datos. Los drivers OLE DB transformarán la consulta SQL en un cursor representado con una estructura XML, que es independiente del motor de la base de datos.

Esto nos permitirá trabajar con múltiples orígenes de datos, de distintos fabricantes e incluso en formatos que no pertenezcan a bases de datos, por ejemplo, ficheros planos u hojas de cálculo, lo que representa un importante punto de compatibilidad y flexibilidad.

La siguiente figura muestra el esquema de un DataSet

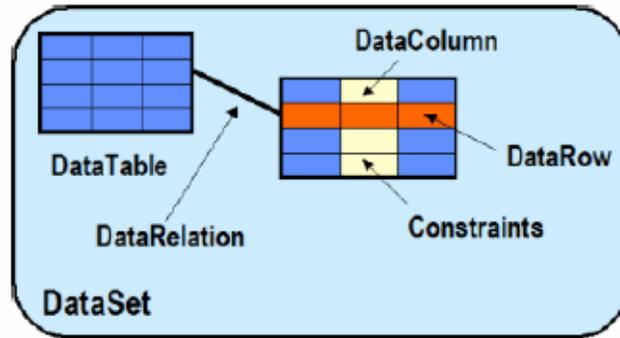


Figura 2.4.1 Esquema de un DATASET.

El formato que emplea ADO .NET para almacenar su estado es XML. Puesto que ya es un estándar de la industria, esta persistencia nos ofrece la siguiente cualidad: la información puede estar accesible para cualquier componente del sistema que entienda XML.

EL PROVEEDOR DE DATOS DE .NET FRAMEWORK

Un proveedor de datos en .NET Framework sirve de puente entre una aplicación DataSet y un origen de datos, es decir, permite conectarse a un origen de datos para recuperar y modificar datos del origen de datos, así mismo, reflejar en ese origen los cambios que se produzcan en los datos.

Los proveedores de datos son los siguientes

- Proveedores de .NET Framework para SQL Server

Se utiliza para bases SQL Server 7.0 o SQL Server 2000.

- Proveedores de .NET Framework para OLE DB

Es uno de los dos proveedores de datos que se instalan junto con .NET Framework. Este proveedor resulta apropiado para las bases de datos como Microsoft SQL Server anteriores a la versión 7 y con bases Oracle así como con cualquier base que utilice el motor Jet 4.0 tales como Access.

- Proveedores de .NET Framework para ODBC

Se utiliza para bases de datos que requieran del proveedor OLE DB para ODBC de Microsoft sirve para trabajar bases de datos como Microsoft SQL Server anteriores a la versión 7 y con bases Oracle.

- Proveedores de .NET Framework para Oracle

Resulta apropiado para bases de datos Oracle. Se recomienda para versiones 8.1.7 y posteriores.

Los proveedores de datos de .Net Framework para ODBC y Oracle no están incluidos en la versión 1.0 de .NET Framework. Si se desean utilizar estos pueden ser descargados de la página de Microsoft.

En la tabla siguiente se describen los cuatro objetos centrales que constituyen un proveedor de datos de .NET Framework.

Objeto	Descripción
Connection	Establece una conexión a un origen de datos determinado.
Command	Ejecuta un comando en un origen de datos.
DataReader	Lee una secuencia de datos de sólo avance y sólo lectura desde un origen de datos.
DataAdapter	Llena un DataSet y realiza las actualizaciones necesarias en el origen de datos.

Es un programa que crea instalaciones profesionales en minutos, en un solo archivo y trabaja con las versiones del Sistema Operativo Windows.

Pertenece a la empresa Indigo Rose Software.

Setup Factory 7.0 esta basado en el lenguaje LUA, el cual es un lenguaje de escritura con toda la programación lógica necesaria para la creación de instaladores de cualquier aplicación.

CARACTERÍSTICAS

- Es una herramienta que trabaja bajo un ambiente gráfico, de manera que su uso resulta sencillo. Incluye radio botones, archivos que pueden ser editados, así como pantallas de acuerdo de licencias, números seriales de verificación, entre otras cosas.
- La ventana principal otorga un control sobre todas las pantallas en tiempo real de tal manera que los cambios pueden verse conforme se van ejecutando. No contiene un desempacador esto permite que la inicialización e instalación de cualquier archivo sea más rápido que con otros instaladores.
- La instalación se lleva a cabo mediante un ayudante el cual permite que el proceso se lleve a cabo dando unos cuantos clicks. Únicamente se genera un archivo ejecutable el que tiene todas las herramientas para su distribución la cual puede ser mediante Web, e-mail, LAN, CD, DVD y hasta en Floppy.
- Tiene docenas de fondos para la ventana de instalación como también que el usuario pueda crear el suyo propio usando la imagen de su preferencia o bien un color en específico. Puede elegirse el papel tapiz de distintas formas, tales como gradientes, imágenes, colores, textos de encabezados y pie de página con efectos 3D. Se puede configurar la fuente (color, tamaño, estilo).

- El programa consta de 250 librerías prediseñadas las cuales permiten una poderosa funcionalidad para hacer cualquier programa. Para realizar una acción simplemente basta con seleccionarla de la categoría de acciones y el ayudante se encargará de hacer alguna característica básica.
- Puede ejecutar funciones DLL, manipular cadenas, copiar archivos, iniciar o finalizar servicios, interactuar con páginas Web, desplegar cuadro de texto, etc.
- Incluye los runtimes para las versiones más populares tales como el runtime de Visual Basic 5, Visual Basic 6, .Net Framework necesario para aplicaciones de Visual Basic.Net, JET, MDAC (Microsoft Data Access Components) entre otros. También permite crear y agregar módulos de dependencias. Da un completo control sobre la detección e instalación de cada uno de los módulos, así puede ajustarse a las necesidades específicas.
- Cuenta con un desinstalador que es muy sencillo el cual simplemente se carga y el desinstalador se encarga de borrar iconos, archivos, módulos, librerías DLL colocados por el mismo instalador.
- El instalador puede cambiar el idioma para ajustarlo a las necesidades de los usuarios puede soportar Inglés, Francés, Español, Alemán, Italiano y también puede agregarse módulos de idiomas.
- Puede crear archivos de reporte de errores y script.

2.5. ANÁLISIS DE LOS REQUERIMIENTOS DE HARDWARE Y SOFTWARE

A Continuación se presenta de forma detallada los requerimientos del equipo que se debe utilizar para trabajar sin ningún problema con el sistema de adquisición de Diagnósticos Clínicos.

HARDWARE

Procesador Pentium II de 450 MHz, AMD K6-II de 450 MHz o equivalente,

Disco duro 100 MB de espacio libre.

Memoria RAM 128 MB

Unidad CD-ROM

Monitor Super VGA (800 x 600) con una resolución de 256 colores.

Mouse

SOFTWARE

Sistema Operativo Windows 98 o superior

Office XP (Access y Excel)

Acrobat Reader 5 o superior.

3.- DISEÑO DEL SISTEMA

Para el diseño del sistema de Adquisición de Diagnósticos Clínicos, se utilizaron diagramas de casos de uso, diagramas de flujo de datos, así como también, diagramas de entidad relación, contando con el apoyo de las herramientas de UML (Unified Modeling Language) para dicho fin.

Los casos de uso han sido tratados en el capítulo anterior, con base en ellos se realizarán los diagramas de flujo de datos los cuales se explicarán a continuación.

DIAGRAMAS DE FLUJO DE DATOS

Los diagramas de flujo de datos, son representaciones gráficas de un conjunto de elementos que han sido empleados para las especificaciones de sistemas de información. Son herramientas muy frecuentes utilizadas para el análisis estructurado y el manejo de los datos de una forma eficiente.

El diagrama del flujo de datos, es una técnica que representa el flujo de la información y las transformaciones que se aplican a los datos, al moverse desde la entrada hasta la salida..

El diagrama de flujo de datos es un modelo que describe:

- Los lugares de origen y destino de los datos (los límites del sistema),
- Las transformaciones a las que son sometidos los datos (los procesos internos),
- Los lugares en los que se almacenan los datos dentro del sistema, y
- Los canales por donde circulan los datos.

El diagrama del flujo de datos posee niveles de desagregación. El Nivel 0 o Diagrama de Contexto es aquel que muestra una sola burbuja y las entidades externas o terminadoras con los que interactúa el sistema.

DESCRIPCIÓN DE CADA UNO DE LOS ELEMENTOS DE UN DIAGRAMA DE FLUJO DE DATOS

Entidad Externa:

Son generalmente clases lógicas de cosas o de personas, las cuales representan una fuente o destino de transacciones, como por ejemplo clientes, empleados, proveedores, etc., con las que el sistema se comunica.

Se representa de la siguiente manera:

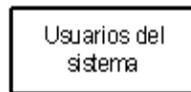


Figura 3.1. Entidad externa.

Proceso:

Son las funciones o procesos que transforman entradas de datos en salidas de información.

Su nombre deberá ponerse mediante una frase imperativa, que consistirá idealmente de un verbo activo seguido por una cláusula objeto, cuanto mas simple mejor.

Los procesos se relacionan con:

- Almacenes
- Entidades externas
- Otros procesos
- Deberán tener al menos una Entrada y una Salida

En este caso se representará con la siguiente figura:

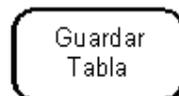


Figura 3.2. Proceso.

Flujo de datos:

Representa un transporte de paquetes de datos desde su origen hasta su destino, es decir, que representa una estructura de datos en movimiento de una parte del sistema a otro.

Un flujo muestra las interfaces entre los elementos del diagrama de flujo de datos.

La flecha indica la dirección del flujo.

- Flecha que llega o sale de un proceso.
- Describe el movimiento de bloques o paquetes de información de una parte del sistema a otro.
- El nombre representa el significado del paquete (un solo tipo de paquetes) que se mueve por el flujo de datos.
- Tiene dirección.
- Flujos divergentes (copias o división en paquetes más simples) y convergentes (unión para formar paquetes más complejos). No los usamos.
- No responde a dudas de procedimiento.



Figura 3.3. Flujo de datos.

Almacén o archivo:

Representa un archivo lógico en donde se agregan o de donde se extraen datos. Es una estructura de datos, pero estática.

Puede ser físicamente un archivo de tarjetas, una microficha, un archivo, o un archivo en cinta o diskette.

- Implica escritura, actualización o borrado de datos.

- Implica lectura o recuperación de información almacenada
- Modela una colección de paquetes en reposo.
- Puede ser requerimiento fundamental o aspecto conveniente de la realización del sistema.
- Ejemplos de aspectos convenientes de interponer un almacén entre dos procesos:

Se representa de la siguiente manera:

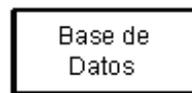
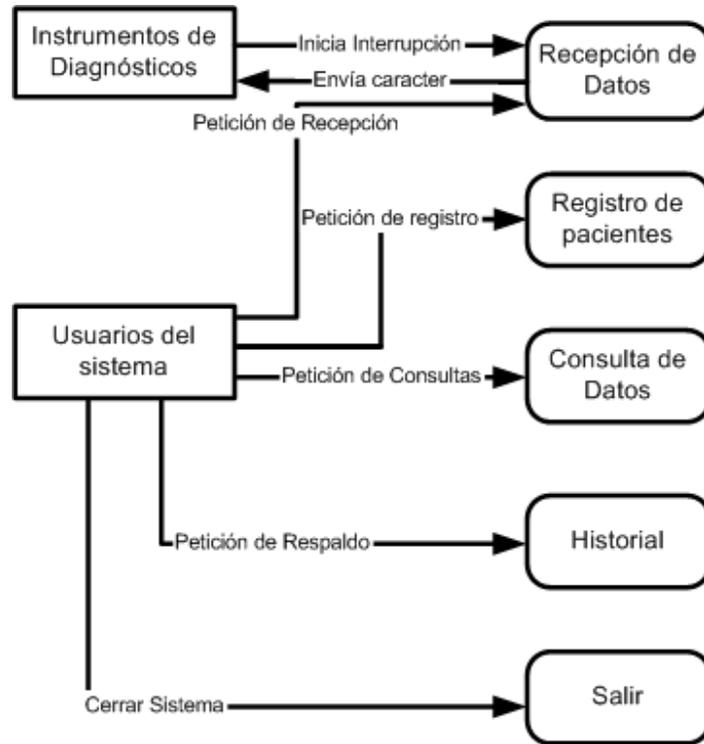


Figura 3.4. Almacén.

DIAGRAMA DE CONTEXTO

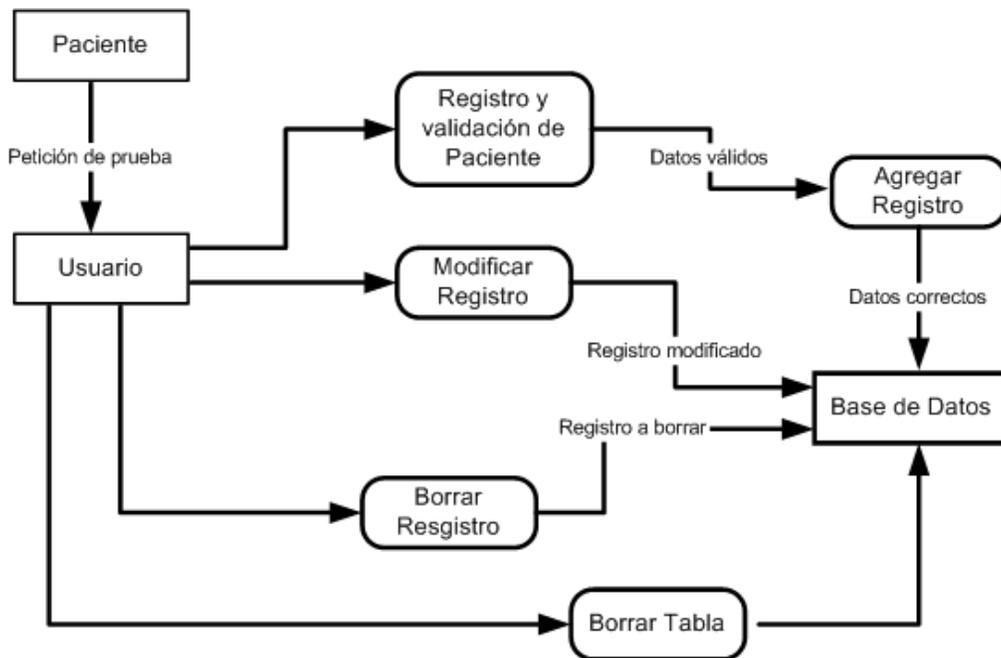
Es el modelo de procesos más general que puede utilizarse en un sistema o aplicación. El sistema o aplicación se representa como un único proceso. Sus interacciones con la empresa, otros sistemas y el mundo exterior se dibujan como flujos de datos de entradas y salidas.

A continuación se presenta el diagrama de contexto del Sistema de Adquisición de Diagnósticos Clínicos.



