

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA DE TELECOMUNICACIONES

**USO DE UN SISTEMA REMOTO AUTOMATIZADO
PARA EL CONTROL GLUCÉMICO DE LA DIABETES**

TRABAJO ESPECIAL DE GRADO

presentado ante la

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

Como parte de los requisitos para optar al título de

INGENIERO EN TELECOMUNICACIONES

REALIZADO POR

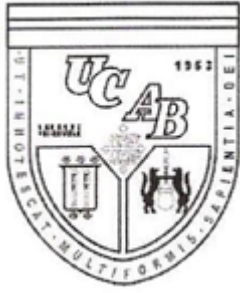
*Juan Manuel Arenas Riquelme.
Gilmar Carolina Rendón Solórzano.*

PROFESOR TUTOR

Iván Escalona

FECHA

Caracas, 13 de Octubre del 2006.



UNIVERSIDAD CATÓLICA ANDRÉS BELLO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE TELECOMUNICACIONES

**USO DE UN SISTEMA REMOTO AUTOMATIZADO
PARA EL CONTROL GLUCÉMICO DE LA DIABETES**

TRABAJO ESPECIAL DE GRADO

presentado ante la

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

Como parte de los requisitos para optar al título de

INGENIERO EN TELECOMUNICACIONES

REALIZADO POR

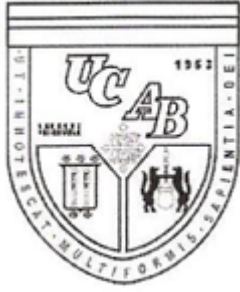
*Juan Manuel Arenas Riquelme.
Gilmar Carolina Rendón Solórzano.*

PROFESOR TUTOR

Iván Escalona.

FECHA

Caracas, 13 de Octubre del 2006.



UNIVERSIDAD CATÓLICA ANDRÉS BELLO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE TELECOMUNICACIONES

**USO DE UN SISTEMA REMOTO AUTOMATIZADO
PARA EL CONTROL GLUCÉMICO DE LA DIABETES**

REALIZADO POR

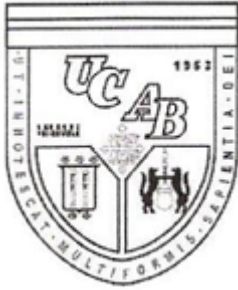
Juan Manuel Arenas Riquelme.
Gilmar Carolina Rendón Solórzano.

PROFESOR TUTOR

Iván Escalona.

FECHA

Caracas, 13 de Octubre del 2006.



UNIVERSIDAD CATÓLICA ANDRÉS BELLO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE TELECOMUNICACIONES

**USO DE UN SISTEMA REMOTO AUTOMATIZADO
PARA EL CONTROL GLUCÉMICO DE LA DIABETES**

Este jurado; una vez realizado el examen del presente trabajo ha evaluado su contenido con el resultado: _____

J U R A D O E X A M I N A D O R

Firma:

Firma:

Firma:

Nombre: José Pirrone

Nombre: María Cristi Stefanelli

Nombre: Iván Escalona

REALIZADO POR

*Juan Manuel Arenas Riquelme.
Gilmar Carolina Rendón Solórzano.*

PROFESOR TUTOR

Iván Escalona.

FECHA

Caracas, 13 de Octubre del 2006.

USO DE UN SISTEMA REMOTO AUTOMATIZADO PARA EL CONTROL GLUCÉMICO DE LA DIABETES

Resumen

La diabetes es una enfermedad de carácter metabólico, que ha penetrado dentro de la sociedad venezolana progresivamente, tornándose en un problema de salud pública. Un control adecuado de la enfermedad permite al paciente tener calidad de vida minimizando los riesgos correspondientes. En el presente trabajo en el área interdisciplinaria de Telemedicina, elaborado con asesoría médica en el Centro de Salud Santa Inés, se desarrolla un prototipo que permite complementar el cuidado continuo y compartido de la diabetes las 24 horas del día a través de un sistema remoto que garantice la comunicación médico-paciente y ayude a disminuir la frecuencia de las consultas médicas ofreciéndole al paciente una autonomía supervisada. El prototipo transmite los datos adquiridos por el glucómetro del paciente hacia un servidor en el centro médico correspondiente que almacena los datos y le permite al médico conocer los niveles de glicemia del paciente en tiempo real, llevar un registro actualizado del paciente, observar su comportamiento de manera gráfica y detectar alarmas de hiper e hipoglicemia que requieran ajustes en el tratamiento. Se seleccionó un glucómetro de los existentes en el mercado venezolano con puerto USB para el proceso de desarrollo de compatibilidad con un PC. El prototipo se llevó a cabo en lenguaje de programación “Visual Basic 6” por ser una herramienta de fácil implementación, otorgando así una interfaz (Programa Paciente) que logra la interoperabilidad entre el glucómetro y el PC, obtiene los datos, permite la comunicación médico-paciente, y desarrolla otra interfaz (Programa Médico) ubicada en el servidor que facilita el seguimiento automatizado del estado del paciente. El sistema ofrece herramientas tanto a médicos como pacientes que permitan capturar, gestionar, visualizar e interpretar datos de monitorización, así como intercambiar datos y mensajes vinculados con el estatus de la enfermedad.

Palabras claves: Diabetes, Glucómetro, PC, Médico, Paciente.

Índice General

Resumen.....	i
Dedicatoria y Agradecimientos.....	vii
Introducción.....	1
I.1 Planteamiento del proyecto.....	2
I.2 Objetivos.....	3
Objetivo General.....	3
Objetivos Específicos.....	3
I.3 Justificación.....	4
I.4 Limitaciones y Alcances.....	5
II.1 Diabetes.....	6
II.2 Glucómetro.....	7
II.3 Protocolo USB.....	9
II.4. Modelo OSI.....	11
II.5. IP. Protocolo Internet.....	14
II.6. TCP. Protocolo de Control de Transmisión.....	17
II.7. Internet.....	22
II.8 Microsoft Visual Studio.....	26

III.1 Fase I. Teórica: Levantamiento de Información.....	27
III.1.1 Estudio de indicadores y parámetros del proyecto.....	28
III.1.2 Estudio y Selección de Equipos	28
III.2 Fase 2. Técnica: Diseño del Prototipo.....	29
III.2.1 Compatibilidad Glucómetro – PC	30
III.2.2 Transmisión PC- Paciente al Servidor - Médico	33
III.2.3 Interfase de Gestión del Médico.....	37
III.2.3.1 Obtención de la data del correo electrónico	38
III.2.3.2 Almacenamiento de la data en la base de datos	38
III.2.3.3 Visualización de los datos de los pacientes.....	38
III.2.3.4 Detección de alarmas de Hipo e Hiperglicemia	39
III.3 Fase III. Pruebas.....	40
IV.1. Programa Paciente.	41
IV.2. Programa Médico.....	45
Capítulo V	54
Conclusiones y Recomendaciones.....	54
Referencias Bibliográficas	58
ANEXO A.....	60
Gama de Glucómetros.....	60

ANEXO B	69
Guía de consulta rápida One Touch Ultra.....	69
APÉNDICE A	71
Manual de Usuario Programa- Paciente, Programa Médico	71
APÉNDICE B.....	83
Códigos desarrollados para las aplicaciones de programa médico y programa paciente	83

Índice de Figuras

Figura 1. Proceso de Transmisión y Recepción de datos.....	8
Figura 2. Representación de la topología del Bus USB.....	10
Figura 3. Niveles del Modelo OSI.....	12
Figura 4. Encabezado IPv4 de un datagrama IP.....	15
Figura 5. Encabezado TCP.....	20
Figura 8. Glucómetro conectado al PC.....	31
Figura 9. Data transmitida y recibida por el PC.....	32
Figura 10. Carácter enviado por el PC.....	32

Figura 11. Bandeja de Entrada del correo.....	34
Figura 12. Correo Recibido por el Médico.....	35
Figura 13. Algoritmo empleado por el programa Médico.....	36
Figura 15. Mensaje arrojado por el programa.....	41
Figura 16. Mensaje arrojado por el programa sin conectar el Glucómetro.....	42
Figura 17 Ventana Programa Paciente.....	42
Figura 18. Pasos Para la ejecución del Programa Paciente.....	43
Figura 19.Paso 3 del programa Paciente.....	44
Figura 20. Correo Electrónico Recibido por el Servidor.....	45
Figura 21. Ventana Principal Programa Médico.....	46
Figura 22. Muestra de alarma.....	48
Figura 23. Opción Nuevo Paciente.....	49
Figura 24. Perfil en Access.....	49
Figura 25. Mediciones del Paciente.....	50
Figura 26. Historia del Paciente.....	50
Figura 27. Opción Ver Gráfica.....	51

Figura 28. Opciones de Visualización. 52

Figura 29. Impresión de las mediciones..... 53

Índice de Tablas

Tabla 1. Algunas opciones del IP. -----17

Tabla 2. Primitivas de Socket para TCP. -----19

Tabla 3. Puertos asignados empleados. -----19

Tabla 4. Los métodos de solicitud HTTP integrados. -----25

Tabla 5. Tabla comparativa de algunos de los glucómetros estudiados. -----29

Tabla 6. Algunos caracteres enviados por el PC. -----33

Tabla 7. Caracteres ASCII enviados por el PC. -----43

Dedicatoria y Agradecimientos

A dios le damos gracias por mantener nuestra fé, valor, dedicación, constancia, esfuerzo y esperanza para alcanzar esta meta.

Con profundo agradecimiento para nuestros padres: Gilmar Solórzano y Agustín Rendón (padres de Gilmar), Rosa Riquelme y Juan Arenas (padres de Juan), quienes nos guiaron a culminar nuestros estudios universitarios, al apoyarnos incondicionalmente en todo momento.

A nuestros hermanos y abuelos por sus consejos.

A nuestros profesores, no encontramos palabras precisas de agradecimiento para con nuestra formación universitaria.

Especial agradecimiento al Profesor Iván Escalona (Tutor) en este trabajo de grado, quien nos sembró esa semillita de ansiedad de querer saberlo todo, de estar a la expectativa y de investigar siempre más adelante.

Orgullosos nos sentimos de haber seleccionado la Universidad Católica Andrés Bello para lograr este objetivo, formando parte de la I promoción de Ingenieros en Telecomunicaciones.

A ustedes queremos dedicarles este éxito, no nos detendremos en esta primera meta, seguiremos con más retos cosechando triunfos.

Finalmente, a nuestros compañeros de la I promoción, por ser personas de gran calidad humana, con quienes compartimos y convivimos los mejores momentos como estudiantes, les dejamos nuestras más sinceros sentimientos de amistad y agradecimientos, encuentren donde se encuentren.

A todos, muchas gracias.

Introducción

El desarrollo del presente Trabajo Especial de Grado en el área interdisciplinaria de Telemedicina surge de la necesidad de registrar, organizar y plasmar indicadores que proporcione una alternativa a centro asistenciales médicos con el fin de controlar y adaptar de manera eficiente un tratamiento a personas que presentan alteraciones características en el metabolismo, por una enfermedad llamada Diabetes.

El Trabajo Especial de Grado esta constituido por 5 capítulos. En el primer capítulo se describe el planteamiento del problema, se detalla el objetivo general y los específicos, se establece la justificación de dicho trabajo culminando con las limitaciones y alcance del mismo. El Capítulo dos, es una revisión de teorías, conocimientos, estudios realizados, que sirven de base y soporte para la ejecución del proyecto. En el tercer capítulo se establece la metodología empleada conjuntamente con el desarrollo del prototipo, describiendo los pasos, actividades, técnicas requeridas para llevar a cabo la elaboración del mismo. En el cuarto capítulo se exponen y analizan los resultados obtenidos, los cuales responden a la metodología y objetivos planteados, y finalmente en el quinto capítulo se dan las conclusiones y recomendaciones como consecuencia final del esfuerzo realizado para la obtención del prototipo.

Diseñar un sistema remoto automatizado que permita un seguimiento del nivel de glucosa en la sangre de pacientes diabéticos requiere de búsqueda de información, indicadores resaltantes, que permitan establecer las fases necesarias para el desarrollo del prototipo haciendo uso del conocimiento adquirido en la carrera de Ingeniería de Telecomunicaciones.

Capítulo I

Planteamiento del Proyecto

I.1 Planteamiento del proyecto

La Diabetes Mellitus, es una enfermedad de carácter metabólico, caracterizada por presentar el organismo, niveles inadecuados de glucosa. A través de los años esta enfermedad ha llegado a penetrar en los distintos rangos sociales, tornándose como un problema de salud pública en la mayoría de los países latinoamericanos.

En Venezuela, esta enfermedad, forma parte de las principales causas de muerte.

La diabetes es una enfermedad que no tiene cura, pero puede ser controlada y para estar en un buen estado de salud, los pacientes deben seguir un estricto tratamiento, el cual les permitirá mantener los niveles de glicemia adecuadas en el organismo.

El tratamiento de la diabetes requiere de un seguimiento exhaustivo en cuanto a la evolución del paciente, en donde se debe realizar un chequeo periódico del mismo. Un control estricto implica recursos no siempre disponibles y difíciles de mantener en una rutina clínica. Además, en algunos casos el control debe ser diario para garantizar la eficiencia del tratamiento. Para dicho control se requiere necesariamente realizar una consulta médica, que en algunos casos se torna difícil de establecer debido a factores de tiempo y disponibilidad tanto del paciente como del médico.

Es por eso que el campo interdisciplinario de la Telemedicina brinda una alternativa que permite realizar el seguimiento remoto de pacientes diabéticos, para que no se tenga que acudir necesariamente a una consulta médica a la hora de llevar a cabo el tratamiento; por otro lado que sea eficiente permitiendo mostrar en tiempo

real el avance y efecto del mismo. Con una forma de control remota de la diabetes, se mejora la calidad de vida del paciente, su progreso en el tratamiento, la comunicación médico-paciente y la detección sintomática de problemas en el mismo.

I.2 Objetivos

Objetivo General

Desarrollar un prototipo que permita realizar un seguimiento continuo de la diabetes a través de un sistema remoto automatizado que garantice la comunicación médico-paciente, ayude a disminuir la frecuencia de visitas médicas del paciente y logre una mejor eficiencia en el tratamiento.

Objetivos Específicos

- Obtener información para el desarrollo del prototipo: Determinar los factores diabéticos incidentes en la determinación del prototipo.
- Realizar una selección de equipos (Glucómetros) entre los disponibles en el mercado nacional, que cumplan con los requerimientos del proyecto.
- Desarrollar un sistema de comunicación entre el glucómetro y el PC, que permita la compatibilidad y una transmisión de información eficaz desde el glucómetro hacia el PC vía Usb.
- Desarrollar un sistema de comunicación para el envío de datos entre el PC del paciente o del ambulatorio y el servidor en el centro médico principal de salud, que permita al médico conocer los niveles de glicemia del paciente frecuentemente y en tiempo real, sin necesidad de que el mismo asista una consulta médica.
- Ofrecer herramientas tanto a médicos como pacientes que permitan capturar, gestionar, visualizar e interpretar datos de seguimiento de la enfermedad, así como intercambiar datos y mensajes vinculados con el estatus de la misma.

- Mejorar la calidad de vida del paciente al disminuir las visitas a consultas médicas.
- Proporcionar al paciente un servicio de “autonomía supervisada”, accediendo de manera remota a un sistema especializado que registra indicadores importantes para poder llevar el control de la enfermedad.

I.3 Justificación

La Diabetes Mellitus es un paradigma de enfermedad crónica, la cual se caracteriza por presentar alteraciones en el metabolismo de las personas, como mostrar altos niveles de glucosa en la sangre, anomalía que puede traer como consecuencias otras complicaciones en el organismo como precipitaciones de accidentes cardiovasculares, ó cerebro vasculares, lesiones neurológicas, coma, riesgos vitales. La Diabetes representa uno de los principales problemas de salud en el mundo. Dicha enfermedad se caracteriza por requerir de un seguimiento exhaustivo de los niveles de glicemia del paciente para lograr implementar un tratamiento eficiente.

Las personas características de esta enfermedad deben recurrir con frecuencia a centros de salud para chequear y controlar estos niveles de glicemia, en donde se deben establecer rutinas clínicas difíciles de mantener, ya que hacen uso de recursos que no siempre se encuentran disponibles (disponibilidad de: médico, paciente, glucómetro). Es por ello que se requiere la implementación de un sistema que permita controlar los niveles de glicemia de manera remota y automatizada, evitando así las frecuentes consultas médicas.

Actualmente, en Venezuela no se dispone de sistemas remotos automatizados que proporcionen este tipo de servicio, es por esto que se busca desarrollar un sistema que brinde estas prestaciones, empleando recursos y herramientas tecnológicas que están disponibles como glucómetros con puertos USB, dispositivos USB que permiten gestionar el mismo, PC's, Internet, componentes que son prácticos, de fácil acceso,

garantizando que dicho sistema, pueda ser usado de igual manera entre personas de distintos estratos sociales.

I.4 Limitaciones y Alcances

El proyecto desarrollado incluye un servicio alámbrico de seguimiento de la diabetes para una clínica u hospital y varios pacientes, empleando un sistema remoto automatizado.

Este trabajo especial de grado, sólo abarcará el diseño y creación de un prototipo adaptado a las necesidades del paciente, médico, requerimientos del sistema y recursos disponibles dentro del mercado venezolano.

Una vez experimentado el prototipo, de acuerdo con los resultados obtenidos de las diferentes pruebas de transmisión al servidor, se garantiza el control de los niveles de glucosa del paciente donde el médico establece un tratamiento de manera eficiente gracias a la ayuda que otorga el proyecto desarrollado.

En un futuro, existen posibilidades de crear módulos adicionales al prototipo inicial con el fin de brindar comodidades y nuevas prestaciones, tanto al paciente como el médico. Un ejemplo de esto es establecer conexión inalámbrica entre el glucómetro y el centro de salud a través del sistema móvil celular con ayuda de la interfaz infrarroja de los equipos. Este módulo no fue implementado porque los glucómetros con puerto infrarrojo en el mercado venezolano no incluyen el software necesario para descifrar, mediante un sniffer, la información enviada por el glucómetro y así desarrollar la aplicación paciente-médico.

Capítulo II

Marco Teórico

II.1 Diabetes

La diabetes Mellitus es una enfermedad que se caracteriza por la presencia de altos niveles de glucosa en la sangre debido a defectos en la secreción de insulina, sustancia producida por el páncreas, la cual convierte los alimentos en energía (en forma de glucosa). Es importante comprender primero el proceso normal de metabolismo de los alimentos (Mediks, 2006, <http://mediks.com/saludyvida/articulo/>). Varios procesos ocurren durante la digestión.

Primero, la glucosa, un azúcar que es fuente de combustible para el cuerpo, entra en el torrente sanguíneo. Luego el páncreas, es el órgano que produce una sustancia llamada insulina la cual es muy importante debido a que es la que transporta la glucosa producida por el páncreas del torrente sanguíneo a los músculos, grasas y células hepáticas, para que estos puedan utilizar la sustancia como combustible permitiendo así su funcionamiento (Grupo Diabetes SAMFyC, 1996, <http://www.cica.es/aliens/samfyc/>).

Las personas que tienen Diabetes se caracterizan por poseer niveles inadecuados (altos o bajos) de glucosa en la sangre lo que ocurre bien porque el páncreas no produce suficiente insulina como para permitir el transporte adecuado de la glucosa del torrente sanguíneo, a los músculos, grasas y células hepáticas; bien porque los músculos, grasas y células hepáticas no responden de manera normal a la glucosa, ó bien porque pueden darse ambos casos simultáneamente.

La diabetes se presenta en diferentes tipos siendo las más comunes:

Tipo I: Se diagnostica en la infancia. El cuerpo no produce o produce muy poca insulina y se necesitan inyecciones diarias de esta para poder sobrevivir y, de no hacerse correctamente, se pueden presentar emergencias médicas.

Tipo II: Es más común que el tipo I y corresponde al 90% de todos los casos de Diabetes y generalmente se presenta en la edad adulta. El páncreas no produce suficiente insulina para mantener los niveles de glucosa en la sangre normales, a menudo, debido a que el cuerpo no responde bien a la insulina. Muchas personas con este tipo de diabetes, incluso no saben que la tienen, a pesar de ser una condición grave (Federación Diabetológica Colombiana, 2006, <http://www.fdc.org.co/>).

Para poder llevar un control de los niveles de glucosa en la sangre, se hace imprescindible una cuantificación de los mismos. Un dispositivo para este fin es el glucómetro y se describe a continuación.

II.2 Glucómetro

Es un dispositivo que mide los niveles de glucosa en la sangre, analizando los cambios en una tira reactiva sobre la que se ha depositado una pequeña muestra de sangre.

El glucómetro trae consigo unas micro-agujas que son desechables así como unas tiras reactivas. El procedimiento es muy sencillo: lo primero es obtener una gota de sangre, pinchando el pulpejo de cualquier dedo de las manos del paciente, obteniendo una gota de sangre, la cual se coloca sobre una tira reactiva. La tira impregnada con sangre se introduce en el glucómetro, se debe esperar algunos segundos para ver en la pantalla digital del glucómetro el nivel de glicemia del paciente que tiene en ese momento.

El resultado no es exacto, pero sí lo suficientemente cercano y fiable como para poder ajustar el tratamiento en función de los valores obtenidos.

Es importante resaltar que la micro aguja es desechable, solo se emplea una vez y jamás debe ser compartida con otra persona.

Una vez obtenido el nivel de glicemia, el paciente debe transmitir dichos datos para que sean recibidos por el médico en el centro hospitalario por medio del siguiente proceso (Ver Figura 1).

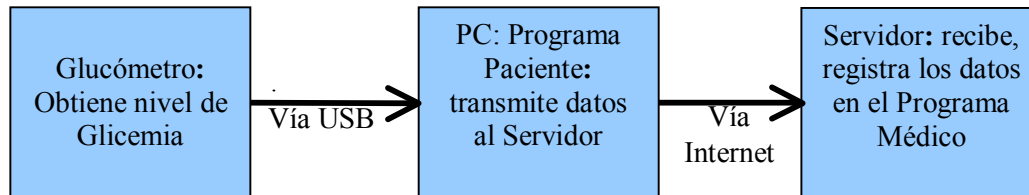


Figura 1. Proceso de Transmisión y Recepción de datos
Fuente: Elaboración Propia.

El paciente al obtener su nivel de glicemia por medio del glucómetro, conecta dicho dispositivo vía usb al PC. El PC por medio del Programa Paciente, recibe los datos y los transmite al servidor ubicado en el centro médico vía Internet.

El servidor recibe y registra los datos por medio del Programa Médico, en donde el médico podrá visualizar, analizar los datos y ajustar el tratamiento adecuado al paciente de manera eficiente.

Para la implementación de los dos programas utilizados tanto por el paciente como por el médico, se emplea en el presente trabajo Microsoft Visual Studio para desarrollar ambas interfaces.

Debido a este proceso de transmisión y recepción de datos, es primordial comprender los conceptos básicos empleados para el desarrollo del prototipo como: Protocolo USB, Modelo OSI, Protocolo Internet (IP), Protocolo de control de transmisión (TCP), Internet, Microsoft Visual Studio, los cuales se describen a continuación.

II.3 Protocolo USB

Universal Serial Bus, mejor conocido como Protocolo USB es una interfaz que permite conectar varios periféricos como mouse, impresoras, cámaras y otras unidades para transmitir una serie de datos de una forma sencilla, rápida basada en comunicación serial. Este tipo de interfaz proporciona altas tasas de transferencia.

El protocolo USB contempla una serie de elementos entre los cuales se encuentran los dispositivos USB (hubs) y el Host USB. La topología conformada por dichos elementos es en forma de estrellas apiladas donde un hub es el centro de cada estrella. Cada segmento de cable es una conexión punto-a-punto entre el host y los hubs o función, o un hub conectado a otro hub o función (Brodín & Giménez, 2006, <http://usuarios.lycos.es/kurganz/introduccion.html>).

