



BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA



FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

TEMA DE TESIS

APLICACIÓN WEB PARA EL MONITOREO DE LA DIABETES

**TESIS PARA OBTENER EL TITULO DE:
LICENCIADO EN CIENCIAS DE LA COMPUTACION**

PRESENTA:

CLAUDIA HERNÁNDEZ PÉREZ

ASESOR:

DR. LUIS ENRIQUE COLMENARES GUILLÉN

PUEBLA, PUEBLA FEBRERO 2013

Dedicatoria

Adriana porque haces que el esfuerzo, la constancia y los desvelos hoy culminen en la realización de este proyecto para ti con mucho cariño.

Agradecimientos

Gracias a mi madre Amalia Hernández Pérez que aunque ya no pudo ver realizado este sueño donde quiera que esté espero se sienta orgullosa y sepa que sus sacrificios valieron la pena.

A mis hermanos José, Aurora y Alfredo, a mis primos que siempre me han brindado su apoyo y comprensión.

A mis profesores del diplomado y en especial a mi asesor Doctor Luis Enrique Colmenares Guillén que me dedicaron su tiempo y conocimientos para llevar a cabo esta tesis.

ÍNDICE GENERAL

INTRODUCCIÓN.....	1
Antecedentes	1
¿Qué es la diabetes?.....	1
Tipos de diabetes	2
Tipo 1.....	2
Tipo 2.....	4
Tipo 3: Gestacional.....	5
Problemática	5
Planteamiento del problema.....	6
Objetivos.....	6
Objetivo general	6
Objetivos específicos	6
Requerimientos funcionales.....	6
CAPÍTULO 1: TRABAJO RELACIONADO	7
1.1. Programa de acompañamiento Sophia.....	7
1.2. Software para el control de diabetes	7
1.3. Software Diabetes.....	8
CAPÍTULO 2: METODOLOGÍA	10
2.1. Ingeniería del sistema y diseño de software.....	10
2.1.1. Metodología RUP (Proceso Unificado Rational).....	10
2.1.2. Características esenciales.....	10
2.1.3. Ciclo de vida.....	10
2.1.4. UML.....	11
2.1.5. Modelo de caso de uso	12
2.1.6. Documentación de casos de uso	14
2.1.7. Diccionario del modelo	15
2.1.8. Diagramas de secuencia.....	15
2.1.9. Diagramas de colaboración.....	18
2.1.10. Glosario	19
2.2. Base de datos.....	20
2.2.1. Introducción	20
2.2.2. Definición de base de datos	20
2.2.3. DBMS	20
2.2.4. Tipos de usuario.....	21
2.2.5. Modelo de datos.....	21
2.2.6. Base de datos relacional	22
2.3. Normalización.....	22
2.3.1. Definición de normalización.....	22

2.3.2. Formas normales	23
2.3.3. Primera Forma Normal	23
2.3.4. Segunda Forma Normal	23
2.3.5. Tercera Forma Normal	23
2.3.6. Cuarta Forma Normal	24
2.4. Normalización de las tablas.....	25
2.5. Entidades y atributos de las tablas.....	26
2.6. Modelo Entidad-relación (E-R)	29
CAPÍTULO 3: PROPUESTA DEL SISTEMA	31
3.1. Implementación	31
3.2. Arquitectura cliente-servidor.....	32
3.3. Código abierto	33
3.3.1. Apache	35
3.3.2. PHP.....	35
3.3.3. XAMP. LAMP, WAMP, MAMP	36
3.3.4. MySQL	37
CAPÍTULO 4: EVALUACIÓN Y PRUEBAS DEL SISTEMA .	38
4.1. Administrador	38
4.2. Doctor.....	40
4.3. Enfermera.....	42
4.4. Paciente	43
Conclusiones y Trabajo a futuro.....	45
Bibliografía.	46

ÍNDICE DE FIGURAS

Figura 1. Comportamiento de la insulina	1
Figura 2. Insulina y las células beta	3
Figura 3. Zonas del cuerpo afectadas por la diabetes.....	3
Figura 4. Diabetes Tipo 2	4
Figura 5. Causas de mortalidad en México	5
Figura 6. SiDiary y los datos de captura.....	8
Figura 7. Pilot	8
Figura 8. Metodología RUP	10
Figura 9. Caso de uso Administrador	12
Figura 10. Caso de uso Doctor.....	12
Figura 11. Caso de uso Enfermera	13
Figura 12. Caso de uso Paciente	13
Figura 13. Caso de uso monitoreo de la diabetes	14
Figura 14. Diagrama de secuencia: Validar usuario.....	16
Figura 15. Diagrama de secuencia: Alta doctor.....	17
Figura 16. Diagrama de secuencia: Enfermera actualiza sus datos.....	18
Figura 17. Diagramas de colaboración.....	19
Figura 18. Pasos del proceso de la normalización	25
Figura 19. Modelo Entidad-Relación	30
Figura 20. Diagrama funcional del sistema	32
Figura 21. Arquitectura cliente - servidor.....	33
Figura 22. Software libre - Código abierto.....	34
Figura 23. Pantalla inicial	38
Figura 24. Pantalla Administrador	38
Figura 25. Pantalla Inserta usuario paciente	39
Figura 26. Pantalla Datos.....	39
Figura 27. Pantalla Historial	40
Figura 28. Pantalla Pacientes-Doctor.....	41
Figura 29. Pantalla Recetar.....	41
Figura 30. Pantalla Paciente-Enfermera	42

Figura 31. Pantalla Información e historial	42
Figura 32. Pantalla Inicio-Paciente.....	43
Figura 33. Pantalla Medicación	43
Figura 34. Pantalla Actualizar lecturas	44

ÍNDICE DE TABLAS

Tabla 1: Doctor.....	26
Tabla 2: Paciente.	26
Tabla 3: Enfermera.....	27
Tabla 4: Entidades.	29

INTRODUCCIÓN

En este capítulo se establecen los antecedentes generales de la diabetes ya que su tendencia a crecer provoca miles de muertes por año.

Antecedentes

Desde 3000 años A.C. se tiene antecedentes de la diabetes y sus síntomas en lugares como China, India, Egipto, Grecia y Roma. Tiempos en los cuales se desconocía la enfermedad y sus causas; pero cuya sintomatología fue lo que más llamó la atención como ganas de orinar, hambre y sed constantes, mismas que eran presentadas en su mayoría por personas comilonas y ricas. El nombre de diabetes viene de los sabios griegos Apolonio de Mileto y Demetrio de Apamea; dicho nombre procede de la palabra diabinex que significa “pasar a través de”.

Era tanta la sed que producía esta enfermedad que a quienes la padecían, se les secaba la boca y el cuerpo, además de que adelgazaban, se desesperaban y morían prematuramente, pues se les sumaban otras enfermedades infecciosas como la tuberculosis o gangrena en los pies.

¿Qué es la diabetes?

La Diabetes es un trastorno del metabolismo, es decir de la forma en que el cuerpo emplea los alimentos digeridos como fuente de energía y crecimiento. En condiciones normales, el aparato digestivo convierte una porción de los alimentos que ingerimos en un azúcar, llamado glucosa. Ésta pasa a la sangre lista para alimentar a las células.

La insulina es una hormona producida por el páncreas para controlar el azúcar en la sangre. La diabetes puede ser causada por muy poca producción de insulina, resistencia a ésta o a ambas.

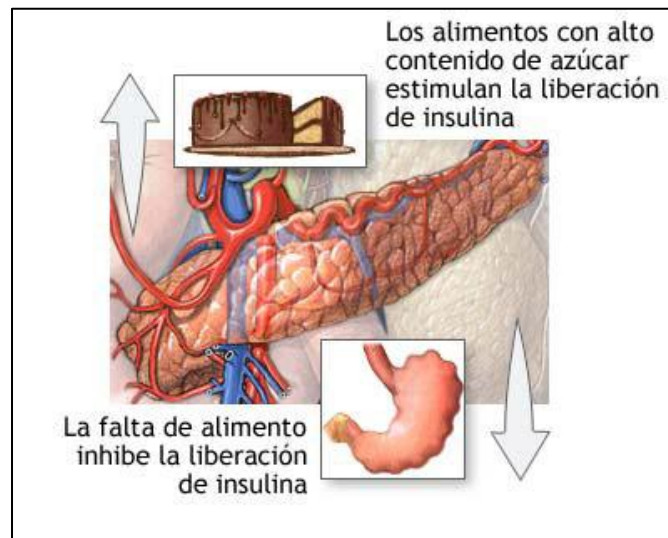


Figura 1. Comportamiento de la Insulina.

Para comprender la diabetes, es importante entender primero el proceso normal por medio del cual, el alimento se descompone y es empleado por el cuerpo para obtener energía. Varias cosas suceden cuando se digiere el alimento:

- Un azúcar llamado glucosa, que es fuente de energía para el cuerpo, entra en el torrente sanguíneo.
- Un órgano llamado páncreas produce la insulina, cuyo papel es transportar la glucosa del torrente sanguíneo hasta los músculos, la grasa y las células hepáticas, donde puede utilizarse como energía.

Las personas con diabetes presentan hiperglucemia, debido a que su cuerpo no puede movilizar el azúcar hasta los adipocitos, hepatocitos y células musculares para almacenarse como energía. Esto se debe a que:

- El páncreas no produce suficiente insulina.
- Sus células no responden de manera normal a la insulina.

Tipos de diabetes

La Organización Mundial de la Salud reconoce tres formas de diabetes mellitus: tipo 1, tipo 2 y diabetes gestacional, cada una con diferentes causas y factores de riesgo.

Tipo 1

Antiguamente llamada Diabetes Insulino, ahora insulino dependiente o diabetes de comienzo juvenil. Se presenta en jóvenes y en adultos también pero con menos frecuencia. En esta enfermedad, el cuerpo no produce o produce poca insulina y se necesitan inyecciones diarias de esta hormona. La causa exacta se desconoce. El diagnóstico es alrededor de los 25 años de edad, y afecta a cerca de 4,9 millones de personas en todo el mundo.

La diabetes tipo 1 se presenta cuando las defensas del propio organismo (sistema inmune) destruyen las células beta del páncreas. Las células beta del páncreas son la "fábrica" donde se produce la insulina. La insulina se necesita para movilizar el azúcar de la sangre (glucosa) hasta las células, donde se almacena y se utiliza después para obtener energía. En la diabetes tipo 1, estas células producen poca o ninguna insulina. Sin la insulina suficiente, la glucosa se acumula en el torrente sanguíneo en lugar de entrar en las células y el cuerpo es incapaz de usarla para obtener energía.

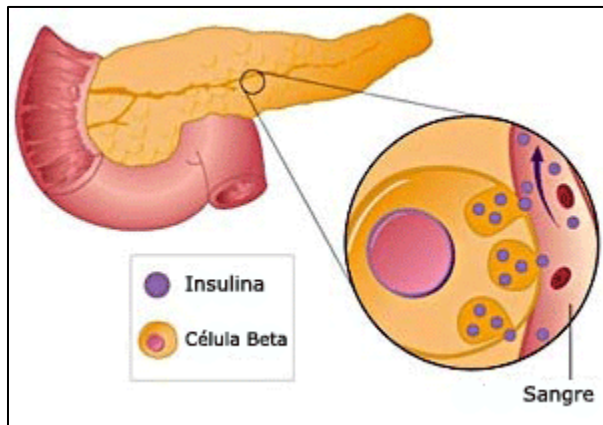


Figura 2. Insulina y las células beta.

Para el año 2000, se estimó que alrededor de 171 millones de personas eran diabéticas en el mundo y que llegarán a 370 millones en 2030. Este padecimiento causa diversas complicaciones, dañando frecuentemente a ojos, riñones, nervios y vasos sanguíneos.

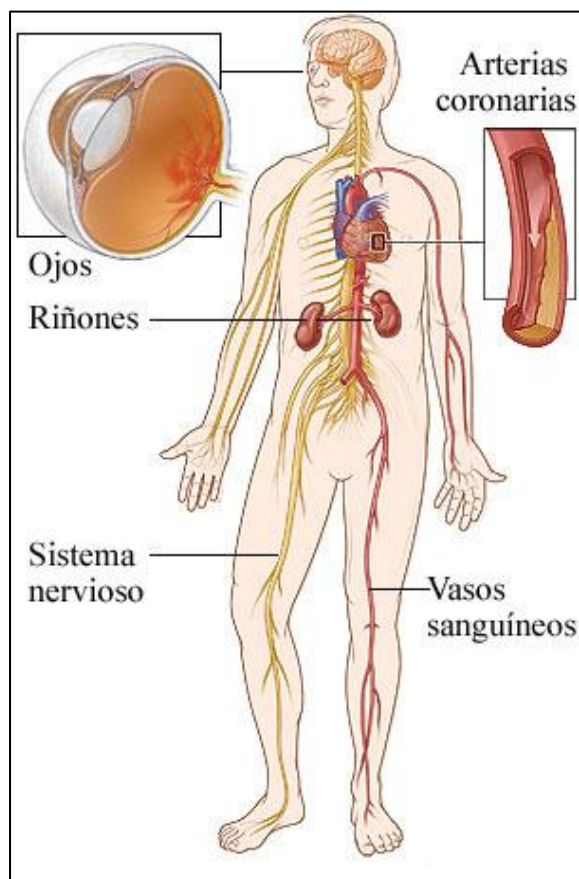


Figura 3. Zonas del cuerpo afectadas por la diabetes

Tipo 2

Representa más del 90% de todos los casos de diabetes. En Diabetes tipo 2, puede que el páncreas produzca suficiente insulina, pero el cuerpo no puede utilizarla adecuadamente. Esto se conoce como resistencia a la insulina. También es posible que en algún momento el páncreas deje por completo de producir insulina, esto causa la elevación del nivel de azúcar y obliga al páncreas a producir aún más insulina. Eventualmente, el páncreas puede colapsar por el esfuerzo extraordinario de producir insulina extra. Entonces, el páncreas posiblemente ya no tendrá la capacidad para producir suficiente insulina y mantener los niveles de azúcar dentro de lo normal. Suele aparecer en adultos, pero puede afectar a personas de cualquier edad.