Cada proceso se explicará en un diagrama:

REGISTRO DE PACIENTES



RECEPCIÓN DE PACIENTES

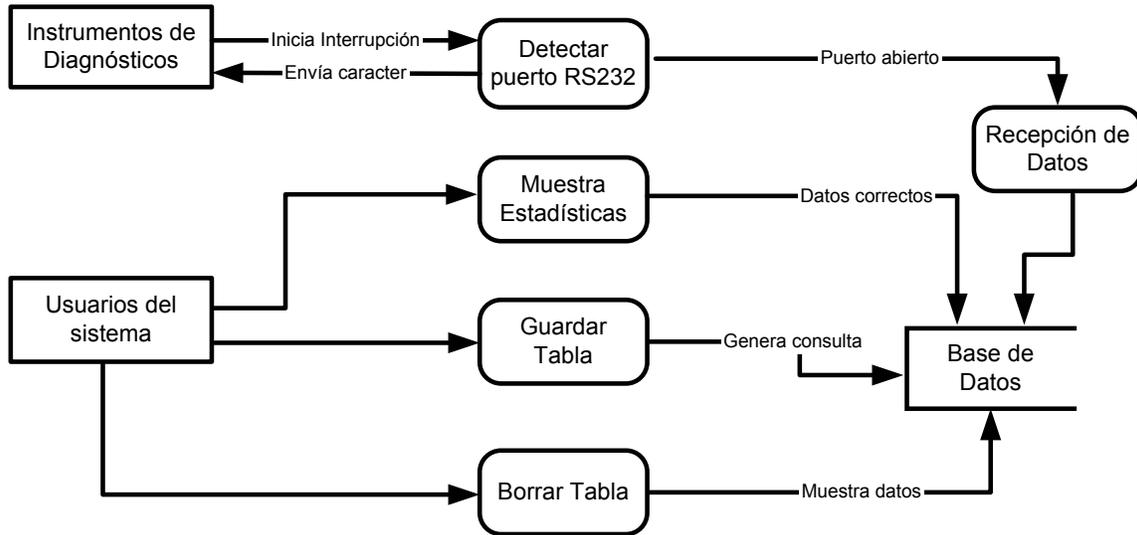


DIAGRAMA DE PRESENTACIÓN DE DATOS

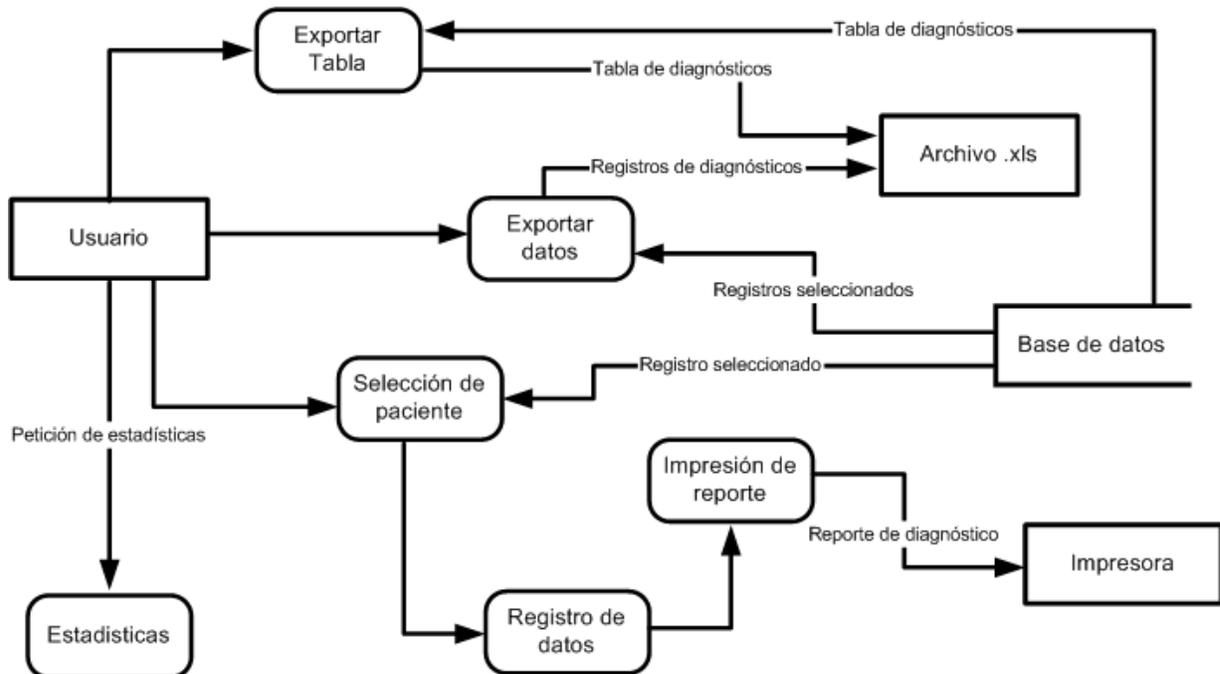


DIAGRAMA DE RESPALDOS

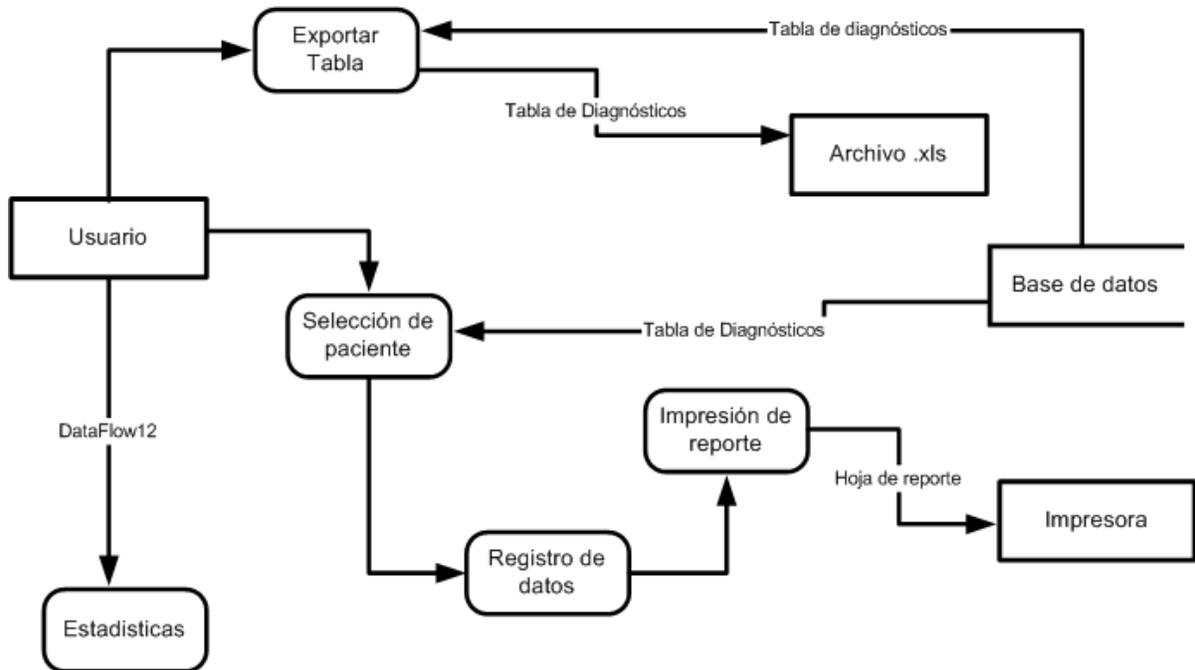


DIAGRAMA DE FUNCIÓN GUARDAR



DIAGRAMA DE FUNCIÓN BORRAR



DIAGRAMAS DE ENTIDAD RELACIÓN

El modelo entidad-relación aporta una herramienta de modelado para representar las entidades, relaciones y atributos.

Su objetivo es proporcionar un modelo independiente de cualquier almacenamiento de datos y método de acceso.

El modelo entidad-relación se representa por un conjunto de entidades que presentan ciertas propiedades llamadas atributos, definidos sobre un cierto dominio de datos. Las entidades se vinculan mediante relaciones que, en ciertas variantes de la notación, pueden también tener sus propios atributos. En principio, estas relaciones pueden ser n-arias, pero en la práctica se trabaja con relaciones binarias.

Los diagramas se representan por medio de la siguiente simbología:

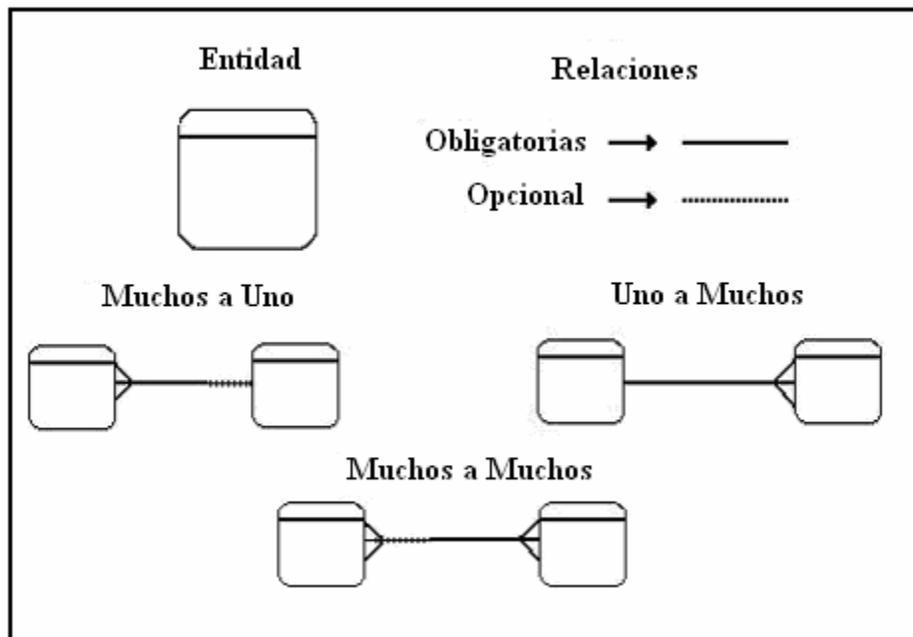


Figura 3.5. Simbología de lo diagramas de entidad relación.

DIAGRAMA DE ENTIDAD RELACIÓN PRIMARIA

Este diagrama representa la relación existente entre las tablas Clínica y General Paciente para la construcción de la Consulta de Diagnósticos, la cual se presentará en la ventana de Tabla de Datos.

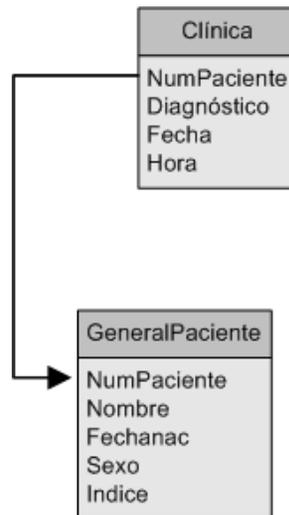
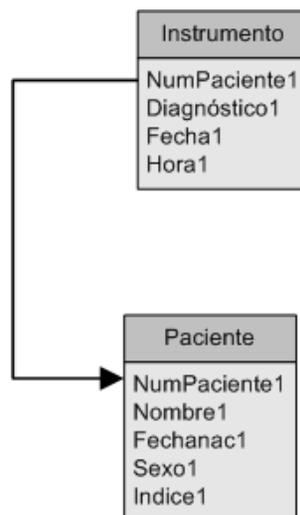


DIAGRAMA DE ENTIDAD RELACIÓN SECUNDARIA

Este diagrama representa la relación existente entre las tablas Instrumento y Paciente para la construcción de la Consulta de Historial, la cual se presentará en la ventana de Historial.



4. DESARROLLO DEL SISTEMA DE ADQUISICIÓN DE DIAGNÓSTICOS CLÍNICOS

DESCRIPCIÓN DEL AMBIENTE

El sistema de Adquisición de Diagnósticos Clínicos tiene como objetivo simplificar la toma de diagnósticos tanto en un hospital, como en un Laboratorio Clínico o cualquier lugar en que se tenga acceso a dicho instrumento. Este sistema es un sistema de propósito específico ya que solamente funciona con el instrumento de Diagnósticos Clínicos.

El ambiente del Sistema de Adquisición de Diagnósticos Clínicos es un ambiente gráfico y de fácil entendimiento y manejo.

A continuación se explicará como funciona cada una de las ventanas que conforman el sistema y como debe utilizarse para para que presente un funcionamiento adecuado.

El sistema se divide en las siguientes ventanas y se deberán trabajar en el siguiente ambiente:

- + Ventana principal
 1. Registro de Pacientes
 2. Recepción de pacientes
 3. Tabla de datos
 4. Historial

VENTANA PRINCIPAL

Es la ventana que aparece cuando se inicializa el programa.

Su función principal es la de poder contener todas las ventanas del sistema.

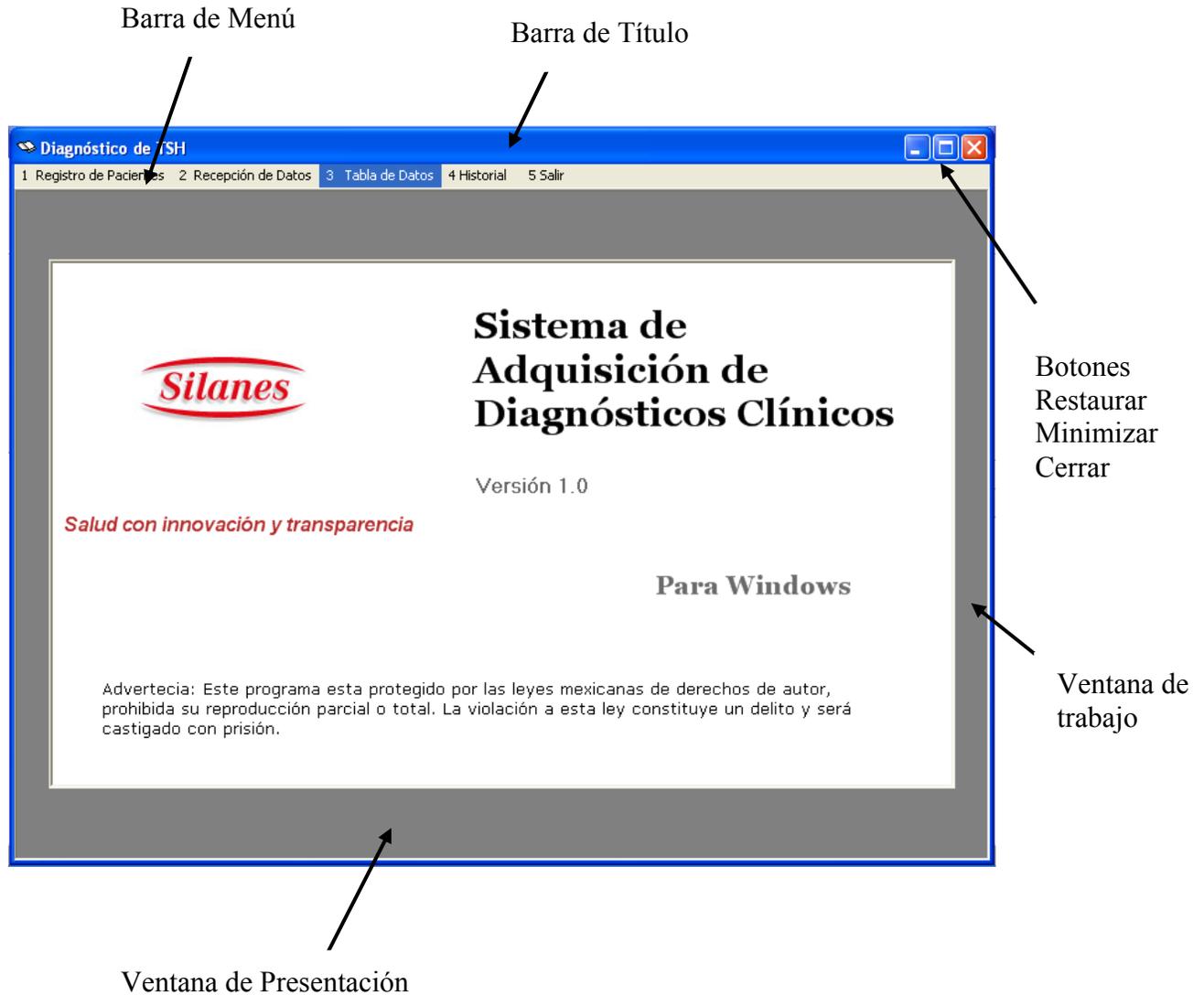


Figura 4.1. Ventana Principal.

Los elementos de la ventana principal son los siguientes:

- **BOTONES DE MINIMIZAR, RESTAURAR Y CERRAR**

Los botones de Minimizar, Restaurar y Cerrar aparecen en la esquina superior derecha de la ventana.

La función de estos botones es la misma que para cualquier aplicación Windows y es la siguiente:

Botón	Descripción
	El botón Minimizar reduce la ventana del sistema a un ícono.
	Restaurar permite llevar la ventana del sistema a pantalla completa y viceversa.
	Cerrar permite salir del sistema.

Cada uno de los botones se puede elegir presionando el botón izquierdo del ratón.