La topología del bus USB se puede dividir en tres partes y se pueden observar en la figura 2:

- La capa física: Como están conectados los elementos físicamente.
- La capa lógica: Los roles y las responsabilidades de los elementos USB.
- La relación software del cliente-función: Como se ven mutuamente el software del cliente y los interfaces de las funciones relacionadas.

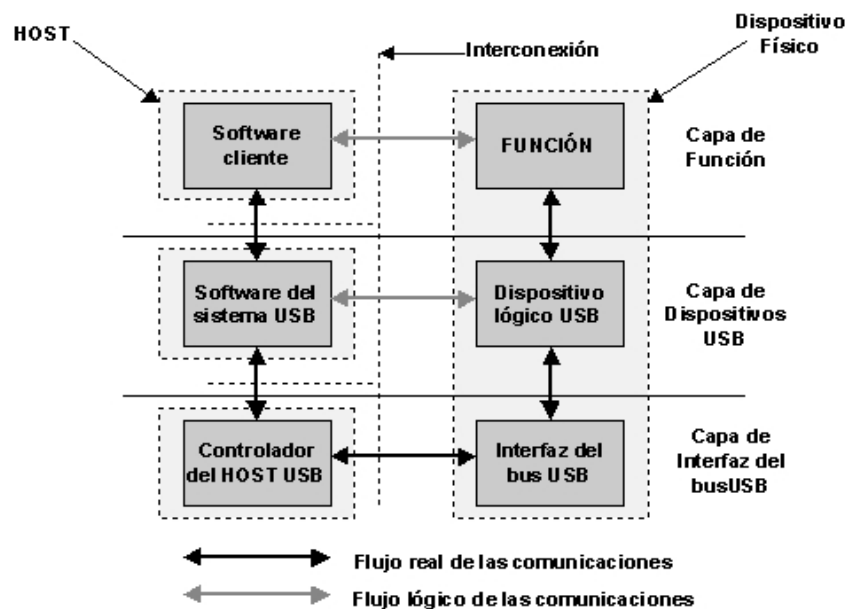


Figura 2. Representación de la topología del Bus USB.

Fuente: Brodín & Giménez, 2006, <http://usuarios.lycos.es/kurganz/introduccion.html>

El host incluye el software y el hardware sobre el cual se sostiene el USB. El host tiene la habilidad de procesar y gestionar los cambios de configuración que puedan ocurrir en el bus durante su funcionamiento. El host gestiona el sistema y los recursos del bus como el uso de la memoria del sistema, la asignación del ancho de banda del bus y la alimentación del bus. El host también ayuda al usuario con la configuración automática de los dispositivos conectados y reaccionando cuando son desconectados.

Un host puede soportar uno o más buses USB. El host gestiona cada bus independientemente de los demás. Los recursos específicos del bus como el ancho de banda asignado son únicos a cada bus. Cada bus está conectado al host a través de un controlador del host.

Sólo hay un host en cualquier sistema USB. Desde la interfaz USB hasta el sistema de host de la computadora es lo que se le llama controlador de host y puede estar implementado como combinación de hardware, o software.

El controlador de host está formado por el hardware y el software que permite a los dispositivos USB ser conectados al host. Este controlador es el agente iniciador del

bus, es decir es el que comienza las transferencias en el bus. El controlador de bus es el maestro en un bus USB.

El host USB interactúa con los dispositivos USB a través del controlador. Las funciones básicas del controlador de host son:

- Detectar la inserción o desconexión de dispositivos USB.
- Gestionar el flujo de control entre el host y los dispositivos.
- Gestionar el flujo de datos entre el host y los dispositivos.
- Coleccionar estadísticas de actividad y estado.
- Proveer una cantidad limitada de energía a los dispositivos conectados.

La transmisión de datos del PC al servidor, se realiza vía Internet por medio del protocolo TCP/IP (Protocolo de Control de Transmisión / Protocolo Internet), los cuales trabajan en diferentes capas del modelo OSI, y se describen a continuación.

II.4. Modelo OSI

OSI (*Open System Interconnection*) es un modelo que permite que dos sistemas diferentes se puedan comunicar independientemente de la arquitectura subyacente (Forouzan, 2002).

El modelo está basado en una arquitectura de 7 niveles (Ver Figura 3): el físico, el de enlace de datos, el de red, el de transporte, el de sesión, el de presentación, y finalmente el de aplicación.

Aplicación	Niveles de soporte al usuario
Presentación	
Sesión	
Transporte	Enlaza los dos soportes
Red	Niveles de soporte de red
Enlace	
Físico	

Figura 3. Niveles del Modelo OSI.
Fuente: Elaboración Propia

Los niveles físico, enlace y red son los niveles que le dan soporte a la red, los niveles de sesión, presentación y aplicación son los niveles de soporte al usuario, y el nivel de transporte enlaza estos dos soportes garantizando confiabilidad a la entrega de los paquetes.

Cuando un mensaje viaja de una máquina de origen a otro de destino, pasa a través de muchos nodos intermedios los cuales operan en los primeros tres niveles del modelo.

Entre cada par de niveles adyacentes existe una interfaz que es la que permite el paso de datos en bajada entre los distintos niveles de la máquina de origen y de subida a través de los niveles de la máquina de destino.

Los niveles que conforman el modelo OSI se describen a continuación:

Nivel Físico: Maneja las funciones necesarias relacionadas con la transmisión de flujo de datos a través del medio físico como especificaciones eléctricas, mecánicas de la interfaz y medio de transmisión. Lleva a cabo procedimientos para que sea posible la transmisión como: codificar el flujo de bits en señales eléctricas u ópticas;

definir la tasa de transmisión; sincronizar los relojes del emisor y receptor; por último definir el modo de transmisión entre ambos dispositivos, si va a ser simplex, semiduplex ó duplex.

Nivel de Enlace de Datos: es el que otorga la confiabilidad al nivel físico, para así supervisar que la información sea entregada de un nodo a otro. Entre las principales funciones de este nivel se tienen: divide el flujo de bits recibidos de la capa superior en tramas; añade una cabecera a la trama definiendo la dirección física del emisor y/o receptor de la trama; impone un mecanismo de control de flujo para prevenir desbordamiento en el receptor si la velocidad de este último es menor a la del emisor; incluye mecanismos para detectar y retransmitir tramas dañadas o perdidas; por último determina en todo momento que dispositivo lleva el control del enlace.

Nivel de Red: Asegura la entrega del paquete de un nodo a otro a través de múltiples redes. Entre las principales funciones de este nivel se contemplan: gestiona problemas de direcciones locales añadiendo a la cabecera de los paquetes provenientes de la capa superior direcciones lógicas que permitan identificar dicha trama si el paquete llega a traspasar otras redes; proporciona mecanismos de encaminamiento en nodos intermedios para que los paquetes lleguen al destino final.

Nivel de Transporte: responsable de que el mensaje llegue al destino de manera correcta y en orden, supervisando tanto el control de errores como el control de flujo. Entre las responsabilidades de este nivel se tienen: incluye en la cabecera una dirección de puerto que especifica a que aplicación dentro de la máquina de destino debe llegar el mensaje; divide el mensaje en segmentos proporcionando un número de secuencia para que estos puedan ser reensamblados de manera correcta en el destino y en el caso de que se pierdan segmentos estos sean reemplazados.

Nivel de Sesión: controla el diálogo de la red, estableciendo, manteniendo y sincronizando la interacción entre sistemas de comunicación. Entre las funciones principales de dicho nivel se tienen: permite el modo de transmisión simplex, semiduplex, ó duplex en el enlace nodo a nodo; permite que un proceso agregue

puntos de prueba en cada segmento del mensaje, para verificar la llegada de cada segmento en el destino.

Nivel de Presentación: se encarga de la sintaxis y semántica de la información. Traduce la información en flujo de bits para que esta pueda ser transmitida, además permite la interoperabilidad entre los distintos métodos de codificación. Este nivel es capaz de asegurar la privacidad, incluyendo métodos de cifrado para transformar la información a otro formato, proceso que se invierte en el destino. También incluye operaciones de compresión para reducir el número de bits a transmitir.

Nivel de Aplicación: Permite al usuario acceder a la red. Proporciona interfaces de usuario y soporte para servicios como el correo electrónico, el acceso y la transferencia de archivos remotos.

En resumen, el modelo OSI proporciona guías para la compatibilidad entre sistemas diferentes.

Antes del modelo OSI, se desarrolló otro modelo que tuvo mas penetración en el mundo Internet como lo es la familia de protocolos TCP/IP.

TCP/IP esta basado en 5 niveles, físico, enlace de datos, red, transporte y aplicación. Este es un protocolo jerárquico donde los niveles superiores son soportados por los niveles inferiores.

TCP/IP define dos protocolos en el nivel de transporte: Protocolo de control de transmisión (TCP) y el protocolo de datagramas de usuario (UDP). En el nivel de red se define el protocolo entre redes (IP).

II.5. IP. Protocolo Internet

El protocolo de Internet es un protocolo no orientado a conexión usado tanto por el origen como por el destino para la comunicación de datos a través de una red de

paquetes conmutados (Tanenbaum, 2003). Este protocolo trabaja a nivel de Capa de red.

En una red IP, los datos son enviados a manera de bloques conocidos como datagramas (Enciclopedia Wikipedia, 2006, [http:// es.wikipedia.org/wiki/ Protocolo_IP](http://es.wikipedia.org/wiki/Protocolo_IP)).

Un datagrama IP posee una parte de encabezado y una parte de texto, cuyo formato se puede observar en la siguiente figura. El encabezado (Ver Figura 4) tiene una parte fija de 20 bytes y una parte opcional de longitud variable.

Versión	IHL	Tipo de Servicio	Longitud Total		
Identificación			D F	M F	Desplazamiento del fragmento
Tiempo de Vida	Protocolo		Suma de Verificación del Encabezado		
Dirección de origen					
Dirección de destino					
Opciones (ó mas palabras)					

Figura 4. Encabezado IPv4 de un datagrama IP.
Fuente: Tanenbaum, 2003.

Entre los campos mostrados en la figura se describen los siguientes:

Versión: Lleva el registro de la versión del protocolo al que pertenece el datagrama.

IHL: Se incluye en el encabezado para indicar la longitud en palabras de 32 bits. El valor mínimo es de 5, que se aplica cuando no hay opciones. El valor máximo es de 4 bits que es 15, limitando el encabezado a 60 bytes y por lo tanto el campo de opciones a 40 bytes.

Tipo de Servicio: En este campo se distinguen las diferentes clases de servicios permitiendo varias combinaciones de velocidades y confiabilidad.

Longitud Total: Incluye todo el datagrama: tanto el encabezado como los datos. La longitud máxima es de 65,535 bytes.

Identificación: Este campo es necesario para que el host de destino determine a que datagrama pertenece el fragmento recién llegado. Luego del campo de Identificación viene un bit sin uso.

DF (*Don't Fragment*): Bandera de un bit, es una orden para los enrutadores para que no fragmenten el datagrama, ya que el destino es incapaz de unir las partes de nuevo.

MF (*More Fragment*): Significa más fragmentos. Todos los fragmentos excepto el último tienen activado esta bandera, la cual es de un bit, y así se determina cuando han llegado todos los fragmentos del datagrama.

Desplazamiento del fragmento: Indica en que parte del datagrama actual va este fragmento. Todos los fragmentos excepto el último del datagrama deben ser múltiplo de 8 bytes, unidad de fragmentos elemental. Fragmentar un datagrama es importante, ya que al conectar redes distintas pueden ocurrir problemas de compatibilidad debido a los diferentes tamaños de paquetes, y es por esta razón que se requiere de una fragmentación.

Tiempo de Vida: Contador que limita la vida del paquete. Se emplea en segundos, permitiendo una vida máxima de 255 seg.

Protocolo: En el host de destino, cuando se reciben todos los fragmentos y la capa de red ensambla dichas partes formando el datagrama completo, este campo indica el protocolo de las capas superiores a la cual debe entregarse el paquete completo.

Suma de verificación del encabezado: Verifica solamente si el encabezado está correcto empleando un algoritmo sumando todas las medias palabras de 16 bits usando aritmética de complemento a uno.

Dirección de Origen y de Destino: Indican el número de red y el número de host.

Opciones: Campo de longitud variable. Este campo se diseñó con el fin de que las versiones subsiguientes del protocolo incluyeran informaciones no presentes en el diseño original, y así permitir a los experimentadores probar ideas nuevas. El campo opciones se rellena para completar múltiplos de 4 bytes. De las opciones que se incluyeron inicialmente se reflejan en la tabla 1.

Opción	Descripción
Seguridad	Especifica que tan secreto es el datagrama
Enrutamiento estricto desde el origen	Indica la ruta completa a seguir
Enrutamiento libre desde el origen	Da una lista de los enrutadores que no deben evitarse
Registrar ruta	Hace que cada enrutador agregue su dirección IP
Marca de Tiempo	Hace que cada enrutador agregue su dirección y su marca de tiempo.

Tabla 1. Algunas opciones del IP.
Fuente: Tanenbaum, 2003.

Este protocolo debido a que trabaja a nivel de capa de red, proporciona servicios a protocolos pertenecientes a la capa de transporte, basándose en datagramas. La idea principal es la de enrutar paquetes del origen al destino.

II.6. TCP. Protocolo de Control de Transmisión.

En Internet se tienen dos protocolos principales que trabajan a nivel de capa de Transporte, uno orientado a conexión y otro no orientado a conexión. El protocolo no orientado a conexión es el UDP (*Protocolo de Datagramas de Usuario*), proporciona aplicaciones que envían datagramas IP encapsulados sin tener que establecer una conexión. UDP puede emplearse para interacciones cliente-servidor multimedia.

El otro protocolo empleado en la capa de transporte, orientado a conexión es TCP (*Protocolo de Control de Transmisión*), el cual se diseñó para proporcionar un flujo de bytes confiable de extremo a extremo a través de una interred no confiable. En una interred se pueden encontrar diferentes topologías, anchos de banda, retardos, tamaños de paquetes variables y otros parámetros, donde TCP se adapta de manera dinámica a dichas propiedades sobreponiéndose a muchos tipos de fallas que puedan ocurrir.

Una entidad de transporte TCP, acepta flujos de datos de usuarios de procesos locales, se dividen en fragmentos que no excedan los 64KB, y cada fragmento es enviado como un datagrama IP independiente. Cuando los datagramas que contienen datos TCP llegan a la máquina de destino se pasan a la entidad TCP, la cual reconstruye los flujos de bytes originales.

Debido a que la capa IP no proporciona garantía de que los datagramas se entreguen de manera apropiada, corresponde a TCP terminar los temporizadores y retransmitir los datagramas conforme sea necesario. TCP reensambla los mensajes en la secuencia apropiada si los datagramas llegan en orden incorrecto, así que dicho protocolo proporciona confiabilidad que la mayoría de los usuarios desean y que IP no proporciona y es por esto que ambos protocolos trabajan conjuntamente.

El servicio TCP se obtiene al hacer que tanto el servidor como el cliente creen puntos terminales llamados sockets. Cada socket posee una dirección IP del host así como un número de 16 bits local al host llamado puerto. Para establecer una conexión entre un socket de la máquina de origen y la máquina de destino se realizan por medio de llamadas de socket (Ver tabla 2).

Primitiva

Significado

SOCKET Crea un nuevo punto Terminal de comunicación

BIND	Adjunta una dirección local a un socket
LISTEN	Anuncia la disposición a aceptar conexiones; indica el tamaño de cola
ACCEPT	Bloquea al invocador hasta la llegada de un intento de conexión
CONNECT	Intenta establecer activamente una conexión
SEND	Envía datos a través de la conexión
RECEIVE	Recibe datos de la conexión
CLOSE	libera la conexión

Tabla 2. Primitivas de Socket para TCP.

Fuente: Tanenbaum, 2003.

Para los números de puertos menores que 1024, se le dicen puertos bien conocidos y se reservan para servicios estándar (Ver tabla 3).

Nº Puertos	Protocolo	Uso
25	SMTP	Correo Electrónico
80	HTTP	World Wide Web
110	POP-3	Acceso remoto al correo electrónico

Tabla 3. Puertos asignados empleados.

Fuente: Tanenbaum, 2003.

La entidad TCP emisora y receptora intercambian datos en forma de segmentos. El segmento consiste en un encabezado fijo TCP de 20 bytes seguido de cero o más bytes de datos. El software TCP decide el tamaño de los segmentos puede acumular datos de diferentes escrituras para formar un segmento o dividir los datos de una escritura en varios segmentos. Los límites que restringen el tamaño del segmento, primero, en cada segmento, incluido el encabezado TCP debe caber una carga útil de 65,515 bytes del IP. Segundo, cada red tiene una unidad máxima de transferencia (MTU) y cada segmento debe caber en la MTU, la MTU generalmente es de 1500 bytes definiendo el límite superior del tamaño del segmento.

Cada segmento comienza con un encabezado de formato fijo de 20 bytes. El encabezado fijo va seguido de opciones de encabezado que va hasta $65,535 - 20 = 65,495$ bytes de datos, donde los primeros 20 se refieren al encabezado IP y los segundos al encabezado TCP (Ver figura 5).

Puerto de Origen				Puerto de Destino			
Número de secuencia							
Número de confirmación de recepción							
Longitud del encabezado TCP	U	/	E	E	S	E	Tamaño de Ventana
	I	C	S	S	N	I	
	C	E	E	T	T	T	
Suma de verificación				Apuntador urgente			
Opciones (0 o más palabras de 32 bits)							
Datos (Opcional)							

Figura 5. Encabezado TCP.
Fuente: Tanenbaum, 2003.

Los campos mostrados en la figura se describen a continuación:

Puerto de origen y de destino: Indican los puntos locales de conexión. La dirección de un puerto mas la dirección IP de su host forman un punto terminal único de 48 bits.

Los campos numero de secuencia y numero de confirmación de recepción ambos son de 32 bits de longitud.

Longitud del encabezado TCP: Indica la cantidad de palabras de 32 bits contenidas en el encabezado TCP. Este campo es necesario ya que el campo opciones es de longitud variable por lo que el encabezado también. Este campo indica el comienzo de los datos del segmento. El campo siguiente a este es de 6 bytes y no se usa.

URG: es un indicador de un 1 bit, que se establece en 1, cuando esta en uso el apuntador urgente el cual sirve para indicar un desplazamiento en bytes a partir del numero actual de secuencia en el que se encuentran los datos urgentes.

ACK: este bit cuando se encuentra en 1, indica que el número de confirmación de recepción es válido. Si el indicador esta en cero significa que no se contiene una confirmación de recepción, y se ignora dicho campo.

PSH: bit que indica datos que se deben transmitir de inmediato.

RST: se usa para establecer una conexión que se ha confundido debido a una caída de host u otra razón, también sirve para rechazar un segmento no valido o un intento de abrir una conexión.

SYN: bit que se utiliza para establecer conexiones.

FIN: bit que se utiliza para liberar una conexión, indica que el emisor no tiene más datos para transmitir.

Tamaño de la ventana: el control de flujo TCP se maneja usando una ventana corrediza de tamaño variable, con este campo se indica la cantidad de bytes que pueden enviarse comenzando por el byte cuya recepción se ha confirmado.

Suma de verificación: Este campo proporciona una suma de verificación del encabezado, los datos y el pseudo encabezado para dar un sentido de confiabilidad en la transmisión.

Opciones: Se adicionan características no cubiertas por el encabezado normal. Entre las opciones se encuentra una donde el host especifica la carga útil TCP máxima que está dispuesto a aceptar.

Los protocolos a nivel de la capa de transporte como UDP, TCP deben ser capaces de administrar conexiones a través de redes no confiables. En la capa de transporte se

deben manejar todas las primitivas de servicio, administrar conexiones y temporizadores y asignar y usar créditos.

TCP proporciona un flujo de bytes bidireccional y confiable. Los enrutadores de Internet pueden fragmentar los segmentos, en donde los hosts deben reensamblarlos.

Las capas que se encuentran por debajo de la capa de aplicación proporcionan transporte confiable pero no el trabajo verdadero para los usuarios, en donde la capa de aplicación tiene gran relevancia.

En la capa de aplicación se necesitan protocolos de apoyo que permitan el funcionamiento de las aplicaciones reales, como el DNS (*Sistema de nombres de dominio*) que maneja asignación de nombres de Internet. En Internet también corren otras aplicaciones reales como correo electrónico, World Wide Web y multimedia.

II.7. Internet

Para Internet se emplea el DNS (*Sistema de nombres de dominio*), cuya esencia es la invención de un esquema de nombres jerárquicos basado en dominios y un sistema de base de datos distribuido para implementar este esquema de nombres. DNS se usa principalmente para relacionar los nombres de host y direcciones de correo electrónico en direcciones IP.

Para el envío de correos electrónicos se emplea el protocolo SMTP (*Protocolo Simple de Transporte de Correo*). En Internet, el correo electrónico se entrega al hacer que la máquina de origen establezca conexión TCP con el puerto 25 de la máquina de destino. SMTP acepta las conexiones de entrada y copia mensajes de ellas en los buzones adecuados.

Una vez que se establece una conexión TCP con el puerto 25, la máquina emisora opera como cliente, espera que la máquina receptora, operando como servidor hable primero. El servidor comienza enviando una línea de texto que proporcionando su

identidad, indicando si esta preparado para recibir el correo. En caso de que no lo este, el cliente libera la conexión.

El correo electrónico se entrega al hacer que el emisor establezca una conexión TCP con el receptor y después que envíe el correo electrónico a través de ella. Este modelo funcionó hasta que la red Internet se empezó a extender, y en el momento en el que el cliente necesitaba enviar un correo electrónico pero el receptor no se encontraba en línea. La solución para esto, es que un agente de transferencia de mensajes en una máquina ISP acepte correo electrónico para sus clientes y lo almacene en sus buzones en una máquina ISP debido a que este agente puede estar en línea todo el tiempo, el correo electrónico puede enviarse las 24 horas del día. Para que el usuario obtenga el correo electrónico del agente de transferencia de mensajes ISP, se creó el protocolo POP3 (*Protocolo de Oficina de Correos Versión 3*).

POP3 inicia cuando el usuario arranca el lector de correo. Éste llama al ISP y establece una conexión TCP con el agente de transferencia de mensajes en el puerto 110. Una vez establecida la conexión, el protocolo POP3 pasa por tres estados de secuencia: Autorización, Transacciones, Actualización.

El estado de autorización se enlaza con el inicio de sesión por parte del usuario. El estado de transacción se relaciona con el hecho de que el usuario colecciona los mensajes de correo electrónico y los marca para eliminación desde el buzón. El estado de actualización se encarga de que los mensajes de correo electrónico se eliminen.

Durante el estado de autorización, el cliente envía su nombre y su contraseña. Una vez establecida la conexión, el cliente puede emplear comandos POP3 al servidor para realizar una operación. El cliente puede enviar el comando LIST, que hace que el servidor realice una lista del contenido del buzón, un mensaje por línea indicando de qué longitud es ese mensaje.

Para recuperar los mensajes el cliente puede emplear el comando RETR y los puede marcar para eliminarlos usando el comando DELE.

El cliente para terminar el estado de transacción y entrar en actualización puede emplear el comando QUIT donde el servidor envía una respuesta y termina la conexión TCP.

Para acceder a documentos distribuidos en múltiples máquinas se vincularon por medio de una gran estructura como lo es *World Wide Web* también conocida como *WEB*, en Internet.

Para desplegar una página web se hace uso de un navegador, y éste concentra los elementos de la página desplegada. Cuando se selecciona un elemento, el navegador sigue el hipervínculo y obtiene la página seleccionada.