Por lo general, la diabetes tipo 2 se desarrolla lentamente con el tiempo. La mayoría de las personas con esta enfermedad tienen sobrepeso en el momento del diagnóstico. El aumento de la grasa le dificulta al cuerpo el uso de la insulina de la manera correcta. La diabetes tipo 2 puede presentarse también en personas delgadas y es más común en los ancianos. Los antecedentes familiares y los genes juegan un papel importante. Un bajo nivel de actividad, una dieta deficiente y el peso corporal excesivo (especialmente alrededor de la cintura) aumentan el riesgo [1].

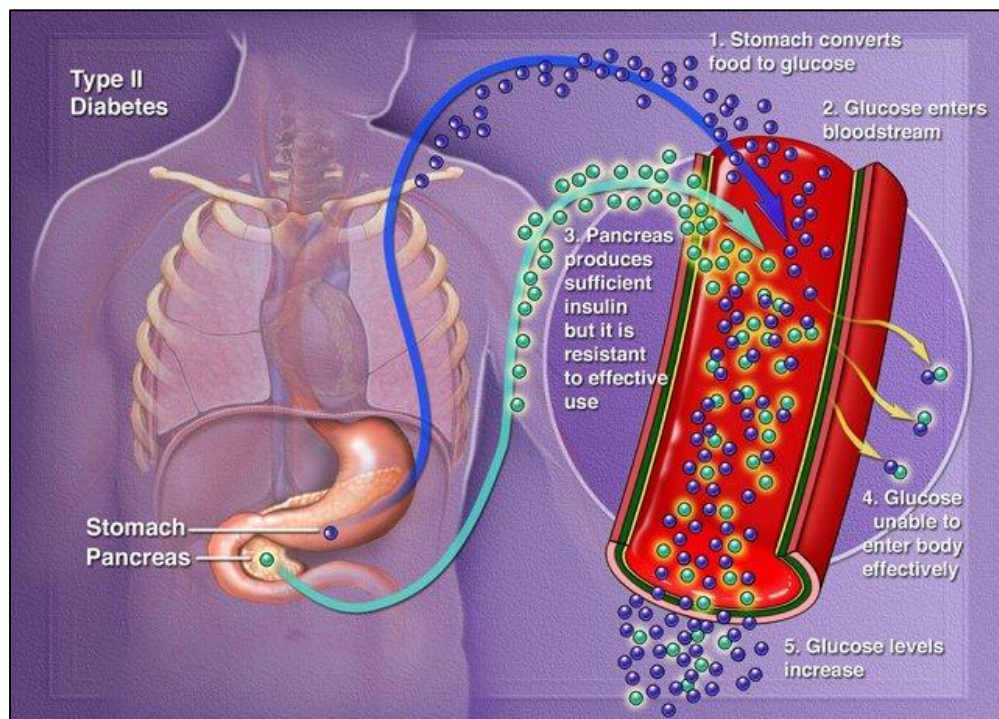
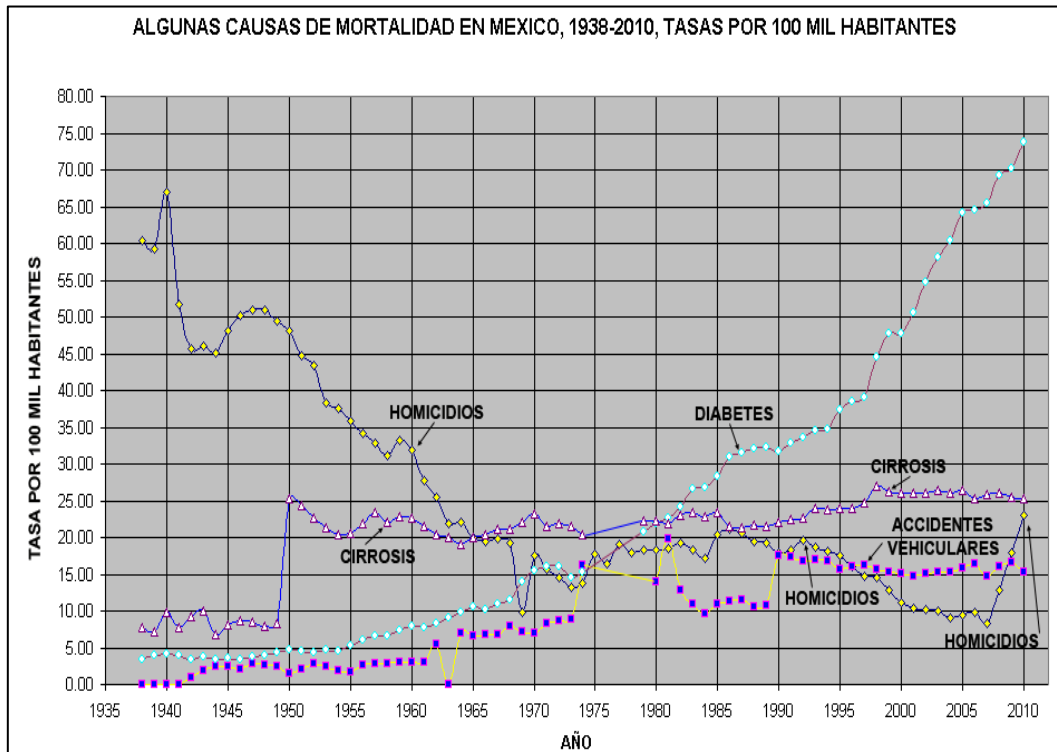


Figura 4. Diabetes Tipo 2

Tipo 3: Diabetes gestacional

Es un tipo de Diabetes que se presenta solamente durante el embarazo, y se denomina de esta manera cuando se detectan cifras elevadas de glucosa POR PRIMERA VEZ en una mujer embarazada. Normalmente este tipo de Diabetes se detecta entre las semanas 24 y 28 del embarazo y en la gran mayoría de los casos, desaparece en cuanto se presenta el parto.



Actualización de Marzo de 2012

Figura 5. Causas de mortalidad en México.

La Diabetes Mellitus de acuerdo con la OMS, Organización Mundial de la Salud, es causa de más de 220 millones de personas enfermas y se estimó que en 2008 fallecieron 1.26 millones de personas como consecuencias directas del exceso de azúcar en la sangre [2].

Problemática

En México, la diabetes sin control es responsable de setenta mil muertes al año y es la primera causa de ceguera, amputación y uso de diálisis por insuficiencia renal, así como una de las principales causas de baja productividad laboral debido a las complicaciones crónicas como: neuropatía, ceguera, insuficiencia renal, pie diabético (que puede llevar inclusive a la amputación de extremidades), infarto del miocardio, accidente vascular cerebral, mayor frecuencia de cáncer y muerte prematura.

Planteamiento del problema

- El paciente en ocasiones olvida tomar sus medicamentos.
- El doctor no tiene un control específico sobre el tipo de diabetes de un paciente.
- El doctor no tiene actualizada la información sobre los índices de glucosa, presión arterial o insulina de los pacientes.
- El doctor y la enfermera sólo tiene contacto con el paciente en citas médicas de la institución de salud.

Objetivos

Desarrollar una aplicación en la World Wide Web (WWW) para monitorear el estado clínico de pacientes con diabetes.

Objetivo general

Crear una base de datos con información personal y clínica de pacientes para establecer un contacto más directo con su doctor y lograr monitorear el estado de salud actualizado de pacientes con diabetes.

Objetivos específicos:

- Crear un sistema que administre los datos.
- El usuario podrá ingresar al sistema con Nombre de Usuario y Password.
- El sistema validará al usuario según el nivel (administrador, doctor, enfermera o paciente) para iniciar una sesión.
- El sistema podrá ingresar, modificar, eliminar y actualizar los registros de usuarios.
- Se podrá tener acceso al sistema desde cualquier dispositivo con acceso a Internet.

Requerimientos funcionales:

El sistema manejará una base de datos que incluya:

- Tipo de usuario (Administrador, Doctor, Enfermera o Paciente).
- Datos personales de los usuarios.
- Datos clínicos de pacientes.

El contenido de esta tesina se presenta de la siguiente manera:

El Capítulo 1, es el trabajo relacionado con aplicaciones que sirven de apoyo a pacientes con diabetes, darle seguimiento y llevar un control personal. En el capítulo 2 se hace el análisis y diseño del sistema creando modelos y diagramas de casos de uso, de secuencia y de colaboración. Además se diseña la base de datos, se crean las tablas, se explica la normalización y relación entre ellas. El capítulo 3 contiene la teoría básica de ingeniería de software que se va a aplicar en el desarrollo del sistema y algunos aspectos del lenguaje utilizado. El capítulo 4 muestra pantallas del funcionamiento del sistema. Y al final se realizan conclusiones y trabajo a futuro del sistema realizado.

CAPÍTULO 1: TRABAJO RELACIONADO

Breve descripción de los elementos y dispositivos que permiten llevar un control clínico de pacientes diabéticos.

1.1. Programa de acompañamiento Sophia

La Assurance-Maladie, el servicio médico francés, lanzó hace tres años una iniciativa –el programa de acompañamiento Sophia – para hacer un seguimiento telefónico a las persona con Diabetes para poder gestionar las rutinas de la enfermedad y poder llevar un buen control de su Diabetes. 100,000 pacientes inscritos han recibido periódicamente llamadas telefónicas de un equipo de 78 consejeros en salud que han complementado el trabajo de sus médicos y les han ayudado a resolver dudas, además de escuchar sus problemas y ofrecerles soluciones. Los consejeros también les ayudaron a establecer objetivos y a adherirse al tratamiento. A estas llamadas telefónicas de seguimiento, se añadieron también materiales bibliográficos de consulta, como guías y libros temáticos. El resultado ha sido muy satisfactorio. En el grupo de referencia que se tomó para evaluar la acción se ha detectado que los pacientes seguían mejor las indicaciones médicas, se hacían más puntualmente las diversas pruebas de control de complicaciones (revisiones oftalmológica, electrocardiogramas, etc.) y lo más importante, el índice de hemoglobina glicosilada de los participantes en el grupo, bajo más que la media del resto de la población diabética [3].

1.2. Software para el control de diabetes

Existen en la WWW algunos programas gratuitos o de prueba como SiDiary, es un programa informático fácil de usar, muy intuitivo y útil para personas diabéticas. Les ayuda a llevar el control de los niveles de glucosa, de lo que comen y de las horas a las que deben tomar la medicación. Lo creó Fernando Rafart, informático de profesión, cuando le diagnosticaron la diabetes a los 25 años.

Cada vez que la persona diabética chequea su nivel de glucosa o se inyecta la dosis de insulina, debe introducir los datos en el programa. El software incorpora una base de datos de alimentos capaz de calcular los efectos de cada comida en los niveles de glucosa y el gasto de energía si se practica deporte. También se pueden introducir el peso, la presión arterial y el pulso. Con todos estos datos, el programa puede hacer análisis estadísticos que permiten al usuario que visualice rápidamente cuál está siendo su evolución. Ante cualquier valor que se salga de la normalidad, el sistema avisa al usuario o bien informa a los padres vía SMS, una opción muy útil si el enfermo es un niño. Además, otra opción permite enviar los datos al médico por correo electrónico, de manera que éste puede tener un seguimiento a tiempo real de la evolución de su paciente.

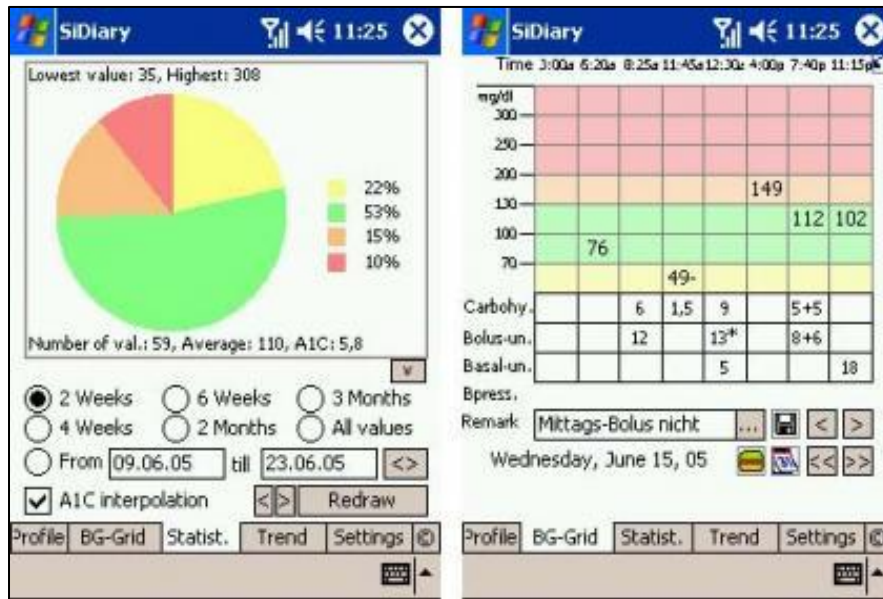


Figura 6. SiDiary y los datos de captura.