- **BARRA DE TÍTULO**

La Barra de Título aparece en la parte superior de la ventana.

- **BARRA DE MENÚ**

Aparece debajo de la barra de título.

La Barra de Menú es el centro de comandos de Sistema de Adquisición de Diagnósticos Clínicos. Cada opción en la Barra de Menú llama a un menú de ventanas que es utilizado para manipular y trabajar tanto en la adquisición de los datos como en la edición de los mismos.

Las funciones de la barra de menú son las siguientes:

Opción	Descripción
1 Registro de pacientes	Muestra la tabla Registro de Pacientes, en la cual pueden darse de alta, modificarse o borrar los registros contenidos en la tabla de datos.
2 Recepción de datos	Muestra la ventana Recepción de datos, en la cual llegan cada uno de los registros tomados por el instrumento.
3 Tabla de Datos	<p>Muestra la ventana Tabla de Datos, en la cual se pueden observar los registros de forma personalizada, es decir, puede verse el nombre del paciente y su respectivo diagnóstico en la misma tabla de datos, así como cada uno de los datos adicionales como son fecha de nacimiento, sexo, hora y fecha en que se tomó el diagnóstico.</p> <p>En esta ventana pueden imprimirse reportes clínicos de cada uno de los registros, guardar estos registros en tablas de archivos de tipo Excel, así como abrir archivos de tipo Excel que han sido guardados anteriormente.</p>
4 Historial	Muestra la ventana Historial, es una ventana similar a la Tabla de Datos con la diferencia es que ésta muestra todos los registros que han sido respaldados desde el primer uso del instrumento.
5 Salir	Sale del programa.

Esta ventana puede ser manipulada mediante el teclado de manera que para poder acceder a cada parte del menú, basta con presionar Alt + Número, por ejemplo si se deseara acceder a la ventana Historial, simplemente se presiona Alt + 4 y la ventana aparecerá.

- **VENTANA DE TRABAJO**

Aparece abajo de la Barra de Menú.

La Ventana de Trabajo es el área en donde el Sistema de Adquisición de Diagnósticos Clínicos despliega las ventanas de manipulación de los datos del instrumento, muestra las ventanas para registrar a los pacientes y las tablas de registros finales. Así también, es el área en la que también puede ser desplegados los cuadros de diálogo que el sistema utiliza para solicitar o mostrar información al usuario.

- **VENTANA DE PRESENTACIÓN**

Aparece dentro de la ventana de Trabajo.

Esta ventana contiene el logotipo de la empresa que tiene los derechos del software, el nombre del sistema, la versión, la especificación de software así como una advertencia de uso, la cual se cumplirá en caso de ser ignorada y a continuación es mostrada.

VENTANAS SECUNDARIAS

Es la ventana que aparece cuando se inicializa el programa, esta ventana contiene todas las ventanas del sistema.

Una vez que se inicializa el sistema aparecerá la ventana principal la cual tendrá contenida a la ventana de presentación, ésta aparecerá unos instantes y desaparecerá, instantes después aparecerá la ventana de Registro de Pacientes.

Registro de pacientes es la ventana que aparece por omisión, pero no necesariamente es la ventana con la que se tiene que trabajar, se puede cambiar la ventana presionando el título

en la barra de menú el nombre de la ventana con la que se desea trabajar y se hace el cambio de ventana.

a) REGISTRO DE PACIENTES

Esta ventana se encarga de agregar los registros de los pacientes. En esta ventana se puede dar de alta datos tales como nombre del paciente, número de paciente, sexo y fecha de nacimiento.

Esta ventana se divide en tres partes:

- Registro de Pacientes
- Barra de navegación
- Edición de Registro de Pacientes

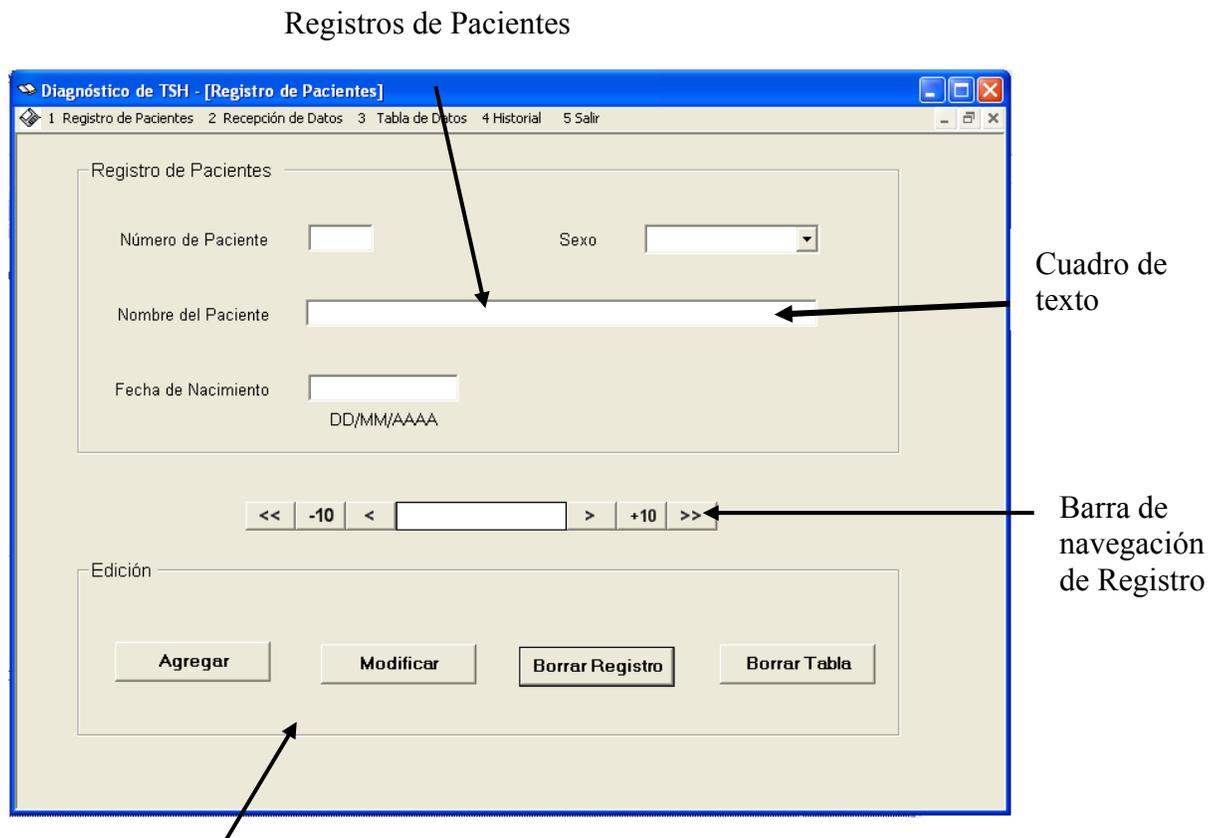


Figura 4.2. Ventana de Registro de Pacientes.

Ahora se describe cada una de las partes:

a.1) REGISTRO DE PACIENTES

Esta sección es en la que se escriben los datos que se van a agregar a una tabla contenida en memoria. Cada cuadro de texto deberá ser llenado con los datos que se requieren.

De igual manera ésta sección sirve para visualizar los datos ya contenidos en la base de datos, esto se hace mediante la barra de navegación.

a.2) BARRA DE NAVEGACIÓN

La barra de navegación como su nombre lo indica sirve para navegar en los registros que están contenidos en memoria. Ésta barra consta de un cuadro de texto y los botones de navegación.

El cuadro de texto indica el número de registro del paciente, el cual es presentado en la sección de Registro de pacientes y el número total de registros que contiene la tabla.

A continuación se presenta la descripción del cuadro de texto:

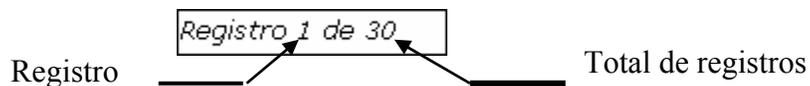


Figura 4.3. Cuadro de texto de Registro de Pacientes.

A continuación se describen los botones de navegación.

Botón	Descripción
	Botón de inicio. Regresa al primer registro que se agregó en la tabla.
	Botón -10. Regresa diez registros a partir del registro en el que se encuentra en ese momento.
	Botón regreso. Regresa un registro.
	Botón de fin. Muestra el último de los archivos que está contenido en la tabla.
	Botón +10. Avanza diez archivos a partir del registro en que se encuentra en ese momento.
	Botón de avance. Avanza un registro.

a.3) EDICIÓN DE REGISTRO DE PACIENTES

Esta sección se encarga de cambiar, agregar o borrar datos que están contenidos en el cuadro de texto de la sección Registro de pacientes. Los botones son los siguientes:

Botón	Descripción
<p data-bbox="289 310 407 342">Agregar</p>	<p data-bbox="699 333 1406 478">Agrega a la tabla de datos, los registros que están contenidos en los cuadros de texto de la sección de recepción de pacientes.</p>
<p data-bbox="282 562 414 594">Modificar</p>	<p data-bbox="691 588 1406 732">Guarda los cambios que se realizaron en los cuadros de texto, en la tabla de datos, esto es debido a que pueden cometerse errores al momento de agregar.</p> <p data-bbox="691 749 1406 842">Nota: los datos que pueden ser cambiados son: nombre, sexo y fecha de nacimiento.</p>
<p data-bbox="241 926 454 957">Borrar Registro</p>	<p data-bbox="691 947 1354 978">Borra los registros no deseados en la tabla de datos.</p>
<p data-bbox="263 1066 436 1098">Borrar Tabla</p>	<p data-bbox="691 1092 1406 1341">Borra toda la tabla de datos completa. Al presionar este botón aparecerá un cuadro de diálogo con la opción el cual preguntará si se desean guardar los cambios. Si presiona la opción “Si” éste enviará los datos a la tabla de respaldos.</p>

El sistema de adquisición de datos tiene que seguir una serie de pasos para su buen funcionamiento, por lo que el primer paso es el de Registrar los pacientes. Este registro se hace de forma simultánea con el instrumento.

El registro de pacientes se hace de la manera siguiente:

- Realizar la prueba con el instrumento. Cada vez que se toma un diagnóstico clínico aparecerá en la pantalla del instrumento el número de registro, diagnóstico, fecha, hora, etc.

- Verificar el número de diagnóstico. Ese mismo número deberá ser escrito en el cuadro de texto cuyo título es el de Número de paciente en la ventana de registros de pacientes dentro del sistema.

Por ejemplo en la pantalla del instrumento tenemos los siguientes datos:



Figura 4.4. Display de datos del Instrumento.

En la ventana del Registro de pacientes los datos se colocarán de la siguiente manera:

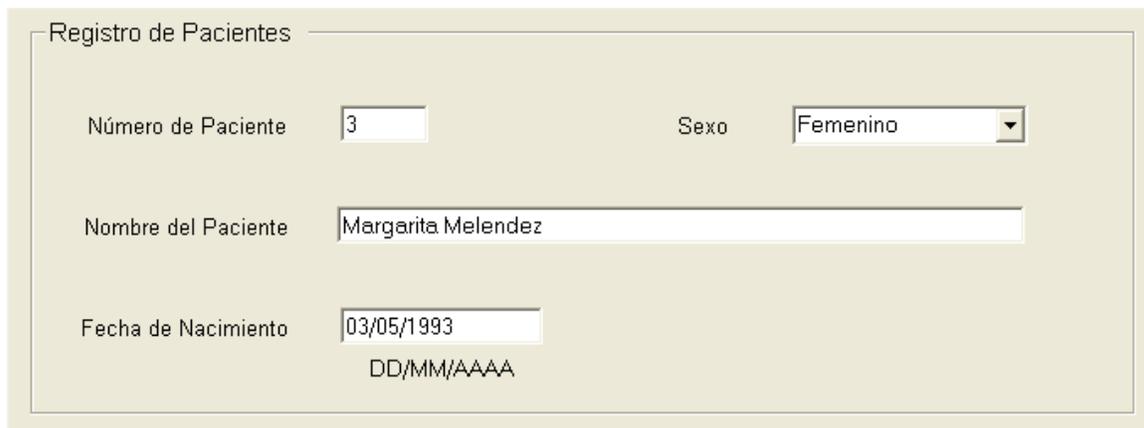
A screenshot of a software window titled "Registro de Pacientes". The window has a light beige background and contains several input fields. The first row has "Número de Paciente" with a text box containing "3" and "Sexo" with a dropdown menu showing "Femenino". The second row has "Nombre del Paciente" with a text box containing "Margarita Melendez". The third row has "Fecha de Nacimiento" with a text box containing "03/05/1993" and the label "DD/MM/AAAA" below it.

Figura 4.5. Registro de Pacientes.

En este caso es importante escribir el número de paciente. Esto es debido a que no es necesario colocar un registro de paciente por cada diagnóstico, es decir, habrá diagnósticos que no se necesiten almacenar o bien que son de prueba, pero aún así, estos diagnósticos son enviados al sistema, pero si no se guarda un nombre o número de paciente para este diagnóstico la tabla de datos no lo desplegará.

Los registros de diagnósticos que no tengan registros de pacientes o bien, registros de pacientes que no tengan registros de diagnósticos no serán desplegados en la ventana de tabla de datos.

Se deberá oprimir el botón de **Agregar**, para que se guarden los campos en la base datos. Y aparecerá el siguiente cuadro de diálogo, el cual notifica que el registro se agregó exitosamente. Si el cuadro no aparece después de pulsar el botón de agregar repita la operación ya que estos datos no fueron guardados.

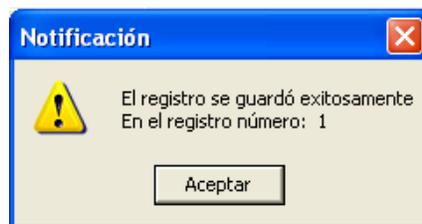


Figura 4.6. Mensaje de Registro Agregado.

Si después de haber agregado los datos se desea cambiar algunos de los campos, esto se puede hacer siguiendo los siguientes pasos:

- Ubicar el cuadro de texto que se desea cambiar:
- Cambiar los datos anteriores por los nuevos datos. Esto se hace simplemente sobrescribiendo los datos nuevos en el cuadro de texto.
- Presionar botón **Modificar**. Aparecerá un cuadro de diálogo, el cual indica cuántos registros han sido modificados. Si el cuadro de diálogo aparece como cero registros modificados eso quiere decir que no se modificó ningún registro y que la operación tiene que repetirse.

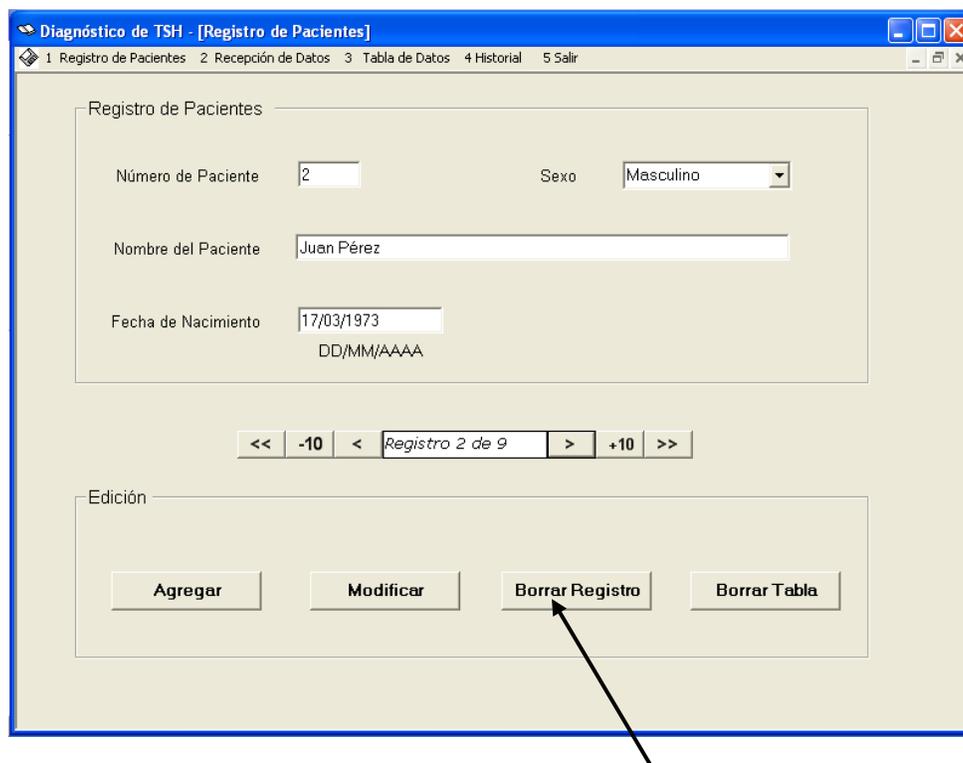


Figura 4.7. Mensaje de registro Modificado.

