



**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE
PUEBLA
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN**

**SISTEMA DE AYUDA MÉDICA:
SAMED**

TESINA PROFESIONAL PARA OBTENER EL TÍTULO DE
LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

LUIS MIGUEL GARCÍA ACA

ASESOR:

M.C. ALFONSO GARCÉS BÁEZ

Puebla, Pue.

Mayo 2014

INDICE

INTRODUCCIÓN

OBJETIVO
JUSTIFICACIÓN
MISIÓN Y VISIÓN

CAPITULO UNO

MARCO TEORICO

1.1 INGENIERÍA DE SOFTWARE.....	1
1.1.1 SOFTWARE.....	2
1.1.2 PROCESO DE SOFTWARE.....	3
1.1.3 MODELO DE PROCESOS DEL SOFTWARE.....	3
1.1.4 ITERACIÓN DE PROCESOS.....	4
1.1.5 MODELO DE CASCADA.....	4
1.2 BASES DE DATOS.....	7
1.2.1 INTRODUCCIÓN A LAS BASES DE DATOS.....	8
1.2.2 CLASIFICACIÓN DE LOS MODELOS DE DATOS.....	8
1.2.3 MODELO ENTIDAD – RELACIÓN.....	10
1.2.4 FORMAS NORMALES.....	14
1.2.5 DIAGRAMA.....	16
1.3 PLATAFORMA CLIENTE SERVIDOR.....	19
1.3.1 SERVIDOR DE BASES DE DATOS.....	20

CAPITULO DOS

ANÁLISIS DEL Sistema de Ayuda Médica (SAMED)

2.1 PLANTEAMIENTO DEL PROBLEMA.....	22
2.2 ESTUDIO DE VIABILIDAD.....	22
2.2.1 VIABILIDAD TECNOLÓGICA.....	22
2.2.2 VIABILIDAD OPERACIONAL.....	23
2.2.3 VIABILIDAD ECONÓMICA.....	23
2.3 ESPECIFICACIÓN DE REQUERIMIENTOS.....	23
2.3.1 DOCUMENTO DE REQUERIMIENTOS DE USUARIO.....	24
2.3.2 DESCRIPCIÓN DETALLADA DEL SISTEMA.....	24
2.3.3 DOCUMENTO DE ESPECIFICACIÓN DEL SISTEMA.....	26
2.3.4 IDENTIFICAR LAS ENTIDADES.....	27
2.4 DESCRIPCIÓN DE LA INFORMACIÓN.....	27
2.4.1 IDENTIFICACIÓN DE CASOS DE USO DEL SISTEMA.....	28
2.4.2 IDENTIFICACIÓN DE RELACIONES.....	30
2.4.3 IDENTIFICACIÓN DE ATRIBUTOS.....	31
2.4.4 IDENTIFICADORES DE LAS ENTIDADES.....	31
2.5 ESQUEMA ENTIDAD – RELACIÓN.....	32
2.5.1 REVISIÓN DEL ESQUEMA CON EL USUARIO.....	32
2.6 DICCIONARIO DE DATOS.....	33

CAPITULO TRES

DISEÑO DEL SAMED

3.1 DISEÑO LÓGICO.....	36
3.2 REVISIÓN DE REQUERIMIENTOS.....	36
3.3 MODELO RELACIONAL.....	37
3.3.1 ELIMINACIÓN MUCHOS A MUCHOS N:M.....	37
3.3.2 VALIDACIÓN ESQUEMAS MEDIANTE LA NORMALIZACIÓN.....	39
3.3.2.1 PRIMERA FORMA NORMAL 1FN.....	40
3.3.2.2 SEGUNDA FORMA NORMAL 2FN DEPENDENCIA PLENA.....	41
3.3.2.3 TERCERA FORMA NORMAL 3FN DEPENDENCIAS TRANSITORIAS.....	42

CAPITULO CUATRO

IMPLEMENTACIÓN Y PRUEBAS SAMED

4.1 IMPLEMENTACION DE LA BASE DE DATOS EN MySQL.....	43
4.2 INTERFAZ DEL SISTEMA.....	45

CONCLUSIONES

BIBLIOGRAFIA

INTRODUCCIÓN

La computación ha realizado grandes cambios benéficos, en todo aquellos con lo que interactúa el hombre.

Las Bases de Datos son un medio de almacenar y controlar pequeñas o grandes cantidades de información, de cualquier tipo. Esto ha servido para la utilización rápida, sencilla y eficaz de datos. Esto aunado con la tecnología del Internet, ha logrado no simplemente el fácil manejo de información, sino que además de ello el poder utilizar dicha información desde cualquier parte del mundo.