El hipervínculo para nombrar cualquier página que se encuentre en la web necesita de URLs (*Localizadores Uniformes de Recursos*), el cual esta conformado por tres partes: el nombre del protocolo (http), el nombre del DNS de la máquina donde se localiza la página y el nombre del archivo que contiene la página. Una vez que el navegador obtiene la dirección IP del servidor, establece una conexión TCP con el puerto 80 del servidor.

Un servidor web en su ciclo principal realiza los siguientes pasos: Acepta una conexión TCP de un cliente (navegador), Obtiene el nombre del archivo solicitado, Obtiene el archivo del disco, Regresa el archivo al cliente, Libera la conexión TCP.

El protocolo utilizado en World Wide Web es HTTP (Protocolo de transferencia de Hipertexto). Dicho protocolo especifica que mensajes pueden enviar los clientes a los servidores y que respuestas se obtienen. Todos los usuarios y servidores deben obedecer este protocolo.

La importancia de TCP se basa en que ni los navegadores ni los servidores deben preocuparse por los mensajes largos, perdidos o duplicados, ni por confirmaciones de recepción.

HTTP se diseñó para utilizarlo en web, posteriormente se le agregaron otro tipo de métodos que estarán orientadas a aplicaciones futuras. Estos métodos son operaciones diferentes a las de solicitar una página Web. En la tabla 4 se pueden listar los métodos integrados empleados en HTTP.

Método	Descripción
GET	Solicita la lectura de una página Web
HEAD	Solicita la lectura del encabezado de una página Web.
PUT	Solicita el almacenamiento de una página Web.
POST	Inserta algo a un recurso con nombre (por ejemplo, una página web).
DELETE	Elimina una página Web.
TRACE	Repite la solicitud entrante
CONNECT	Reservado para uso futuro
OPTIONS	Consulta ciertas opciones.

Tabla 4. Los métodos de solicitud HTTP integrados.
Fuente: Tanenbaum, 2003.

El método HEAD puede emplearse para obtener la fecha de la última modificación de la página Web

El método PUT permite la construcción de una colección de páginas Web en un servidor remoto.

El método POST, al igual que PUT transporta un URL pero en lugar de reemplazar los datos existentes se insertan nuevos datos.

El método TRACE es solo para la depuración, indica al servidor que regrese la solicitud cuando las solicitudes no se procesan de manera correcta y así el usuario sabe cual es la solicitud que realmente ha recibido del servidor.

El método CONNECT no se utiliza en la actualidad y por último el método OPTIONS sirve para consultar al servidor sobre las propiedades que tiene el usuario ó de archivos específicos (TanenBaum, 2003).

II.8 Microsoft Visual Studio

Tanto el paciente como el médico emplearan interfaces desarrolladas en Microsoft Visual Studio.

Conformada por varios lenguajes de programación, entre ellos Visual C++ y Visual Basic. Este último es un lenguaje de programación basado en el viejo lenguaje BASIC, brinda un sistema completo para el desarrollo de aplicaciones para el sistema operativo Microsoft Windows. Visual Basic permite escribir, editar, y probar aplicaciones de Microsoft Windows (Microsoft Visual Studio, 2006, http://es.wikipedia.org/wiki/Microsoft_Visual_Studio).

Capítulo III

Marco Metodológico y Desarrollo

En este capítulo se describe de manera detallada, las actividades y procedimientos que se llevaron a cabo para lograr cada una de las fases de investigación de acuerdo a los objetivos planteados, estas son (Ver Figura 6):

Fase I Teórica: Comprendió todo lo relacionado con el levantamiento de información con respecto a los conceptos necesarios para la realización del prototipo

Fase II Técnica: Contempló todo lo relacionado con el diseño del prototipo, las actividades que se realizaron para lograr el mismo.

Fase III Pruebas: En esta fase se realizaron pruebas al prototipo con el fin de mejorar el mismo y lograr los objetivos planteados.



Figura 6. Fases de Investigación.
Fuente: Elaboración Propia

Como se observa en la figura, la culminación de una fase conlleva a la otra, están debidamente enlazadas para así lograr el objetivo planteado, que es la obtención del prototipo.

III.1 Fase I. Teórica: Levantamiento de Información

En esta fase se llevó a cabo la documentación y el levantamiento de información para obtener los indicadores necesarios con el fin de diseñar el prototipo a través de artículos, documentos, consultas en línea, libros, etc.

Las actividades y procedimientos que se contemplaron esta fase se describen a continuación.

III.1.1 Estudio de indicadores y parámetros del proyecto

Se realizó un estudio detallado de los indicadores y parámetros resaltantes los cuales son imprescindibles para la elaboración del proyecto, dentro de los cuales se puede mencionar los conceptos básicos de Diabetes “*Mellitus*” y todo lo concerniente a la enfermedad: causas, síntomas, tipos, tratamientos, indicadores de la enfermedad; se estudió el funcionamiento del glucómetro, se investigó a fondo el protocolo USB, dispositivos USB, Protocolo IP.

En cuanto a la enfermedad, una vez estudiado los tipos de diabetes, se requirió de apoyo médico profesional mediante una reunión con médicos especialistas (endocrinos) del centro de salud Santa Inés para la definición del tipo de paciente con quien se podría usar el prototipo, aquí se concretó que el diseño estaría enfocado a pacientes con Diabetes tipo dos, debido a que un 90% de los casos de diabetes representan a este tipo, además de que dichos pacientes no requieren inyecciones diarias de insulina en comparación con los pertenecientes al tipo I.

Para el glucómetro se estudiaron los pasos necesarios para la obtención del nivel de glucosa, y a su vez se recibió apoyo médico profesional para una mejor comprensión del uso del glucómetro.

III.1.2 Estudio y Selección de Equipos

Se realizó una búsqueda dentro del mercado venezolano de glucómetros con dispositivos USB cuyas prestaciones cumplieran con los requisitos y parámetros necesarios para la elaboración del prototipo. Dichos dispositivos deben ser de fácil acceso y prácticos, además de ser de costos razonables para así brindar facilidades al paciente.

Entre los distintos glucómetros estudiados (Ver Anexo A, Gama de glucómetros), estos debían poseer puerto USB para así permitir la interconexión de dicho dispositivo con el PC y también deben traer el software para posteriormente obtener el patrón de transmisión del dispositivo. En la siguiente tabla se comparan algunos de los glucómetros estudiados.

Glucómetro	Disponible en mercado Venezolano	Software incorporado	transmisión de datos a PC
Accu – Chek Active	Sí	No	Vía Infrarrojo
Accu – Chek Go	Sí	No	Vía Infrarrojo
One Touch Ultra	Sí	Sí	Vía USB

Tabla 5. Tabla comparativa de algunos de los glucómetros estudiados.
Fuente: Elaboración Propia.

El glucómetro escogido fue el OneTouch Ultra de marca Lifescan, cuyo manual de usuario resultó muy práctico y se puede observar en el Anexo B.

III.2 Fase 2. Técnica: Diseño del Prototipo.

En esta fase se contemplan tres actividades fundamentales, las cuales son: (Ver Figura 7).

- Actividad 1. Compatibilidad Glucómetro – PC: en esta actividad se logra la compatibilidad entre el glucómetro escogido y la computadora personal.
- Actividad 2. Transmisión PC-Paciente al Servidor-Médico: se transmite la información necesaria enviada por el paciente al computador (servidor) del médico.
- Actividad 3. Interfase de Gestión del Médico: Por último en esta actividad se procede a la creación de una interfase en el computador del médico de fácil manejo, que permita almacenar la información de cada paciente en forma segura, y otorgue informes escritos y gráficos de todas y cada una de las

mediciones realizadas por el paciente, para que así el médico pueda llevar un seguimiento de la enfermedad y proporcionar un tratamiento que se adapte a las necesidades del paciente, así como dar mensajes de alerta al médico en caso de que las medidas se salgan de límites preestablecidos.

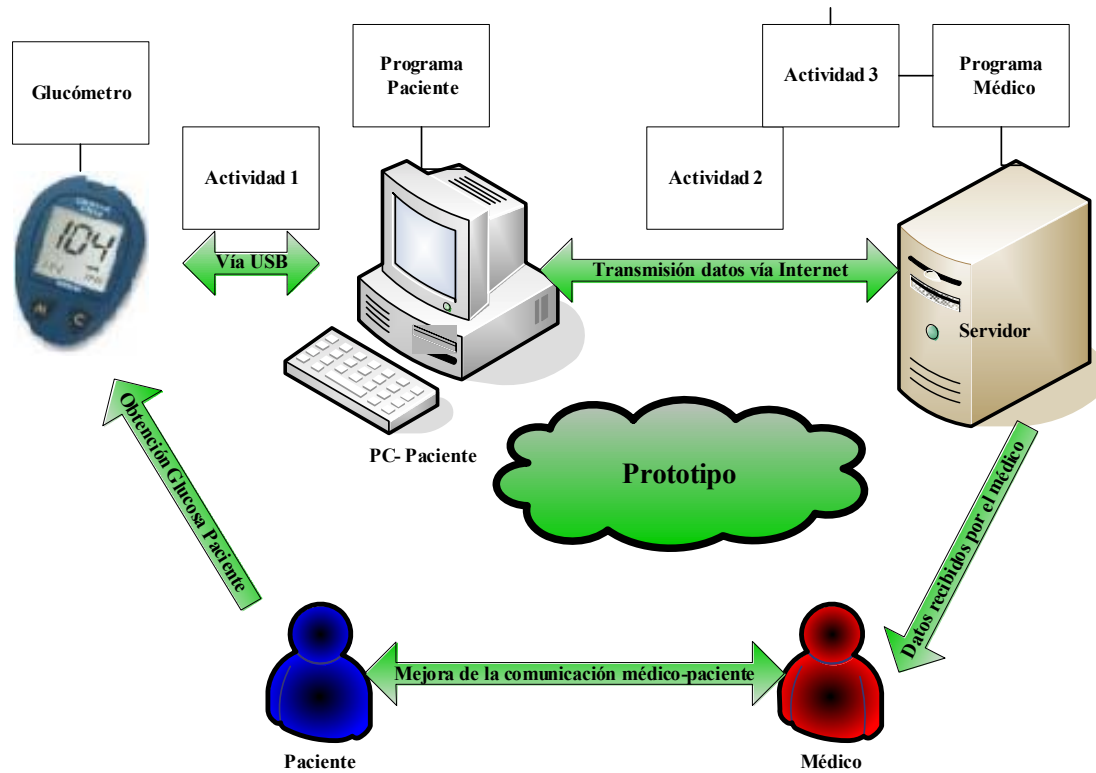


Figura 7. Actividades de la fase Técnica.
Fuente: Elaboración Propia

III.2.1 Compatibilidad Glucómetro – PC

Antes de explicar el proceso de compatibilidad entre dispositivos, es preciso aclarar que el glucómetro posee un cable USB para la conexión con el computador. Por configuración de controladores (*drivers*), el puerto USB donde se conecta dicho cable es reconocido como un puerto COM, específicamente COM 4. Es por ello que se

trató al glucómetro como un dispositivo conectado por puerto serial a lo largo de todo el proyecto.

Las características de este puerto son: velocidad de transmisión 9600 baudios, paridad nula, 8 bits de data por trama y un bit de stop. De esta forma se pudo sincronizar el intercambio de información con el glucómetro

Tomando en cuenta lo explicado anteriormente, se procedió a descifrar el código que emplea el glucómetro *One Touch Ultra* para establecer comunicación con el PC. Aquí se utilizaron dos programas, el *Software One Touch* adquirido con el glucómetro y un monitor del puerto *COM COM Port Sniffer*. Este último es una herramienta que permite visualizar toda la data que transita por un puerto serial determinado en código binario, hexadecimal y código ASCII.

Una vez realizada una medición de prueba de glucosa en la sangre, se procedió a conectar el glucómetro al PC mediante el cable USB como se muestra en la Figura 8.



Figura 8. Glucómetro conectado al PC.

Fuente: Elaboración Propia.

Posteriormente, se ejecutaron los programas *Software One Touch* y *COM Port Sniffer*, de manera que el monitor del puerto COM pudiera detectar toda la data transmitida y recibida por el PC desde el glucómetro. En la figura 9 se muestra el *COM Port Sniffer* con las entradas y salidas de data correspondientes.

Seq	Type	Time	Request	I/O	Device Object	IRP	Status
47	URB	10:59:13:187	BULK_OR_INTERRUPT_TRANSFER	IN	0x81C8D030	0x8227B5B8	STATUS_SUCCESS
48	URB	10:59:13:187	BULK_OR_INTERRUPT_TRANSFER	OUT	0x81944DE8	0x822A3940	STATUS_SUCCESS
49	URB	10:59:13:187	BULK_OR_INTERRUPT_TRANSFER	OUT	0x81C8D030	0x822A3940	STATUS_SUCCESS
50	URB	10:59:13:187	BULK_OR_INTERRUPT_TRANSFER	IN	0x81C8D030	0x8195C2B8	STATUS_PENDING
51	URB	10:59:13:187	BULK_OR_INTERRUPT_TRANSFER	IN	0x81C8D030	0x822A3940	STATUS_SUCCESS
52	URB	10:59:13:187	BULK_OR_INTERRUPT_TRANSFER	OUT	0x81944DE8	0x8227B5B8	STATUS_SUCCESS
53	URB	10:59:13:187	BULK_OR_INTERRUPT_TRANSFER	OUT	0x81C8D030	0x8227B5B8	STATUS_SUCCESS
54	URB	10:59:13:187	BULK_OR_INTERRUPT_TRANSFER	IN	0x81C8D030	0x8195C2B8	STATUS_PENDING
55	URB	10:59:13:187	BULK_OR_INTERRUPT_TRANSFER	IN	0x81C8D030	0x8227B5B8	STATUS_SUCCESS
56	URB	10:59:13:187	BULK_OR_INTERRUPT_TRANSFER	OUT	0x81944DE8	0x822A3940	STATUS_SUCCESS
57	URB	10:59:13:187	BULK_OR_INTERRUPT_TRANSFER	OUT	0x81C8D030	0x822A3940	STATUS_SUCCESS
58	URB	10:59:13:187	BULK_OR_INTERRUPT_TRANSFER	IN	0x81C8D030	0x8195C2B8	STATUS_PENDING
59	URB	10:59:13:187	BULK_OR_INTERRUPT_TRANSFER	IN	0x81C8D030	0x822A3940	STATUS_SUCCESS
60	URB	10:59:13:187	BULK_OR_INTERRUPT_TRANSFER	OUT	0x81944DE8	0x8227B5B8	STATUS_SUCCESS
61	URB	10:59:13:187	BULK_OR_INTERRUPT_TRANSFER	OUT	0x81C8D030	0x8227B5B8	STATUS_SUCCESS
62	URB	10:59:13:203	BULK_OR_INTERRUPT_TRANSFER	IN	0x81C8D030	0x8195C2B8	STATUS_PENDING
63	URB	10:59:13:203	BULK_OR_INTERRUPT_TRANSFER	IN	0x81C8D030	0x8227B5B8	STATUS_SUCCESS
64	URB	10:59:13:203	BULK_OR_INTERRUPT_TRANSFER	OUT	0x81944DE8	0x822A3940	STATUS_SUCCESS
65	URB	10:59:13:203	BULK_OR_INTERRUPT_TRANSFER	OUT	0x81C8D030	0x822A3940	STATUS_SUCCESS
66	URB	10:59:13:203	BULK_OR_INTERRUPT_TRANSFER	IN	0x81C8D030	0x8195C2B8	STATUS_PENDING
67	URB	10:59:13:203	BULK_OR_INTERRUPT_TRANSFER	IN	0x81C8D030	0x822A3940	STATUS_SUCCESS
68	URB	10:59:13:203	BULK_OR_INTERRUPT_TRANSFER	OUT	0x81944DE8	0x8227B5B8	STATUS_SUCCESS
69	URB	10:59:13:203	BULK_OR_INTERRUPT_TRANSFER	OUT	0x81C8D030	0x8227B5B8	STATUS_SUCCESS
70	URB	10:59:13:203	BULK_OR_INTERRUPT_TRANSFER	IN	0x81C8D030	0x8195C2B8	STATUS_PENDING
71	URB	10:59:13:203	BULK_OR_INTERRUPT_TRANSFER	IN	0x81C8D030	0x8227B5B8	STATUS_SUCCESS
72	URB	10:59:13:203	BULK_OR_INTERRUPT_TRANSFER	OUT	0x81944DE8	0x822A3940	STATUS_SUCCESS

Figura 9. Data transmitida y recibida por el PC.

Fuente: Elaboración Propia.

La línea de operación resaltada en la figura 9, muestra los caracteres enviados por el PC al glucómetro para establecer la conexión, cuya descripción se muestra en la figura 10.

Offset	Hex Data	Ascii
00000000	43	C

Figura 10. Carácter enviado por el PC.

Fuente: Elaboración Propia.

Analizando este tráfico de data, se dedujo que la parte del código ASCII más importante estaba formada por los caracteres que se muestran en la tabla 5.

ASCII	HEXADECIMAL	DECIMAL
DC1 (device control 1)	11	17
Enter	D	13
D	44	68
M	4D	77

P	50	80
P	50	80

Tabla 6. Algunos caracteres enviados por el PC.

Fuente: Elaboración Propia.

Estos caracteres son la petición de transmisión de data desde el PC hacia el glucómetro. La respuesta que generan es la fecha, la hora, el día, el valor de medición de glicemia, el modelo de glucómetro y el código que identifica a cada medición. Estos datos de respuesta por el glucómetro vienen dados mediante el siguiente formato:

```

¥P 005,"TWC8B7CRT","MG/DL " 05E0
P "TUE","07/04/06","10:55:32 "," 084 ", 00 0835
P "TUE","07/04/06","18:51:56 "," 069 ", 00 083A
P "SAT","03/18/06","05:45:40 "," 081 ", 00 0826
P "MON","03/13/06","09:24:16 "," 098 ", 00 0828
P "MON","03/13/06","10:16:08 "," 116 ", 00 0821
¥
    
```

Obtenido el patrón, se puede apreciar en Azul, el modelo del glucómetro y las unidades (mg/dl) en que vienen dadas las mediciones. En Verde el día, fecha y hora de cada medición. En Rojo la cantidad de glicemia en la sangre y en Naranja el código de cada evento de análisis de glicemia.

III.2.2 Transmisión PC- Paciente al Servidor - Médico

Una vez obtenida la data del glucómetro en crudo, se procedió a transmitirla al medico vía correo electrónico. Para ello se creó un correo electrónico en un servidor POP 3, que garantizara acceso remoto, no restringido, por parte del programa que se está empleando. Existen muchos servidores POP que sólo permiten ingresar a

usuarios que entren directamente desde la página web correspondiente. Ejemplos de ello encontramos en servidores como *Gmail* y *Yahoo*.

El servidor seleccionado fue CANTV.NET, el nombre de usuario utilizado fue “proyectodiabetes” y la contraseña “12345678”. Estos datos son utilizados por el programa desarrollado en Visual Basic, para ingresar al correo y depositar la información correspondiente cada vez que el usuario cargue la información de las mediciones en el computador.

En las figuras 11 y 12 se observa un ejemplo de data enviada desde el programa del paciente hacia el correo “proyectodiabetes@cantv.net”. Donde se puede apreciar la dirección del emisor y el título del correo, este último es el mismo para todos los pacientes, con el fin de filtrar otros posibles correos no relacionados con el proyecto.

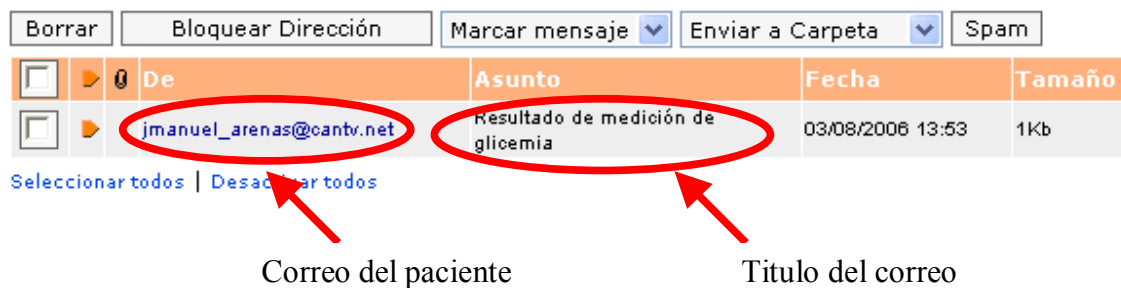


Figura 11. Bandeja de Entrada del correo.
Fuente: Elaboración Propia.

Responder Responder a todos Reenviar Borrar Traducir Enviar a Carpeta ▾

[Versión imprimible](#) | [Encabezado](#) < Anterior | Siguiente > | [Volver a Mensajes](#)

De:	jmanuel_arenas [jmanuel_arenas@cantv.net]	Bloquear Remitente	Guardar Contacto
Para:	proyectodiabetes@cantv.net		Guardar Contacto
Cc:			
Asunto:	Resultado de medición de glicemia		
Fecha:	03/08/2006 13:53:06		

Juan Manuel Arenas
17123935
WP 005,"TWC8B7CRT","MG/DL " 05E0
P "TUE","07/04/06","18:55:32 "," 084 ", 00 0835
P "TUE","07/04/06","18:51:56 "," 069 ", 00 083A
P "SAT","03/18/06","13:45:40 "," 081 ", 00 0826
P "MON","03/13/06","10:24:16 "," 098 ", 00 0828
P "MON","03/13/06","10:16:08 "," 116 ", 00 0821

Figura 12. Correo Recibido por el Médico.
Fuente: Elaboración Propia.

Una vez ubicada la data en el correo electrónico, es tomada por el programa del médico a través de algoritmo mostrado en la figura 13.

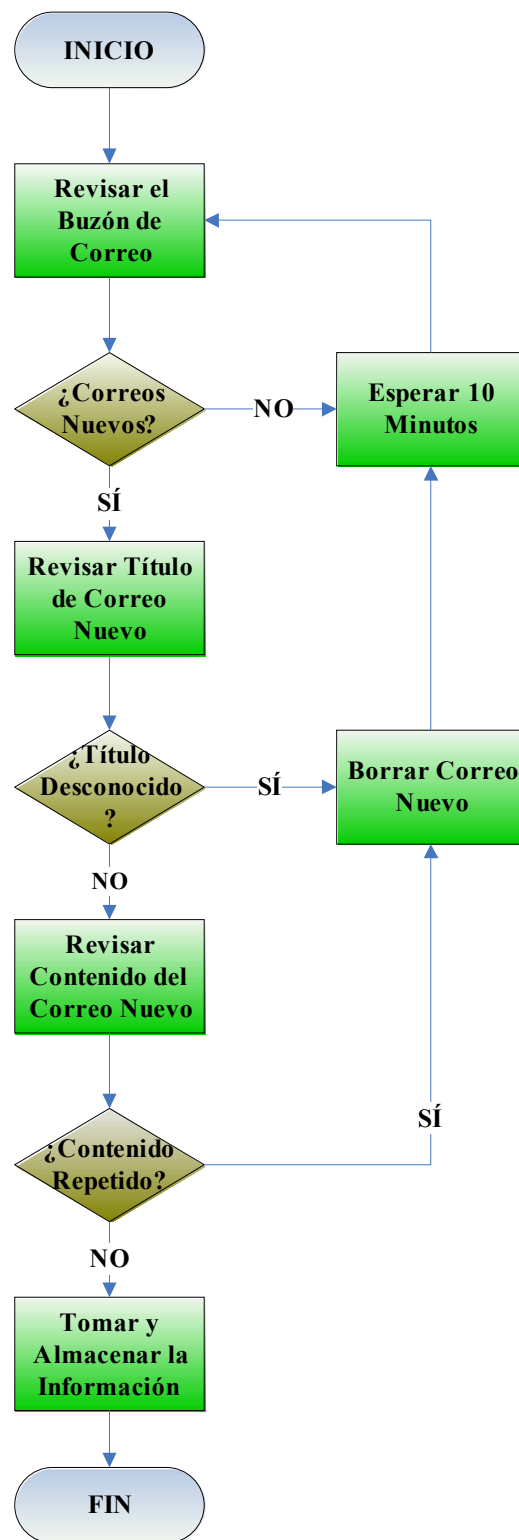


Figura 13. Algoritmo empleado por el programa Médico
Fuente: Elaboración Propia.