Los campos Nombre, Sexo, Fecha de Nacimiento, no tienen ningún problema para ser modificados; sin embargo, Número de paciente puede ser cambiado solamente que al modificar el campo, si existe un registro con ese mismo número de campo, el registro modificado sustituirá al registro anterior.

Se puede borrar cualquier registro contenido en esta tabla, para lograr esto, lo único que debe hacerse es colocarse en el registro que se desea borrar, es decir, que en los cuadros de texto aparezcan los datos que se desean borrar; por último se presiona el botón **Borrar**.

La siguiente imagen muestra como borrar un registro:



Presionar botón para borrar el
archivo número 2.

Figura 4.8. Borrar Registro.

Aparecerá un cuadro de diálogo notificando que el registro ha sido borrado, si este cuadro no aparece, debe repetirse la operación



Figura 4.9. Mensaje de registro Borrado.

Para ubicar los registros, es decir, avanzar o retroceder se puede hacer con las barras de navegación.

Una vez almacenados los registros deseados en la ventana Registro de Pacientes, se pueden recibir los registros de diagnósticos. Esto se puede hacer en la siguiente ventana:

b) RECEPCIÓN DE DATOS

Esta ventana se encarga de recibir los datos del instrumento.

Tiene como característica la de poder cambiar el puerto de comunicación serial.

La ventana consta de tres secciones y son las siguientes:

- Barra de botones de adquisición
- Tabla de datos
- Barra de estado

Barra de botones de adquisición

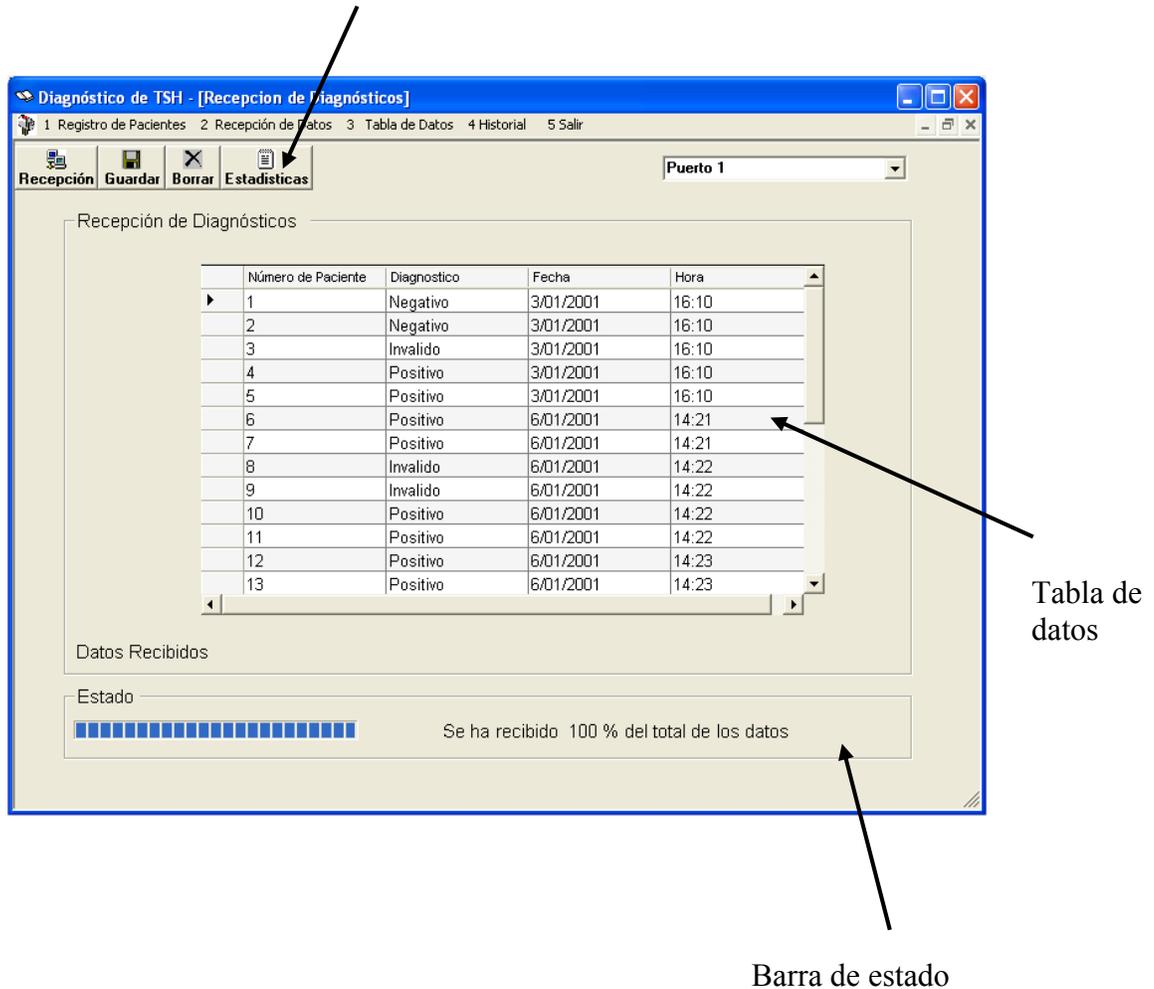


Figura 4.10. Ventana Recepción de Datos.

b.1) BARRA DE BOTONES DE ADQUISICIÓN

Estos botones son usados para el momento en que los datos son llamados.

Los datos recibidos son guardados en la base de datos, la cual es dinámica, por lo que puede ser editada por los botones de la barra de adquisición.

Botón	Descripción
	<p>Al oprimir éste botón se inicia la comunicación entre la computadora y el instrumento de diagnósticos clínicos. Una vez que se ha oprimido este botón, la computadora empezará a recibir los datos en la tabla. Si la tabla esta vacía, los datos entrarán sin ningún problema, si la tabla está llena aparecerá un cuadro de diálogo en el cual pregunta si se desean guardar los datos previos y tendrá tres botones (SI, NO, Cancelar). Si la opción es SI, mandará los datos a la tabla en la cual se contendrán los registros de respaldo (Ventana de Historial) y borrará los registros para recibir los nuevos. Si la opción es NO entonces simplemente borrará los registros anteriores.</p>
	<p>Copia los registros contenidos en ese momento a la tabla de respaldo.</p>
	<p>Este botón borra los registros contenidos en la tabla de datos.</p>
	<p>Muestra cuántos de los registros entrantes tienen resultados positivos, negativos o inválido.</p>
	<p>Permite seleccionar los diferentes puertos con los que puede trabajar una computadora. Con este cuadro de se pueden elegir los puertos del 1 al 10, pero antes de elegir el puerto debe asegurarse que el cable al instrumento está conectado.</p>

Por lo general las máquinas tienen uno o dos puertos, pero existen también cables que convierten de salida, serial a entrada USB, lo cual significa que se puede abrir otro puerto.

b.2) TABLA DE DATOS

Presenta los datos del instrumento.

Con esta tabla se puede verificar si los datos llegaron en forma correcta.

	Número de Paciente	Diagnostico	Fecha	Hora
▶	1	Invalido	7/02/2003	10:09
	2	Positivo	9/06/2004	1:01
	3	Negativo	9/08/2001	4:07
	4	Negativo	11/12/2001	5:01
	5	Negativo	11/03/2008	4:255

Figura 4.11. Tabla de Recepción de Datos.

En la ventana de tabla de datos se tiene un letrero que dice “Recepción de datos”, el cual, en el momento en que se inicia la transmisión, cambia a Recibiendo datos... Este es un indicador de que los datos realmente están llegando. Si la etiqueta no cambia, los datos no están llegando.

b.3) BARRA DE ESTADO

Indica el porcentaje de datos que se han recibido. Cuando la barra muestra 100%, los datos se han terminado de adquirir.

Para recibir los datos se siguen los siguientes pasos:

b.4) PUERTO DE COMUNICACIÓN

Antes de recibir los datos se tiene que elegir el puerto de comunicación por el cual dichos datos serán recibidos. Las computadoras de escritorio hoy en día solamente tienen un puerto de comunicación y por lo tanto en este caso se elige el puerto 1, así también, la mayoría de las computadoras portátiles no tiene puerto serial por lo que se tiene que conectar por medio de un cable que adapta el puerto serial a un puerto USB, entonces el puerto tiene otro número el cual se tiene que verificar. El sistema puede abrir hasta diez puertos de comunicación.

Se tiene que elegir un puerto de comunicación al iniciar esta ventana en caso contrario aparece el siguiente cuadro de error:



Figura 4.12. Mensaje puerto no detectado.

Para desaparecer este mensaje simplemente se presiona el botón aceptar y desaparece, este mensaje aparecerá siempre que no elija puerto.

Para elegir puerto de comunicación simplemente se presiona la flecha que aparece a un lado del cuadro de texto que se encuentra en la barra de botones de adquisición y este desplegará una lista de puertos que pueden ser elegidos. Para elegir el puerto de comunicación, se tendrá que dar clic con el botón izquierdo del mouse sobre el puerto que se desea utilizar, de esta manera la opción siempre quedará a la vista y sabrá que puerto está activo en esos momentos.

Si se desea cambiar el puerto simplemente se repite la operación anterior.

Así también se debe asegurar que el puerto que se ha elegido, sea un puerto de comunicación que exista en la computadora que se está utilizando, de lo contrario presentará el siguiente error.



Figura 4.13. Mensaje puerto no detectado.

Una vez que el puerto está bien conectado, el instrumento requiere establecer la comunicación, espera un momento y si la comunicación no se establece, aparecerá el siguiente cuadro de diálogo.



Figura 4.14. Mensaje de comunicación no establecida.

Nota: Es recomendable presionar el botón de reset del instrumento de diagnósticos cada vez que se cambie el puerto de comunicación.

b.5) RECEPCIÓN DE DATOS

Para recibir los datos enviados por el instrumento, simplemente se presiona el botón de Recepción, el cual está ubicado en la barra de botones de adquisición. Este botón iniciará la comunicación entre el instrumento y la PC.

Si el sistema es usado por primera vez, inmediatamente se recibirán los datos sin ningún problema.

Si el sistema ha sido utilizado anteriormente antes de iniciar la recepción de datos aparecerá el siguiente cuadro de texto:

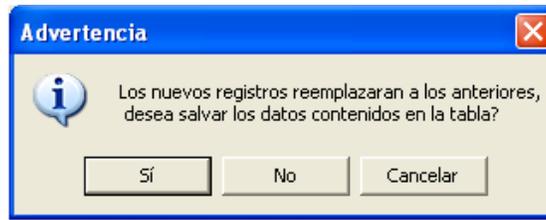


Figura 4.15. Salvar datos recibidos.

Este cuadro indica que la tabla contiene datos, lo cual no permite que los nuevos registros puedan ser almacenados. Por lo tanto pregunta si se desea guardar dichos registros. Esto indica que los datos serán enviados a la tabla de respaldos para que estos puedan ser consultados más adelante en la ventana de Historial.

El cuadro de diálogo esperará una respuesta que puede ser cualquiera de las siguientes.

- **Si.** Esta opción borra los datos de la tabla de registros y los envía a la tabla de respaldos para que estos puedan ser consultados más adelante.
- **No.** Esta opción simplemente borra los datos. Y no los envía a respaldos.
- **Cancelar.** Simplemente cierra el cuadro de diálogo. Si presiona esta opción los datos seguirán contenidos en la tabla lo cual no permitirá que los datos nuevos puedan ser agregados.

El tiempo de recepción puede ser desde 30 seg. hasta 60 seg.

b.6) FUNCIONES ADICIONALES

En algunas ocasiones puede desearse que los datos sean guardados sin necesidad de adquirir nuevos datos, para poder realizar esta acción existe el botón que se encuentra en la barra de botones de adquisición llamado **Guardar**, esta opción no borra los datos contenidos en la tabla, para ello se encuentra el botón llamado **Borrar**, el cual se encuentra en la barra de botones de adquisición.

El botón **Estadísticas** muestra cuantos diagnósticos son positivos, cuantos son negativos y cuantos son inválidos.

Después de haber recibido los datos y de haber registrado los pacientes, estos datos quedan contenidos en la tabla de datos, y se muestran en la ventana de Tabla de datos.

Se divide en tres partes que son:

- Barra de botones de la tabla de datos
- Tabla de datos
- Barra de navegación de la tabla de datos

c) TABLA DE DATOS

Muestra los datos personalizados de cada uno de los registros de la tabla, de manera que estando en esta ventana podemos verificar cada uno de los nombres de los pacientes con su respectivo diagnóstico.

Barra de botones de Tabla de datos

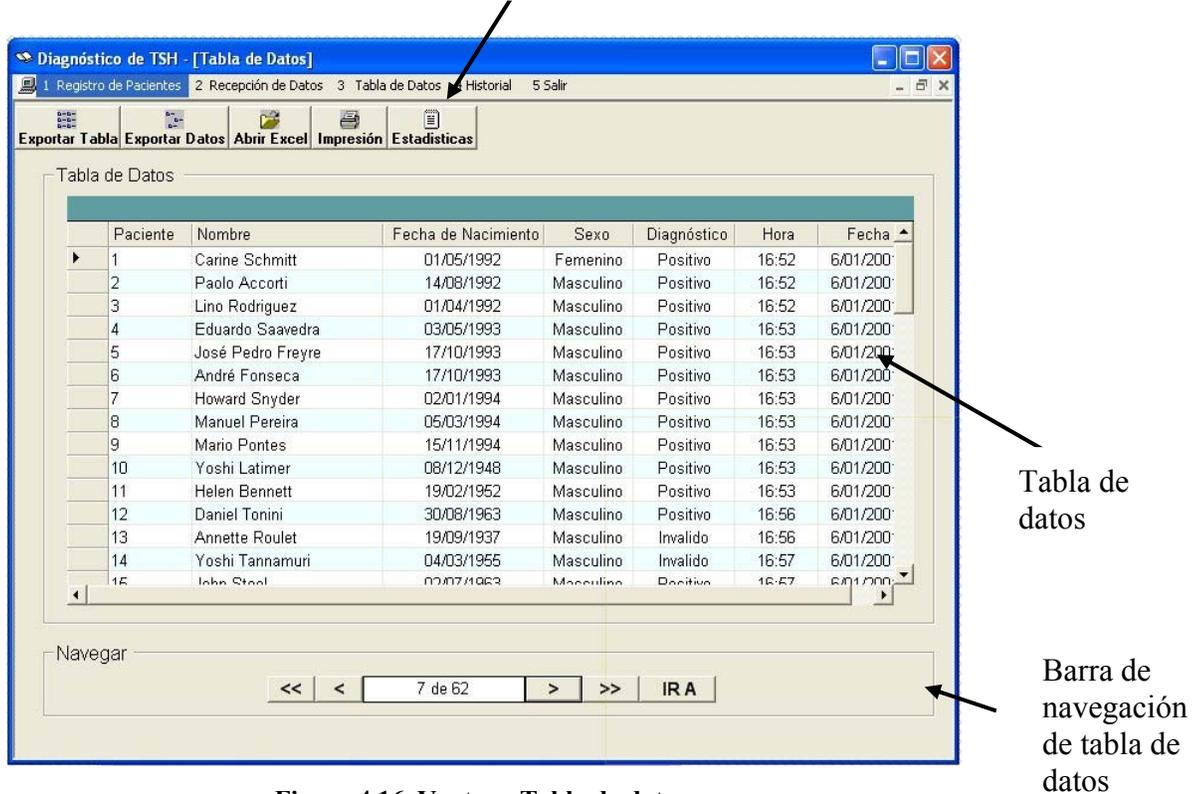


Figura 4.16. Ventana Tabla de datos.