El programa es gratis el primer mes y luego tiene un coste entre los 15 y 40 euros, según la versión que se elija. Se puede instalar en ordenadores portátiles y de sobremesa, agendas electrónicas, teléfonos móviles y si se utiliza en varios dispositivos, se puede sincronizar [4].

1.3. Software Diabetes



Figura 7. Pilot.

Diabetes Pilot [™] es un software que hace más fácil el control de la diabetes, toma el lugar de cuadernos de papel tradicionales, pero también hace mucho más:

- Almacena los niveles de glucosa, insulina y otros medicamentos, comidas, ejercicio, presión arterial, resultados de pruebas, y otras notas.
- Ayuda a ver las tendencias en los niveles de azúcar en sangre, medicamentos, dieta y otras áreas con diferentes informes y gráficos.
- Le permite transferir los datos de varias formas para su posterior análisis y la comunicación con su equipo de atención de la salud [5].

CAPÍTULO 2: METODOLOGÍA

Se plantea la problemática que se pretende resolver y las posibles hipótesis, métodos y herramientas. Además se muestra a detalle el diseño del sistema, el modelo relacional y la normalización del mismo.

2.1. Ingeniería del sistema y diseño de software

2.1.1. Metodología RUP (Proceso Unificado de Rational)

Es un proceso utilizado para el desarrollo de software, que define quien, como, cuando, y que debe hacerse en el proyecto. Aplica varias de las mejores prácticas en el desarrollo moderno de software en una forma que se adapta a un amplio rango de proyectos y organizaciones. Su principal objetivo es asegurar la producción de software de alta calidad, que cumpla las necesidades de sus usuarios finales, que sea realizado en las fechas acordadas y con el presupuesto disponible [6].

2.1.2. Características esenciales

- Forma disciplinada de asignar tareas y responsabilidades.
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo.
- Administración de requisitos.
- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software.

2.1.3. Ciclo de Vida

Esta metodología cuenta con 4 fases:



Figura 8. Metodología RUP.

- **Fase de Inicio:** Tiene como propósito definir y acordar el alcance del proyecto, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones posteriores.
- **Fase de elaboración:** Se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.
- **Fase de construcción:** El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requisitos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto.
- **Fase de transición:** El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto.

En cada etapa y en cada iteración se perfeccionan los productos previos que hayan requerido algún cambio, todo eso mientras se intentan conseguir los objetivos concretos de la fase. De esta forma el ciclo de vida RUP sigue un modelo adaptativo de desarrollo de software [7].

2.1.4. UML

UML, por sus siglas en inglés, Unified Modeling Language, es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad, es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

Nació en 1994 cubriendo los aspectos principales de todos los métodos de diseño antecesores y, precisamente, los padres de UML son Grady Booch, autor del método Booch; James Rumbaugh, autor del método OMT e Ivar Jacobson, autor de los métodos OOSE y Objectory. La versión 1.0 de UML fue liberada en Enero de 1997 y ha sido utilizado con éxito en sistemas construidos para toda clase de industrias alrededor del mundo: hospitales, bancos, comunicaciones, aeronáutica, finanzas, entre otras.

UML describe un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

2.1.5. Modelo de casos de uso

El diagrama de casos de uso representa la forma en, como un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso) [8].

Un diagrama de casos de uso consta de los siguientes elementos:

- Actor.
- Casos de Uso.
- Relaciones de Uso.

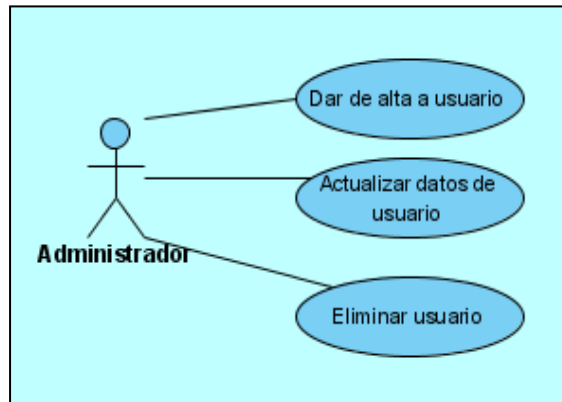


Figura 9. Caso de uso Administrador.

El actor administrador será el encargado de insertar, eliminar y modificar datos en las tablas.

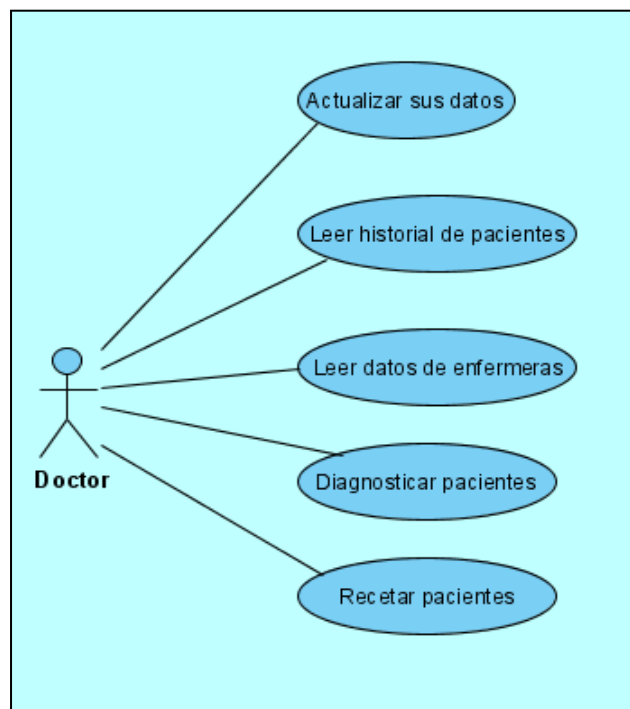


Figura 10. Caso de uso Doctor.

El usuario doctor podrá leer los datos de sus pacientes una vez que haya sido validado por el sistema. El doctor podrá actualizar sus datos como nombre, teléfono personal y del trabajo en caso de que cambie de lugar de trabajo o de teléfono personal.

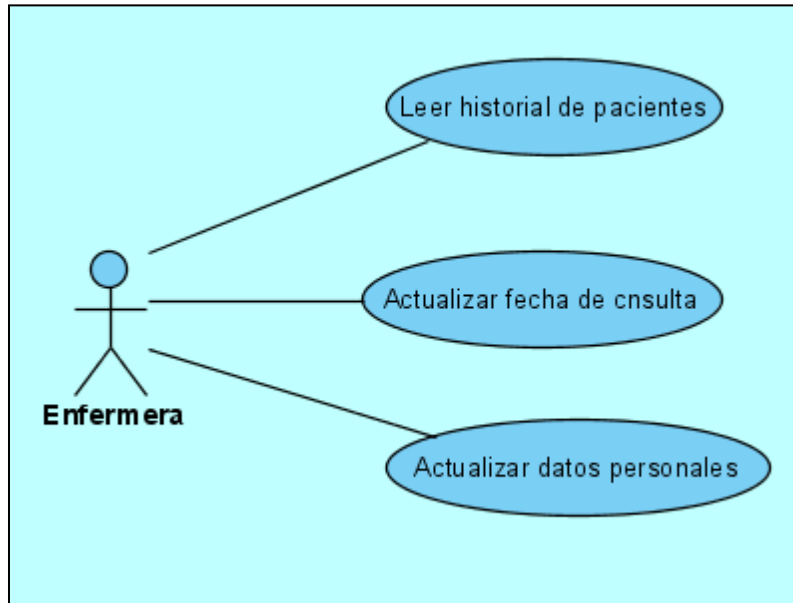


Figura 11. Caso de uso Enfermera.

Las operaciones de enfermera serán las de leer el historial de los pacientes de los cuales es responsable, cambiar la fecha de la próxima cita del paciente y modificar su teléfono personal; en caso de que cambie de trabajo este dato también deberá actualizarse en la base de datos y el doctor pueda leerlos en su sesión.

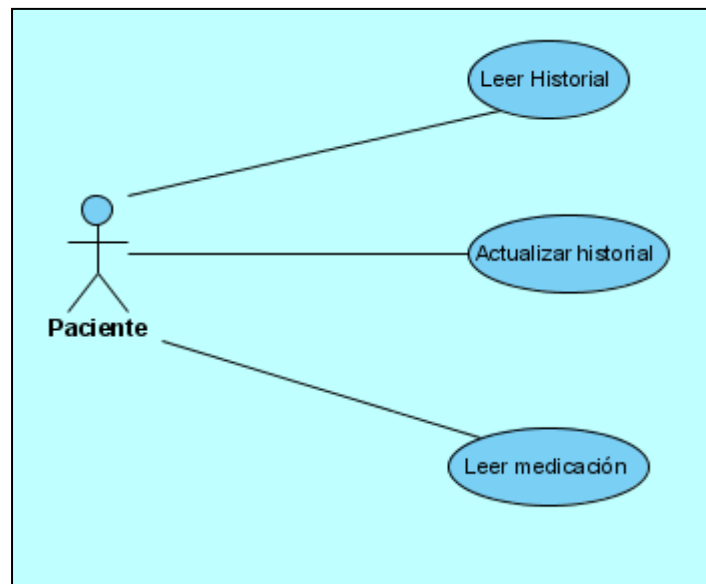


Figura 12. Caso de uso Paciente.

Las operaciones de usuario paciente serán leer historial, donde el paciente podrá leer en una pantalla sus datos personales e historial clínico; también podrá leer su medicación y escribir los datos más recientes respecto a sus niveles de glucosa y presión arterial para mantener actualizada la base de datos y esta información sea revisada por el actor doctor y enfermera .

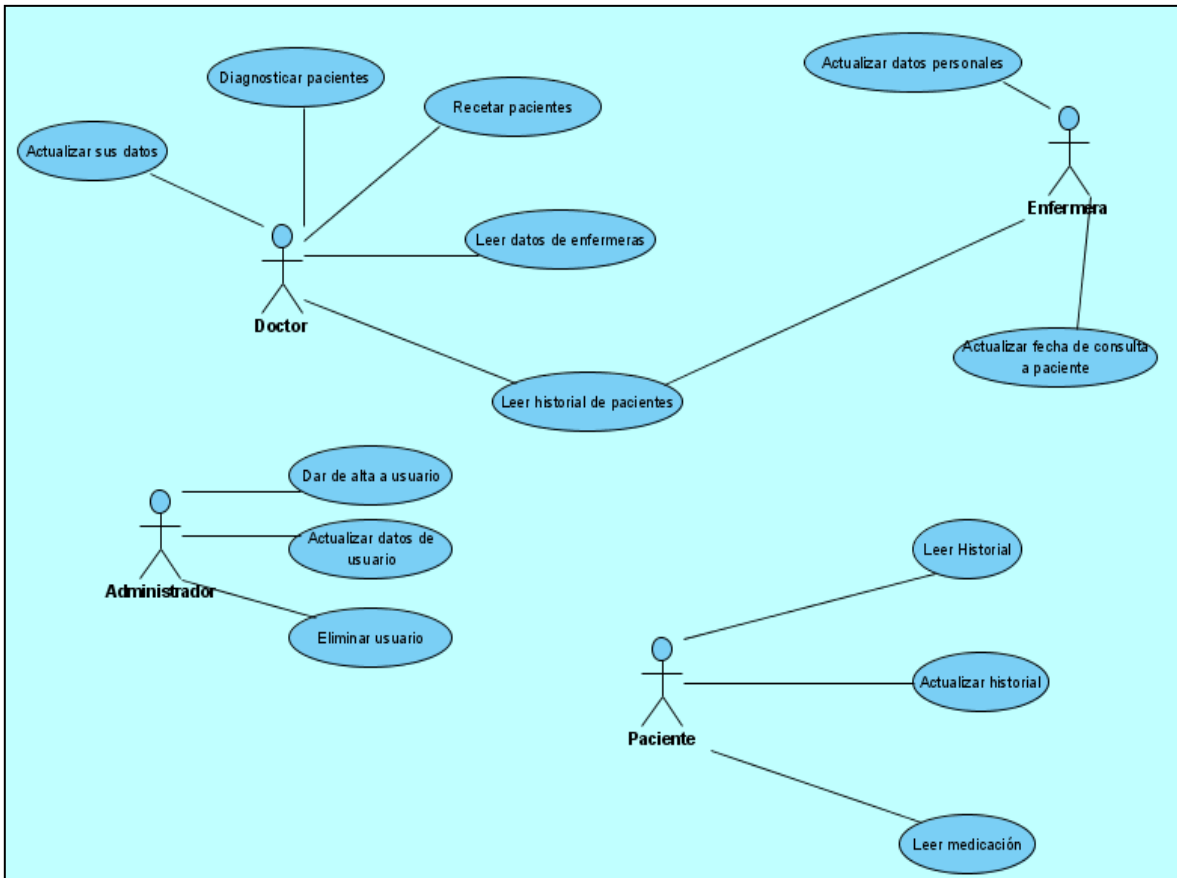


Figura 13. Sistema monitoreo de la diabetes

En el diagrama se identifican a los actores y los roles que representan, todos ellos al ingresar al sistema deben ser validados para efectuar las funciones que les corresponden.