III.2.3 Interfase de Gestión del Médico.

El programa del médico fue desarrollado en cuatro etapas principales (Ver Figura 14):

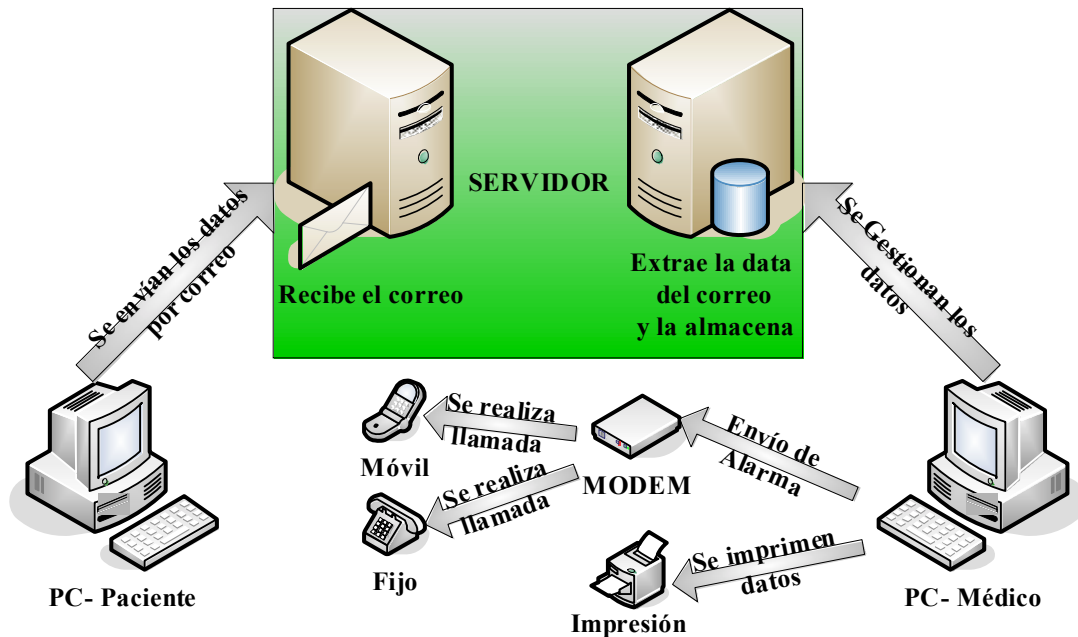


Figura 14. Etapas Principales de la Interfase del Médico.
Fuente: Elaboración Propia.

Una vez que el paciente envía los datos por correo, y una vez que el correo sea recibido por el servidor del médico, los módulos principales que forman parte de la Interfase del médico según la figura se encuentran:

- Obtención de la data del correo electrónico
- Almacenamiento de la data en la base de datos
- Visualización de forma gráfica y escrita de todos y cada uno de los datos de los pacientes registrados en el sistema. Brindando la opción de impresión en papel.
- Detección de alarmas de Hipo e Hiperglicemia indicando día, fecha, hora y valor de glicemia de la medición correspondiente.

III.2.3.1 Obtención de la data del correo electrónico

El programa del médico fué diseñado con el fin de realizar actualizaciones de información provenientes de los pacientes periódicamente. De esta forma es posible garantizar que el tiempo transcurrido entre la transmisión de información desde el paciente hacia el médico sea de máximo diez minutos, este tiempo fué establecido así en el momento en el que se desarrolló el código empleado.

El protocolo empleado para realizar la búsqueda de información en el servidor POP 3 CANTV es el STMP. Aquí el programa introduce el nombre de usuario “proyectodiabetes@cantv.net” y la contraseña “12345678” para poder ingresar a la data. Una vez logrado esto, revisa si hay correos nuevos, en caso de haber alguno, analiza entonces el título y el contenido de dicho correo. Si este es nuevo y es reconocido como perteneciente al proyecto entonces es almacenado en la base de datos.

Una vez terminado este proceso, el programa borra el correo leído y cierra la sesión “proyectodiabetes@cantv.net”.

III.2.3.2 Almacenamiento de la data en la base de datos

El proceso de almacenaje en la base de datos fue implementado bajo Microsoft Access y SQL, donde se toma la información proveniente del correo electrónico y se almacena en caso de no estar repetida, de lo contrario es desechada.

Para este fin fue creada una base de datos denominada “proyectodiabetes.mdb” en formato Windows 2000 para garantizar compatibilidad con Visual Basic 6, la cual fue ubicada en la carpeta C:/proyectodiabetes para su fácil acceso.

III.2.3.3 Visualización de los datos de los pacientes.

El programa del médico fue diseñado con las opciones de visualizar la data recién llegada escrita y gráficamente, observar la historia de los pacientes registrados en el

sistema escrita y gráficamente, registrar un nuevo paciente en el sistema y señalar las posibles alarmas que puedan ocurrir o hayan ocurrido en algún momento de la historia del paciente. Todo bajo la interfaz gráfica de Visual Basic 6.

Este programa fué desarrollado con la idea de mantenerse en ejecución las 24 horas del día ya que de esta forma puede recibir data y trabajarla en cualquier momento sin necesidad de la presencia del médico a toda hora,

La visualización fué formulada tanto en el PC como a nivel de impresión, esto quiere decir que brinda la posibilidad de observar y analizar los resultados tanto en la computadora como en una hoja impresa, que luego puede ser otorgada al paciente para que posea constancia de su estado de salud.

III.2.3.4 Detección de alarmas de Hipo e Hiperglicemia

La detección de alarmas fué uno de los puntos más importantes en el diseño del programa del médico, ya que representa una de las principales funciones para la cual fué desarrollado.

Consta de indicadores de alarma, visualización gráfica y escrita de las diferentes alarmas ocurridas y un módulo extra implementado posteriormente que incluye llamadas en tiempo real vía Modem, al celular del médico en caso de presentarse alguna. Este módulo efectúa de tres a cinco repiques al teléfono celular médico, quien deberá tener identificado al número llamante como “Alarma STRD”. Este número llamante se encuentra en el consultorio, es por eso que siempre será el mismo.

Esta llamada es efectuada sólo cuando se esta en presencia de alguna alarma proveniente de la data recién llegada del correo electrónico.

Para poder identificar cuando un valor de glicemia esta fuera de los rangos normales, el programa del médico solicita los límites mínimo y máximo de glicemia establecidos para cada paciente, a la hora de ser registrado en el sistema. De esta

manera se tienen como datos fijos y de referencia almacenados en la base de datos en Access.

III.3 Fase III. Pruebas.

La idea principal de esta fase de experimentación, es la de que, a medida que se realicen pruebas sobre el prototipo diseñado se puedan corregir los errores que se obtengan, mejorando los módulos realizados, plantear nuevos métodos para darle más valor agregado al prototipo.

Mediante reuniones continuas con los médicos del Centro Asistencial Santa Inés, así también con el Tutor, se realizaron demostraciones del prototipo y consultas con el fin de obtener sugerencias y mejoras en los módulos desarrollados cumpliendo con los objetivos propuestos.

Capítulo IV

Análisis de Resultados

En este capítulo se exponen los resultados, describiendo cada uno de ellos, de acuerdo al prototipo diseñado, respondiendo a la metodología y objetivos planteados.

Los resultados obtenidos convergen en dos aplicaciones desarrolladas.

- El “programa del paciente”.
- El “programa del médico”.

IV.1. Programa Paciente.

Al ejecutar el “programa del paciente”, éste envía un mensaje como el que se muestra en la figura 15.

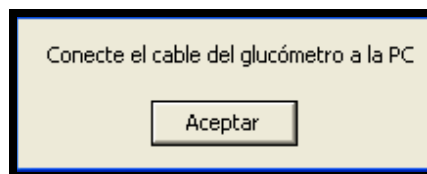


Figura 15. Mensaje arrojado por el programa.
Fuente: Elaboración Propia.

De esta manera se asegura una transferencia de información exitosa entre el glucómetro y el PC. En caso de no conectarse el dispositivo, el programa cierra automáticamente y envía un mensaje de “vuelva a ejecutar el programa” como se muestra en la siguiente figura.

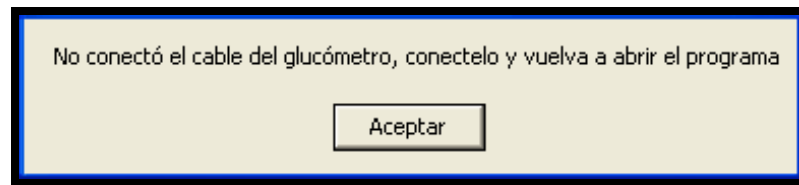


Figura 16. Mensaje arrojado por el programa sin conectar el Glucómetro.
Fuente. Elaboración Propia.

Una vez conectado el glucómetro al PC, vía USB se desprende la ventana principal del programa del paciente, el cual se muestra en siguiente figura.

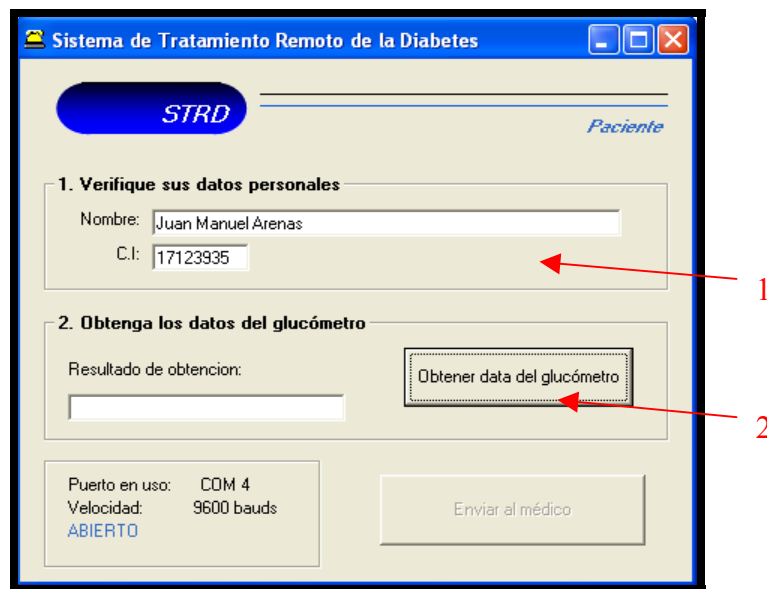


Figura 17 Ventana Programa Paciente.
Fuente: Elaboración Propia.

Aquí se dispone de una interfaz de fácil uso, la cual requiere de sólo tres pasos para ejecutar la transmisión de información y se pueden detallar en la siguiente Figura.

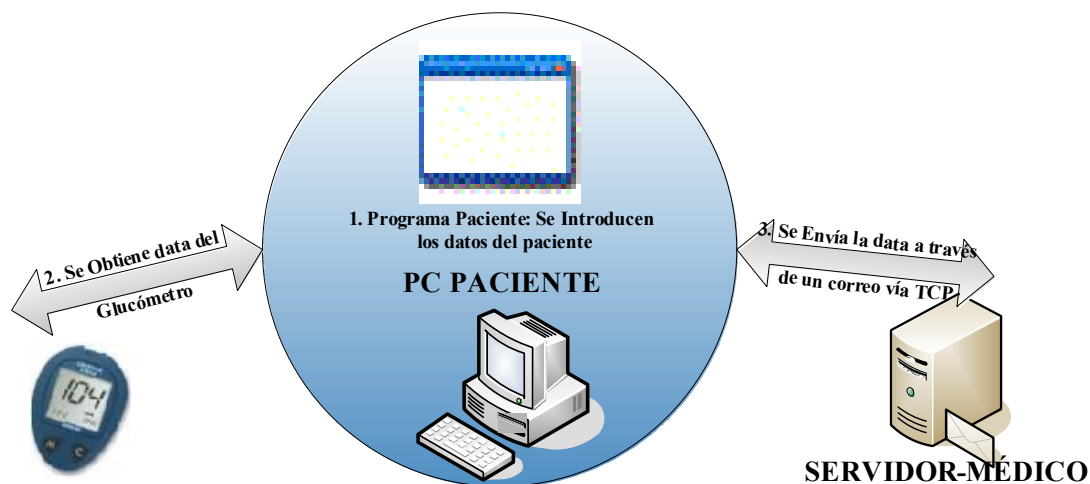


Figura 18. Pasos Para la ejecución del Programa Paciente.

Fuente: Elaboración Propia.

Primero, el usuario debe introducir su nombre completo y su cédula, esto es primordial ya que es la manera de cómo el programa del médico identifica y posteriormente almacena los datos del paciente en cuestión en la base de datos.

Segundo, se procede a obtener la data del glucómetro presionando el botón “Obtener data del glucómetro”. Este procedimiento tomará de dos a cinco segundos en efectuarse.

El código en ASCII que requiere el glucómetro *ONE TOUCH ULTRA* para enviar la data al computador personal esta formado por los caracteres:

ASCII	HEXADECIMAL	DECIMAL
DC1 (<i>device control 1</i>)	11	17
Enter	D	13
D	44	68
M	4D	77
P	50	80
P	50	80

Tabla 7. Caracteres ASCII enviados por el PC.

Fuente: Elaboración Propia.

Estos caracteres deben ser enviados desde el PC hacia el glucómetro secuencialmente. La respuesta a este código viene dada de la siguiente forma:

```
¥P 005,"TWC8B7CRT","MG/DL " 05E0 P "TUE","07/04/06","10:55:32 "," 084 ",  
00 0835 P "TUE","07/04/06","18:51:56 "," 069 ", 00 083A P  
"SAT","03/18/06","05:45:40 "," 081 ", 00 0826 P "MON","03/13/06","09:24:16  
"," 098 ", 00 0828 P "MON","03/13/06","10:16:08 "," 116 ", 00 0821 ¥
```

Donde se puede apreciar en Azul, el modelo del glucómetro y la unidades (mg/dl) en que vienen dadas las mediciones. En Verde el día, fecha y hora de cada medición. En Rojo la cantidad de glicemia en la sangre y en Naranja el código de cada evento de análisis de glicemia. El carácter P indica separación entre las mediciones y el carácter indica inicio y fin de la transmisión

Tercero, una vez obtenida la información del glucómetro, es transmitida vía correo electrónico al médico presionando el botón “Enviar al médico” como lo indica la siguiente figura.

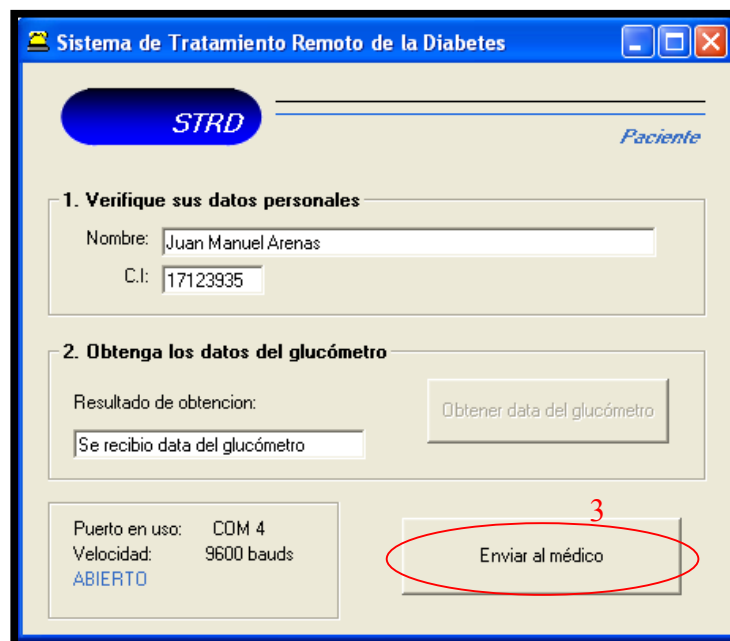


Figura 19.Paso 3 del programa Paciente.
Fuente: Elaboración Propia.

El proceso de establecimiento del puerto COM4 para la transferencia de data desde el glucómetro hacia el PC se efectúa automáticamente al ejecutar el programa, así como también la determinación de la velocidad de transmisión en baudios, los bits de paridad, los bits de data y bits de stop.

Una vez enviada la data al médico es almacenada temporalmente en el correo electrónico tal y como se muestra en la siguiente figura.

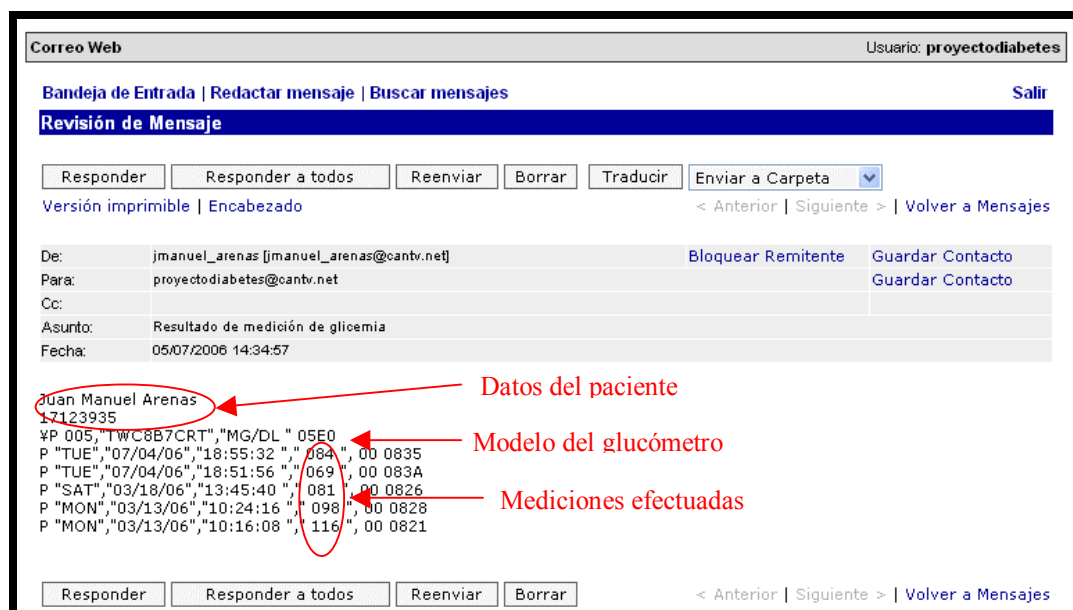


Figura 20. Correo Electrónico Recibido por el Servidor.

Fuente: Elaboración Propia.

Una vez almacenada la información del paciente en el correo electrónico, el “programa del paciente” culmina su ejecución y es aquí donde comienza a tomar parte el “programa del medico” en la recepción de información.

IV.2. Programa Médico

Este programa se encarga de tomar la data desde el correo sin modificaciones y almacenarla en la base de datos en Access a través de comandos SQL, siempre y cuando la data no este repetida y pertenezca al proyecto STRD “Sistema de

Tratamiento Remoto de la Diabetes”, en caso contrario el programa la borra del buzón de correo y la omite.

Al ejecutar el “programa del médico” se dispone de una ventana principal como la mostrada en la figura.

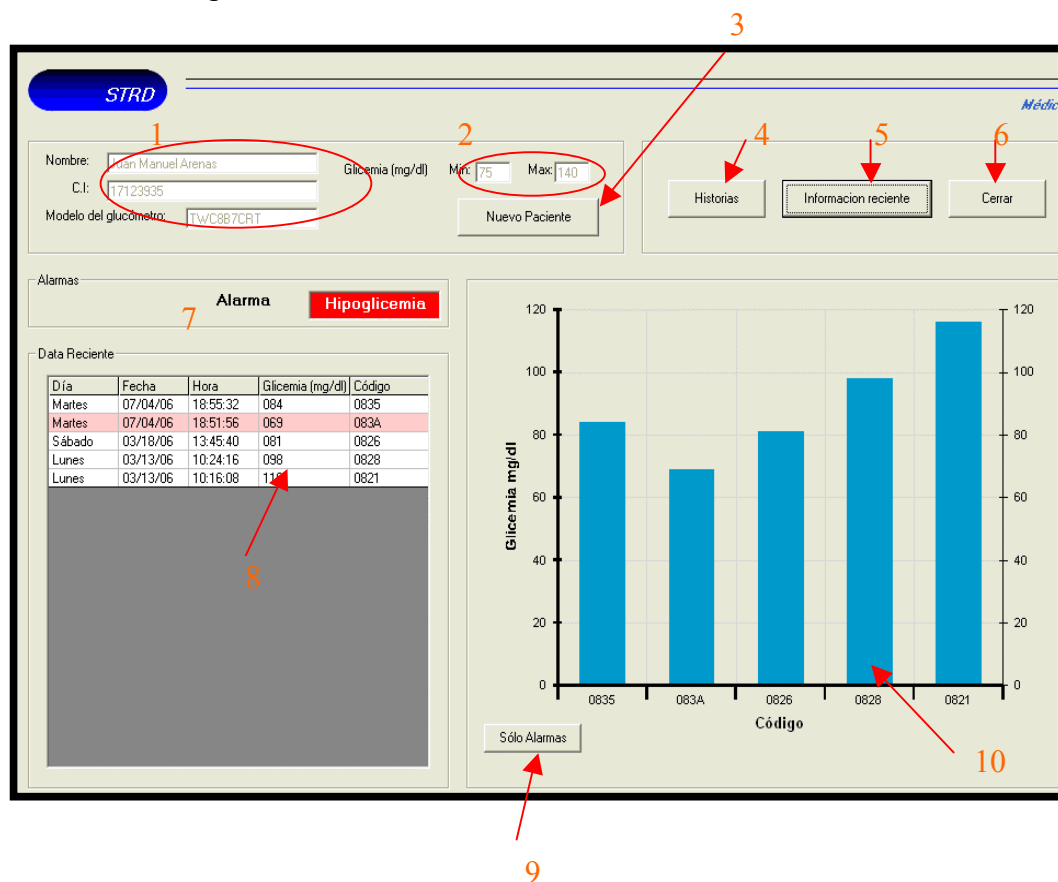


Figura 21. Ventana Principal Programa Médico.

Fuente: Elaboración Propia.

Dentro de las opciones que posee esta interfaz son las siguientes:

1. Información del paciente: que incluye nombre completo, número de cédula y tipo de glucómetro que emplea el paciente.
2. Valores Límite de glicemia del paciente en cuestión: muestra los mg/gl de glicemia en la sangre que se toman como referencia para determinar una hiper o hipoglucemia en caso de existir.

3. Botón nuevo paciente: Permite al médico registrar un nuevo usuario en el sistema. Si no se realiza este proceso, el paciente no dispondrá del servicio del sistema ya que sus datos no serán tomados en cuenta al ser recibidos vía correo electrónico.
4. Botón Historias: Muestra las historias de todos y cada uno de los pacientes registrados así como también los casos de alarma que hayan podido presentarse.
5. Botón Información reciente: Actualiza la información mostrada en la pantalla principal de forma manual, en caso de que el medico no desee esperar el tiempo que tarda el programa en refrescarla automáticamente cada diez minutos.
6. Botón cerrar: Cierra y guarda los cambios realizados en la aplicación
7. Indicador de alarmas: Se enciende en caso de existir alguna Hiper o hipoglucemia.
8. Tabla informativa: Posee la información del día, fecha, hora, valor de glicemia y código de cada una de las mediciones realizadas por el paciente en cuestión. Señalando en fondo rosado los casos de alarma tal y como se muestra en la figura anterior.
9. Botón sólo alarmas: Muestra gráficamente sólo las alarmas que hayan podido presentarse, (ver figura 22). En caso de no haber alarmas, dicho botón queda deshabilitado.
10. Gráfica: muestra en gráfica de barras cada una de las mediciones encontradas en la tabla.