En esta ventana puede generarse un reporte con los datos.

Esta ventana tiene como característica la de presentar solamente los registros en los cuales están dados de alta los números tanto de pacientes como de diagnósticos. Esto es, si por alguna razón la PC adquirió del instrumento el registro número 26 con todas sus características tales como número de paciente, diagnóstico hora y fecha, y no se registró dicho paciente número 26 en la ventana de Registro de pacientes, este registro no aparecerá en la tabla de datos. Puede suceder algo similar registrando el número de paciente y no adquiriendo ese número por medio de la computadora.

Una vez que dichos datos están desplegados en la tabla, pueden realizarse otras tareas que a continuación se describirán.

c.1) BARRA DE BOTONES DE TABLA DE DATOS

Estos botones son utilizados para la manipulación de los registros de tabla de datos. Estos botones se describen a continuación:

Botón	Descripción
	<p>Envía completamente los datos contenidos en la tabla a archivos de tipo Excel. Cuando la transmisión de estos finaliza, se abre de forma automática la ventana de Excel para mostrar los datos. A partir del momento que se abre la hoja de Excel, se podrán utilizar todas sus propiedades.</p>
	<p>Envía un determinado número registros de la tabla de datos a un archivo de tipo Excel.</p>
	<p>Botón que permite abrir los archivos de la tabla de datos que han sido guardados con anterioridad.</p>
	<p>Imprime un reporte de cada uno de los registros de la tabla de datos, con los datos del paciente y su diagnóstico. Para imprimir los datos del paciente, primero se tiene que presionar el botón del mouse en el registro del cual se desea la impresión del reporte y posteriormente presionar este botón.</p> <p>El reporte contiene los datos personales del paciente y su respectivo diagnóstico, así como el nombre a quien va dirigido dicho reporte, el nombre de la persona</p>

responsable de tomar dicho reporte, así como también el nombre de la institución en la que se realizó el análisis.



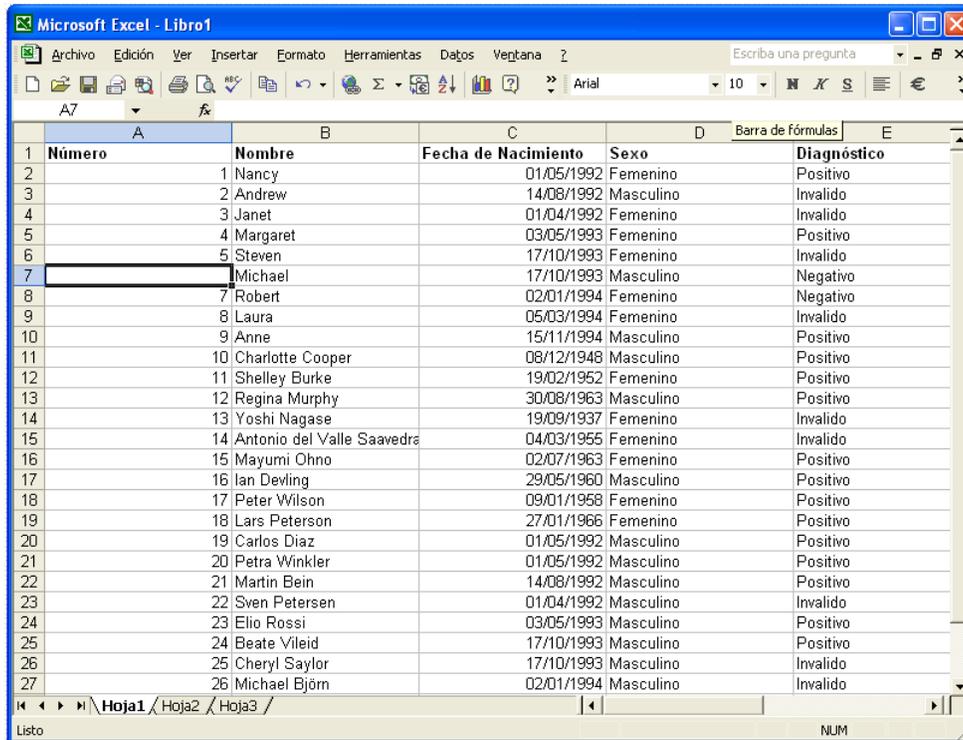
Esta opción muestra el número de diagnósticos positivos, negativos e inválidos, así como cuántos pacientes son femeninos y cuántos masculinos.

Estas estadísticas son las correspondientes a la tabla de datos y solamente se han tomado en cuenta los registros que aparecen en dicha tabla.

c.2) EXPORTAR TABLA DE DATOS A EXCEL

El botón exportar tabla a Excel de la barra de botones de tabla de datos sirve para enviar la tabla de datos completa a un archivo de tipo Excel. El objetivo es hacer un archivo de respaldo, para que este pueda ser transferido a otras máquinas o simplemente pueda ser modificado para fines específicos.

La tabla de datos transferida a archivos de tipo Excel se observa de la siguiente manera:



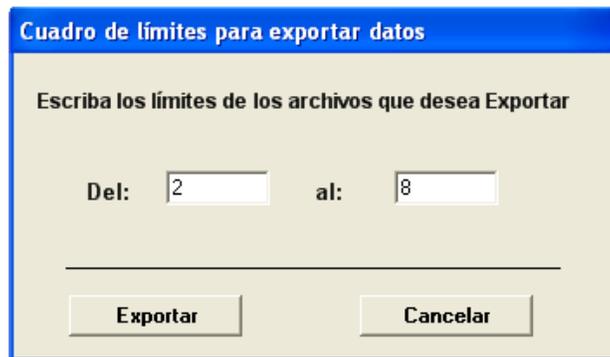
Número	Nombre	Fecha de Nacimiento	Sexo	Diagnóstico
1	Nancy	01/05/1992	Femenino	Positivo
2	Andrew	14/08/1992	Masculino	Invalido
3	Janet	01/04/1992	Femenino	Invalido
4	Margaret	03/05/1993	Femenino	Positivo
5	Steven	17/10/1993	Femenino	Invalido
6	Michael	17/10/1993	Masculino	Negativo
7	Robert	02/01/1994	Femenino	Negativo
8	Laura	05/03/1994	Femenino	Invalido
9	Anne	15/11/1994	Masculino	Positivo
10	Charlotte Cooper	08/12/1948	Masculino	Positivo
11	Shelley Burke	19/02/1952	Femenino	Positivo
12	Regina Murphy	30/08/1963	Masculino	Positivo
13	Yoshi Nagase	19/09/1937	Femenino	Invalido
14	Antonio del Valle Saavedra	04/03/1955	Femenino	Invalido
15	Mayumi Ohno	02/07/1963	Femenino	Positivo
16	Ian Devling	29/05/1960	Masculino	Positivo
17	Peter Wilson	09/01/1958	Femenino	Positivo
18	Lars Peterson	27/01/1966	Femenino	Positivo
19	Carlos Diaz	01/05/1992	Masculino	Positivo
20	Petra Winkler	01/05/1992	Masculino	Positivo
21	Martin Bein	14/08/1992	Masculino	Positivo
22	Sven Petersen	01/04/1992	Masculino	Invalido
23	Elio Rossi	03/05/1993	Masculino	Positivo
24	Beate Vileid	17/10/1993	Masculino	Positivo
25	Cheryl Saylor	17/10/1993	Masculino	Invalido
26	Michael Björn	02/01/1994	Masculino	Invalido

Figura 4.17. Datos Exportados a Excel.

c.3) EXPORTAR DATOS

Esta función hace algo parecido a la de exportar tabla, la única diferencia es que esta función solamente exporta un intervalo de datos.

Para ejecutar esta función simplemente se presiona el botón de **Exportar datos** y aparecerá el siguiente cuadro de mensaje:



Cuadro de límites para exportar datos

Escriba los límites de los archivos que desea Exportar

Del: al:

Figura 4.18. Límites de exportación de datos.

Se deberá escribir el intervalo de los datos que desea enviar. Por ejemplo se desea enviar a partir del segundo registro hasta el registro número ocho, entonces se deberá escribir en el cuadro del: el número dos y en el registro a: el número ocho. Después se deberá presionar el botón **Exportar** para poder enviar los datos.

Si se presiona botón **Cancelar** se suspenderá la función y se cerrará el cuadro de diálogo.

Los límites son el intervalo de registros que se desean enviar, mas no son los números de paciente que se quieren enviar.

c.4) ABRIR EXCEL

En esta función se permite abrir documentos de Excel que se han guardado anteriormente. Es decir, es una especie de respaldo.

Para poder acceder a esta función basta con presionar el botón **Abrir Excel** el cual se encuentra en la barra de botones de tabla de datos. Aparecerá automáticamente un cuadro de diálogo para abrir archivos, el cual contiene todos los archivos. Se debe seleccionar el archivo que se desea abrir y presionar en el botón de **Abrir** e inmediatamente abrirá la hoja seleccionada.

Esta opción, a pesar de que muestra todos los archivos, únicamente abre archivos Excel.

La ventana se muestra en seguida:

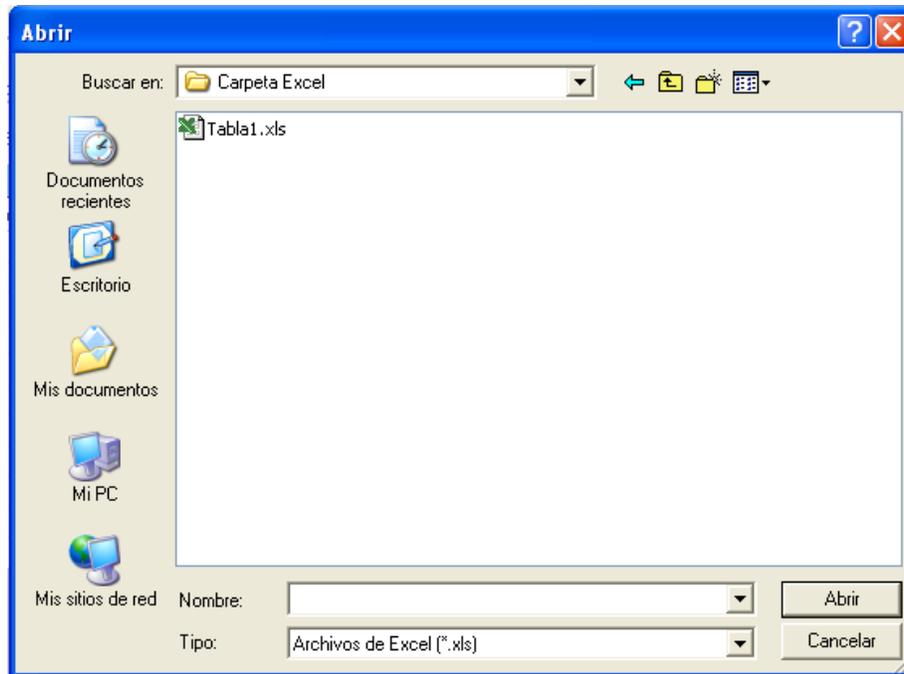


Figura 4.19. Ventana de abrir archivo.

c.5) REPORTE DE DIAGNÓSTICO CLÍNICO

El sistema es capaz de generar un sencillo reporte sobre el diagnóstico de cada paciente.

Para generar el reporte se deben seguir los siguientes pasos:

1. Se debe seleccionar dentro de la tabla de datos el registro del cual se desea el reporte. Esto se hace dando clic con el botón izquierdo del mouse sobre el registro que se desea, dentro de la tabla
2. Se debe presionar el botón **Generar reporte** en la barra de botones de tabla de datos.

Aparecerá el cuadro de diálogo siguiente:

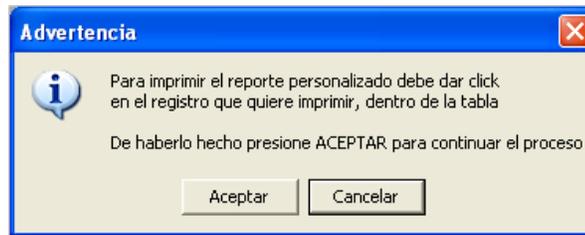


Figura 4.20. Mensaje para imprimir reporte.

Este cuadro simplemente reafirma lo que se debió haber realizado en el paso 1. Se debe dar clic en **Aceptar** si ya ha elegido el registro del paciente del que se desea hacer el reporte, si todavía no se elige se deberá presionar el botón **Cancelar** y realizar las instrucciones del paso 1.

3. Llenar el cuadro diálogo que aparece.

A dialog box titled "Cuadro de Datos de Impresión" with a blue header. It contains four text input fields: "Dirigido a:" with the value "Dr. Juan Pérez", "Institución" with "HOSPITAL GENERAL", "Tipo de Examen" with "ona estimulante de la Tiroides", and "Atentamente" with "Dr. Horacio Peña". At the bottom, there are three buttons: "Imprimir", "Vista Previa", and "Cancelar".

Figura 4.21. Cuadro de datos de impresión.

4. Seleccionar algunos de los botones.

- **Imprimir.** Manda el documento directamente a impresión.
- **Vista previa:** Envía el reporte a la ventana de vista previa de cómo quedará el documento. A partir de la ventana vista previa puede mandarse imprimir dicho documento. Esta opción es la recomendada ya que puede simplificar errores y economizar tinta y papel en caso de existir un error.
- **Cancelar.** Sale del cuadro de diálogo.

El reporte que se genera tiene el formato siguiente:

HOSPITAL GENERAL

México D.F. a Sábado, 27 de Agosto de 2005

Dr. Juan Pérez

P R E S E N T E

Se ha practicado a Jorge Paniagua, nacido(a) el 20 de Julio de 1968 de sexo Masculino, el examen de Hormona estimulante de la Tiroides obteniendo un resultado de:

NEGATIVO

Dicho examen fue realizado el 15 de Agosto de 2005

A T E N T A M E N T E

Dr. Horacio Peña

c.6) ESTADÍSTICAS

El botón estadísticas muestra un reporte de los resultados de cada uno de los diagnósticos especificando cuantos fueron positivos, negativos o inválidos, así como cuantos pacientes son femeninos y cuantos masculinos.

A continuación se muestra un ejemplo de la tabla de estadísticas:

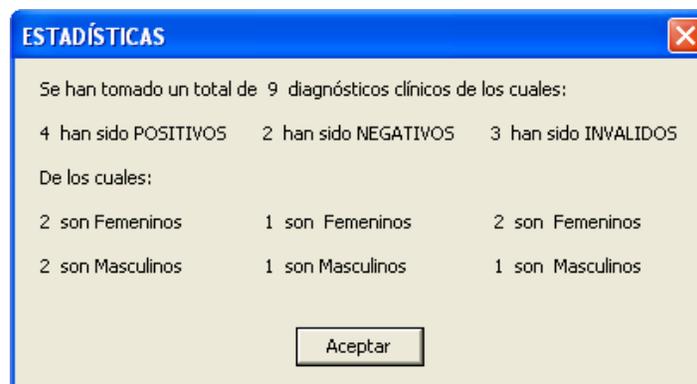


Figura 4.22. Mensaje de estadísticas.

d) HISTORIAL

Esta ventana tiene características similares a la ventana Tabla de datos, esta ventana contiene todos los registros guardados, es decir, contiene la historia de los registros tomados por el instrumento.