2.1.6. Documentación de casos de uso

Caso de uso: **Validar.**

Actor: Paciente y sistema.

Precondiciones: El actor debe estar registrado en el sistema.

Flujo básico:

- El sistema muestra la pantalla principal.
- El actor escribe su id y su Password.
- Termina Caso de uso.

Flujo de excepción:

- El actor no escribe correctamente el Id o el password, entonces se muestra en pantalla el mensaje de regresar a la pantalla de principal.
- El actor no existe, el sistema informa con un mensaje de error.

Caso de uso: Actualizar historial.

Actores: Paciente y sistema.

Precondiciones: El usuario debe estar registrado en el sistema.

Flujo básico:

- El sistema muestra la pantalla principal.
- El actor escribe su Id y Password.
- El sistema valida al actor.
- El sistema muestra una pantalla con los datos clínicos del actor.
- El actor escribe los nuevos datos en los campos necesarios y elige la opción Actualizar.
- Se guardan los cambios.
- Termina el Caso de uso.

Flujo de excepción:

- El actor no escribe correctamente el Id o password, entonces se mostrará un mensaje de error.
- El actor presiona el botón Reset, entonces no se hacen cambios en el registro.

2.1.7. Diccionario del modelo

Nombre: Administrador.

Definición de Trabajo: Sistema de base de datos que llevará a cabo tareas como insertar, actualizar y eliminar registros.

Nombre: Paciente.

Definición de Trabajo: Información acerca de una persona registrada para realizar su control clínico.

Nombre: Doctor.

Definición de Trabajo: Información acerca de una persona registrada que hará el control clínico de un paciente.

Nombre: Enfermera.

Definición de Trabajo: Información acerca de una persona registrada que verificará el estado clínico de un paciente.

2.1.8. Diagramas de secuencia

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. El diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados entre los objetos.

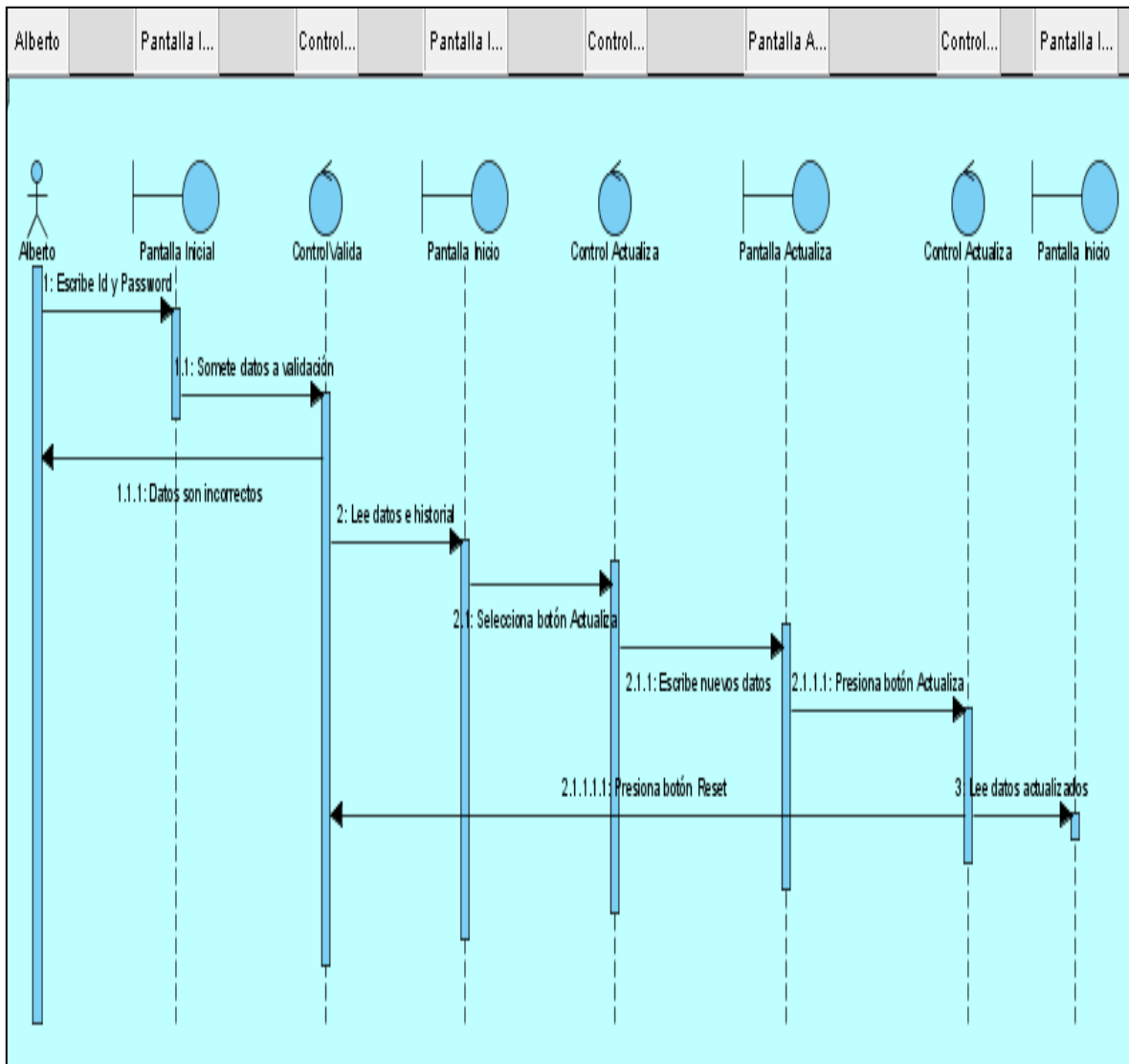


Figura 14. Diagrama de secuencia Validar usuario.

Escenario: El paciente Alberto actualiza sus datos.

- El paciente Alberto ingresa a la pantalla inicial y escribe su Id y password.
- El sistema valida al actor.
- El sistema muestra la pantalla de sesión Paciente con opciones.
- El actor elige Actualizar.
- El sistema muestra un formulario con los campos a modificar.
- El actor escribe los nuevos datos y presiona el botón Actualizar.
- El sistema actualiza el registro del actor.
- Termina el Caso de Uso.

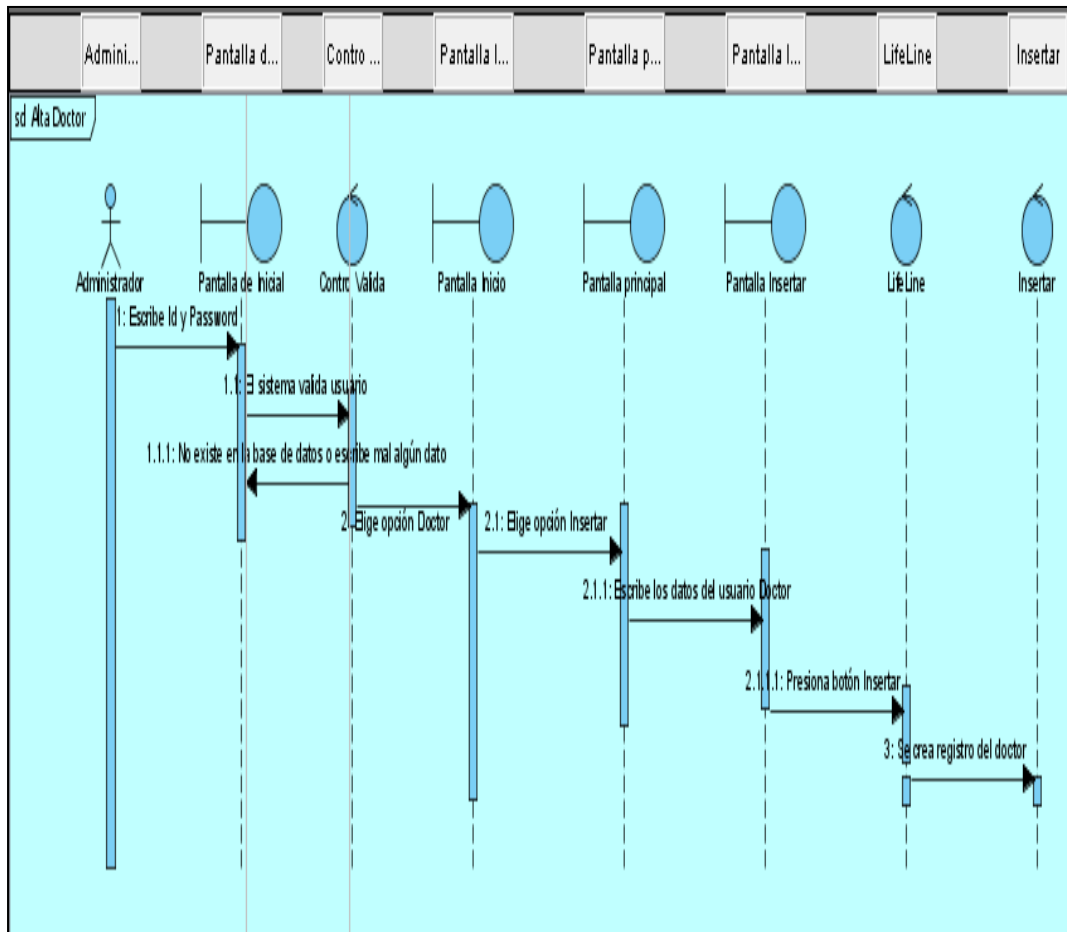


Figura 15. Diagrama de secuencia Alta Doctor.

Escenario: El administrador da de alta en el sistema al doctor Roberto.

- El actor administrador escribe en la pantalla inicial su Id y password.
- El sistema valida al actor.
- El sistema muestra la pantalla de sesión Administrador con opciones.
- El actor elige la opción Doctor.
- El sistema muestra un formulario a llenar con los datos del doctor.
- El actor presiona el botón Insertar.
- El sistema da de alta el registro del actor.
- Termina el Caso de Uso.

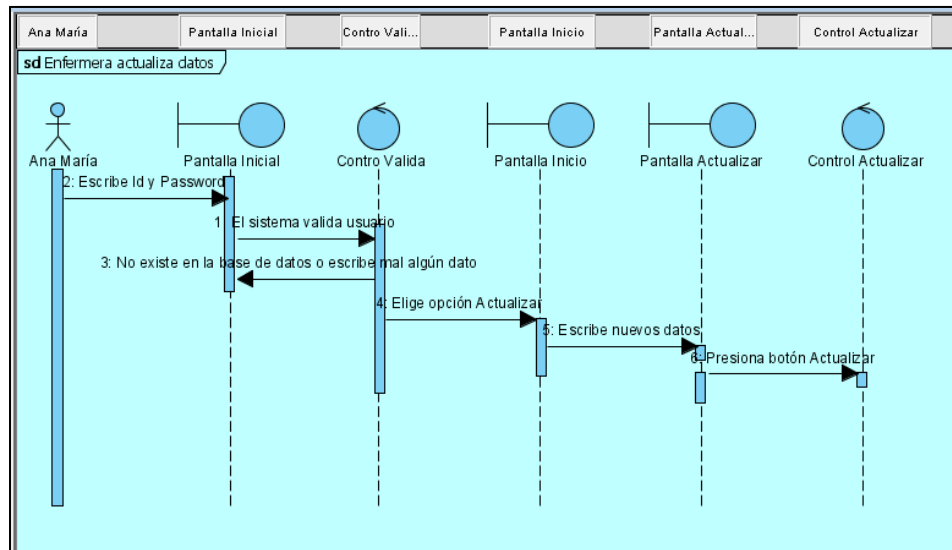


Figura 16. Diagrama de secuencia: Enfermera actualiza sus datos.

Escenario: La enfermera Ana María actualiza sus datos.

- La Enfermera Ana María escribe en la pantalla inicial su Id y password.
- El sistema valida al actor.
- El sistema muestra la pantalla de sesión Enfermera con los datos de su cuenta.
- El actor elige la opción Actualizar.
- El sistema muestra un formulario a llenar.
- El actor presiona el botón Actualizar.
- El sistema actualiza el registro del actor.
- Termina el Caso de Uso.

2.1.9. Diagramas de colaboración

Un diagrama de colaboración en las versiones de UML 1.x es esencialmente un diagrama que muestra interacciones organizadas alrededor de los roles. A diferencia de los diagramas de secuencia, los diagramas de colaboración, también llamados diagramas de comunicación, muestran explícitamente las relaciones de los roles. Por otra parte, un diagrama de comunicación no muestra el tiempo como una dimensión aparte, por lo que resulta necesario etiquetar con números de secuencia tanto la secuencia de mensajes como los hilos concurrentes [8].

- Muestra cómo las instancias específicas de las clases trabajan juntas para conseguir un objetivo común.
- Implementa las asociaciones del diagrama de clases mediante el paso de mensajes de un objeto a otro. Dicha implementación es llamada "enlace".