Cabe destacar que cada uno de estos elementos desplegados en la ventana principal, se actualizan a medida que llega data al servidor.

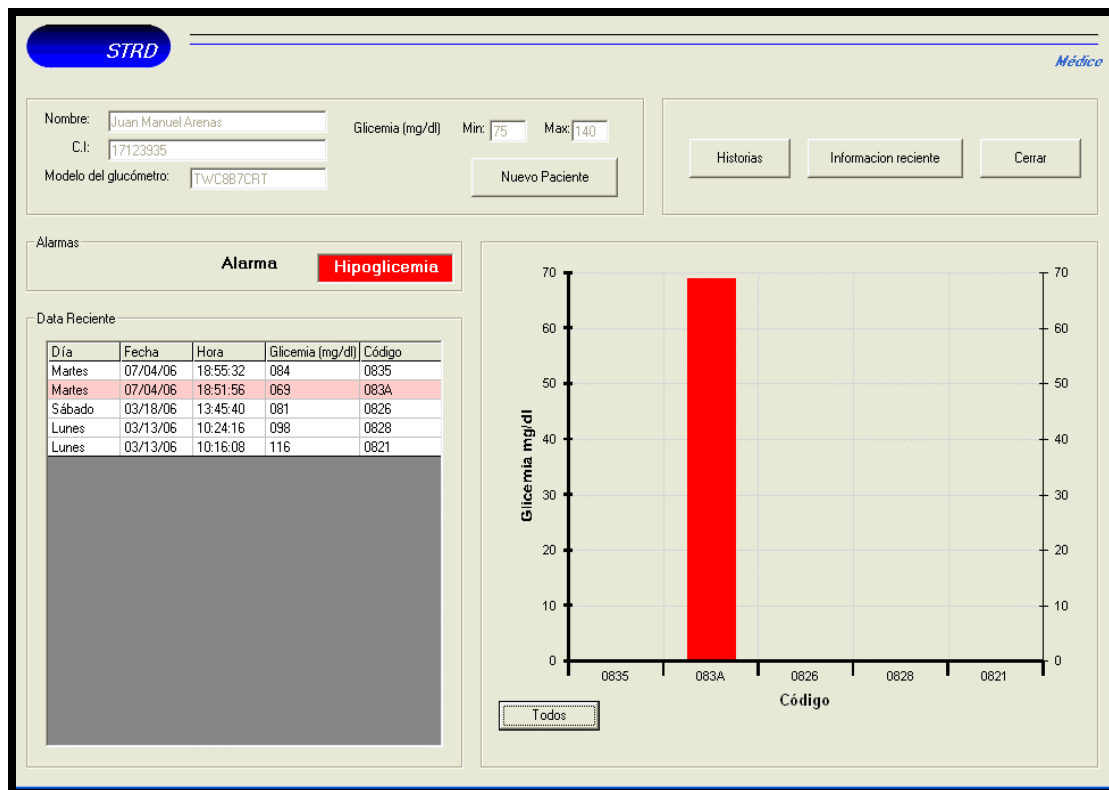


Figura 22. Muestra de alarma.
Fuente: Elaboración Propia.

Al presionar el botón “Nuevo paciente” se abre una nueva ventana en la cual se introducen datos del paciente a registrar como nombre completo, cédula de identidad, teléfono, dirección, ciudad y valores límite de glicemia máximo y mínimo. Por otro lado se seleccionan la fecha de nacimiento y el tipo de diabetes (tipo 1 o tipo 2) del mismo. (Ver figura 23)

Nuevo Paciente:

STRD Médico

Nombre:

CI:

Teléfono:

Dirección:

Ciudad:

Fecha de nacimiento: Día / Mes / Año

Tipo de diabetes: Seleccione

Valores críticos de glicemia: (mg/dl)

Min: Máx:

Figura 23. Opción Nuevo Paciente.
Fuente: Elaboración Propia.

Una vez terminada la introducción de datos, se habilita el botón “Almacenar”, el cual sólo guardara la información suministrada si ésta está completa. La función de éste botón es crear un nuevo perfil en Access, llenando cada columna de la tabla de la base de datos con un dato diferente, esto se muestra en la figura 24.

Nombre	CI	Teléfono	Dirección	Ciudad	Fecha_de_naci	Tipo_de_diabeti
Juan Manuel Ar	17123935	2433735	Calle13, Edif St	Caracas	2/4/1984	Tipo 2
Juan Manuel Ar	17123935	2433735	Calle13, Edif St	Caracas	2/4/1984	Tipo 2
Juan Manuel Ar	17123935	2433735	Calle13, Edif St	Caracas	2/4/1984	Tipo 2
Juan Manuel Ar	17123935	2433735	Calle13, Edif St	Caracas	2/4/1984	Tipo 2
Juan Manuel Ar	17123935	2433735	Calle13, Edif St	Caracas	2/4/1984	Tipo 2
Juan Manuel Ar	17123935	2433735	Calle13, Edif St	Caracas	2/4/1984	Tipo 2

Figura 24. Perfil en Access.
Fuente: Elaboración Propia

Valor_min_glice	Valor_max_glic	Dia_de_medicion	fecha_de_medicion	hora_de_medicion	glicemia	codigo
80	90				0	
80	90	Martes	07/04/2006	09:55:32	84	0835
80	90	Martes	07/04/2006	18:51:56	69	083A
80	90	Sábado	18/03/2006	05:45:40	81	0826
80	90	Lunes	13/03/2006	10:24:16	98	0828
80	90	Lunes	13/03/2006	10:16:08	116	0821

Figura 25. Mediciones del Paciente.

Fuente: Elaboración Propia.

Haciendo “clic” en el botón “Historias” en la ventana principal, aparece una nueva ventana mostrando todos los datos personales y las mediciones de los pacientes previamente registrados en el sistema tal y como se observa en la figura 25.

Esta parte del proyecto realiza una consulta en la base de datos “*query*”, en donde solicita la información del paciente a partir de su cédula, ya que es posible que existan varios con el mismo nombre.

Figura 26. Historia del Paciente.

Fuente: Elaboración Propia.

Esta consulta se ejecuta al seleccionar el paciente haciendo “clic” en la pestaña mostrada en la figura anterior.

Una vez ubicados los datos en Access, se cargan en la pantalla. Esta información contiene todas y cada una de las mediciones del paciente, con sus respectivas alarmas en caso de que las haya, desde el momento en que fue registrado así como también los datos personales del mismo.

Esta ventana además de mostrar alarmas de hipo e hiperglicemia, resalta aquellas mediciones que han sido efectuadas en ayuno. De esta manera se facilita el análisis del resultado por parte del médico.

El botón “Imprimir” permite al médico otorgar un informe escrito de la historia del paciente. Esta opción no requiere de configuraciones adicionales ya que con sólo tener una impresora asignada al PC, es suficiente.

Al presionar el botón “Ver gráfica” el programa del médico muestra una pantalla como la expuesta en la figura 27.

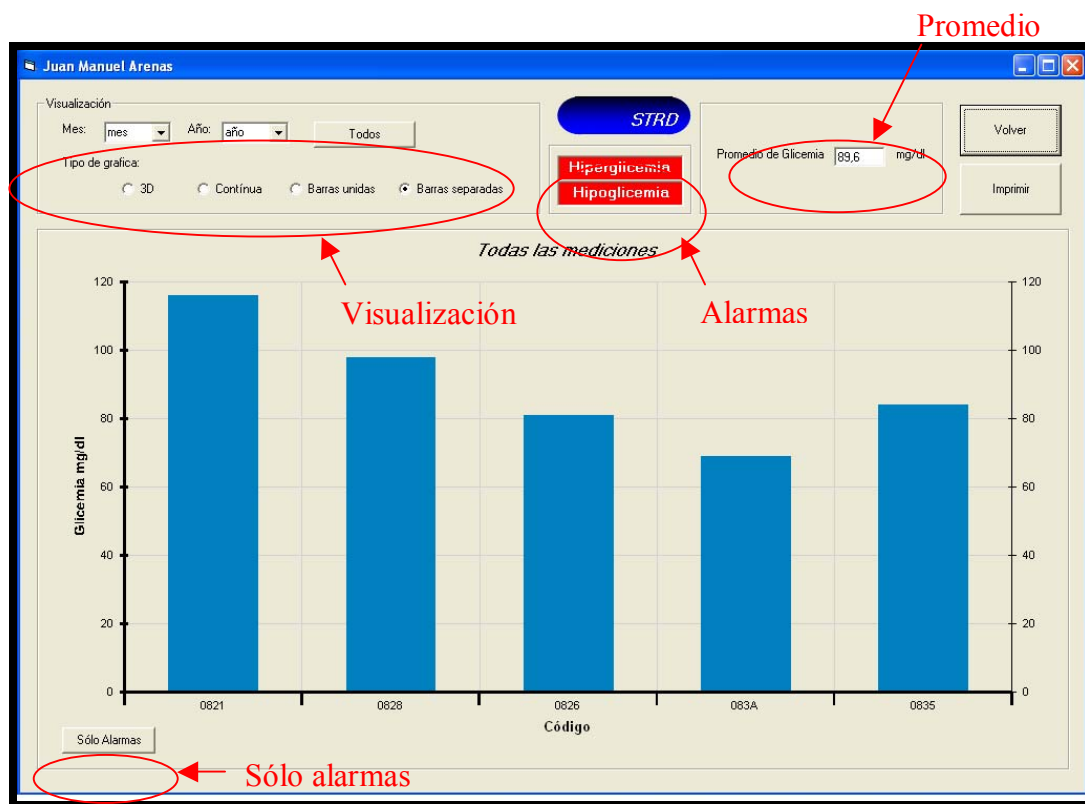
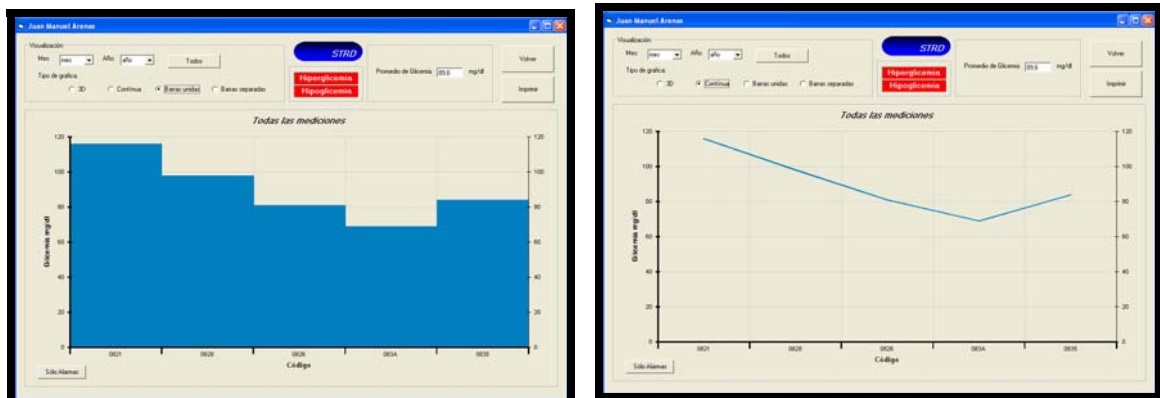


Figura 27. Opción Ver Gráfica.
Fuente: Elaboración Propia.

Donde se visualiza gráficamente los valores existentes en la tabla de la ventana “Historia del paciente”.

Al igual que en la ventana principal, aquí se dispone también del botón “Solo alarmas”, que resalta únicamente las mediciones que se encuentren fuera de los rangos normales de glicemia.

En la parte superior izquierda se encuentran las opciones de visualización, en donde se dispone de observación por mes y año de los datos de la historia del paciente o bien de todos. También existe la posibilidad de variar la forma de la gráfica para una mejor apreciación de valores, ver figura 28.



(a)

(b)

Figura 28. Opciones de Visualización.

(a) Barras Unidas, (b) Continúa.

Fuente: Elaboración Propia.

En la parte superior central se aprecian los indicadores de alarma, a su derecha el promedio en mg/dl de todos los valores mostrados en la gráfica. En la parte superior derecha se encuentra el botón “Imprimir” el cual además de imprimir los datos del paciente y todas sus mediciones de manera escrita, refleja en una segunda hoja la gráfica correspondiente como se muestra en la figura 29.

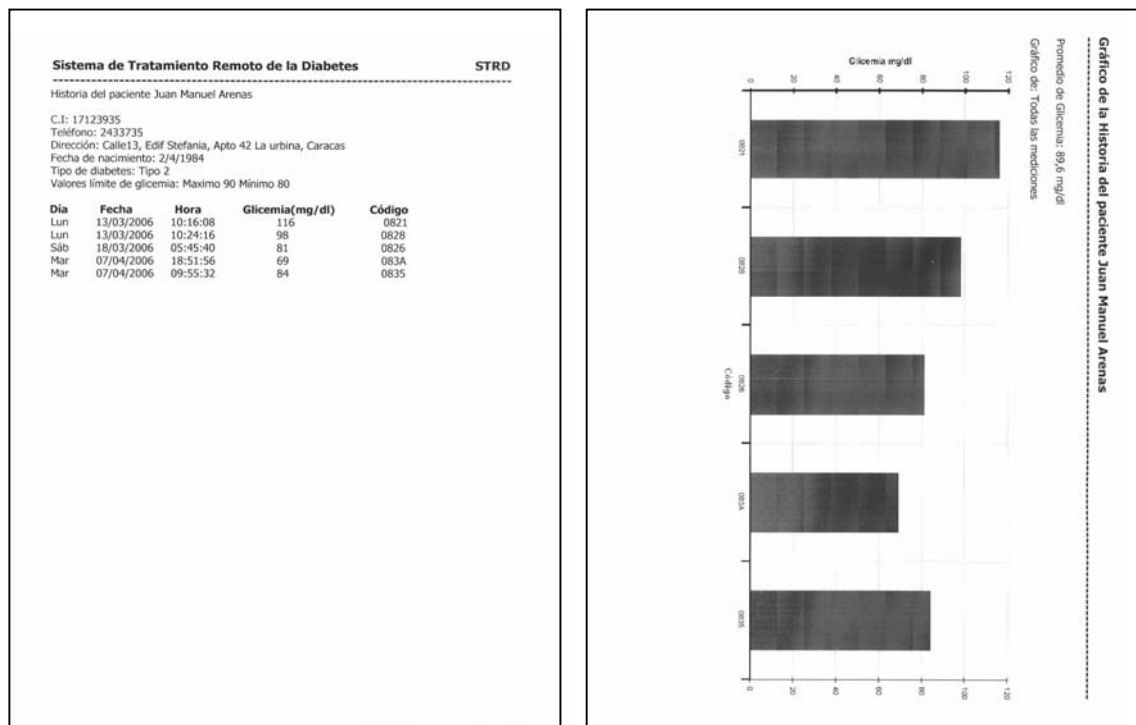


Figura 29. Impresión de las mediciones.
Fuente: Elaboración Propia.

Para un correcto funcionamiento tanto del programa paciente como del médico, se deben seguir una serie de pasos para el éxito del sistema, además el PC debe tener ciertos requerimientos para un mejor desenvolvimiento. Por consiguiente para una mejor comprensión en cuanto al uso del programa médico y paciente se puede observar el manual de usuario para ambos programas en el Apéndice A.

Una vez finalizado el prototipo, se presentó el mismo ante los médicos del Centro de salud Santa Inés con el fin de realizar pruebas con pacientes diabéticos, que frecuentan el centro de salud.

Capítulo V

Conclusiones y Recomendaciones

La diabetes es una enfermedad de carácter metabólico, que ha penetrado dentro de la sociedad venezolana progresivamente, tornándose un problema de salud pública, llegando a formar parte de las principales causas de muerte en el país, además de acarrear otras enfermedades de alto riesgo. Llevar un control de manera eficiente resulta así importante para la calidad de vida del paciente. El área interdisciplinaria de telemedicina contribuye significativamente con este control.

El sistema desarrollado permite llevar un control de los niveles de glucosa del paciente en tiempo real de manera automatizada, es decir una vez que las mediciones de glicemia son realizadas por medio del glucómetro, el paciente envía los datos hacia el servidor ubicado en el centro hospitalario a través de Internet, con esta información el médico puede hacer un seguimiento periódico del paciente y ajustar el tratamiento según los resultados.

El diseño del prototipo se llevó a cabo por medio de tres fases: teórica, técnica y de experimentación, debidamente planteadas, las cuales determinaron de manera satisfactoria la culminación del sistema cumpliendo con los objetivos propuestos.

En la fase teórica mediante investigaciones previas al diseño del prototipo se determinaron indicadores imprescindibles para la elaboración del sistema como son los parámetros resaltantes relacionados con la enfermedad, así como el funcionamiento del glucómetro y protocolos a emplear (USB, TCP/IP, STMP, HTTP). Hay que destacar que gracias al apoyo profesional de los especialistas del Centro de Salud Santa Inés (Doctora Luz Silva y Gerente General Bernardo Guinand) se estableció el tipo de paciente al cual estaría enfocado el prototipo.

Los equipos y herramientas que forman parte del prototipo se escogieron de acuerdo a los requerimientos del sistema y necesidades tanto del paciente como del médico considerando que dichos dispositivos deben ser de fácil acceso, prácticos, de costos razonables además de estar dentro del mercado venezolano proporcionando facilidad y comodidad.

Dentro de los aspectos que se contemplaron en la fase técnica, para la compatibilidad glucómetro- PC, se estableció que tanto el glucómetro como el PC, debían tener puerto USB, ya que es una interfaz que transmite datos de forma sencilla, basada en comunicación serial por puerto serial, la cual es capaz de proporcionar altas tasas de transferencia en bits por segundo. Para la efectiva transmisión de datos entre ambos dispositivos a través del protocolo USB, es necesaria la presencia de un dispositivo “host” (PC), el cual tiene la habilidad de procesar y gestionar los cambios de configuración que puedan ocurrir durante la transmisión.

Para poder manipular los datos provenientes del glucómetro, se descifró el patrón con código ASCII enviado por el glucómetro al PC mediante el programa *COM PORT SNIFFER* permitiendo visualizar la data que transitaba por el puerto serial y de esta manera se procedió a la elaboración de una interfaz desarrollada en *MICROSOFT VISUAL BASIC*, que procesó la data en crudo para su correcto envío hacia el servidor ubicado en el centro hospitalario.

El lenguaje de programación empleado en las aplicaciones: programa paciente, programa médico, fué *MICROSOFT VISUAL BASIC*; herramienta caracterizada por poseer gran cantidad de librerías de código y módulos para el desarrollo de aplicaciones en Microsoft Windows, brindando un sistema completo y de interfaz gráfica de fácil comprensión. Adicionalmente por medio de este lenguaje y mediante una librería específica, se pudo acceder remotamente, sin restricción, a datos en un correo electrónico de un servidor POP3, donde se almacenan los datos enviados por la aplicación paciente y así la aplicación médico, ubicado en el centro hospitalario,

tiene también acceso a dichos datos, para que luego sean procesados y almacenados por el programa médico.

Por su parte la aplicación desarrollada para el médico, es capaz de realizar actualizaciones de datos provenientes de los pacientes cada diez minutos, además permite registrar, detectar alarmas de hipo e hiperglicemia, visualizar de manera gráfica las mediciones de glicemia que lleva el paciente, brindando opciones de impresión de todos los datos almacenados.

El programa se desarrolló con la idea de llevar un seguimiento diario de los niveles de glicemia las 24 horas del día, a fin de reducir el nivel de consultas médicas y dar mayor flexibilidad tanto al médico como al paciente.

Una fase muy importante fué la de experimentación, debido a que en esta se realizaron pruebas continuas al prototipo, con el fin de obtener mejoras y de corregir errores en caso de fallas. Mediante demostraciones ante el tutor y los médicos especialistas del Centro de Salud Santa Inés (Doctora Luz Silva, Gerente General Bernardo Guinand y Gerente Médico Carlos Paradisi) se obtuvieron sugerencias para darle más utilidad y valor agregado al prototipo como la de incluir en el programa médico la opción imprimir. Para una futura implementación del prototipo en dicho centro de salud, para realizar pruebas directas con pacientes, se requiere de una configuración previa de los equipos “PC” donde se va a instalar la herramienta, con el propósito de garantizar un acceso no restringido al correo electrónico “POP3 utilizado en el proyecto” por parte de los servidores que controlan la red.

Como recomendaciones que surgieron en el desarrollo del sistema, están:

- a) Creación de módulos adicionales al prototipo inicial con el fin de brindar nuevas prestaciones tanto al paciente como el médico.
- b) El establecimiento de una conexión inalámbrica entre el glucómetro y el Centro de Salud a través de un sistema móvil celular con ayuda de la interfaz

infrarroja de los equipos para así lograr una mayor penetración en zonas de difícil acceso. Este módulo no fue implementado porque los glucómetros con puerto infrarrojo en el mercado venezolano no incluyen el software necesario para descifrar, mediante un sniffer, la información enviada por el glucómetro y así desarrollar la aplicación paciente-médico. En nuestro país estos glucómetros sólo pueden descargar la información almacenada directamente hacia una impresora, mediante un equipo que sólo se le suministra a los distribuidores de los glucómetros y no al público en general.

- c) La migración a un sistema de software libre, ya que este prototipo inicial es solo soportado por el S.O Windows.

Referencias Bibliográficas

Accu-chek (2006). *Testing blood Sugar* [Artículo en línea], disponible: www.accu-chek.com/us/rewrite/content/en-us/4.1:10/article/ACCM_general_article-2410.htm. [Consulta: 2006, Febrero 15]

Accu-chek (2005). *Productos Accu-chek* [Información en línea], disponible: http://www.accu-chek.com.pe/04_product/AccuChekGo.htm [Consulta: 2005, Noviembre15]

American Diabetes Association (2001). *Como medir la glucosa en la sangre* [Artículo en línea], disponible www.diabetes.org [Consulta 2006, Febrero10]

Arenas, J. & Rendón, G. (2006). *Manual de Usuario Programa Paciente, Programa Médico* [Elaboración Propia].

Brodín, E. & Giménez, A. (2006). *El bus USB* [Documentación en línea], disponible: <http://usuarios.lycos.es/kurganz/introduccion.html> [Consulta 2006, Abril 4].

DiabetesGold (2005). *GlucoLeader* [Información en línea], disponible: http://www.diabetesgold.com/sp_glucometro.htm. [Consulta: 2005, Noviembre 20]

Enciclopedia Wikipedia (2006). *Protocolo de Internet* [Información en Línea], disponible: http://es.wikipedia.org/wiki/Protocolo_IP. [Consulta: 2006, Julio 10]

Federación Diabetológica Colombiana (2006). *Diabetes Control y Prevención* [Consulta en Línea], disponible en <http://www.fdc.org.co/> [Consulta 2005, Noviembre 5]

Forouzan, B (2002). *Transmisión de datos y redes de comunicaciones* (2da ed.). Mac Graw Hill. Madrid. Páginas: 41-54.

Grupo Diabetes SAMFyC (1996). *Recomendaciones para el autoanálisis de Glucemia* [Consulta en Línea], disponible en <http://www.cica.es/aliens/samfyc/> [Consulta 2005, Noviembre 15]

Lifescan (2002). *Guía rápida de consulta de OneTouchUltra*. Milpitas, CA USA.