Barra de botones de Historial

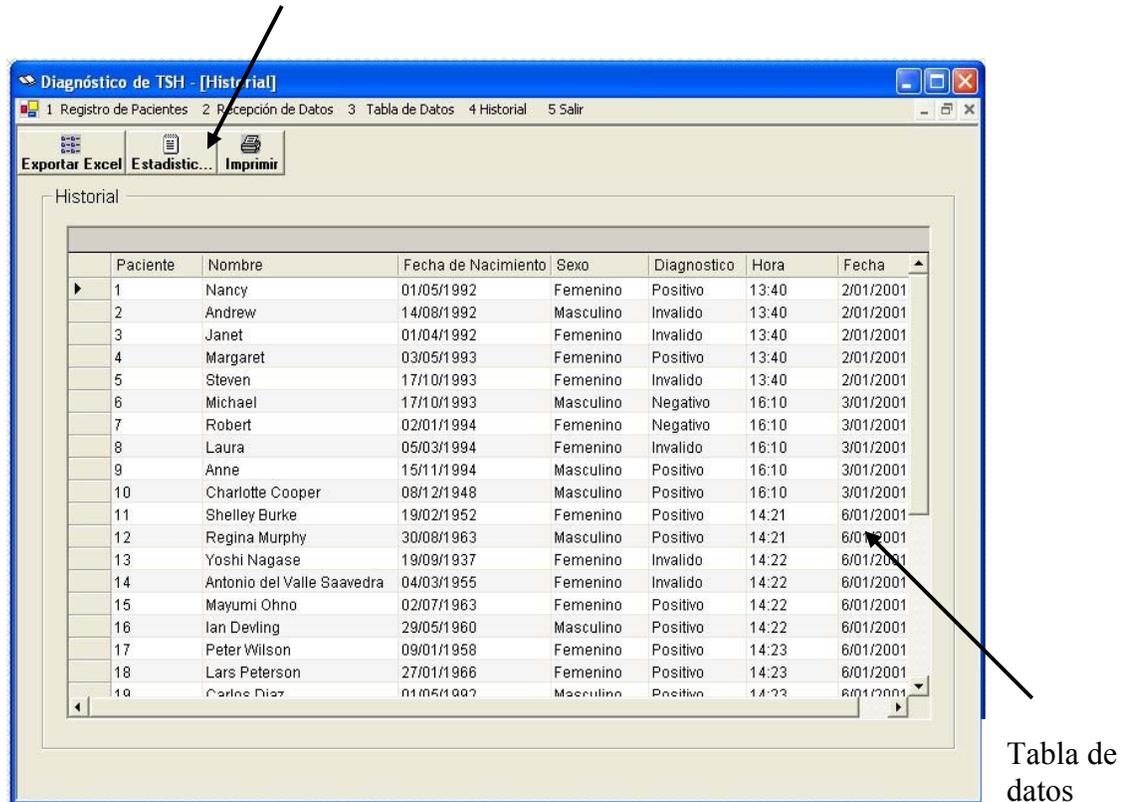


Figura 4.23. Tabla de Historial.

d.1) BARRA DE BOTONES DE HISTORIAL

Estas barras ya no son descritas porque sus funciones son exactamente iguales a las de la ventana de datos.

e) SALIR

Sale completamente del programa.

5.- FASE DE PRUEBAS Y CORRECCIÓN

Las pruebas son parte importante en el desarrollo de sistemas. Se realizan con el propósito de encontrar fallas y se establecen para mejorar la calidad del sistema. En las pruebas se requiere que se descarten las ideas acerca de lo que se desarrolló en el sistema, ya que al descubrir los errores, se logra superar cualquier conflicto en el sistema.

La prueba ideal de un sistema sería exponerlo en todas las situaciones posibles, así encontraríamos hasta la última falla, de tal manera que es posible garantizar su respuesta ante cualquier caso que se le presente en la ejecución real. Esto es imposible desde todos los puntos de vista: humano, económico e incluso matemático.

Dado que todo es finito en programación (el número de líneas de código, el número de variables, el número de valores en un tipo, etc.) cabe pensar que el número de pruebas posibles es finito. Esto deja de ser cierto en cuanto entran en juego bucles, en los que es fácil introducir condiciones para un funcionamiento sin fin. Aún en el caso de que el número de posibilidades fuera finito, el número de combinaciones posibles es tan enorme que se hace imposible su identificación y ejecución a todos los efectos prácticos.

Los objetivos de las pruebas son:

- Asegurar la obtención y formalización de los requerimientos del usuario y verificar que son adquiridos de una manera completa y correcta.
- Mostrar errores no descubiertos en los requerimientos establecidos.
- Obtener un conjunto de pruebas que tengan la mayor probabilidad de descubrir los defectos del sistema.
- Ejecutar el programa con la intención de descubrir un error.

Las pruebas se centran en los procesos lógicos internos del programa, asegurando que todas las sentencias se han comprobado, realizándose para poder detectar errores, cuando se produzcan resultados reales de acuerdo con los resultados requeridos.

Las pruebas pueden dividirse en distintas categorías, estas pueden ser: Caja negra, Caja blanca, Unitarias, de Integración, y de Regresión; mismas que se enuncian a continuación:

PRUEBAS DE CAJA NEGRA

También denominada prueba de comportamiento, la pruebas de caja negra se emplean conociendo la función específica para lo que fue diseñado el producto, Se puede llevar a cabo pruebas que demuestren que cada función es completamente operativa y al mismo tiempo buscando errores en cada función.

Estas pruebas son diseñadas para validar los requisitos funcionales sin fijarse en el funcionamiento interno del programa.

Las pruebas de caja negra están especialmente indicadas en aquellos módulos que van a ser interfaz con el usuario (en sentido general: teclado, pantalla, fichero, canales de comunicación, etc.).

Las pruebas de caja negra intentan encontrar errores de las siguientes categorías:

1. Funciones incorrectas o ausentes.
2. Errores de interfaz.
3. Errores de estructuras de datos o en acceso a bases de datos externas.
4. Errores de rendimiento.
5. Errores de inicialización.

El probador se limita a suministrarle datos como entrada y estudiar la salida.

LIMITACIONES DE LAS PRUEBAS DE CAJA NEGRA

Lograr una buena cobertura con pruebas de caja negra es un objetivo deseable pero no suficiente a todos los defectos. Un programa puede pasar con una holgura millones de pruebas y sin embargo tener defectos internos que surgen en el momento más inoportuno.

PRUEBAS DE CAJA BLANCA

Se centra en la estructura de control del programa, basado en un minucioso examen de detalles de procedimientos donde se puede verificar el estado del programa en varios puntos para determinar si el real coincide con el esperado. Con las pruebas de caja blanca se puede comprobar que:

1. Todos y cada uno de los caminos se llevan a cabo de manera correcta.
2. Se lleva a cabo cada una de las decisiones lógicas.
3. Se ejecuten todos los bucles en sus límites operacionales.
4. Se ejerciten las estructuras internas de datos para asegurar su validez

LIMITACIONES DE LAS PRUEBAS DE CAJA BLANCA

Lograr una buena cobertura con pruebas de caja blanca es un objetivo deseable; pero no suficiente a todos los efectos. Un programa puede estar perfecto en todos sus términos, y sin embargo no servir a la función que se pretende ejecutar.

Las pruebas de caja blanca nos convencen de que un programa hace bien lo que hace pero no de que haga lo que el usuario necesita.

PRUEBAS UNITARIAS

Son las pruebas que centran el proceso de verificación sobre un programa o módulo con la finalidad de encontrar problemas en la lógica y problemas técnicos en el código.

PRUEBAS DE INTEGRACIÓN

La prueba de integración, es una técnica sistemática para que la estructura del programa esté de acuerdo con lo que dictó el diseño.

PRUEBAS DE REGRESIÓN

Ayudan a asegurar que los cambios (debidos a las pruebas o por otros motivos) no introducen un comportamiento no deseado de errores adicionales.

Con este tipo de pruebas se permite detectar fallas que se hayan introducido durante las modificaciones a un sistema, permitiendo verificar que dichas modificaciones no impacten y que sigan cumpliendo con los requerimientos planteados.

CONSIDERACIONES PARA LA EJECUCIÓN DE LAS PRUEBAS

Es importante tomar en cuenta algunas consideraciones al realizar las pruebas, para así obtener mejores resultados en la detección de errores cometidos al desarrollar el sistema.

Riesgos.- Los riesgos son aquellos que pueden afectar negativamente la ejecución de las pruebas.

Suposiciones.- Son las premisas que puedan afectar positiva o negativamente la ejecución de las pruebas complicando o facilitando las actividades de las pruebas.

Condiciones.- Lleva a cabo la definición de datos, funciones, y comportamiento al implementar el sistema, así como la información sobre el rendimiento y limitación que delimitan el sistema.

Restricciones.- Están relacionadas con el desarrollo del proyecto en sí: la tecnología de pruebas, el estado del ambiente de pruebas.

PRUEBAS APLICADAS

PRUEBAS DE CAJA NEGRA:

Esencialmente se probaron los módulos en los que el usuario tiene que interactuar introduciendo datos, esto se hace con la finalidad de observar si la adquisición de los datos del usuario por el sistema es correcta.

La primer prueba de caja negra realizada fue la de correr el programa desde su ícono de inicio en el escritorio de la PC, una vez realizado el programa ejecutable, si la prueba falla entonces se debe regresar a la parte de la elaboración del empaquetado.



Figura 5.1. Prueba de inicio.

La ventana a la que se le hicieron la mayor parte de las pruebas fue la ventana en la que se registran los datos del paciente, ya que ésta es la ventana en la que el usuario interactúa con el programa y por lo tanto pueden existir más probabilidades de fallas en el programa, en esta ventana se validaron cada una de las casillas de admisión de datos, de manera que se valida la fecha (si el usuario comete errores al introducir la fecha el programa será capaz de

especificarlos), el número de paciente (marca error si se introduce un número existente en la base o un carácter no numérico), el sexo está limitado solamente a dos tipos de entradas que son femenino o masculino y el único registro que no se limitó fue el de nombre, este es capaz de aceptar todo tipo de caracteres.

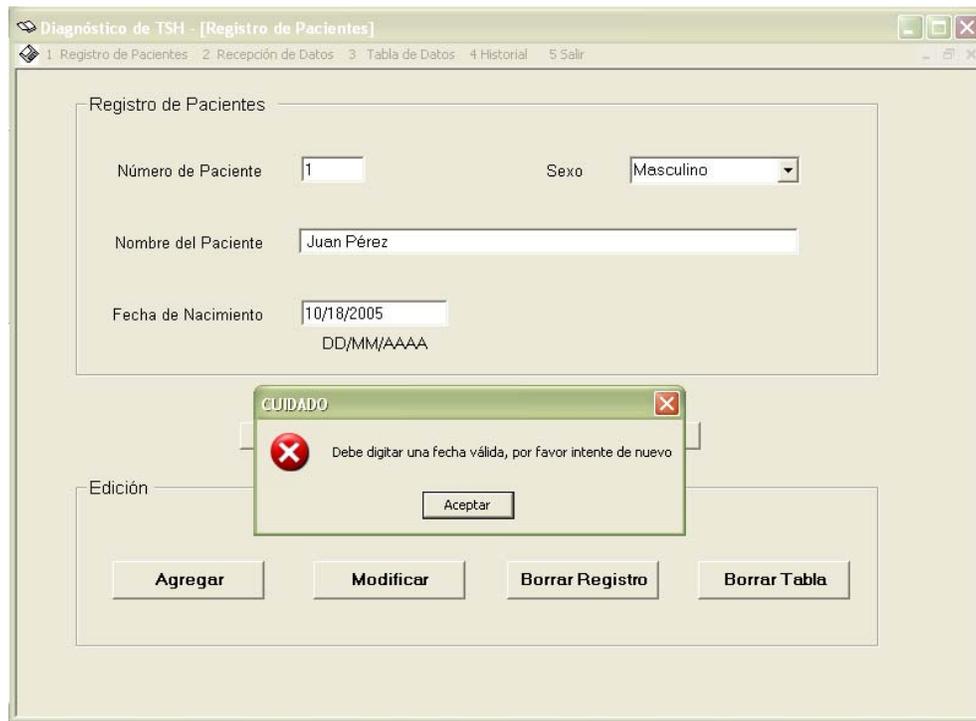


Figura 5.2. Validación de Fecha de Nacimiento.

Así mismo se validó la edición de los registros, de manera que si el registro muestra que fue agregado exitosamente, se compruebe en la Base de Datos que realmente fue agregado. De igual manera se hace la comprobación con Modificar, Borrar Registro y Borrar Tabla.

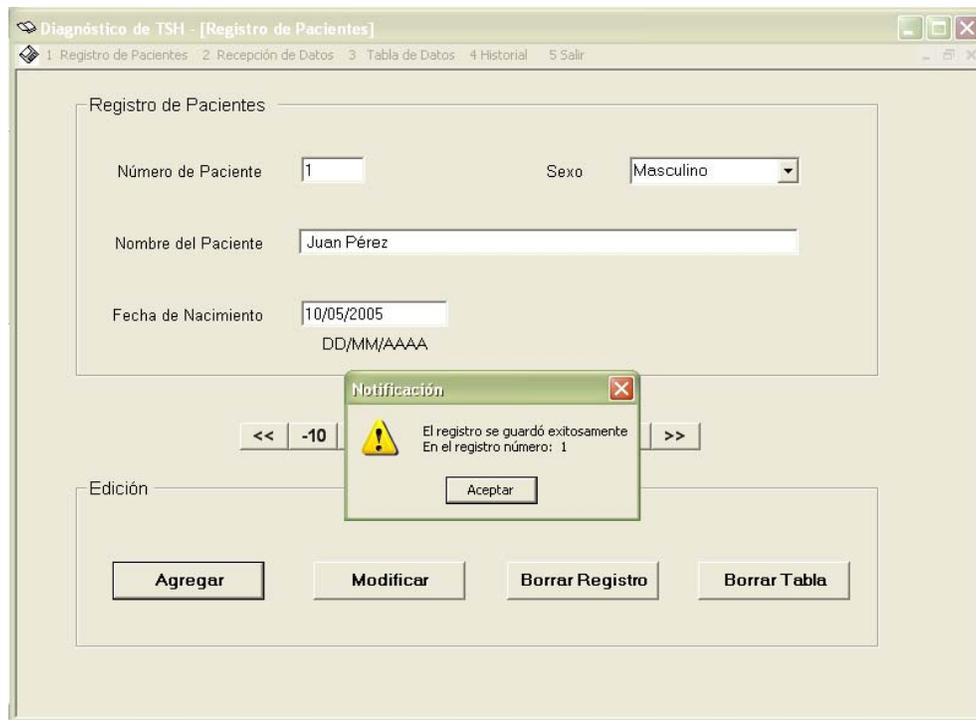


Figura 5.3. Validación de Edición.

En la tabla de datos de Access se verificó que el registro realmente sea guardado.

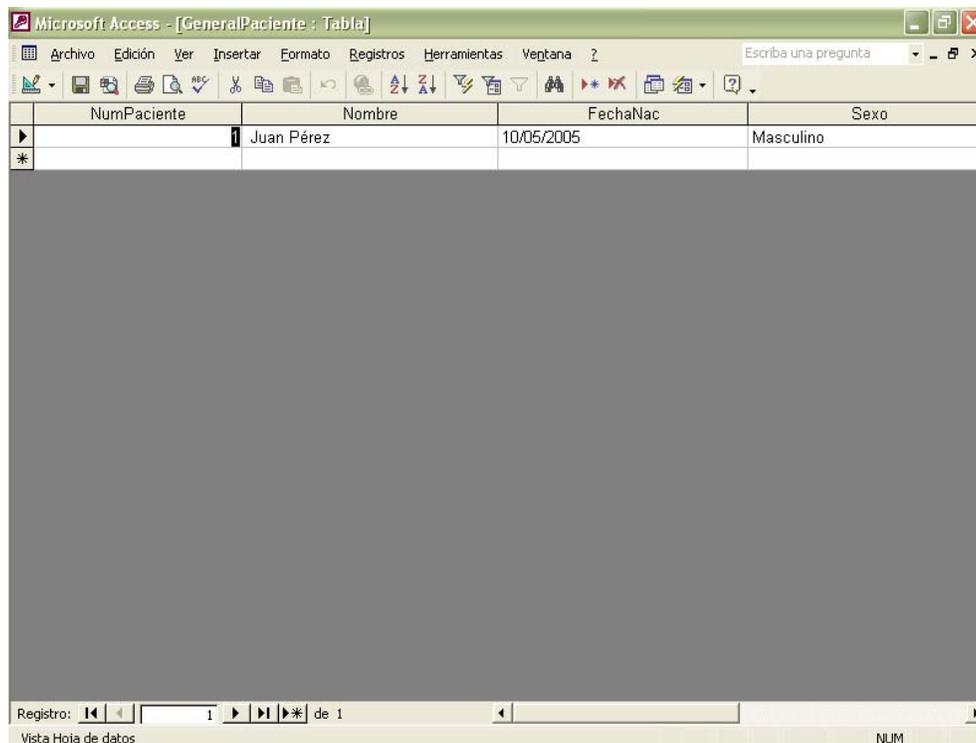


Figura 5.4. Validación de un registro Agregado en Access.