Un diagrama de comunicación es también un diagrama de clases que contiene roles de clasificador y roles de asociación en lugar de sólo clasificadores y asociaciones. Los roles de clasificador y los de asociación describen la

configuración de los objetos y de los enlaces que pueden ocurrir cuando se ejecuta una instancia de la comunicación. Cuando se instancia una comunicación, los objetos están ligados a los roles de clasificador y los enlaces a los roles de asociación. El rol de asociación puede ser desempeñado por varios tipos de enlaces temporales, tales como argumentos de procedimiento o variables locales del procedimiento. Los símbolos de enlace pueden llevar estereotipos para indicar enlaces temporales.

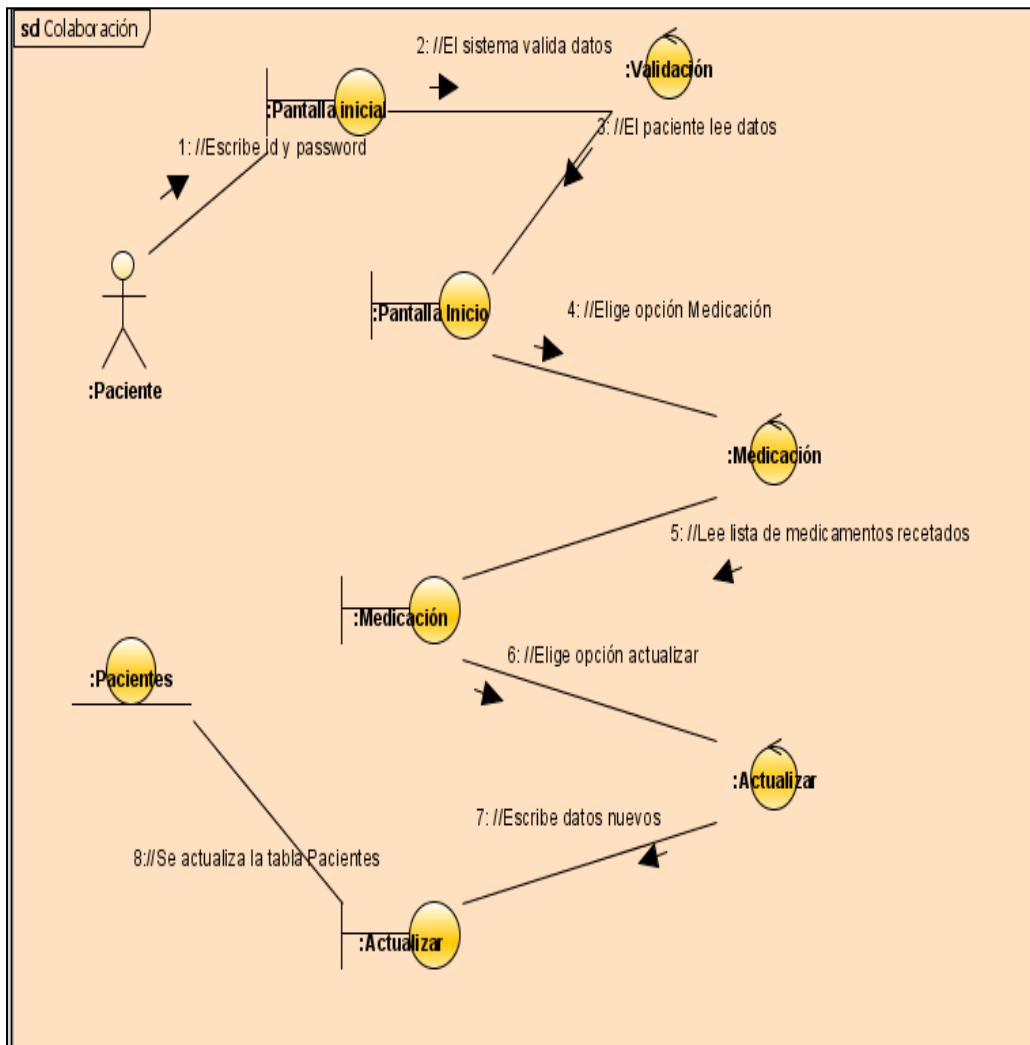


Figura 17. Diagrama de colaboración: Paciente lee medicación y actualiza historial.

2.1.10. Glosario

Actor.- representa cualquier cosa que interactúa con el sistema, puede ser una persona, una máquina o un sistema que a su vez representan un rol (administrador, usuario, banco).

Base de datos.- es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una

biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

Caso de uso.- Diagrama que representa la forma como interactúan los actores con el sistema.

Doctor.- Persona que interactúa con el sistema para hacer un control clínico de pacientes.

Enfermera.- Persona encargada de monitorear el estado clínico del paciente.

Formulario.- Pantalla donde se capturan los datos.

Paciente.- Persona con riesgo de tener diabetes o que ya la tiene a la que se le apoyará en el diagnóstico y control.

2.2. Base de datos

2.2.1. Introducción

Desde la antigüedad el hombre ha tratado de mantener la información ordenada y almacenada en archivos de papel para poder explotarla de una forma potencial, con la aparición de la computación este modo de mantener la información almacenada y ordenada ha cambiado notablemente, con la ayuda del equipo de computo es más fácil mantener esta información, más aun con la aparición de distintos modelos de datos no solo el mantener la información almacenada sino también explotarla de una forma más eficiente [9].

2.2.2. Definición de Base de Datos.

Una base de datos consiste en una colección de datos relacionados entre sí, que contiene información acerca de una empresa determinada. Toda base de datos se diseña, se construye y prueba con datos para un propósito específico. Esta dirigida a un grupo de usuarios y tiene ciertas aplicaciones preconcebidas que interesan a dichos usuarios. Las ventajas de utilizar un sistema de bases de datos, es que una base de datos es compacta, rápida, menos laboriosa y actual. Ofreciendo a la empresa un control centralizado de la información [9].

Una base de datos relacional consiste en un conjunto de tablas, a cada una de las cuales se le asigna un nombre exclusivo. Cada fila de la tabla representa una relación entre un conjunto de valores. El usuario del sistema de bases de datos puede consultar esas tablas, insertar nuevas tuplas, borrar y modificar las tuplas. Las operaciones más importantes y básicas de una base de datos son:

- Insertar datos.
- Actualizar datos.
- Recuperar datos.
- Borrar datos.

2.2.3. DBMS

Los sistemas de gestión de bases de datos (database management system, DBMS) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

Existen distintos objetivos que deben cumplir los DBMS:

- **Abstracción de la información.** Los DBMS ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.
- **Independencia.** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Consistencia.** En aquellos casos en los que no se ha logrado eliminar la redundancia, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
- **Seguridad.** La información almacenada en una base de datos puede llegar a tener un gran valor. Los DBMS deben garantizar que esta información se encuentra segura de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- **Manejo de transacciones.** Una transacción es un programa que se ejecuta como una sola operación. Esto quiere decir que luego de una ejecución en la que se produce una falla es el mismo que se obtendría si el programa no se hubiera ejecutado.
- **Tiempo de respuesta.** Lógicamente, es deseable minimizar el tiempo que el DBMS demora en proporcionar la información solicitada y en almacenar los cambios realizados.

2.2.4. Tipos de usuario

Podemos definir a los usuarios como toda persona que tenga todo tipo de contacto con el sistema de base de datos desde que éste se diseña, elabora, termina y se usa.

- Usuario Final: es la persona que utiliza los datos, esta persona ve datos convertidos en información.
- Desarrollador de Aplicaciones: persona que desarrolla los sistemas que interactúan con la Base de Datos.
- DBA: es la persona que asegura integridad, consistencia, redundancia, seguridad este es el Administrador de Base de Datos quien se encarga de realizar el mantenimiento diario o periódico de los datos.

Las personas que tienen acceso al DBMS se clasifican de la siguiente manera:

- Usuarios Ingenuos: Son aquellos que interactúan con el sistema por medio de aplicaciones permanentes.
- Usuarios Sofisticados: Son aquellos con la capacidad de acceder a la información por medios de lenguajes de consulta.

2.2.5. Modelo de Datos

Además de la clasificación por la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de administración de datos. Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para

almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos y conceptos matemáticos.

2.2.6. Base de datos relacional

Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". Pese a que ésta es la teoría de las bases de datos relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar. Esto es pensando en cada relación como si fuese una tabla que está compuesta por registros (las filas de una tabla), que representarían las tuplas y campos (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a las bases de datos relacionales es el lenguaje de consultas estructurado SQL (Structured Query Language) es un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales. Durante su diseño, una base de datos relacional pasa por un proceso al que se le conoce como normalización de una base de datos.

2.3. Normalización

2.3.1. Definición de Normalización

Es el proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas, que además de ser más simples y más estables, son más fáciles de manipular. La normalización hace las cosas fáciles de entender. Los seres humanos tenemos la tendencia de simplificar las cosas al máximo, además la gran ventaja de la normalización de base de datos es el consumo de espacio.

La normalización es una ayuda que proporciona un procedimiento riguroso para el diseño de bases de datos. Una base de datos mal diseñada puede funcionar inicialmente pero puede mostrar anomalías en el almacenamiento debidas al agrupamiento de los campos cuando se efectúan las operaciones de inserción, actualización o eliminación.

Las relaciones normalizadas se agrupan en cuatro categorías llamadas formas normales FN siendo cada nivel una descomposición más completa de una relación del nivel anterior.

El proceso de normalización empieza con la combinación de todos los datos de la base en una relación, la que a su vez se descompone en dos o más relaciones más pequeñas. Se efectúan descomposiciones sucesivas de las relaciones intermedias hasta que todas las relaciones obtenidas pertenecen a la cuarta forma normal (4FN) [10].

2.3.2. Formas Normales

Son las reglas que debemos cumplir para evitar:

- existencia redundancia de datos
- problemas de actualización de la base de datos
- mantener la integridad de las tablas que componen la base de datos

Redundancia: duplicidad de datos

Integridad: es la manera en la que evitamos las malas relaciones y aseguramos que la base de datos contiene elementos verdaderos y completos.

2.3.3. Primera Forma Normal

Toda entidad en una base de datos debe tener una clave única y principal o asignarle un identificador.

Una relación está en 1FN si todos los campos en cada registro contienen un solo valor tomado de sus dominios respectivos, es decir en cada campo solo debe de existir un valor.

Por ejemplo nosotros los seres humanos tenemos un nombre, pero puede que el mismo nombre lo tenga otra persona, entonces para ser identificados de forma única nos asignan un número el cual nos representa, como la CURP, esta será entonces nuestra clave principal.

2.3.4. Segunda Forma Normal

Las relaciones entre cada uno de los atributos de la entidad deben ser uno a uno.

Una relación es segunda forma normal si es 1FN y cada atributo no-clave de la relación es total y funcionalmente dependiente de su clave principal.

Por ejemplo: los estudiantes deben tener un número de identificación, un género, etcétera, pero encontramos que el tipo de número de identificación puede ser de la cédula, del seguro social, de un pasaporte, en este caso no se cumpliría la segunda forma normal ya que el número de identificación estaría relacionado con varios tipos de identificación al que puede pertenecer.

2.3.5. Tercera Forma Normal

Eliminar los atributos de cada una de las entidades que no cumpla con la Segunda Forma Normal.

Una relación es 3FN, si es 2FN y ningún atributo no-clave en la relación es funcionalmente dependiente de algún otro atributo no clave.

Por ejemplo: un estudiante tiene un género al que pertenece, pero este puede ser femenino o masculino, no hay un solo género definido para todas las personas, entonces no cumple con la segunda forma normal, debe ser eliminado de la tabla o entidad a la que pertenece.

2.3.6. Cuarta Forma Normal

La Cuarta Forma Normal se asegura de que las dependencias multivaluadas independientes estén correcta y eficientemente representadas en un diseño de base de datos.

La cuarta forma normal es el siguiente nivel de normalización después de la forma normal de Boyce-Codd (BCNF).

Forma Normal de Boyce/Codd: Una relación es BFNC si cada determinante en la relación es una clave aspirante. Si existe algún atributo que resulte ser total y funcionalmente dependiente de otro se le llama determinante.

Los campos eliminados en la tercera forma normal, se evalúan y si son necesarios los convertiremos en nuevas entidades, y aplicaremos nuevamente todas las formas normales.

Terminología relacional equivalente

- Relación = tabla o archivo.
- Registro = fila, renglón o tupla.
- Atributo = columna o campo.
- Clave = llave o código de identificación.
- Clave Primaria = clave candidata elegida.
- Clave Ajena (o foránea) = clave externa o clave foránea.
- Dependencia multivaluada = dependencia multivalor.

Los términos relación, tupla y atributo derivan del álgebra y cálculo relacional, que constituyen la fuente teórica del modelo de base de datos relacional [11].