MediKatálogo (2005). *Glucómetros (Equipos de diagnóstico)* [Información en línea], disponible <http://www.medikatalogo.com.mx/boletines/026/index.html> [Consulta: 2005, Noviembre 20]

Mediks (2006). *La diabetes* [Artículo en línea], disponible en <http://mediks.com/saludyvida/articulo/> [Consulta 2006, Febrero 15]

Microsoft Visual Studio (2006). *Wikipedia* [Información en Línea], disponible: http://es.wikipedia.org/wiki/Microsoft_Visual_Studio. [Consulta: 2006, Agosto 5].

OneTouch ULTRA (2006). *Productos Sanborns* [Consulta en línea], disponible en <http://www.sanborns.com.mx/sanborns/> [Consulta 2006, Abril 7]

Tanenbaum, A (2003). *Redes de Computadoras* (4ta ed.). Prentice Hall. México. Páginas: 602- 630

ANEXO A

Gama de Glucómetros

Dentro de la gama de glucómetros existentes en el mercado venezolano se encuentran:

GlucóLeader

Características:

- Requiere de una muestra muy pequeña de sangre; apenas 1.5µl
- Obtiene resultados exactos en menos de 15 segundos
- Función de recuperación de memoria
- 1.7.14.28 días promedio de despliegue
- Capacidad para descargar información de su computadora personal (descarga gratis de software para el manejo de información de la diabetes)
- Despliegue en grande
- Símbolos de mensaje de error



Figura 1. Glucómetro Glucóleader

Fuente: Diabetes Gold, 2005, http://www.diabetesgold.com/sp_glucometro.htm.

Accu-Chek Sensor (R) Glucómetro con manual.

Características

- Medidor más pequeño con laterales antiderrapantes para un mejor manejo
- Amplio rango de medición: 10 a 600 mg/dl.
- Transferencia de datos a PC
- Utiliza la tira reactiva Accu-Chek Sensor Comfort Glucose (R)
- Curva que se adapta a la forma del dedo al momento de dosificar
- Ventana de verificación para garantizar su correcta dosificación
- Única con electrodos de paladio, para garantizar una correcta cuantificación del resultado
- Enciende automáticamente al insertar la tira
- Promedios de 7.14 y 30 días
- 480 memorias con fecha y hora



Figura 2. Glucómetro Accu-chek Sensor

Fuente: Accu- Chek, 2005, http://www.accu-chek.com.pe/04_product/AccuChekGo.htm

Accu-Chek Active (R) Glucómetro con manual.

Características

- Tamaño mínimo, discreción máxima.
- 200 memorias con fecha y hora.
- Promedio de los últimos 7 y 14 días.
- Resultado en 5 seg. Aproximadamente.
- Transferencia inalámbrica de datos a PC a través de puerto infrarrojo.
- Amplio rango de medición de glucosa (10-600 mg/dl) y temperatura ambiental (10 a 40 ° centígrados).
- Confiabilidad y precisión de alta tecnología.
- Se enciende al introducir la tira reactiva.



Figura 3. Glucómetro Accu-chek Active

Fuente: Accu-Chek, 2005, http://www.accu-chek.com.pe/04_product/AccuChekGo.htm

Ascensia ELITE Glucómetro

Medidor de glucosa en sangre

Contiene

- 1 Ascensia ELITE (instrumento glucómetro)
- 1 caja con 25 tiras reactivas Ascensia ELITE
- 1 dispositivo para punción con 5 lancetas
- 1 manual con sección en español
- 1 tira control
- 1 estuche de vinyl
- 1 agenda de resultados DEXTROLOG
- 1 registro de garantía
- 1 guía rápida

La última innovación de Bayer no es simplemente un Medidor de Glucosa en Sangre o una Tira Reactiva para medir el Nivel de Glucosa en Sangre. Es un enfoque fresco hacia el cuidado de la diabetes y está siendo llamado Ascensia™. Bayer reconoce que la necesidad de las personas con diabetes y de los profesionales de la salud varían extensamente. Compartiendo con esas personas para recavar suficiente información, Bayer está desarrollando los productos, servicios y los sistemas de apoyo para ajustarse a las necesidades del individuo. El enfoque fresco ha permitido a Bayer crear una red de productos amistosos para el usuario en la línea Ascensia que vienen en un paquete de sistemas de soporte y servicios enfocados al cliente.



Figura 4. Glucómetro Ascensia Elite

Fuente: Diabetes Gold, 2005, http://www.diabetesgold.com/sp_glucometro.htm.

Accu-chek Go



Figura 5. Glucómetro Accu-chek-Go

Fuente: Accu-Chek 2006, www.accu-chek.com/us/rewrite/content/en-us/4.1:10/article/ACCM_general_article-2410.htm.

Cada Kit Active New contiene:

- Medidor y estuche
- Manual de usuario
- Lancetero Softclix
- 10 Lancetas Softclix con hoja de instrucciones
- Tubo con 10 tiras reactivas y chip de codificación

- Certificado de Garantía

Características

- Medición de 5 segundos de duración
- Botón de eyección de tira reactiva
- 300 posiciones de medición (con hora y fecha)
- Necesidad de un gota muy pequeña de sangre
- Promedio de los últimos 7, 14 y 30 días
- Alarma recordatoria para medición de glucosa o colocación de insulina
- Alarma de fecha de expiración de tiras
- Alarma de hipoglucemia
- Capacidad de reutilización de tira en caso se haya realizado la prueba con muy poca sangre
- Capacidad de transferencia de datos a través de puerto infrarrojo para la comunicación con programas de análisis de PC adecuados (p.ej. Accu-chek Compass)
- Sujetadores de goma.

OneTouch ULTRA



Figura 6. Glucómetro OneTouch Ultra

Fuente: OneTouch ULTRA, 2006, <http://www.sanborns.com.mx/sanborns/>

Cada kit contiene:

- Medidor OneTouch Ultra para la determinación de glucosa en la sangre.
- 10 tiras reactivas.
- Dispositivo de punción regulable.
- Solución de control 3,75 ml.
- 10 Lancetas estériles.
- Manual de Usuario.
- Guía rápida de Uso.
- Pila de litio 3.0 V (instalada).
- Cápsula para analizarse el brazo.

Características

- Gran pantalla con números más grandes de fácil lectura.

- 150 memorias con fecha y hora.
- Promedios automáticos de 14 y 30 días.
- Puede transferir datos al programa In Touch™ para visualizar gráficamente e identificar tendencias con puerto USB.
- También permite tomar muestras de sangre del brazo.

ANEXO B

Guía de consulta rápida One Touch Ultra


CODIFICACIÓN DEL MEDIDOR

Codifique el medidor antes de usarlo por primera vez y cada vez que cambie de tiras de tiras reactivas OneTouch® Ultra.

PASO 1
Entre en el modo de codificación. Comience con el medidor apagado. Inserte la tira reactiva para encender el medidor. Aparecerá todos los elementos en la pantalla, seguidos por el número de código.

PASO 2
Haga coincidir los números de código. Compare el número de código que aparece en la pantalla del medidor con el del frasco de tiras reactivas. Si coinciden, comience el análisis. Si no coinciden, consulte el Paso 3.

PASO 3
Codifique el medidor. Pulse el botón C hasta que los números de código coincidan. Cuando aparezca Δ , el medidor estará listo para comenzar el análisis.



AW060381724*

PASO 4
Resultados listos en sólo 5 segundos.

UTILIZACIÓN DE LA MEMORIA DEL MEDIDOR

PASO 1
Entre en el modo de memoria. Comience con el medidor apagado. Pulse el botón M. Cuando vea el promedio de 14 días seguido por el 0: 90 días, está en el modo de memoria.

PASO 2
Recupere los resultados del análisis. Tres segundos después, aparecerá el resultado más reciente junto con la fecha y la hora. Pulse el botón C una vez y aparecerá el resultado del segundo análisis más reciente. Si continúa pulsando el botón C, podrá ver los resultados de los últimos 15P análisis por orden cronológico.

PASO 3
Baja del modo de memoria. Pulse el botón M para apagar el medidor.

SÍMBOLOS Y MENSAJES DE LA PANTALLA

Cuando aparezca la hora de modo intermitente en la pantalla, significa que el medidor ha perdido energía momentáneamente y que ha entrado en la modalidad de ahorro. Reajuste la hora y su zona, la fecha y, sobre todo, la unidad de medida, según si fuera necesario. Después de reajustar el medidor, introduzca la tira reactiva. Si en la pantalla aparece \rightarrow , muestre el código del medidor. Los resultados de los análisis guardados en la memoria del medidor al ser encendido se reinician en el que se reinician. No tenga en cuenta los promedios de los últimos 14 y 30 días. Llame al Servicio de Atención al Cliente 900 100 228 (línea gratuita).

El Medidor nuevo sin código. (Codificado) o 2) El medidor se ha quedado sin energía momentáneamente y ha entrado en la modalidad de reposo. Reajuste el número del código. Llame al Servicio de Atención al Cliente 900 100 228. Posible problema con la memoria del medidor: no utilice los promedios de los últimos 14 y 30 días.

Resultado del análisis con la sugerencia de verificar el nivel de tiempos reactivos.

RESUMEN DEL ANALISIS

Antes de comenzar el análisis, limpie los dedos con agua tibia y jabón. Desespójese y seque bien.

PASO 1
Inserte la tira reactiva. Presione la zona que luego seque. El medidor se encenderá automáticamente.

PASO 2
Aplique la muestra. Cuando aparezca el símbolo Δ , asegúrese de que la gota de sangre sea el tamaño correcto del fondo superior de la tira reactiva.

PASO 3
Inserte la muestra de codificación. Asegúrese de que la muestra está bien antes de que el medidor complete la cuenta atrás. Si la muestra no se ha liberado antes de completar a contar el medidor, repita la prueba con una tira.

HI El nivel de glucosa en sangre es superior a 500 mg/dL (33,3 mmol/L).

LO El nivel de glucosa en sangre es inferior a 20 mg/dL (1,1 mmol/L).

LOE La temperatura es demasiado baja para efectuar el análisis. Repita el análisis cuando la temperatura del medidor sea superior a 6°C.


HIE La temperatura es demasiado alta para efectuar el análisis. Repita el análisis cuando la temperatura del medidor sea inferior a 37°C.

--- La gota está demasiado baja de carga para efectuar el análisis. Reintente de inmediato.

ONE TOUCH® Ultra

SISTEMA PARA LA DETERMINACIÓN DE GLUCOSA EN SANGRE
SISTEMA DE CONTROL DE LA GLUCOSA EN SANGRE

GUIA RÁPIDA
GUIA DE CONSULTA RÁPIDA



IMPORTANTE: Lea las instrucciones detalladas en el Manual de Usuario de OneTouch® Ultra antes de realizar un análisis. Consulte el medidor para conocer el procedimiento de análisis y las instrucciones para utilizar el medidor.

IMPORTANTE: Lea las instrucciones detalladas en el Manual de Usuario de OneTouch® Ultra antes de realizar un test. Vaya en orden de embalajes como otras más instrucciones sobre a configuración de medidor o a procedimiento de test.

LIFESCAN
glucosímetros

Problema con el medidor. Repita el análisis con una tira nueva.

Er-1

La tira reactiva estaba ya usada o hay algún problema con el medidor. Repita el análisis con una tira nueva.

Er-2

La muestra de sangre o la solución de control se ha aplicado antes de aparecer en pantalla. Repita el análisis con una tira nueva.

Er-3

Podría deberse a tres causas:
El posible nivel alto de glucosa en su lugar como temperatura con presión al fondo inferior de temperatura de funcionamiento del medidor (6 - 37 °C). Repita el análisis en su lugar más adecuado. O bien, 2) La tira reactiva puede estar deteriorada o haberse pasado de tiempo de uso. Repita el análisis con una tira reactiva nueva. O bien, 3) La muestra se ha aplicado de modo incorrecto. Compruebe la aplicación de la muestra y repita el análisis con una tira reactiva nueva.

Er-4

Tira reactiva dañada o veneno de contaminación en tira. Repita el análisis con una tira nueva.

Er-5

Si el problema persiste, llame a Lifescan o Servicio de Atención al Cliente: 900 100 228.

Figura 1. Guía rápida de One touch ultra
Fuente: Lifescan, 2002

APÉNDICE A

Manual de Usuario Programa- Paciente, Programa Médico

Requerimientos del Sistema

Para que el programa se ejecute de manera adecuada, el sistema debe contener:

- Programa paciente.exe
- MSCOMM MEDICO RECEPTOR EMAIL.exe
- Medico programa principal 7.1.exe
- La carpeta “Proyectodiabetes” almacenada en c:\, aquí esta la base de datos, la voz pregrabada y el archivo txt. que utiliza el programa
- Software del cable del glucómetro “ que lo ubica en el COM 4
- Modem, ubicado en el COM 3
- Conexión a Internet establecida (sin firewall)

Nota: el correo al cual se envía y recibe la información es:

Login: “proyectodiabetes@cantv.net

Password: 12345678

Manual del Usuario

El sistema consta principalmente de dos programas “programa del paciente” y “programa del medico”.

Antes de explicar el uso del programa, es preciso aclarar que el glucómetro posee un cable USB para la conexión con el computador. Por configuración de controladores (*drivers*), el puerto USB donde se conecta dicho cable es reconocido como un puerto COM, específicamente COM 4. Es por ello que se trató al glucómetro como un dispositivo conectado por puerto serial a lo largo de todo el proyecto.

Programa del paciente

1. Antes de realizar cualquier conexión, instale el controlador (driver) del cable del glucómetro en la PC.
2. Una vez instalado, proceda a efectuar la medición de insulina siguiendo las instrucciones indicadas en el manual del glucómetro.
3. Conecte el glucómetro a la PC como se muestra en la figura 1.



Figura 1. Glucómetro conectado al PC.
Fuente: Elaboración Propia.

4. Ejecute el programa “programa del paciente.exe” y espere que aparezca la siguiente ventana.

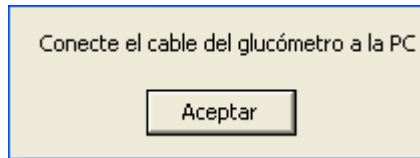


Figura 2. Mensaje arrojado por el programa.
Fuente: Elaboración Propia.

5. Como ya el glucómetro ha sido conectado, presione “Aceptar” y le aparecerá una ventana como la siguiente.

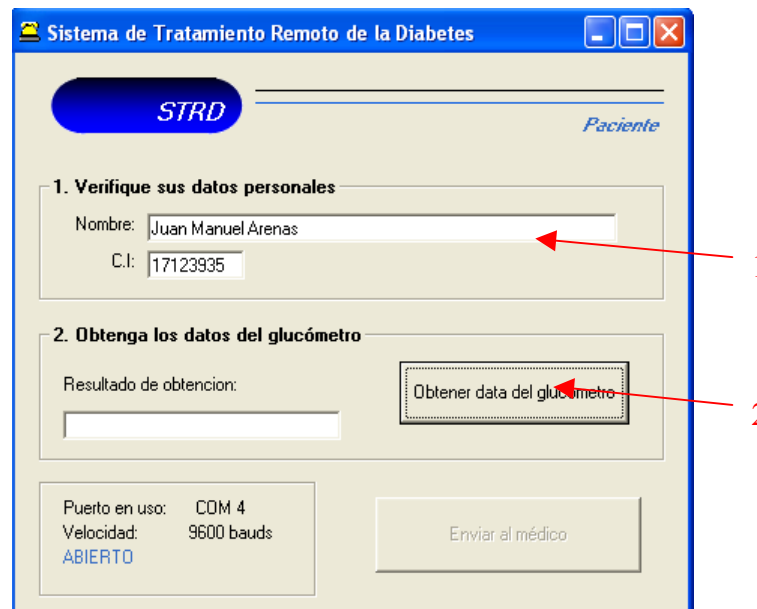


Figura 3. Ventana Programa Paciente.
Fuente: Elaboración Propia.

Aquí, el usuario debe introducir su nombre completo y su cédula. Luego debe proceder a obtener la data del glucómetro presionando el botón “Obtener data del glucómetro”. Este procedimiento tomará de dos a cinco segundos en efectuarse.

Una vez obtenida la información del glucómetro, es transmitida vía correo electrónico al médico presionando el botón “Enviar al médico” como lo indica la figura 4.

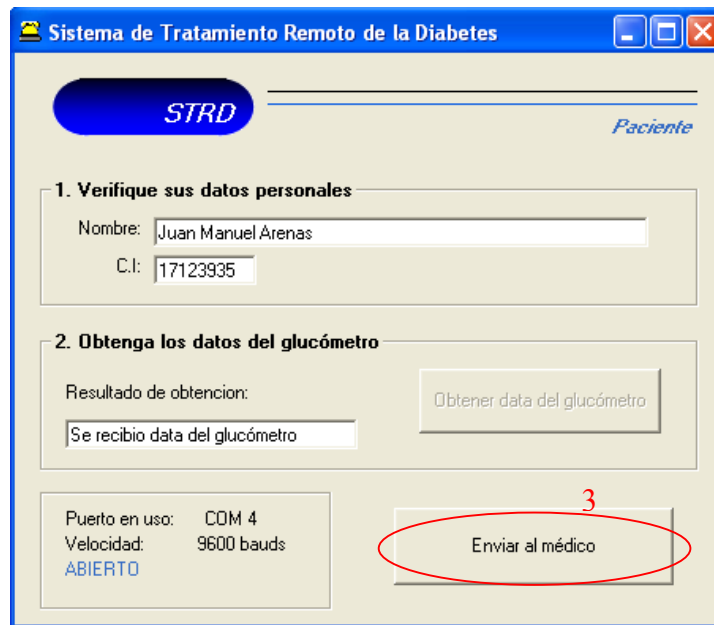


Figura 4. Envío de la data al médico.
Fuente: Elaboración Propia.

Programa del médico

Al ejecutar el “programa del médico” se dispone de una ventana principal como la mostrada en la figura 5.

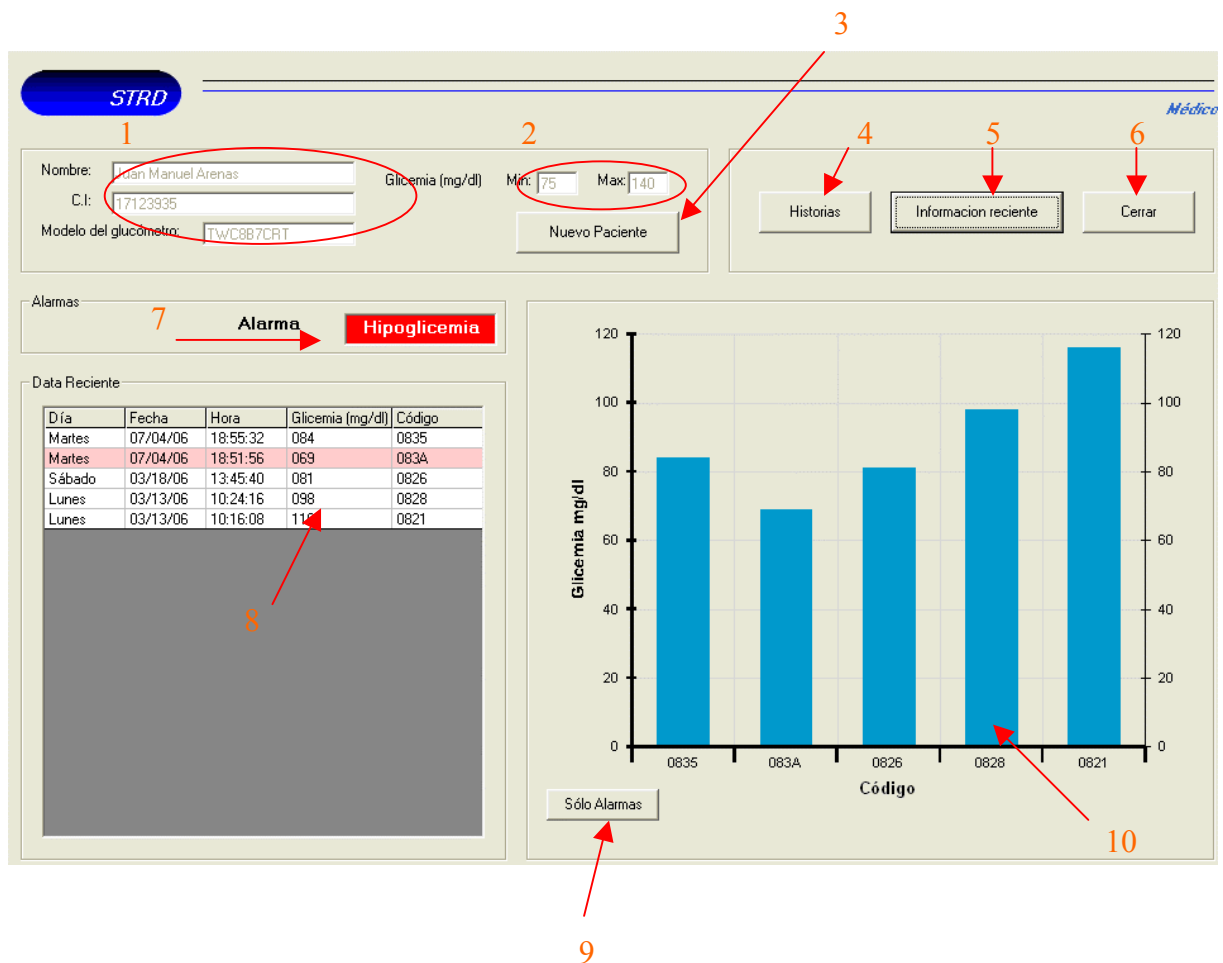


Figura 5. Ventana Principal Programa Médico.

Fuente: Elaboración Propia.

Las opciones que posee esta interfaz son las siguientes:

11. Información del paciente: que incluye nombre completo, número de cédula y tipo de glucómetro que emplea el paciente.
12. Valores Límite de glicemia del paciente en cuestión: muestra los mg/gl de glicemia en la sangre que se toman como referencia para determinar una hiper o hipoglucemia en caso de existir.

13. Botón nuevo paciente: Permite al médico registrar un nuevo usuario en el sistema. Si no se realiza este proceso, el paciente no dispondrá del servicio del sistema ya que sus datos no serán tomados en cuenta al ser recibidos vía correo electrónico.
14. Botón Historias: Muestra las historias de todos y cada uno de los pacientes registrados así como también los casos de alarma que hayan podido presentarse.
15. Botón Información reciente: actualiza la información mostrada en la pantalla principal de forma manual, en caso de que el medico no desee esperar el tiempo que tarda el programa en refrescarla automáticamente cada diez minutos.
16. botón cerrar: cierra y guarda los cambios realizados en la aplicación
17. Indicador de alarmas: se enciende en caso de existir alguna Hiper o hipoglucemia.
18. Tabla informativa: posee la información del día, fecha, hora, valor de glicemia y código de cada una de las mediciones realizadas por el paciente en cuestión. Señalando en fondo rosado los casos de alarma tal y como se muestra en la figura anterior.
19. Botón sólo alarmas: muestra gráficamente sólo las alarmas que hayan podido presentarse, (ver figura siguiente). En caso de no haber alarmas, dicho botón queda deshabilitado.
20. Gráfica: muestra en gráfica de barras cada una de las mediciones encontradas en la tabla.

Cabe destacar que cada uno de estos elementos en la ventana principal se actualiza al variar la data que este llegando recientemente.