De igual manera se verificó que la instrucción Salir dentro del menú principal funcionara de manera correcta.

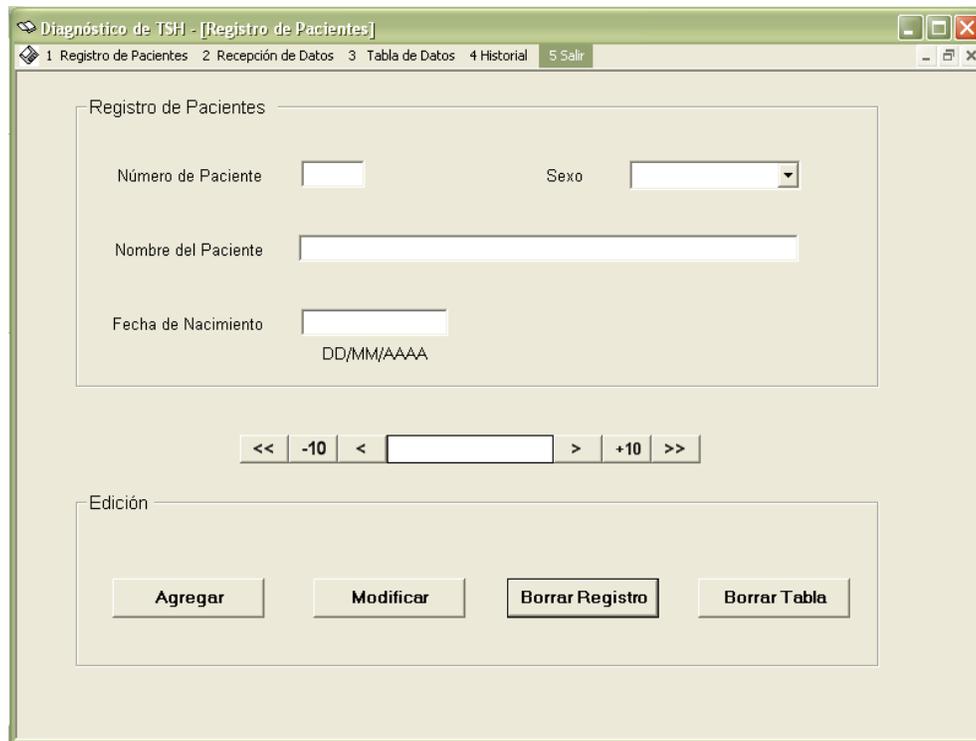


Figura 5.5. Validación del menú Salir.

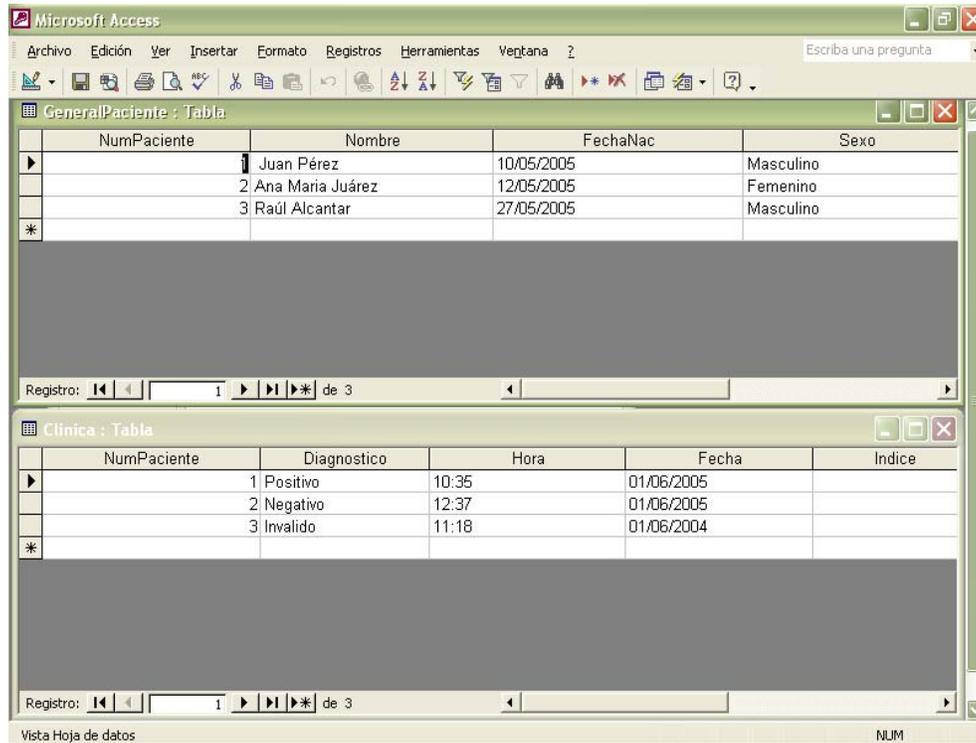
PRUEBAS DE CAJA BLANCA

Estas pruebas fueron aplicadas durante el desarrollo del software, en cada una de las actividades mientras éstas iban siendo programadas. Estas pruebas también se volvieron a hacer en las pruebas unitarias.

PRUEBAS UNITARIAS

Estas pruebas se aplicaron a cada unidad o módulo del programa para verificar su buen funcionamiento. De manera que se revisó cada una de las funciones del menú principal, cada botón de cada una de las ventanas, la introducción correcta de los datos y se validaron los botones. Así también, se verificó la parte de recepción de datos con el instrumento, comprobando que el botón trabajara de manera correcta y sincronizada con el instrumento.

En la vista de Tabla de Datos se realizó esta prueba para verificar que la relación entre las tablas se llevará a cabo de manera correcta



The screenshot shows the Microsoft Access interface with two data tables displayed in a 'Data Sheet View'. The top table, 'GeneralPaciente', has columns for patient ID, name, birth date, and sex. The bottom table, 'Clínica', has columns for patient ID, diagnosis, time, date, and index. Both tables show three records, demonstrating a one-to-many relationship.

NumPaciente	Nombre	FechaNac	Sexo
1	Juan Pérez	10/05/2005	Masculino
2	Ana Maria Juárez	12/05/2005	Femenino
3	Raúl Alcantar	27/05/2005	Masculino

NumPaciente	Diagnostico	Hora	Fecha	Indice
1	Positivo	10:35	01/06/2005	
2	Negativo	12:37	01/06/2005	
3	Invalido	11:18	01/06/2004	

Figura 5.6. Verificación de relación de tablas de datos en Access.

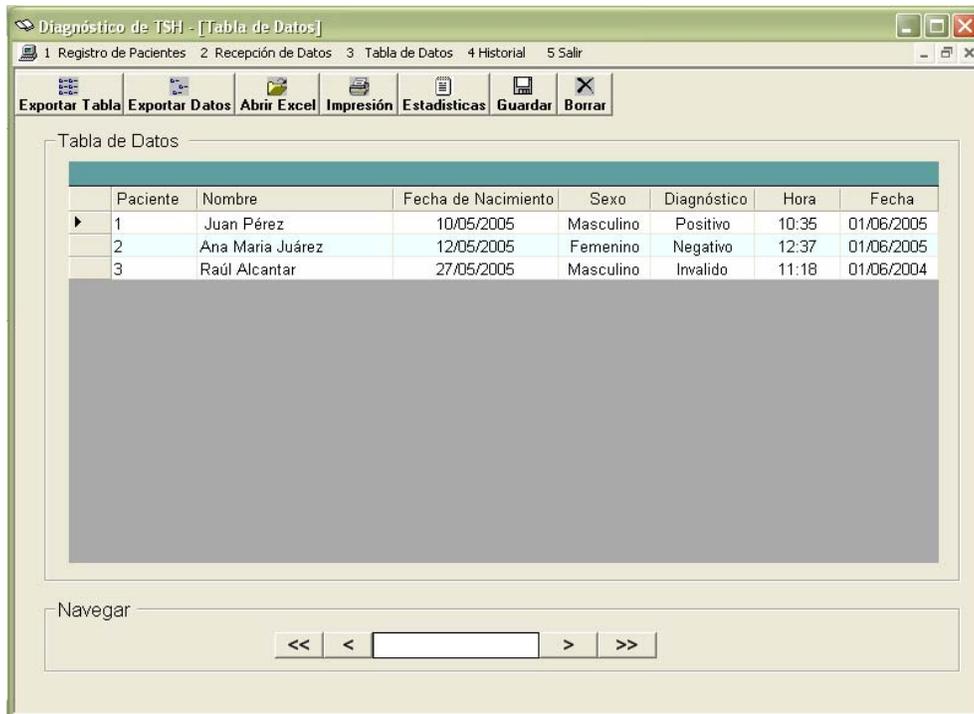


Figura 5.7. Verificación de relación datos en la ventana de Tabla de Datos.

De la misma manera se verificó que los datos se transmitieran de manera correcta a la tabla de Historial.

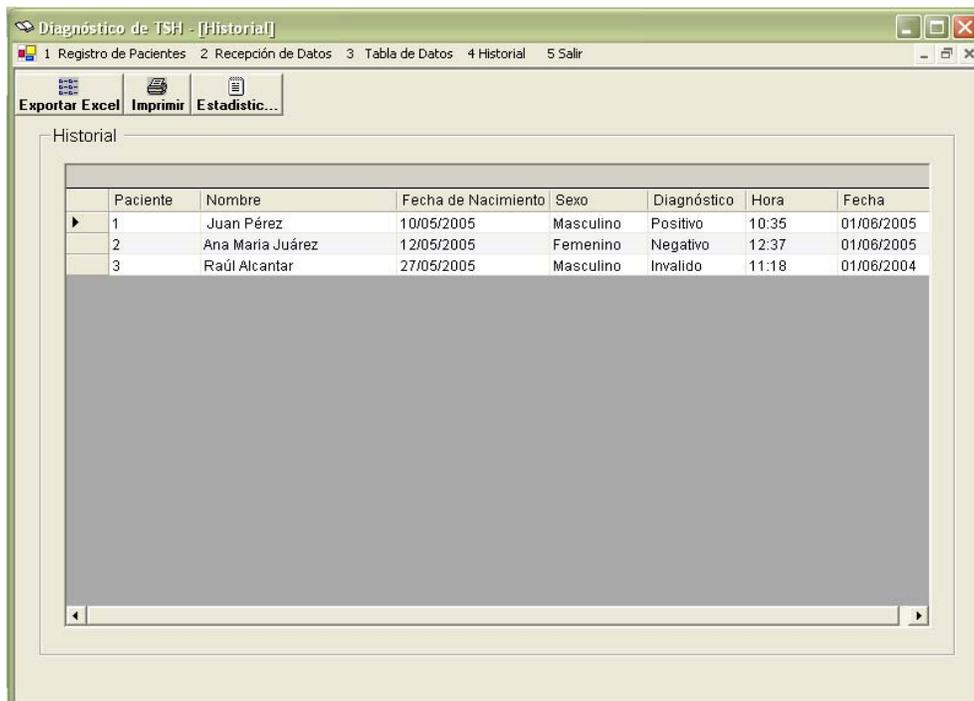


Figura 5.8. Verificación Historial.

Se verificó el botón de estadísticas, éste deberá desplegar datos reales y correctos.

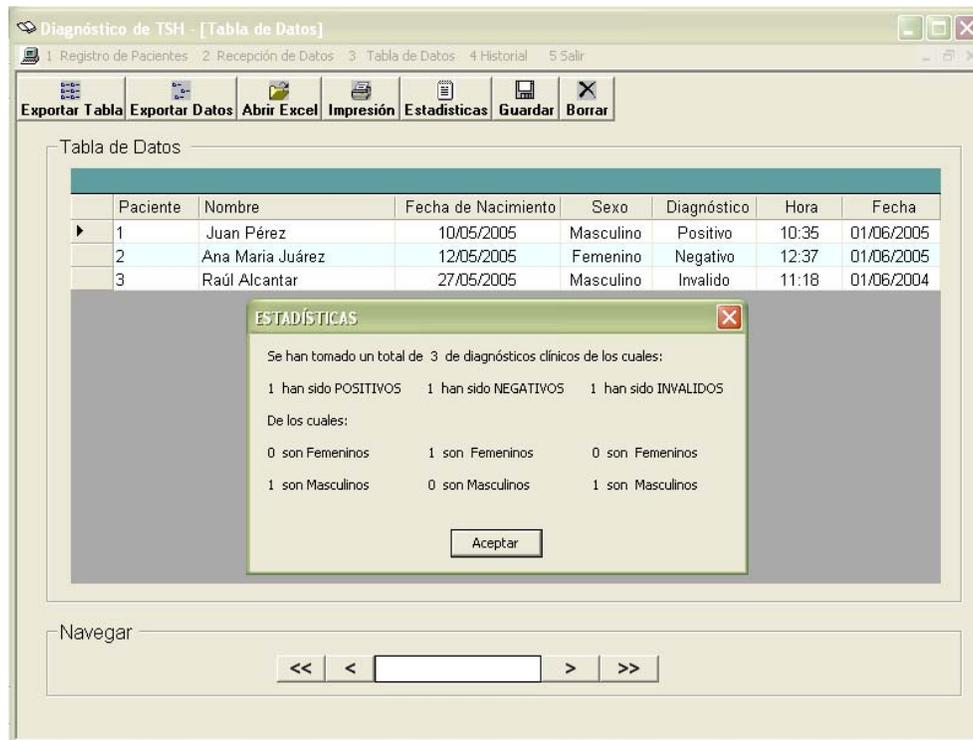


Figura 5.9. Verificación de botón de Estadísticas.

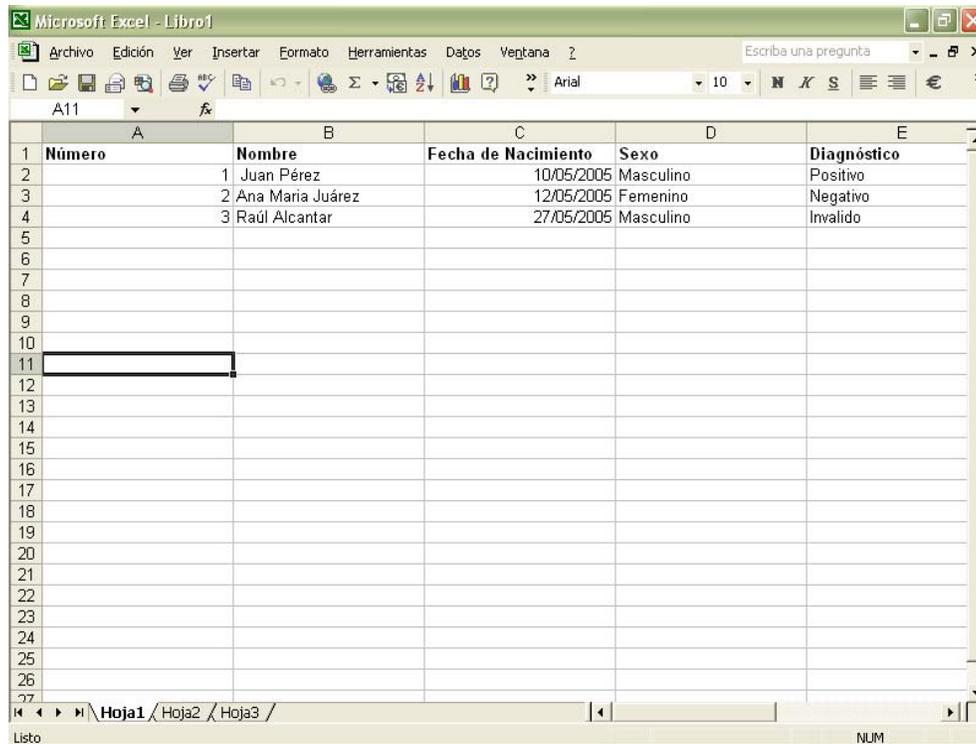
PRUEBAS DE INTEGRACIÓN

En esta fase de pruebas se verificó que cada una de las ventanas trabajará correctamente aún después de unir las en el proyecto final.