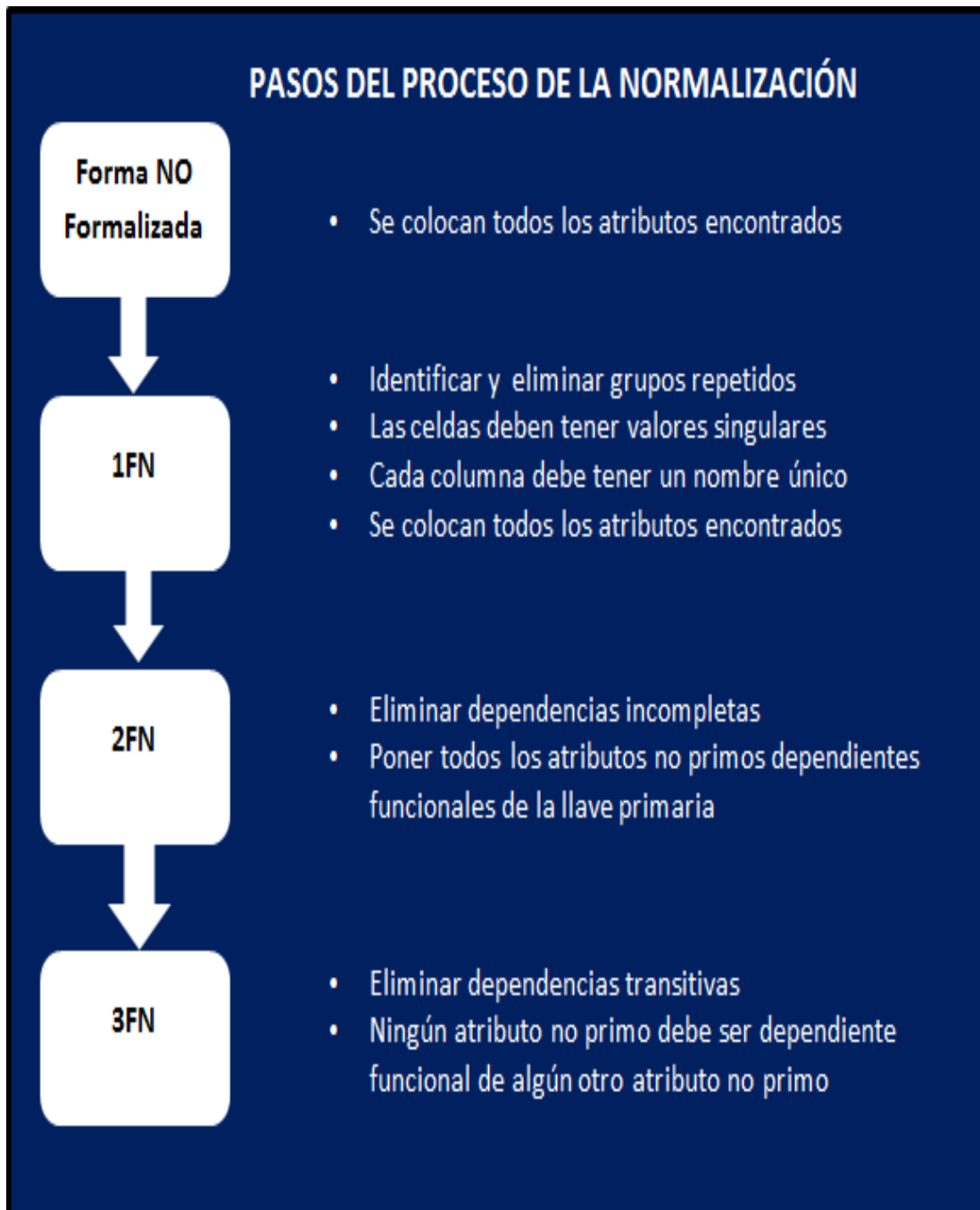


Figura 18. Pasos para el proceso de la normalización.

2.4. Normalización de las tablas

En primer lugar se identifican todas las entidades y atributos que conforman la base de datos para organizarlas en las tablas. Se eliminan grupos repetidos aplicando las reglas de la normalización y se reduce hasta tener una estructura más simple y estable.

2.5. Entidades y atributos de las tablas (Tablas no formalizadas).

Doctor	
#*	Id
*	nombre
*	appat
*	apmat
*	dir
*	tel
*	nombretrabajo
*	dirtrabajo
*	teltrabajo

Los campos sombreados son multivaluados.

- # Clave principal
- * Campo requerido
- o Campo opcional

Un doctor puede tener más de un teléfono dónde localizarlo, puede trabajar en varias instituciones médicas con sus respectivas direcciones y teléfonos del lugar de trabajo.

Tabla 1.

Paciente	
#*	Id
*	nombre
*	appat
*	apmat
*	dir
*	tel
*	sexo
*	fnac
*	peso
o	estatura
o	antecedentesfam
o	tipodiabetes
o	nivel_glucosa
o	presión_arterial
o	fecha_consulta_anterior
o	FechaConsultaActual
o	enfermedades
o	medicamentos
o	dosis

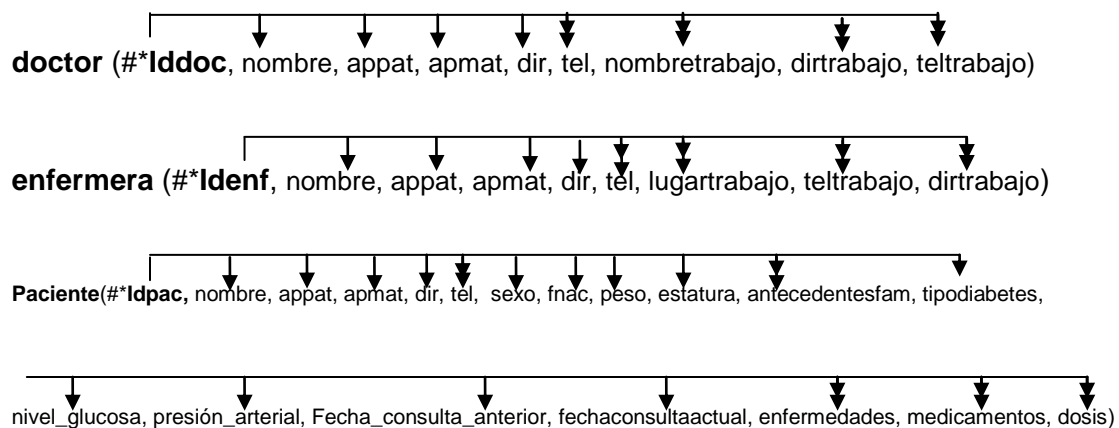
Un paciente puede tener más de un teléfono donde localizarlo, puede heredar de varios familiares una o más enfermedades, tomar más de un medicamento con su respectiva dosis.

Tabla 2.

Enfermera	
#*	Id
*	nombre
*	appat
*	apmat
*	dir
*	tel
*	nombretrabajo
*	dirtrabajo
*	teltrabajo

Tabla 3.

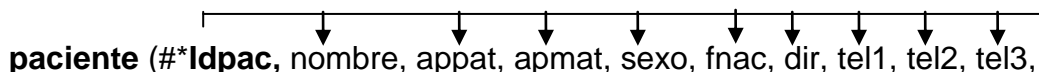
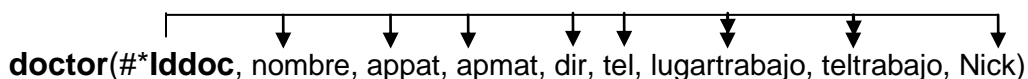
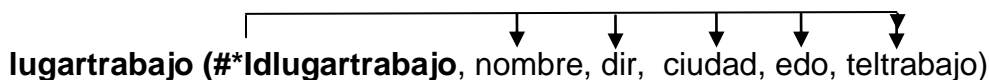
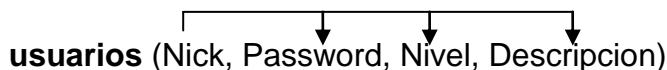
Una enfermera puede tener más de un teléfono dónde localizarla, puede trabajar en varias instituciones médicas y por lo tanto tener más de una dirección de lugar de trabajo y a su vez varios teléfonos del lugar de trabajo.



De aquí entonces se deben crear nuevas tablas y relacionarlas.

1FN:

Se requiere de una tabla que almacene la información de los usuarios y el sistema pueda validarlos e iniciar sesión.



Nick, idtipodiabetes, iddoc)

historial (* Idhistorial, peso, nivel_glucosa, presion_arterial, enfermedades, ascendencia, fecha_consulta_anterior, enfermera, doctor, fecha_consulta_actual, estatura, medicamentos, dosis)

La tabla historial tendrá los datos clínicos del paciente y es la resultante de la relación entre las tablas paciente, enfermera y doctor.

enfermera(*Idenf, nombre, appat, apmat, dir, tel, teltrabajoenf, lugartrabajo, Nick)

2FN:

enfermedad (*Id, Nombre)

teltrabajo (*Idteltrabajoenf, tel, ext, idlugartrabajo)

lugartrabajo (*Idlugartrabajo, nombre, dir, cp, ciudad, edo)

ascendencia (ascendencia, idpac, idenfermedad, tipodiabetes)

medicamentos (* Idmedicamento, principioactivo, nombre, presentacion)

dosis (descripcion, idmedicamento, idhistorial)

enfermera(*Idenf, nombre, appat, apmat, dir, tel, teltrabajoenf, lugartrabajo, Nick)

doctor(*Iddoc, nombre, appat, apmat, dir, tel, , idteltrabajo, Nick)

3FN:

historial(* Idhistorial, peso, nivel_glucosa, presion_arterial,

Fecha_consulta_anterior, idenf, idpac, fecha_consulta_actual, estatura, doctor)

2.6. Modelo Entidad – Relación (E-R)

El diseño conceptual se lleva a cabo a través de modelos formales de datos, como el modelo entidad relación, el cual permite tener una idea clara de cuales son las entidades, atributos e interrelaciones incluidas en el sistema.

En el sistema existen 12 entidades, cada una de ellas se explica a continuación.

Entidad	Descripción
Usuarios	Se requiere esta entidad para iniciar sesión de administrador, doctor, enfermera o paciente según sea el tipo se habilitarán pantallas específicas.
Paciente	Contiene los datos personales y clínicos de los pacientes.
Doctor	Es una de las entidades principales ya que estará a cargo del control clínico de los pacientes.
Enfermera	Contiene información personal para que el doctor o paciente la contacten.
Historial	Se requiere para almacenar los datos clínicos de los pacientes y relacionar esta entidad con la de doctor y enfermera.
Ascendencia	Establece una relación entre el tipo de diabetes del paciente y si la heredó de algún familiar, esta información permitirá el doctor hacer el diagnostico de otras enfermedades.
Enfermedad	Almacena algunas de las enfermedades asociadas a la diabetes.
Tipodiabetes	Contiene un identificador y descripción de los tipos de diabetes.
Dosis	Esta entidad se deriva de la tabla medicamentos e historial y contiene la dosis de cada medicamento recetado.
Medicamentos	Aquí se almacena el identificador, principio activo, nombre y presentación de los medicamentos.
Teltrabajo	Almacena el número telefónico del lugar de trabajo del doctor y la enfermera.
Lugartrabajo	Tiene los datos del lugar de trabajo del doctor y de la enfermera.

Tabla 4: Entidades del sistema.

Después del análisis de las entidades y la forma de relacionar los campos en cada tabla se llegó al modelo de la figura 19, algunas de las tablas coincidían en el contenido de la información como la del teléfono del trabajo del doctor y de la enfermera así que se optó por definir un campo llamado teltrabajoenf que se usará en la tabla del doctor y de la enfermera, a su vez se definieron claves primarias en cada tabla y claves foráneas heredadas de otras tablas; por ejemplo la tabla historial además de tener su propio identificador definido como idhistorial, hereda de la tabla paciente el identificador idpac, de la tabla enfermera el identificador idenf y de la tabla doctor el campo iddoc.