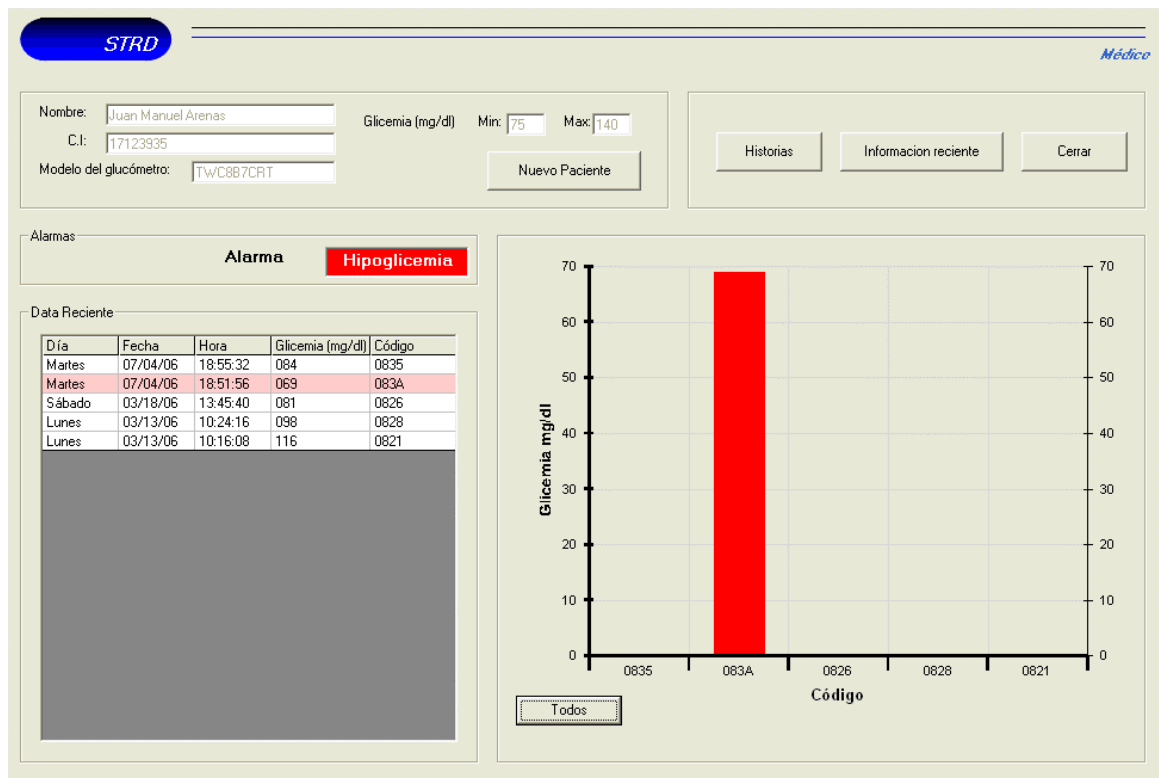


Figura 6. Muestra de alarma.

Fuente: Elaboración Propia.

Al presionar el botón “Nuevo paciente” se abre una nueva ventana en la cual se introducen datos del paciente a registrar como nombre completo, cédula de identidad, teléfono, dirección, ciudad y valores límite de glicemia máximo y mínimo. Por otro lado se seleccionan la fecha de nacimiento y el tipo de diabetes (tipo 1 o tipo 2) del mismo.

The screenshot shows a window titled "Nuevo Paciente:" with a blue header bar. Inside the window, there is a logo for "STRD" and the word "Médico" in the top right corner. The form contains the following fields and controls:

- Nombre:
- CI:
- Teléfono:
- Dirección:
- Ciudad:
- Fecha de nacimiento: / /
- Tipo de diabetes:
- Valores críticos de glicemia: (mg/dl)
Min: Máx:

At the bottom of the form, there are two buttons: "Almacenar" and "Volver".

Figura 7. Opción Nuevo Paciente.
Fuente: Elaboración Propia.

Una vez terminada la introducción de datos, se habilita el botón “Almacenar”, el cual sólo guardara la información suministrada si ésta está completa. La función de éste botón es crear un nuevo perfil en Access, llenando cada columna de la tabla de la base de datos.

Haciendo “clic” en el botón “Historias” en la ventana principal, aparecerá una nueva mostrando todos los datos personales y las mediciones de los pacientes previamente registrados en el sistema tal y como se observa en la figura siguiente.

Historia del paciente

STRD

Paciente: Juan Manuel Arenas

C.I.: 17123935

Teléfono: 2433735

Dirección: Calle13, Edif Stefania, Apto 42 La Urbina

Ciudad: Caracas Fecha de nacimiento: 2/4/1984

Tipo de diabetes: Tipo 2

Valores de glicemia: Max 90 Hiperglicemia

Min 80 Hipoglicemia

Datos

Día	Fecha	Hora	Glicemia (mg/dl)	Código
Lunes	13/03/2006	10:16:08	116	0821
Lunes	13/03/2006	10:24:16	98	0828
Sábado	18/03/2006	05:45:40	81	0826
Martes	07/04/2006	18:51:56	69	083A
Martes	07/04/2006	09:55:32	84	0835

Imprimir Ver Gráfica Volver

Figura 8. Historia del Paciente.
Fuente: Elaboración Propia

Esta consulta se ejecuta al seleccionar el paciente haciendo “clic” en la pestaña mostrada en la figura anterior.

Una vez ubicados los datos en Access, se cargan en la pantalla. Esta información contiene todas y cada una de las mediciones del paciente, con sus respectivas alarmas, desde el momento en que es registrado así como también los datos personales del mismo.

Esta ventana además de mostrar alarmas de hipo e hiperglicemia, resalta aquellas mediciones que hayan sido efectuadas en ayuno. De esta manera se facilita el análisis del resultado por parte del médico.

El botón “Imprimir” permite al médico otorgar un informe escrito de la historia del paciente. Esta opción no requiere de configuraciones adicionales ya que con sólo tener una impresora asignada al PC, es suficiente.

Al presionar el botón “Ver gráfica” el programa del médico muestra una pantalla como la expuesta en la siguiente figura.



Figura 9. Opción Ver Gráfica.

Fuente: Elaboración Propia.

Donde se visualiza gráficamente los valores existentes en la tabla de la ventana “Historia del paciente”.

Al igual que en la ventana principal, aquí se dispone también del botón “Solo alarmas”, que resalta únicamente las mediciones que se encuentren fuera de los rangos normales de glicemia.

En la parte superior izquierda se encuentran las opciones de visualización, en donde se dispone de observación por mes y año de los datos de la historia del paciente o bien de todos. También existe la posibilidad de variar la forma de la gráfica para una mejor apreciación de valores, ver figura siguiente.

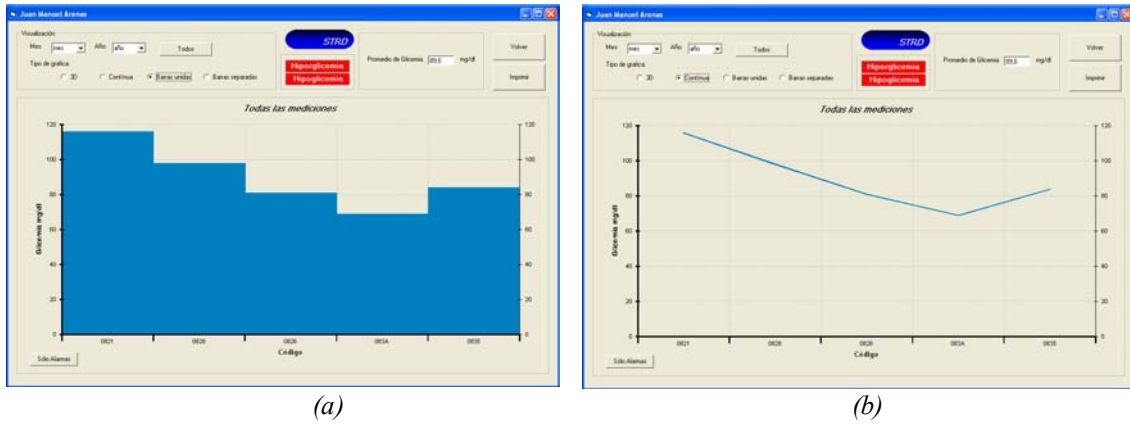


Figura 10. Opciones de Visualización.

(a) Barras Unidas, (b) Continua.

Fuente: Elaboración Propia.

En la parte superior central se aprecian los indicadores de alarma, a su derecha el promedio en mg/dl de todos los valores mostrados en la gráfica. En la parte superior derecha se encuentra el botón “Imprimir” el cual además de imprimir los datos del paciente y todas sus mediciones de manera escrita, plasma en una segunda hoja la gráfica correspondiente como se muestra en la figura siguiente.

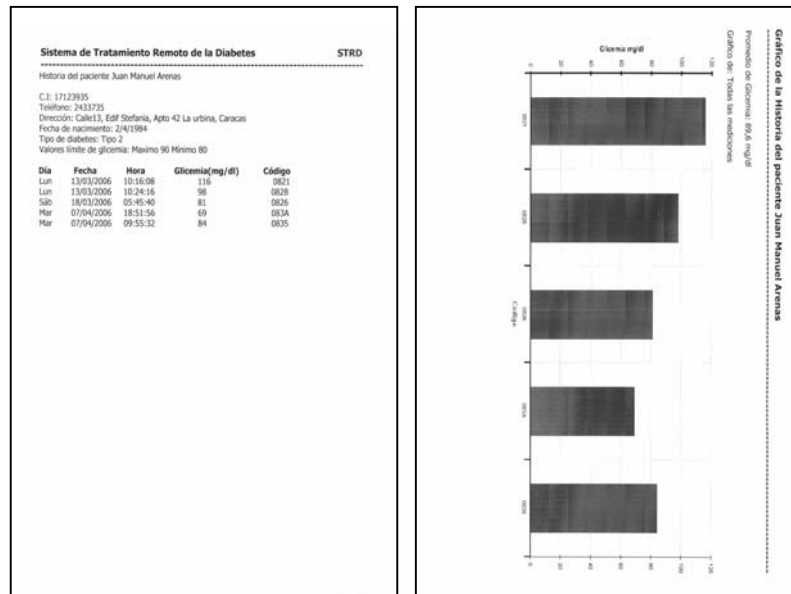


Figura 11. Impresión de las mediciones.

Fuente: Elaboración Propia.

APÉNDICE B

Códigos desarrollados para las aplicaciones de programa médico y programa paciente

Medico programa principal:

Form1

```
Private Sub Form_Load()
Text4.Visible = False
Text5.Visible = False
Text9.Visible = False
Label7.Caption = "Ninguna"
Modem_detectado = False
DetectarModem
MSFlexGrid1.TextMatrix(0, 0) = "Día"
MSFlexGrid1.TextMatrix(0, 1) = "Fecha"
MSFlexGrid1.TextMatrix(0, 2) = "Hora"
MSFlexGrid1.ColWidth(3) = 1250
MSFlexGrid1.TextMatrix(0, 3) = "Glicemia (mg/dl)"
MSFlexGrid1.ColWidth(4) = 1250
MSFlexGrid1.TextMatrix(0, 4) = "Código"
For alineacion = 0 To 4
MSFlexGrid1.ColAlignment(alineacion) = 0
Next alineacion
Timer1.Enabled = True
contador = 0
Shell "c:\Proyectodiabetes\MSCOMM MEDICO RECEPTOR EMAIL.exe",
vbMinimizedFocus
Timer5.Enabled = True
End Sub
Private Sub MSChart1_OLEStartDrag(Data As MSChart20Lib.DataObject,
AllowedEffects As Long)
If MSChart1.Visible = True Then
MSChart1.Visible = False
Else
MSChart1.Visible = True
End If
End Sub
Private Sub Timer1_Timer()
contador = contador + 1
If contador > 30 Then
Shell "c:\Proyectodiabetes\MSCOMM MEDICO RECEPTOR EMAIL.exe",
vbMinimizedFocus
contador = 0
Timer5.Enabled = True
End If
Timer1.Enabled = False
Timer2.Enabled = True
End Sub
Private Sub Timer2_Timer()
```

```
Timer1.Enabled = True
Timer2.Enabled = False
End Sub
Private Sub Command2_Click()
Command5.Visible = False
Text4.Visible = False
Text5.Visible = False
Label7.Caption = "Ninguna"
MSFlexGrid1.BackColor = &H8000000E
alarma2 = False
contadortxt = 0
rows = 1
MSFlexGrid1.rows = 1
Open "c:\Proyectodiabetes\proyectodiabetes.txt" For Append As 1
Close #1
Open "c:\Proyectodiabetes\proyectodiabetes.txt" For Input As 1
Do While Not EOF(1)
Line Input #1, lineatxt
If StrComp(lineatxt, "Vacío") <> 0 And StrComp(lineatxt, "") <> 0 And
StrComp(lineatxt, "¥") <> 0 Then
If contadortxt = 0 Then
Text1.Text = lineatxt
If StrComp(Nombre_anterior, lineatxt) <> 0 Then
alarma2 = True
End If
Nombre_anterior = lineatxt
contadortxt = contadortxt + 1
ElseIf contadortxt = 1 Then
Text2.Text = lineatxt
contadortxt = contadortxt + 1
ElseIf contadortxt = 2 Then
palabratxt = Left(lineatxt, 17)
Text3.Text = Right(palabratxt, 9)
contadortxt = contadortxt + 1
ElseIf contadortxt > 2 Then
MSFlexGrid1.rows = MSFlexGrid1.rows + 1
palabratxt = Left(lineatxt, 6)
palabratxt = Right(palabratxt, 3)
If StrComp(palabratxt, "MON") = 0 Then 'traduzco a español
MSFlexGrid1.TextMatrix(rows, 0) = "Lunes  "
ElseIf StrComp(palabratxt, "TUE") = 0 Then
MSFlexGrid1.TextMatrix(rows, 0) = "Martes  "
ElseIf StrComp(palabratxt, "WED") = 0 Then
MSFlexGrid1.TextMatrix(rows, 0) = "Miércoles"
ElseIf StrComp(palabratxt, "THU") = 0 Then
```



```

        MSFlexGrid1.TextMatrix(rows, 0) = "Jueves  "
    ElseIf StrComp(palabratxt, "FRI") = 0 Then
        MSFlexGrid1.TextMatrix(rows, 0) = "Viernes  "
    ElseIf StrComp(palabratxt, "SAT") = 0 Then
        MSFlexGrid1.TextMatrix(rows, 0) = "Sábado  "
    ElseIf StrComp(palabratxt, "SUN") = 0 Then
        MSFlexGrid1.TextMatrix(rows, 0) = "Domingo  "
    End If
    palabratxt = Left(lineatxt, 17)
    MSFlexGrid1.TextMatrix(rows, 1) = Right(palabratxt, 8)
    palabratxt = Left(lineatxt, 28)
    MSFlexGrid1.TextMatrix(rows, 2) = Right(palabratxt, 8)
    palabratxt = Right(lineatxt, 14)
    MSFlexGrid1.TextMatrix(rows, 3) = Left(palabratxt, 3)
    MSFlexGrid1.TextMatrix(rows, 4) = Right(palabratxt, 4)
    rows = rows + 1
    End If
ElseIf StrComp(lineatxt, "Vacio") = 0 Then
    Close #1
    Text1.Text = "No hay registros nuevos"
    Text2.Text = " ----- "
    Text3.Text = " ----- "
    Exit Sub
End If
Loop
Close #1
MSChart1.RowCount = 0
For Row = 1 To MSFlexGrid1.rows - 1
    MSChart1.RowCount = MSChart1.RowCount + 1
    MSChart1.Row = MSChart1.RowCount
    MSChart1.RowLabel = MSFlexGrid1.TextMatrix(Row, 4)
    MSChart1.Data = MSFlexGrid1.TextMatrix(Row, 3)
Next Row
proyectodiabetes.mdb en la tabla "Pacientes"
Set BDD = OpenDatabase("c:\Proyectodiabetes\proyectodiabetes.mdb")
SQL = "SELECT * FROM Pacientes "
SQL = SQL + "WHERE ((Pacientes.CI)=(" & Text2.Text & ")); "
Set TBL = BDD.OpenRecordset(SQL)
If TBL.EOF = True Then
    Form5.Visible = True
    Form5.Timer1.Enabled = True
    Exit Sub
End If
TBL.Close
BDD.Close

```

```
For rows = 1 To MSFlexGrid1.rows - 1
    Set BDD = OpenDatabase("c:\Proyectodiabetes\proyectodiabetes.mdb")
    SQL = "SELECT * FROM Pacientes"
    Set TBL = BDD.OpenRecordset(SQL)
    TBL.MoveFirst
    repetido = 0
    Do Until TBL.EOF
        If Val(Text2.Text) = TBL("CI") And
StrComp(MSFlexGrid1.TextMatrix(rows, 4), TBL("codigo")) = 0 Then
            repetido = 1
                Text6.Text = TBL("Valor_min_glicemia")
                Text7.Text = TBL("Valor_max_glicemia")
            End If
            TBL.MoveNext
        Loop
        TBL.MoveFirst
        If repetido = 0 Then
            Do Until TBL.EOF
                If Val(Text2.Text) = TBL("CI") Then
                    A = TBL("Nombre")
                    b = TBL("CI")
                    c = TBL("Teléfono")
                    d = TBL("Dirección")
                    e = TBL("Ciudad")
                    f = TBL("Fecha_de_nacimiento")
                    g = TBL("Tipo_de_diabetes")
                    h = TBL("Valor_min_glicemia")
                    i = TBL("Valor_max_glicemia")
                    Text6.Text = TBL("Valor_min_glicemia")
                    Text7.Text = TBL("Valor_max_glicemia")
                End If
                TBL.MoveNext
            Loop
            SQL = "INSERT INTO Pacientes
(Nombre,CI,Teléfono,Dirección,Ciudad,Fecha_de_nacimiento,Tipo_de_diabetes,Val
or_min_glicemia,Valor_max_glicemia,Dia_de_medicion,fecha_de_medicion,hora_de
_medicion,glicemia,codigo) VALUES('" & A & "','" & b & "','" & c & "','" & d & "','"
& e & "','" & f & "','" & g & "','" & h & "','" & i & "','" &
MSFlexGrid1.TextMatrix(rows, 0) & "','" & MSFlexGrid1.TextMatrix(rows, 1) &
','" & MSFlexGrid1.TextMatrix(rows, 2) & "','" & MSFlexGrid1.TextMatrix(rows,
3) & "','" & MSFlexGrid1.TextMatrix(rows, 4) & "'"")"
            BDD.Execute SQL
        End If
        TBL.Close
        BDD.Close
    End If
```

```
Next rows
max xon la fila 3 de la tabla
alarma = False
For rows = 1 To MSFlexGrid1.rows - 1
    If Val(MSFlexGrid1.TextMatrix(rows, 3)) > Val(Text7.Text) Then
        Command5.Visible = True
        Text4.Visible = True
        Label7.Caption = " Alarma "
        For Column = 0 To 4
            MSFlexGrid1.Row = rows
            MSFlexGrid1.Col = Column
            MSFlexGrid1.CellBackColor = &HC0C0FF ' rosado
            alarma = True
        Next Column
    ElseIf Val(MSFlexGrid1.TextMatrix(rows, 3)) < Val(Text6.Text) Then
        Command5.Visible = True
        Text5.Visible = True
        Label7.Caption = " Alarma "
        For Column = 0 To 4
            MSFlexGrid1.Row = rows
            MSFlexGrid1.Col = Column
            MSFlexGrid1.CellBackColor = &HC0C0FF ' rosado
            alarma = True
        Next Column
    End If
Next rows
If alarma = True And alarma2 = True Then
End If
MSChart2.RowCount = 0
For Row = 1 To MSFlexGrid1.rows - 1
    If Val(MSFlexGrid1.TextMatrix(Row, 3)) > Val(Text7.Text) Or
Val(MSFlexGrid1.TextMatrix(Row, 3)) < Val(Text6.Text) Then
        MSChart2.RowCount = MSChart2.RowCount + 1
        MSChart2.Row = MSChart2.RowCount
        MSChart2.RowLabel = MSFlexGrid1.TextMatrix(Row, 4)
        MSChart2.Data = MSFlexGrid1.TextMatrix(Row, 3)
    Else
        MSChart2.RowCount = MSChart2.RowCount + 1
        MSChart2.Row = MSChart2.RowCount
        MSChart2.RowLabel = MSFlexGrid1.TextMatrix(Row, 4)
        MSChart2.Data = 0
    End If
Next Row
For rows = 1 To MSFlexGrid1.rows - 1
```

```
    If Left(MSFlexGrid1.TextMatrix(rows, 2), 2) > 6 And
Left(MSFlexGrid1.TextMatrix(rows, 2), 2) < 10 Then
        Text9.Visible = True
        MSFlexGrid1.Row = rows
        MSFlexGrid1.Col = 0
        MSFlexGrid1.CellBackColor = &H80000013 ' azul
    End If
Next rows
End Sub
Private Sub Timer3_Timer()
mciExecute ("Play c:\Proyectodiabetes\voz.wav")
Timer4.Enabled = True
Timer3.Enabled = False
End Sub
Private Sub Timer4_Timer()
Timer4.Enabled = False
End Sub
Private Sub Timer5_Timer()
Command2_Click
Timer5.Enabled = False
End Sub
Private Sub Command1_Click()
Form5.Visible = False
Form2.Visible = True
Form2.Text1.Text = Form1.Text1.Text
Form2.Text2.Text = Form1.Text2.Text
End Sub
Private Sub Command2_Click()
Form5.Visible = False
Form1.Visible = True
End Sub
Private Sub Timer1_Timer()
Command2_Click
Timer1.Enabled = False
End Sub
Form2
Private Sub Form_Load()
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Combo1.Text = "Dia"
```

```
Combo2.Text = "Mes"
Combo3.Text = "Año"
Combo4.Text = "Seleccione"
Command1.Enabled = True
  For dia = 1 To 31
    Combo1.AddItem dia
  Next dia
  For mes = 1 To 12
    Combo2.AddItem mes
  Next mes
  año = 2006
  Do While año >= 1900
    Combo3.AddItem año
    año = año - 1
  Loop
  Combo4.AddItem "Tipo 1"
  Combo4.AddItem "Tipo 2"
End Sub
Private Sub Command1_Click()
Form2.Visible = False
Form1.Visible = True
If StrComp(Text1.Text, "") = 0 Or StrComp(Text2.Text, "") = 0 Or
StrComp(Text3.Text, "") = 0 Or StrComp(Text4.Text, "") = 0 Or
StrComp(Text5.Text, "") = 0 Or StrComp(Text6.Text, "") = 0 Or
StrComp(Text7.Text, "") = 0 Or StrComp(Combo1.Text, "Dia") = 0 Or
StrComp(Combo2.Text, "Mes") = 0 Or StrComp(Combo3.Text, "Año") = 0 Or
StrComp(Combo4.Text, "Seleccione") = 0 Then
  MsgBox " Introduzca todos los datos personales correctamente "
  GoTo terminar
End If
fecha = Combo1.Text + "/" + Combo2.Text + "/" + Combo3.Text
Set BDD = OpenDatabase("c:\Proyectodiabetes\proyectodiabetes.mdb")
A = Text1.Text
b = Text2.Text
c = Text7.Text
d = Text3.Text
e = Text4.Text
f = fecha
g = Combo4.Text
h = Text5.Text
i = Text6.Text
SQL = "INSERT INTO Pacientes
(Nombre,CI,Teléfono,Dirección,Ciudad,Fecha_de_nacimiento,Tipo_de_diabetes,Val
or_min_glicemia,Valor_max_glicemia) VALUES(" & A & "," & b & "," & c &
"," & d & "," & e & "," & f & "," & g & "," & h & "," & i & ")"
```

```
BDD.Execute SQL
BDD.Close
Set BDD = OpenDatabase("c:\Proyectodiabetes\proyectodiabetes.mdb")
SQL = "SELECT * FROM Pacientes "
SQL = SQL + "WHERE ((Pacientes.CI)=(" & Form1.Text2.Text & ")); "
Set TBL = BDD.OpenRecordset(SQL)
If TBL.EOF = True Then
    Exit Sub
End If
TBL.Close
BDD.Close
For rows = 1 To Form1.MSFlexGrid1.rows - 1
    Set BDD = OpenDatabase("c:\Proyectodiabetes\proyectodiabetes.mdb")
    SQL = "SELECT * FROM Pacientes"
    Set TBL = BDD.OpenRecordset(SQL)
    TBL.MoveFirst
    repetido = 0
    Do Until TBL.EOF
        If Val(Form1.Text2.Text) = TBL("CI") And
StrComp(Form1.MSFlexGrid1.TextMatrix(rows, 4), TBL("codigo")) = 0 Then
            repetido = 1
            Form1.Text6.Text = TBL("Valor_min_glicemia")
            Form1.Text7.Text = TBL("Valor_max_glicemia")
        End If
        TBL.MoveNext
    Loop
    TBL.MoveFirst
    If repetido = 0 Then
        Do Until TBL.EOF
            If Val(Form1.Text2.Text) = TBL("CI") Then
                A = TBL("Nombre")
                b = TBL("CI")
                c = TBL("Teléfono")
                d = TBL("Dirección")
                e = TBL("Ciudad")
                f = TBL("Fecha_de_nacimiento")
                g = TBL("Tipo_de_diabetes")
                h = TBL("Valor_min_glicemia")
                i = TBL("Valor_max_glicemia")
                Form1.Text6.Text = TBL("Valor_min_glicemia")
                Form1.Text7.Text = TBL("Valor_max_glicemia")
            End If
            TBL.MoveNext
        Loop
    End If
End For
```

```
SQL = "INSERT INTO Pacientes
(Nombre,CI,Teléfono,Dirección,Ciudad,Fecha_de_nacimiento,Tipo_de_diabetes,Val
or_min_glicemia,Valor_max_glicemia,Dia_de_medicion,fecha_de_medicion,hora_de
_medicion,glicemia,codigo) VALUES(" & A & "," & b & "," & c & "," & d & ","
& e & "," & f & "," & g & "," & h & "," & i & "," &
Form1.MSFlexGrid1.TextMatrix(rows, 0) & "," &
Form1.MSFlexGrid1.TextMatrix(rows, 1) & "," &
Form1.MSFlexGrid1.TextMatrix(rows, 2) & "," &
Form1.MSFlexGrid1.TextMatrix(rows, 3) & "," &
Form1.MSFlexGrid1.TextMatrix(rows, 4) & ")
BDD.Execute SQL
End If
TBL.Close
BDD.Close
Next rows
End Sub
Form 3
Private Sub Form_Load()
Command1.Enabled = False
Command3.Enabled = False
Text11.Visible = False
rowanterior = 9999
rowanterior2 = 9999
rowanterior3 = 9999
MSFlexGrid1.rows = 0
rows = 0
Set BDD = OpenDatabase("c:\Proyectodiabetes\proyectodiabetes.mdb")
SQL = "SELECT distinct CI, Nombre FROM Pacientes"
Set TBL = BDD.OpenRecordset(SQL)
If TBL.EOF = True Then
MsgBox " No hay usuarios registrados"
Command3.Enabled = False
Combo1.Text = " No hay pacientes "
Exit Sub
End If
TBL.MoveFirst
Do Until TBL.EOF
MSFlexGrid1.rows = MSFlexGrid1.rows + 1
MSFlexGrid1.TextMatrix(rows, 0) = TBL("Nombre")
MSFlexGrid1.TextMatrix(rows, 1) = TBL("CI")
rows = rows + 1
TBL.MoveNext
Loop
TBL.Close
BDD.Close
```

```
Combo1.Text = MSFlexGrid1.TextMatrix(0, 0)
For rows = 0 To MSFlexGrid1.rows - 1
    Combo1.AddItem MSFlexGrid1.TextMatrix(rows, 0)
Next rows
Combo1.Enabled = True
Combo1.Text = "Selecione"
End Sub
```