La primera prueba de integración fue la de acoplar el instrumento con el programa. De manera que los datos obtenidos fuesen coherentes y correctos. Para ello aparece la tabla de datos en la ventana de recepción.

Se verificó que los datos fueran enviados a la tabla de Excel y que no se perdiera ninguno de ellos en la exportación de datos.

De la ventana de Historial de la figura 5.8. exportamos los datos



The screenshot shows a Microsoft Excel window titled "Microsoft Excel - Libro1". The menu bar includes "Archivo", "Edición", "Ver", "Insertar", "Formato", "Herramientas", "Datos", and "Ventana". The toolbar shows various icons for file operations and editing. The active cell is A11. The table below is the main content of the spreadsheet.

	A	B	C	D	E
1	Número	Nombre	Fecha de Nacimiento	Sexo	Diagnóstico
2		1 Juan Pérez	10/05/2005	Masculino	Positivo
3		2 Ana María Juárez	12/05/2005	Femenino	Negativo
4		3 Raúl Alcantar	27/05/2005	Masculino	Invalido
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					

Figura 5.10. Prueba de Integración de la venta de Historia con Access.

Se verificaron los botones de guardar y recibir de todas y cada una de las ventanas. Estos botones tienen las funciones globales del sistema.

Y por último se verificó que cada una de las funciones para la generación del reporte trabajara de manera correcta.

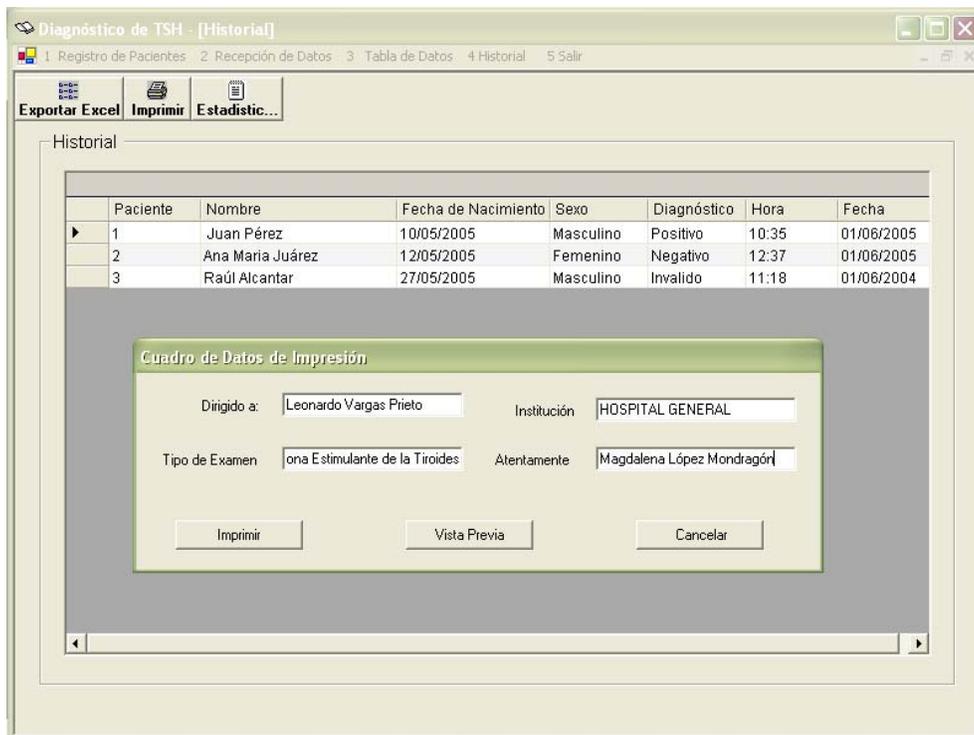


Figura 5.11. Verificación de generación de reporte de diagnósticos.

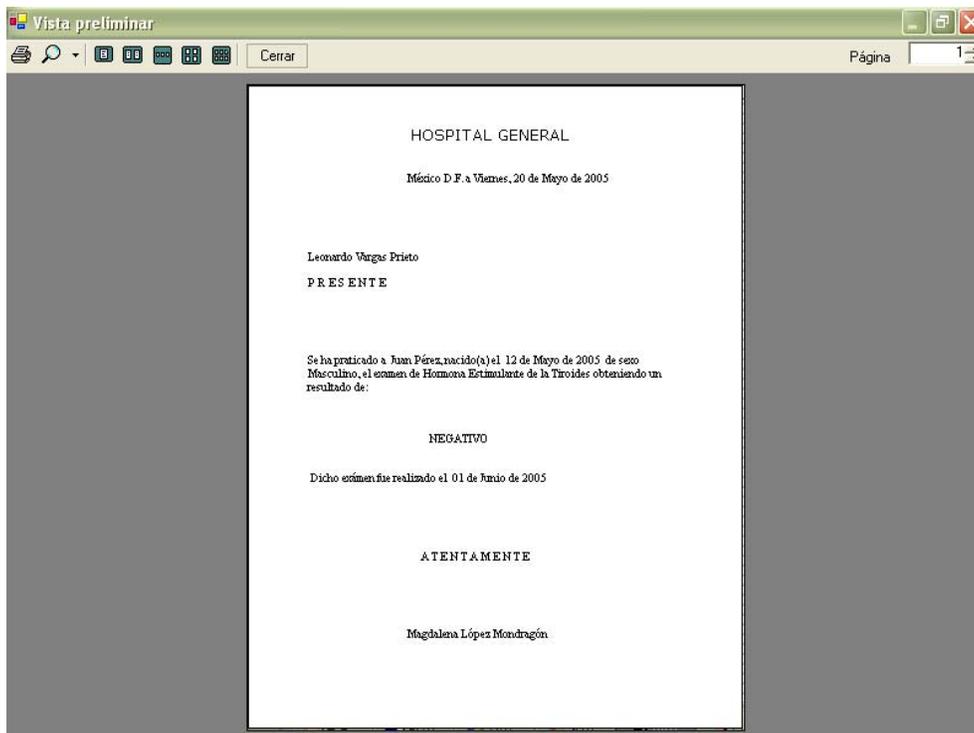


Figura 5.7. Verificación del reporte generado.

6.- MANTENIMIENTO

El mantenimiento es la última fase del proceso de Ingeniería de software. Puede ser la parte más costosa ya que a esta parte se le puede dedicar hasta un 70% de los esfuerzos.

Se llevan a cabo cuatro tipos de mantenimiento y son los siguientes:

1. Mantenimiento correctivo. Existen algunos errores en el programa que no fueron detectados durante la fase de pruebas pero solo han sido encontrados con el tiempo y uso del sistema.
2. Mantenimiento adaptativo. Es una actividad en que se cambian, mejoran o modifican los equipos periféricos y otros elementos del sistema debido a nuevas versiones de sistemas operativos de manera que el sistema pueda interactuar de manera correcta con el entorno cambiante.
3. Mantenimiento perfectivo. Esta actividad se da cuando a medida de que el usuario va utilizando el sistema, se da cuenta de que este puede mejorar, se pueden agregar nuevas funciones etc., por lo que el software se mejora. También esta actividad se da cuando se cambia el software para que éste se pueda adecuar a mejoras futuras.

El mantenimiento se asocia a algunos problemas: muchas veces la falta de disciplina durante el desarrollo casi siempre trae problemas en la etapa de mantenimiento.

Los problemas que trae consigo son los siguientes:

- Algunas veces resulta difícil seguir una evolución de software. A través de varias versiones ya que algunos cambios no son documentados.
- Resulta difícil la comprensión de un programa realizado por alguien más y esa persona no se encuentra para poder explicar cómo es que se elaboró el sistema.
- No existe una documentación adecuada.
- Algunas veces el software no se diseña pensando en un futuro cambio.

El mantenimiento puede ser algo riesgoso, se debe tener cuidado con las sentencias que se cambian, ya que el cambio de alguna de estas puede cambiar completamente el concepto del sistema.

Alguna de las desventajas que se presentan en la etapa tanto de pruebas como de mantenimiento además de la de costos son las siguientes:

- Insatisfacción del cliente cuando una petición de reparación o de modificación aparentemente legítima no se puede atender en un tiempo razonable.
- Disminución de la calidad global del software debido a los errores latentes que introducen los cambios en el software.
- Trastornos en otros esfuerzos de desarrollo al tener que “poner” a trabajar a la plantilla de mantenimiento.

A pesar de que el mantenimiento es una fase complicada y costosa es necesario aplicar ese paso ya que la mayoría de los sistemas necesitan modificaciones y actualizaciones.

CONCLUSIONES

El sistema de Adquisición de Diagnósticos Clínicos, que permite efectuar el proceso de control de registro y seguimiento a diagnósticos de cualquier tipo, tiene como propósito general el de proveer un sistema que adquiera y guarde los diagnósticos personalizados y permita la elaboración de reportes clínicos.

Es un sistema realizado con una sencilla interfaz basada en el lenguaje de programación Visual Basic.Net.

Este sistema guía al usuario en el transcurso de las diversas fases que conforman la adquisición de los diagnósticos y que además, permite llevar un proceso transparente, lo cual permite administrar y seguir ágilmente cada una de las etapas de cada uno de los diagnósticos adquiridos.

Otra ventaja, es la poca utilización de papel pues los diagnósticos podrán ser visualizados en un solo espacio sin necesidad de tener un impreso de cada uno de ellos.

De manera general el sistema de Adquisición de Diagnósticos Clínicos permite:

- Agilizar la operación de los procesos para los reportes de diagnósticos clínicos.
- Unificar el proceso de adquisición de datos, personalización y generación de reporte de diagnósticos.
- Aprovechar los registros de diagnósticos anteriores disminuyendo el tiempo de generación de los mismos.
- Facilitar la consulta y elección de los diagnósticos clínicos a los usuarios del sistema.
- Generar reportes e información estratégica, válida, confiable y oportuna.
- Garantiza la congruencia de la información.

El desarrollo del sistema tuvo algunas complicaciones y se mencionarán a continuación:

Entre las versiones existentes de Visual Basic se seleccionó la versión .NET debido a que:

- Hoy en día es difícil encontrar licencia de versiones anteriores de Visual Basic.
- Se pretende que el programa de alguna manera pueda ser mejorado significativamente, Visual Basic.NET permite grandes mejoras.
- Esta basado en la plataforma .NET Framework, el cual puede ser utilizado para aplicaciones de Internet.
- Porque al estar basado en la plataforma .Net Framework, el sistema puede ser mejorado en cualquiera de los programas basados en esta plataforma y que son compatibles con Visual Basic.NET.

Sin embargo, el desarrollo del sistema tuvo algunas complicaciones, ya que Visual Basic.NET no contiene los controles ActiveX necesarios para el desarrollo de ciertas aplicaciones tales como la comunicación serial. En este caso fue necesario buscar alternativas para establecer la comunicación serial y se consideraron las siguientes:

- Desarrollar una clase propia con las características necesarias para establecer la comunicación serial.
- Comprar una aplicación ya desarrollada y adaptarla a las necesidades del programa.
- Llamar el control de versiones anteriores a la versión de Visual Basic.NET.

Por razones de tiempo y dinero se tomó la última opción, pero aún así, el control tenía sus complicaciones, ya que algunas de las propiedades de dicho control no funcionan al 100% en Visual Basic.NET., por lo que algunas líneas de código se desarrollaron en Visual Basic 6 y fueron adaptadas a Visual Basic.NET.

El sistema de Adquisición de Diagnósticos Clínicos, ha sido realizado para ser distribuido comercialmente, por esta razón se debe crear un archivo de instalación del sistema, el cual debe contener tanto el archivo ejecutable como el archivo de desinstalación, debe ser manejado de manera sencilla y si es posible debe incluir un ayudante de instalación como cualquier programa de uso comercial.

Para poder hacer un archivo de instalación para un sistema que fue realizado bajo la plataforma .NET, es necesario incluir dentro del archivo ejecutable módulos de herramientas como son el .NET Framework, el MDAC (Microsoft Data Access Components), entre otras, las cuales son herramientas que hoy en día muchas de máquinas no las tienen y son indispensables para que el sistema trabaje correctamente.

Se probaron diversos instaladores tanto gratuitos como comerciales, pero estos no permitían trabajar con los componentes que son necesarios para el empaquetado de programas basados en la plataforma .NET Framework, no son programas gráficos por los que tiene que ser programados por medio de scripts y algunos de ellos no incluían una guía de usuario lo suficientemente clara.

Finalmente fue utilizado el Setup Factory 7.0, el cual es un programa que sirve para crear instaladores gráficos de manera simple. Este programa fue comprado, lo cual representa una desventaja, ya que eleva el precio del sistema, sin embargo, el uso de este software redujo esfuerzos en momento de realizar el empaquetado, ya que a diferencia de otros el Setup Factory 7.0 permite incluir los módulos que son necesarios para la distribución del sistema, tales como los componentes de la plataforma, .NET Framework y el MDAC, así como muchas otras aplicaciones que son utilizadas para mejorar la presentación de la instalación.

Por último cabe mencionar que todos los objetivos fueron cubiertos totalmente, ya que el sistema almacena y procesa los datos recibidos del instrumento, permite la consulta de dichos datos y genera un reporte de cada uno de los pacientes y su respectivo diagnóstico.

BIBLIOGRAFÍA

Barwell, Fred, Blair Richard et al. Profesional VB.Net. USA, Wrox, 2003, 2ª Edición.

Blair, Richard; Crossland, Jonathan, et al. Beginning VB.NET. USA, Wrox, 2002, 2ª edición.

Blanco, Luis Miguel. Programación en Visual Basic. Net. España, Grupo Eidos, 2002.

Cassel, Paul y Palmer, Pamela. Aprendiendo Microsoft Access 2000. Enrique Mercado, México, Prentice Hall, 1999.

Cohen, Daniel, Sistemas de información para la toma de decisiones. España, Mc Graw Hill, 1996, 4ª Edición.

Date, C., J., Introducción a los sistemas de bases de datos. México, Adison Wesley Iberoamericana, 1995.

Dobson, Rick. Programación en Microsoft Visual Basic.Net para bases de datos Microsoft Access. José Antonio Bautista Montejo, España, McGraw Hill. 2003.

Evjen, Hill; Beres Jason, et al. Visual Basic.Net Bible. USA, Hungry Minds Inc. 2002.

Laudon, Kenneth; Laudon Jane, Sistemas de información gerencial organización y tecnología de la empresa conectada a red. México Prentice Hall, 2002, 6ª Edición.

Mackenzie, Duncan; Sharkey Kent. Aprendiendo Visual Basic. Net. A. David Garza Marín. México, Pearson Educación. 2003.

Pressman, Roger S. Ingeniería de Software, un enfoque práctico. Carlos Cervigón Ruchaüer, Luis Hernández Yánez. España, Mc GrawHill, 1993, 3ª Edición.

Petroutsos, Evangelos. Mastering Visual Basic.Net. USA. SYBEX. 2002.

Tanenbaum, A. S., Sistemas Operativos Modernos. México, Prentice Hall, México. 1993.

Tsai, Y. h. Alice, Sistemas de bases de datos: Administración y uso. México, Prentice Hall, 1988.

Shmuller, Joseph Aprendiendo UML México, Prentice Hall, México. 2002.

PAGINAS WEB

<http://www.microsoft.com>

<http://latam.msn.com/>

<http://es.wikipedia.org>

<http://p2p.wrox.com/>

<http://vb-mundo.foros.st>