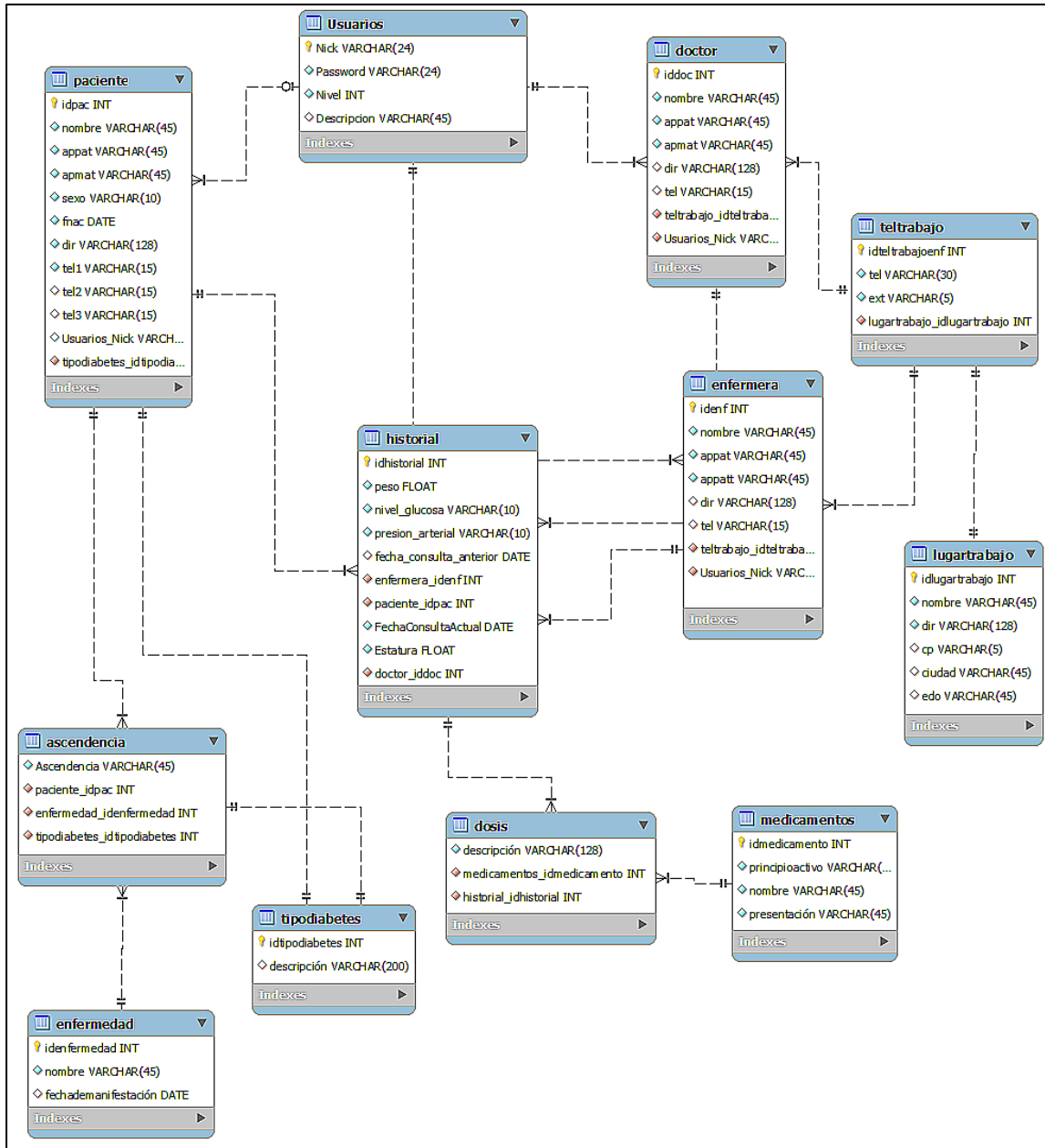


Figura 19. Modelo Entidad – Relación.

CAPITULO 3: PROPUESTA DEL SISTEMA

Contiene la descripción de la arquitectura del sistema, conceptos teóricos y algunos aspectos del software utilizado.

3.1. Implementación

El sistema estará formado por una base de datos que concentre información personal y clínica de pacientes, datos personales y del lugar de trabajo del doctor, medicamentos para que el doctor pueda recetar a los pacientes las dosis correspondientes y sobre la enfermera los datos para poder localizarla en caso de alguna complicación en el estado de salud de los pacientes de los que es responsable.

El sistema admite 4 tipos de usuarios:

- Administrador
- Doctor
- Enfermera
- Paciente

El Administrador es el encargado de autorizar el acceso, coordinar y vigilar la utilización de la base de datos, es decir regular cuáles son las partes de la base de datos a la que van a tener acceso los usuarios.

Funciones del Doctor:

- Actualizar sus datos personales.
- Leer historial de paciente.
- Leer datos de enfermera.
- Diagnosticar paciente.
- Recetar a pacientes.

Funciones de la Enfermera:

- Actualizar sus datos personales.
- Leer historial de paciente.
- Actualizar consulta a paciente.

Funciones del Paciente:

- Leer su historial.
- Actualizar sus lecturas de glucosa en la sangre y presión arterial.
- Revisar su medicación.

El sistema muestra la pantalla inicial donde se deberá ingresar el nombre de Usuario y Password, dependiendo del tipo de usuario identificado por 'nivel' se iniciará la sesión con las funciones asociadas como se ve en la figura 20.

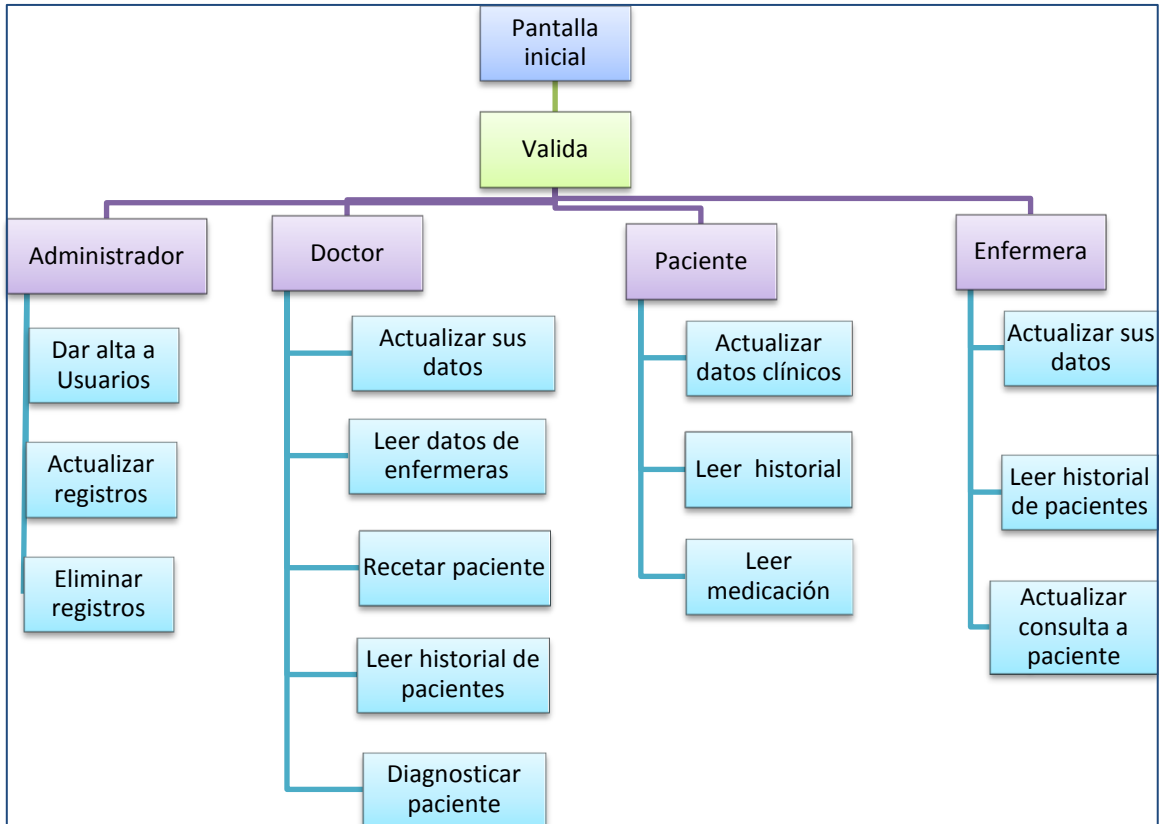


Figura 20. Diagrama funcional del sistema.

3.2. Arquitectura cliente – servidor

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, que le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.



Figura 21. Arquitectura Cliente - Servidor.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Una disposición muy común son los *sistemas multicapa* en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

La *arquitectura cliente-servidor* sustituye a la *arquitectura monolítica* en la que no hay distribución, tanto a nivel físico como a nivel lógico.

La red cliente-servidor es aquella red de comunicaciones en la que todos los clientes están conectados a un servidor, en el que se centralizan los diversos recursos y aplicaciones con que se cuenta; y que los pone a disposición de los clientes cada vez que estos son solicitados. Esto significa que todas las gestiones que se realizan se concentran en el servidor, de manera que en él se disponen los requerimientos provenientes de los clientes que tienen prioridad, los archivos que son de uso público y los que son de uso restringido, los archivos que son de sólo lectura y los que, por el contrario, pueden ser modificados, etc. Este tipo de red puede utilizarse conjuntamente en caso de que se este utilizando en una red mixta.

3.3. Código abierto

Código abierto es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de poder acceder al código, que a las cuestiones éticas y morales las cuales se destacan en el software libre.



Figura 22. Software libre – Código abierto

Al igual que el software libre, el *código abierto* u *open source* tiene una serie de requisitos necesarios para que un programa pueda considerarse dentro de este movimiento, éstos son:

- Libre redistribución: el software debe poder ser regalado o vendido libremente.
- Código fuente: el código fuente debe estar incluido u obtenerse libremente.
- Trabajos derivados: la redistribución de modificaciones debe estar permitida.
- Integridad del código fuente del autor: las licencias pueden requerir que las modificaciones sean redistribuidas sólo como parches.
- Sin discriminación de personas o grupos: nadie puede dejarse fuera.
- Sin discriminación de áreas de iniciativa: los usuarios comerciales no pueden ser excluidos.
- Distribución de la licencia: deben aplicarse los mismos derechos a todo el que reciba el programa.
- La licencia debe ser tecnológicamente neutral: no debe requerirse la aceptación de la licencia por medio de un acceso por clic de ratón o de otra forma específica del medio de soporte del software.

3.3.1. Apache

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo.

Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. En inglés, a patchy server (un servidor "parcheado") suena igual que Apache Server.

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años.

La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache.

La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache [12].

3.3.2. PHP

PHP es un lenguaje de programación interpretado o framework para HTML, diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

Características

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador web y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.

- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).
- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.

Inconvenientes

- Como es un lenguaje que se interpreta en ejecución, para ciertos usos puede resultar un inconveniente que el código fuente no pueda ser ocultado. La ofuscación es una técnica que puede dificultar la lectura del código pero no necesariamente impide que el código sea examinado.
- Debido a que es un lenguaje interpretado, un script en PHP suele funcionar considerablemente más lento que su equivalente en un lenguaje de bajo nivel, sin embargo este inconveniente se puede minimizar con técnicas de cache tanto en archivos como en memoria [13].

3.3.3. XAMPP, LAMP, WAMP, MAMP

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. El programa está liberado bajo la licencia GNU y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP esta disponible para Microsoft Windows, GNU/Linux, Solaris, y MacOS X.

LAMP presenta una funcionalidad parecida a XAMP, pero enfocada en Linux, y WAMP lo hace enfocado en Windows.

Principales sitios desarrollados con PHP

PHP es utilizado en millones de sitios, entre los más destacados se encuentran wikipedia.org, facebook.com y Wordpress.com.

3.3.4. MySQL

MySQL ha crecido, lo que durante un tiempo se consideró como un sencillo juguete para su uso en sitios Web, se ha convertido en la actualidad en una solución viable y de misión crítica para la administración de datos. Antes, MySQL se consideraba como la opción ideal para sitios Web; sin embargo, ahora incorpora muchas de las funciones necesarias para otros entornos y conserva su gran velocidad. MySQL supera desde hace tiempo a muchas soluciones comerciales en velocidad y dispone de un sistema de permisos elegante y potente, y ahora, además, la versión 4 incluye el motor de almacenamiento InnoDB compatible con ACID. MySQL 4 es rápido, dispone de funciones de volcado online e incorpora una gran cantidad de funciones nuevas. Son pocas las razones para desechar MySQL como solución de base de datos. MySQL AB, la compañía responsable del desarrollo de MySQL, dispone de un sistema de asistencia eficiente y a un precio razonable y, como ocurre con la mayor parte de las comunidades de código abierto, encontrará una gran cantidad de ayuda en la Web. Las funciones estándar no incluidas todavía en MySQL (como las vistas y los procedimientos almacenados) están en fase de desarrollo y es posible que estén disponibles para cuando lea estas líneas. Son muchas las razones para escoger MySQL como solución de misión crítica para la administración de datos.

- **Coste:** El coste de MySQL es gratuito para la mayor parte de los usos y su servicio de asistencia resulta económico.
- **Asistencia:** MySQL AB ofrece contratos de asistencia a precios razonables y existe una nutrida y activa comunidad MySQL.
- **Velocidad:** MySQL es mucho más rápido que la mayor parte de sus rivales.
- **Funcionalidad:** MySQL dispone de muchas de las funciones que exigen los desarrolladores profesionales, como compatibilidad completa con ACID, compatibilidad para la mayor parte de SQL ANSI, volcados online, duplicación, funciones SSL e integración con la mayor parte de los entornos de programación. Así mismo, se desarrolla y actualiza de forma mucho más rápida que muchos de sus rivales, por lo que prácticamente todas las funciones estándar de MySQL todavía no están en fase de desarrollo.
- **Portabilidad:** MySQL se ejecuta en la inmensa mayoría de sistemas operativos y, la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.
- **Facilidad de uso:** MySQL resulta fácil de utilizar y de administrar. Gran parte de las viejas bases de datos presentan problemas por utilizar sistemas obsoletos, lo que complica innecesariamente las tareas de administración [14].

CAPITULO 4: EVALUACIÓN Y PRUEBAS

Se describe las pruebas del sistema, también algunas funciones principales de los usuarios.

Después del análisis y diseño de la base de datos se desarrolló la implementación del sistema usando HTML, PHP y MySQL. Se muestran las pantallas de cada actor y las tareas asociadas.