Form4

```
Private Sub Form_Load()
    Option4.Value = True
End Sub

Private Sub Option1_Click()
    Option1.Value = True
    Option2.Value = False
    Option3.Value = False
    Option4.Value = False
    MSChart1.chartType = VtChChartType3dBar
    MSChart2.chartType = VtChChartType3dBar
End Sub

Private Sub Option2_Click()
    Option1.Value = False
    Option2.Value = True
    Option3.Value = False
    Option4.Value = False
    MSChart1.chartType = VtChChartType2dLine
    MSChart2.chartType = VtChChartType2dLine
End Sub

Private Sub Option3_Click()
    Option1.Value = False
    Option2.Value = False
    Option3.Value = True
    Option4.Value = False
    MSChart1.chartType = VtChChartType2dStep
    MSChart2.chartType = VtChChartType2dStep
End Sub

Private Sub Option4_Click()
    Option1.Value = False
    Option2.Value = False
    Option3.Value = False
    Option4.Value = True
    MSChart1.chartType = VtChChartType2dBar
    MSChart2.chartType = VtChChartType2dBar
End Sub
```

Form5

```
Private Sub Command1_Click()
```



```
Form5.Visible = False
Form2.Visible = True
Form2.Text1.Text = Form1.Text1.Text
Form2.Text2.Text = Form1.Text2.Text
End Sub
Private Sub Command2_Click()
Form5.Visible = False
Form1.Visible = True
End Sub
Private Sub Form_Load()
End Sub
Private Sub Timer1_Timer()
Command2_Click
Timer1.Enabled = False
End Sub
Declare Function mciExecute Lib "winmm.dll" (ByVal lpstrCommand As String) As
Long
Medico receptor email:
Private Sub Frame1_DragDrop(Source As Control, X As Single, Y As Single)
End Sub
Private Sub lvMessages_ItemClick(ByVal Item As ComctlLib.ListItem)
txtBody = m_colMessages(Item.Key).MessageBody
End Sub
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
Dim strData As String
Static intMessages As Integer
Static intCurrentMessage As Integer
Static strBuffer As String
Winsock1.GetData strData
Debug.Print strData
If Left$(strData, 1) = "+" Or m_State = POP3_RETR Then
Select Case m_State
Case POP3_Connect
intMessages = 0
m_State = POP3_USER
Winsock1.SendData "USER " & txtUserName & vbCrLf
Debug.Print "USER " & txtUserName
Case POP3_USER
m_State = POP3_PASS
Winsock1.SendData "PASS " & txtPassword & vbCrLf
Debug.Print "PASS " & txtPassword
Case POP3_PASS
m_State = POP3_STAT
Winsock1.SendData "STAT" & vbCrLf
Debug.Print "STAT"
```

```
Case POP3_STAT
  intMessages = CInt(Mid$(strData, 5, _
    InStr(5, strData, " ") - 5))
  mensajes_a_borrar = intMessages
  If intMessages > 0 Then
    m_State = POP3_RETR
    intCurrentMessage = intCurrentMessage + 1
    Winsock1.SendData "RETR 1" & vbCrLf
    Debug.Print "RETR 1"
  Else
    m_State = POP3_QUIT
    Winsock1.SendData "QUIT" & vbCrLf
    Debug.Print "QUIT"
  End If
Case POP3_RETR
  strBuffer = strBuffer & strData
  If InStr(1, strBuffer, vbLf & "." & vbCrLf) Then
    strBuffer = Mid$(strBuffer, InStr(1, strBuffer, vbCrLf) + 2)
    strBuffer = Left$(strBuffer, Len(strBuffer) - 3)
    Set m_oMessage = New CMessage
    m_oMessage.CreateFromText strBuffer
    m_colMessages.Add m_oMessage, m_oMessage.MessageID
    Set m_oMessage = Nothing
    strBuffer = ""
  If intCurrentMessage = intMessages Then
    m_State = POP3_DELE
    If mensajes_a_borrar > 0 Then
      Winsock1.SendData "DELE " & mensajes_a_borrar & vbCrLf
      Debug.Print "DELE " & mensajes_a_borrar
    End If
  Else
    intCurrentMessage = intCurrentMessage + 1
    m_State = POP3_RETR
    Winsock1.SendData "RETR " & _
      CStr(intCurrentMessage) & vbCrLf
    Debug.Print "RETR " & intCurrentMessage
  End If
End If
Case POP3_DELE
  m_State = POP3_QUIT
  Winsock1.SendData "QUIT" & vbCrLf
  Debug.Print "QUIT"
Case POP3_QUIT
  Winsock1.Close
  Call ListMessages
```

```
End Select
Else
    Winsock1.Close
    MsgBox "POP3 Error: " & strData, _
        vbExclamation, "POP3 Error"
End If
End Sub
Private Sub Winsock1_Error(ByVal Number As Integer, Description As String,
    ByVal Scode As Long, ByVal Source As String, ByVal HelpFile As String, ByVal
    HelpContext As Long, CancelDisplay As Boolean)
    MsgBox "Winsock Error: #" & Number & vbCrLf & _
        Description
End Sub
Private Sub ListMessages()
    Dim oMes As CMessage
    Dim lvItem As ListItem
    For Each oMes In m_colMessages
        Set lvItem = lvMessages.ListItems.Add
        lvItem.Key = oMes.MessageID
        lvItem.Text = oMes.From
        lvItem.SubItems(1) = oMes.Subject
        titulo = oMes.Subject
        lvItem.SubItems(2) = oMes.SendDate
        lvItem.SubItems(3) = oMes.Size
        If StrComp(titulo, "Resultado de medición de glicemia") = 0 Then
            txtBody = m_colMessages(lvItem.Key).MessageBody
        Else
            txtBody = "Vacio"
        End If
    Next
    Open "C:\Proyectodiabetes\proyectodiabetes.txt" For Append As 1
    Close #1
    Kill "C:\Proyectodiabetes\proyectodiabetes.txt"
    Open "C:\Proyectodiabetes\proyectodiabetes.txt" For Append As 1
    If EOF(1) Then
        Close #1
        Open "C:\Proyectodiabetes\proyectodiabetes.txt" For Output As 1
        Print #1, txtBody.Text
        Close #1
    End If
End
End Sub
Private m_strReturnPath As String
Private m_strReceived As String
Private m_strSendDate As String
```

```

Private m_strMessageID    As String
Private m_strMessageTo    As String
Private m_strFrom         As String
Private m_strSubject      As String
Private m_strReplyTo      As String
Private m_strSender       As String
Private m_strCC           As String
Private m_strBCC         As String
Private m_strInReplyTo    As String
Private m_strReferences   As String
Private m_strKeywords     As String
Private m_strComments     As String
Private m_strEncrypted    As String
Private m_strMessageText  As String
Private m_strMessageBody  As String
Private m_strHeaders      As String
Private m_lSize           As Long
Public Sub CreateFromText(strMessage As String)
    Dim intPosA           As Integer
    Dim vHeaders          As Variant
    Dim vField            As Variant
    Dim strHeader         As String
    Dim strHeaderName     As String
    Dim strHeaderValue    As String
    intPosA = InStr(1, strMessage, vbCrLf & vbCrLf)
    If intPosA Then
        m_strHeaders = Left$(strMessage, intPosA - 1)
        m_strMessageBody = Right$(strMessage, Len(strMessage) - intPosA - 3)
        m_strMessageText = strMessage
    Else
        Err.Raise vbObjectError + 512 + 101, "CMessage.CreateFromText", _
            "Invalid message format"
    Exit Sub
End If
vHeaders = Split(m_strHeaders, vbCrLf)
For Each vField In vHeaders
    strHeader = CStr(vField)
    intPosA = InStr(1, strHeader, ":")
    If intPosA Then
        strHeaderName = LCase(Left$(strHeader, intPosA - 1))
    Else
        strHeaderName = ""
    End If
    strHeaderValue = Trim$(Right$(strHeader, Len(strHeader) - intPosA))
    Select Case strHeaderName

```

```
Case "return-path"
    m_strReturnPath = strHeaderValue
Case "received"
    m_strReceived = strHeaderValue
Case "from"
    m_strFrom = strHeaderValue
Case "sender"
    m_strSender = strHeaderValue
Case "reply-to"
    m_strReplyTo = strHeaderValue
Case "to"
    m_strMessageTo = strHeaderValue
Case "cc"
    m_strCC = strHeaderValue
Case "bcc"
    m_strBCC = strHeaderValue
Case "message-id"
    m_strMessageID = strHeaderValue
Case "in-reply-to"
    m_strInReplyTo = strHeaderValue
Case "references"
    m_strReferences = strHeaderValue
Case "keywords"
    m_strKeywords = strHeaderValue
Case "subject"
    m_strSubject = strHeaderValue
Case "comments"
    m_strComments = strHeaderValue
Case "encrypted"
    m_strEncrypted = strHeaderValue
Case "date"
    m_strSendDate = strHeaderValue
End Select
Next
End Sub
Public Function CombineMessage() As String
End Function
Public Property Let Headers(ByVal vData As String)
    m_strHeaders = vData
End Property
Public Property Get Headers() As String
    Headers = m_strHeaders
End Property
Public Property Let MessageBody(ByVal vData As String)
    m_strMessageBody = vData
```

```
End Property
Public Property Get MessageBody() As String
    MessageBody = m_strMessageBody
End Property
Public Property Let MessageText(ByVal vData As String)
    m_strMessageText = vData
End Property
Public Property Get MessageText() As String
    MessageText = m_strMessageText
End Property
Public Property Let Encrypted(ByVal vData As String)
    m_strEncrypted = vData
End Property
Public Property Get Encrypted() As String
    Encrypted = m_strEncrypted
End Property
Public Property Let Comments(ByVal vData As String)
    m_strComments = vData
End Property
Public Property Get Comments() As String
    Comments = m_strComments
End Property
Public Property Let Keywords(ByVal vData As String)
    m_strKeywords = vData
End Property
Public Property Get Keywords() As String
    Keywords = m_strKeywords
End Property
Public Property Let References(ByVal vData As String)
    m_strReferences = vData
End Property
Public Property Get References() As String
    References = m_strReferences
End Property
Public Property Let InReplyTo(ByVal vData As String)
    m_strInReplyTo = vData
End Property
Public Property Get InReplyTo() As String
    InReplyTo = m_strInReplyTo
End Property
Public Property Let BCC(ByVal vData As String)
    m_strBCC = vData
End Property
Public Property Get BCC() As String
    BCC = m_strBCC
```

```
End Property
Public Property Let CC(ByVal vData As String)
    m_strCC = vData
End Property
Public Property Get CC() As String
    CC = m_strCC
End Property
Public Property Let Sender(ByVal vData As String)
    m_strSender = vData
End Property
Public Property Get Sender() As String
    Sender = m_strSender
End Property
Public Property Let ReplyTo(ByVal vData As String)
    m_strReplyTo = vData
End Property
Public Property Get ReplyTo() As String
    ReplyTo = m_strReplyTo
End Property
Public Property Let Subject(ByVal vData As String)
    m_strSubject = vData
End Property
Public Property Get Subject() As String
    Subject = m_strSubject
End Property
Public Property Let From(ByVal vData As String)
    m_strFrom = vData
End Property
Public Property Get From() As String
    From = m_strFrom
End Property
Public Property Let MessageTo(ByVal vData As String)
    m_strMessageTo = vData
End Property
Public Property Get MessageTo() As String
    MessageTo = m_strMessageTo
End Property
Public Property Let MessageID(ByVal vData As String)
    m_strMessageID = vData
End Property
Public Property Get MessageID() As String
    MessageID = m_strMessageID
    indice = m_strMessageID
End Property
Public Property Let SendDate(ByVal vData As String)
```

```
    m_strSendDate = vData
End Property
Public Property Get SendDate() As String
    SendDate = m_strSendDate
End Property
Public Property Let Received(ByVal vData As String)
    m_strReceived = vData
End Property
Public Property Get Received() As String
    Received = m_strReceived
End Property
Public Property Let ReturnPath(ByVal vData As String)
    m_strReturnPath = vData
End Property
Public Property Get ReturnPath() As String
    ReturnPath = m_strReturnPath
End Property
Public Property Get Size() As Long
    Size = Len(m_strMessageText)
End Property
Private mCol As Collection
Public Sub Add(oMessage As CMessage, Optional sKey As String)
    If Len(sKey) = 0 Then
        mCol.Add oMessage
    Else
        mCol.Add oMessage, sKey
    End If
End Sub
Public Property Get Item(vntIndexKey As Variant) As CMessage
    Set Item = mCol(vntIndexKey)
End Property
Public Property Get Count() As Long
    Count = mCol.Count
End Property
Public Sub Remove(vntIndexKey As Variant)
    mCol.Remove vntIndexKey
End Sub
Public Property Get NewEnum() As IUnknown
    Set NewEnum = mCol.[_NewEnum]
End Property
Private Sub Class_Initialize()
    Set mCol = New Collection
End Sub
Private Sub Class_Terminate()
    Set mCol = Nothing
End Sub
```



```
End Sub
Programa paciente:
Public velocidad As Integer
Public nro_puerto As Integer
Public incluye_salto_carro As Boolean
Public cadena_entrada As Variant
Public mensajes_cambios As Boolean
Dim contador As Integer
Private Enum SMTP_State
    MAIL_CONNECT
    MAIL_HELO
    MAIL_FROM
    MAIL_RCPTTO
    MAIL_DATA
    MAIL_DOT
    MAIL_QUIT
End Enum
Private m_State As SMTP_State
Private Sub Boton_Enviar_Click()
    Boton_Enviar.Enabled = False
    Texto_Mensajes.Text = "Obteniendo..."
    Timer7.Enabled = True
    On Error GoTo Error_Enviando
    contador = 0
    Timer1.Enabled = True
    GoTo Salir
Error_Enviando:
    MsgBox "Ocurrió un error al intentar enviar el texto"
    MsgBox "Visual Basic detectó: " + Err.Description
    Texto_Mensajes.Text = "Ocurrió error al enviar"
    Resume Salir
Salir:
End Sub
Private Sub Command1_Click()
    Command1.Enabled = False
    Winsock1.Connect Trim$("mail.cantv.net"), 25
    m_State = MAIL_CONNECT
End Sub
Private Sub Form_Load()
    Command1.Enabled = False
    Timer1.Interval = 250
    Timer2.Interval = 250
    Timer3.Interval = 250
    Timer4.Interval = 250
    Timer5.Interval = 250
```

```
Timer6.Interval = 250
Timer1.Enabled = False
Timer2.Enabled = False
Timer3.Enabled = False
Timer4.Enabled = False
Timer5.Enabled = False
Timer6.Enabled = False
mensajes_cambios = False
velocidad = 300 '
nro_puerto = 9600
mensajes_cambios = True
MSComm1.InputLen = 0
MSComm1.RThreshold = 1
MSComm1.SThreshold = 1
    MsgBox "Conecte el cable del glucómetro a la PC"
Etiqu_estado.ForeColor = &HFF&
Etiqu_estado.Caption = "CERRADO"
On Error GoTo manejar_errores
    MSComm1.CommPort = 4
    MSComm1.Settings = Str$(9600) + ",n,8,1"
    MSComm1.PortOpen = True
    MsgBox ("Glucómetro detectado correctamente")
    Etiqu_puerto.Caption = "COM" + Str$(4)
    Etiqu_velocidad.Caption = Str$(9600) + " bauds"
    Etiqu_estado.ForeColor = &H8000000D 'Color azul
    Etiqu_estado.Caption = "ABIERTO"
    GoTo Salir 'Todo Ok, continuar.
manejar_errores:
MsgBox ("No conectó el cable del glucómetro, conéctelo y vuelva a abrir el
programa")
End
Resume Salir
Salir:
End Sub
Private Sub Form_Unload(Cancel As Integer)
    End
End Sub
Private Sub MSComm1_OnComm()
Dim envio As Boolean
envio = False
If MSComm1.CommEvent = comEvReceive Then
Texto_Recibido.Text = Texto_Recibido.Text + MSComm1.Input
End If
If MSComm1.CommEvent = comEvSend Then
    Texto_Mensajes = "Se recibió data del glucómetro"
```

```
End If
End Sub
Private Sub Timer1_Timer()
MSComm1.Output = Chr$(17)
Timer1.Enabled = False
Timer2.Enabled = True
End Sub
Private Sub Timer2_Timer()
MSComm1.Output = Chr$(13)
Timer2.Enabled = False
Timer3.Enabled = True
End Sub
Private Sub Timer3_Timer()
MSComm1.Output = Chr$(68)
Timer3.Enabled = False
Timer4.Enabled = True
End Sub
Private Sub Timer4_Timer()
MSComm1.Output = Chr$(77)
Timer5.Enabled = True
Timer4.Enabled = False
End Sub
Private Sub Timer5_Timer()
MSComm1.Output = Chr$(80)
Timer6.Enabled = True
Timer5.Enabled = False
End Sub
Private Sub Timer6_Timer()
MSComm1.Output = Chr$(80)
Timer6.Enabled = False
    If contador = 0 Then
        Timer1.Enabled = True
        contador = contador + 1
    End If
End Sub
Private Sub Timer7_Timer()
Command1.Enabled = True
Timer7.Enabled = False
End Sub
Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
    Dim strServerResponse As String
    Dim strResponseCode As String
    Dim strDataToSend As String
    Winsock1.GetData strServerResponse
    Debug.Print strServerResponse
```

```
strResponseCode = Left(strServerResponse, 3)
If strResponseCode = "250" Or _
strResponseCode = "220" Or _
strResponseCode = "354" Then
  Select Case m_State
    Case MAIL_CONNECT
      m_State = MAIL_HELO
      strDataToSend = Trim$("jmanuel_arenas@cantv.net")
      strDataToSend = Left$(strDataToSend, _
        InStr(1, strDataToSend, "@") - 1)
      Winsock1.SendData "HELO " & strDataToSend & vbCrLf
      Debug.Print "HELO " & strDataToSend
    Case MAIL_HELO
      m_State = MAIL_FROM
      Winsock1.SendData "MAIL FROM:" &
Trim$("jmanuel_arenas@cantv.net") & vbCrLf
      Debug.Print "MAIL FROM:" & Trim$("jmanuel_arenas@cantv.net")
    Case MAIL_FROM
      m_State = MAIL_RCPTTO
      Winsock1.SendData "RCPT TO:" & Trim$("proyectodiabetes@cantv.net")
& vbCrLf
      Debug.Print "RCPT TO:" & Trim$("proyectodiabetes@cantv.net")
    Case MAIL_RCPTTO
      m_State = MAIL_DATA
      Winsock1.SendData "DATA" & vbCrLf
      Debug.Print "DATA"
    Case MAIL_DATA
      m_State = MAIL_DOT
      Winsock1.SendData "Subject:" & "Resultado de medición de glicemia" &
vbLf
      Debug.Print "Subject:" & "Resultado de medición de glicemia"
      Dim varLines As Variant
      Dim varLine As Variant
      varLines = Split(Text1.Text + Chr$(13) + Text2.Text + Chr$(13) +
Texto_Recibido.Text, vbCrLf)
      For Each varLine In varLines
        Winsock1.SendData CStr(varLine) & vbLf
        Debug.Print CStr(varLine)
      Next
      Winsock1.SendData "." & vbCrLf
      Debug.Print "."
    Case MAIL_DOT
      m_State = MAIL_QUIT
      Winsock1.SendData "QUIT" & vbCrLf
      Debug.Print "QUIT"
```

```
Case MAIL_QUIT
    Winsock1.Close
End Select
Else
    Winsock1.Close
    If Not m_State = MAIL_QUIT Then
        MsgBox "SMTP Error: " & strServerResponse, _
            vbInformation, "SMTP Error"
    Else
        MsgBox "Información enviada al médico", vbInformation
    End If
End If
End Sub
Private Sub Winsock1_Error(ByVal Number As Integer, Description As String,
    ByVal Scode As Long, ByVal Source As String, ByVal HelpFile As String, ByVal
    HelpContext As Long, CancelDisplay As Boolean)
    MsgBox "Winsock Error number " & Number & vbCrLf & _
        Description, vbExclamation, "Winsock Error"
End Sub
```