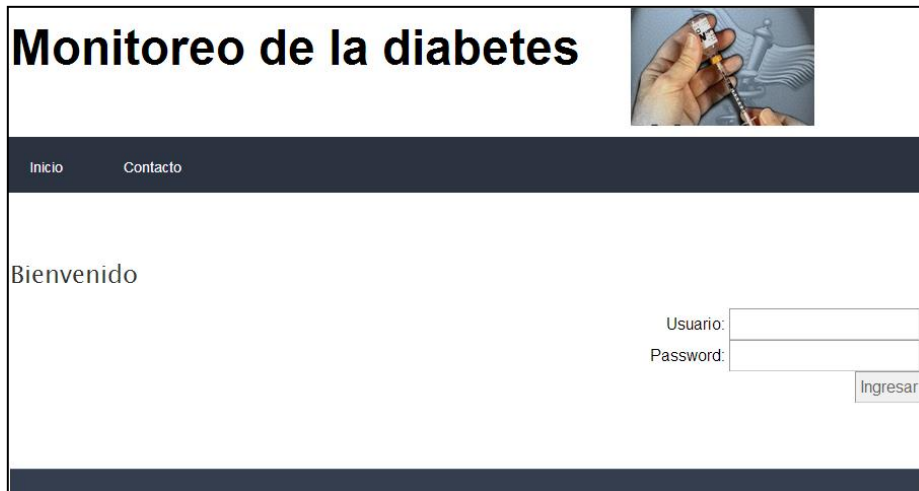


Figura 23. Pantalla Inicial

Se muestra un mensaje de bienvenida además de las opciones de Inicio y Contacto. El usuario proporciona su nombre de usuario y password, el sistema lo validará para iniciar sesión.

En caso de que el usuario exista en la base de datos pero no escriba correctamente su nombre de usuario o password se mostrará un mensaje indicando que debe regresar a la pantalla inicial (Figura 23).

4.1. Administrador



Figura 24. Pantalla Administrador

Al ingresar al sistema como administrador (Figura 24), se listan las opciones del tipo de usuario 0) Administrador, 1) Doctor, 2) Enfermera y 3) Paciente y regular las partes de la base de datos a las cuales tendrá acceso el usuario. La opción Salir regresa a la pantalla Inicial (Figura 23) en todas las sesiones.

Monitoreo de la diabetes	
Inicio Insertar Actualizar Eliminar Salir	
USUARIO PACIENTE	
Nick:	<input type="text" value="Georgina"/>
Password:	<input type="password" value="...."/>
<input type="button" value="Actualizar"/>	

Figura 25. Pantalla Inserta usuario paciente.

El administrador debe escribir el nombre de usuario y contraseña (Figura 25), al hacer clic en el botón Actualizar se muestra la pantalla indicando que la base de datos se ha actualizado. Enseguida se muestran las opciones Datos (Figura 26) e Historial.

Monitoreo de la diabetes	
Inicio Insertar Salir	
USUARIO PACIENTE	
Nombre:	<input type="text" value="Georgina"/>
Apellido Paterno:	<input type="text" value="Juárez"/>
Apellido Materno:	<input type="text" value="Robles"/>
Sexo:	<input type="text" value="Mujer"/>
Fecha de Nacimiento (Año/Mes/Día):	<input type="text" value="1958/4/6"/>
Dirección:	<input type="text" value="Ave. Morelos No.12, Co"/>
Teléfono 1:	<input type="text" value="2457896"/>
Teléfono 2:	<input type="text" value="1693626"/>
Teléfono 3:	<input type="text" value="5789462"/>
Nick:	<input type="text" value="Georgina"/>
Tipo de Diabetes:	<input type="text" value="1"/>
Id Doctor:	<input type="text" value="2"/>
<input type="button" value="Actualizar"/>	

Figura 26. Pantalla Datos.

Para terminar con el registro del paciente se deberá llenar el formulario del la Figura 27, al hacer clic en el botón Actualizar los datos son almacenados en la tabla historial.

Monitoreo de la diabetes	
Inicio Insertar Salir	
USUARIO PACIENTE	
Peso (Kg.) :	<input type="text"/>
Nivel glucosa (dl/mm) :	<input type="text"/>
Presión arterial(mm/Hg) :	<input type="text"/>
Fecha consulta anterior (Año/Mes/Día) :	<input type="text"/>
Id Enfermera :	<input type="text"/>
Id Paciente:	<input type="text"/>
Fecha consulta actual (Año/Mes/Día):	<input type="text"/>
Estatura (m) :	<input type="text"/>
Id Doctor:	<input type="text"/>
<input type="button" value="Actualizar"/>	

Figura 27. Pantalla Historial.

En el caso de los usuarios doctor y enfermera el formulario a llenar tendrá los campos de datos personales y teléfonos de localización.

4.2. Doctor

El usuario doctor debe ingresar su nombre de usuario y password correcto para que el sistema muestre la pantalla de validación; si el sistema no reconoce el nombre de usuario o el password se mostrará un mensaje indicando que debe regresar a la pantalla Inicial (Figura 23).

La primera pantalla, es la de Cuenta que muestra la información personal del doctor, además puede actualizar los números de localización haciendo clic en el icono Actualizar. En la opción Enfermeras se despliegan los datos de las enfermeras dadas de alta en el sistema para poder tener un contacto más directo ya que ellas serán las responsables de hacer modificaciones en la fecha de consulta del paciente por indicaciones del mismo doctor.

En la opción Pacientes se muestra la pantalla con información de pacientes asignados a este doctor (Figura 28), se muestran las opciones Información e Historial y Recetar. En la pantalla Información e historial el doctor puede leer los datos personales y clínicos del paciente.

Monitoreo de la diabetes



Pacientes Enfermeras Cuenta Salir

Bienvenido

Id	Nombre	Tipo Diabetes	Opciones
1	Alberto	1	 
2	Fabiola	3	 
4	Lola	3	 
5	Rosa	1	 

Figura 28. Pantalla Pacientes sesión Doctor.

Al hacer clic en el icono Recetar el sistema muestra los campos a llenar (Figura 29), al presionar el botón Recetar la base de datos se actualiza, si por el contrario se presiona el botón Reset no se efectúan cambios.

Monitoreo de la diabetes



Pacientes Enfermeras Cuenta Salir

Bienvenido

Principio activo	<input type="text" value="Clorpropamida"/>
Nombre	<input type="text" value="Diabinese"/>
Presentación	<input type="text" value="250 mg. 30 tabletas"/>
Descripción	<input type="text" value="1 cada 8 hrs."/>

Figura 29. Pantalla Recetar.

4.3. Enfermera

La pantalla inicial muestra los datos de la Cuenta (Nombre y teléfonos de localización) además el icono que permite modificarlos como ya se explicó para el usuario Doctor. Al seleccionar la opción Paciente se puede leer la información básica del paciente (Figura 30), el ícono de opciones muestra la pantalla con información detallada del paciente (Figura 31). La enfermera puede cambiar la fecha de la siguiente consulta si el doctor así se lo indica haciendo clic en el icono de Actualizar fecha y aceptar los cambios se actualiza la base de datos.



Figura 30. Pantalla Pacientes-Enfermera

Nombre	Alberto Martínez Gómez			
Sexo	Hombre			
Telefono	1234567			
Tipo Diabetes	Tipo 1: insulino dependiente o diabetes de comienzo juvenil. Se presenta en jóvenes y en adultos también pero con menos frecuencia. El cuerpo no produce o produce poca insulina y se necesitan inyecciones diarias de esta hormona			
Peso (Kg)	Nivel de la Glucosa (dl/mm)	Presion Arterial (mg/Hg)	Fecha Consulta Anterior	Fecha Consulta Actual
90	12	95/100	2012-09-07	2013-02-11 12:30:00

Figura 31. Pantalla Información e historial

4.4. Paciente

La pantalla inicial de esta sesión muestra los datos personales y clínicos del paciente (Figura 32). En el enlace medicación el paciente verá en pantalla la lista y descripción de los medicamentos recetados por el doctor (Figura 33).

Monitoreo de la diabetes



Inicio Actualizar Salir

Bienvenido

Nombre	Alberto Martínez Gómez
Sexo	Hombre
Telefono	1234567
Tipo Diabetes	Tipo 1: insulino dependiente o diabetes de comienzo juvenil. Se presenta en jóvenes y en adultos también pero con menos frecuencia. El cuerpo no produce o produce poca insulina y se necesitan inyecciones diarias de esta hormona

Peso (Kg)	Nivel de la Glucosa (dl/mm)	Presion Arterial (mg/Hg)	Fecha Consulta Anterior	Fecha Consulta Actual	Medicación
100	10	100/95	2012-09-07	2012-10-08 00:00:00	

Figura 32. Pantalla de Inicio-Paciente

Monitoreo de la diabetes



Inicio Actualizar Salir

PRINCIPIO ACTIVO	NOMBRE	PRESENTACIÓN	DOSIS
Metformina	Dianven	800 mg. 50 comprimidos	1 cada 8 hrs.

Figura 33. Pantalla Medicación

La opción actualizar muestra un formulario donde el paciente escribirá los valores clínicos más recientes tomados por él o por su enfermera en caso de que sea atendido en casa (Figura 34) y así la base de datos se actualice.



Monitoreo de la diabetes

Inicio Actualizar Salir

Bienvenido

Peso (Kg)	<input type="text" value="95"/>
Nivel glucosa (dl/mm)	<input type="text" value="12"/>
Presión arterial (mg/Hg)	<input type="text" value="95/100"/>

Figura 34. Pantalla Actualizar lecturas.

CONCLUSIONES Y TRABAJO A FUTURO

El objetivo principal en este trabajo de tesina consistió en realizar una aplicación Web para el monitoreo de la Diabetes, además se realizó el análisis y diseño de una base de datos para que pacientes con diabetes logren un control sobre su estado clínico estando en comunicación directa con su médico o enfermera.

Fueron atendidos todos y cada uno de los objetivos planteados; analizando, diseñando e implementado una estrategia de solución para la creación del sistema para finalmente realizar su implementación en HTML y PHP y la utilización de un gestor de base de datos MySQL.

La realización de este sistema es de gran ayuda ya que cada año se incrementa la cantidad de personas con esta enfermedad debido a la falta de atención y de tiempo por las actividades laborales.

El uso de tecnologías web es fundamental para el desarrollo de este tipo de aplicaciones. En lo particular, el desarrollo del sistema me permitió aplicar los conocimientos adquiridos en el diplomado referente a la ingeniería y desarrollo de software así como el uso de lenguajes de programación para aplicaciones web. Sin duda alguna fue una experiencia satisfactoria que resultará fundamental para el desarrollo y realización de futuros proyectos en el ámbito de la informática.

Como trabajo a futuro cabe mencionar que el sistema puede crecer, implementando funciones como la generación de una alerta para el doctor si un paciente tiene lecturas críticas en el nivel de glucosa, manejo de estadísticas y gráficos sobre el avance de la enfermedad y de la cantidad de pacientes por cada doctor o por hospital. Además, se podría realizar una interfaz de la aplicación web que ayude a generar la sincronización de la información con los dispositivos móviles y poder extender a las redes sociales actuales.

Se estima que utilizar un sistema de aplicación web como el presentado en este trabajo puede ayudar a disminuir el número de muertes o detección a tiempo de esta enfermedad.

Se pretende que se extienda su uso a hospitales, clínicas o sanatorios públicos y privados así como también se use como herramienta personal para médicos especialistas en la diabetes que pueden tener una cantidad de pacientes muy alta.

BIBLIOGRAFÍA

[1] ¿Qué es la diabetes?

<https://www.accu-check.com.mx/mx/diabetes/queesladiabetes.htm>, año de visualización: 2012.

[2] MEXICO, PRINCIPALES CAUSAS DE MORTALIDAD DESDE 1938

Recopilación: Ing. Manuel Aguirre Botello, con datos de INEGI, OMS y SINAIS

<http://www.mexicomaxico.org/Voto/MortalidadCausas.htm>, año de visualización: 2012.

[3] Una diabetes mejor controlada.

<http://www.sugarfree.com.es/?tag=educacion>, año de visualización 2012.

[4] Software para diabéticos: controlar el nivel de glucosa

<http://www.vitonica.com/enfermedades/software-para-diabeticos-controlar-el-nivel-de-glucosa>. Año de visualización: 2012.

[5] Diabetes Software: Make Diabetes Easier

<http://www.diabetespilot.com/>, año de visualización: 2012.

[6] Modelo RUP en ISW Rational Unified Process (RUP)/ Proceso Racional Unificado.

<http://rupenishw.blogspot.mx/2010/11/que-es-rup-su-meta-principal-es.html>, año de visualización 2012.

[7] RUP (PROCESO RACIONAL UNIFICADO)

<http://erzh.hostzi.com/index/rup.html>, año de visualización:2012.

[8] UML AND THE UNIFIED PROCESS PRACTICAL OBJECTS-ORIENTES ANALYSIS & DESIGN. Segunda edición. Jim Arlow, Ila Neustadt, John Wiley & Sons. (2005).

[9] Base de datos.

http://es.wikipedia.org/wiki/Base_de_datos, año de visualización: 2012.

[10] http://es.wikipedia.org/wiki/Normalizaci%C3%B3n_de_bases_de_datos, año de visualización: 2012.

[11] Normalización de bases de datos, fecha de creación: 29 Mayo del 2003.

<http://www.eet2mdp.edu.ar/alumnos/MATERIAL/MATERIAL/info/infonorma.pdf>.

Por mysql-hispano.org/, año de visualización 2012.

[12] Servidor Apache

http://es.wikipedia.org/wiki/Servidor_HTTP_Apache. Año de visualización: 2012.

[13] PHP.

<http://es.wikipedia.org/wiki/Php> año de visualización: 2012.

[14] La biblia MySQL Ian Gilfillan Anaya Multimedia. Primera edición 2003.