Existe una gran variedad de software para el control de Bases de Datos y su manejo en Internet. En este proyecto me he inclinado por la utilización de PHP y MySQL, ya que se trata de software libre con excelentes resultados en el desarrollo de sistemas tanto para instituciones publicas como privadas.

En la actualidad nuestro país sufre de problemas alimenticios demasiado graves. Tanto de sobrepeso, como desnutrición y este problema lo encontramos desde el nacimiento de un ser humano. Esto es muy grave ya que en los primeros 2 años de vida el hombre percibe los conocimientos cognitivos y sin una buena alimentación, estos podrían ser erróneos.

El Sistema de Ayuda Médica (SAMED) va enfocado a la nutrición en etapa preescolar, es decir en niños de 0 a 5 años y tiene como único propósito colaborar con el nutriólogo y no la de servir como un autodiagnóstico. Y por ello dicho sistema se encarga de:

- Realizar un diagnóstico con respecto a los síntomas, ingresados por el nutriólogo, indicando el tipo de vitamina necesario para remediar dicho malestar.
- Indica el tipo de vitaminas que contiene cada nutriente, para que el nutriólogo pueda seleccionar el indicado.
- Mostrar una tabla acorde a los datos clínicos del niño e indicando si se encuentra en el peso ideal.
- Y como anexo contiene un historial clínico para cada paciente, recordando que es en esta etapa da comienzo dicho historial, quien sirve para posteriores consultas, no simplemente para el nutriólogo, sino para cualquier especialista.

OBJETIVO

El **Objetivo General** de este trabajo de tesina es el de: “Analizar, Diseñar e Implementar un Sistema para el apoyo Médico”, empleando el modelo de cascada para la Ingeniería de Software, PHP, MySQL para la Base de Datos y Photoshop.

SAMED es un sistema quien como único propósito tiene la de colaborar con el nutriólogo y no la de servir como un autodiagnóstico. Y por ello dicho sistema se encarga de:

Realizar un diagnóstico con respecto a los síntomas, ingresados por el nutriólogo, indicando el tipo de vitamina necesario para aliviar dicho malestar.

Indica el tipo de vitaminas que contiene cada nutriente, para que el nutriólogo pueda seleccionar el indicado.

Los **Objetivos Particulares** que se han planteado para alcanzar el objetivo general ya antes mencionado, son los siguientes:

- I Recopilación de los Requisitos.
- II Recopilación y Análisis de la información relacionada con la Ingeniería de Software y con la teoría de la Base de Datos.
- III Realizar un Análisis profundo de los requisitos para realizar la especificación de los requisitos del sistema SAMED
- IV Realizar un Diseño de la Base de Datos
- V Realizar la Implantación del Sistema y de la Base de Datos, y dar a conocer los resultados de las pruebas realizadas en el Sistema

JUSTIFICACIÓN

Este proyecto de tesina se justifica ya que me permite; proporcionar información de apoyo, aplicar los conocimientos obtenidos y adquirir cierta experiencia:

- Para la Facultad: Este documento tendrá importancia para los alumnos de la facultad ya que les permitirá realizar una extensión, alguna mejora, o simplemente servirles como apoyo para algún nuevo proyecto si no igual, al menos con ciertas similitudes.
- Para la sociedad: El uso de la computación se ha servido de gran apoyo a la sociedad en general ya sea de manera directa e indirectamente, por ello debemos día a día permitir esta colaboración sin pensar en esto como un reemplazo al ser humano por una computadora. Así pues las Bases de Datos Han venido sirviendo para un control más ágil y eficaz de grandes cantidades de información.
- Para mi Persona: Me permite aplicar los conocimientos obtenidos durante mi formación académica a lo largo de mi carrera en esta Facultad de Ciencias de la Computación BUAP. Del mismo modo adquirir experiencia en la elaboración de Bases de Datos acordes a los requerimientos del usuario.

MISIÓN

Trabajar de manera conjunta con el Nutriólogo para corroborar en la toma de decisiones con respecto a las condiciones y requerimientos para la obtención del diagnóstico del paciente.

VISIÓN

Ayudar en la superación del nutriólogo, estando un paso más adelante e ir de la mano con la tecnología teniendo una estructura propia.

CAPITULO UNO

MARCO TEÓRICO

La función principal es la de mostrar la teoría que engloba al proyecto como es; la ingeniería de software, el ciclo de vida de un proyecto, lo que son las bases de datos, hablando específicamente de las Bases de Datos Relacionales, y por ultimo se hace mención del concepto cliente servidor.

1.1 INGENIERÍA DE SOFTWARE

Un proyecto de software como todo desarrollo tiene un inicio y un fin aunque teóricamente podría ejecutarse en las computadoras por siempre. El proyecto para crear el software también tiene un ciclo de vida (Figura. 1.1), el cual esta delimitado por el tiempo más sin en cambio la operación puede ser continúa.

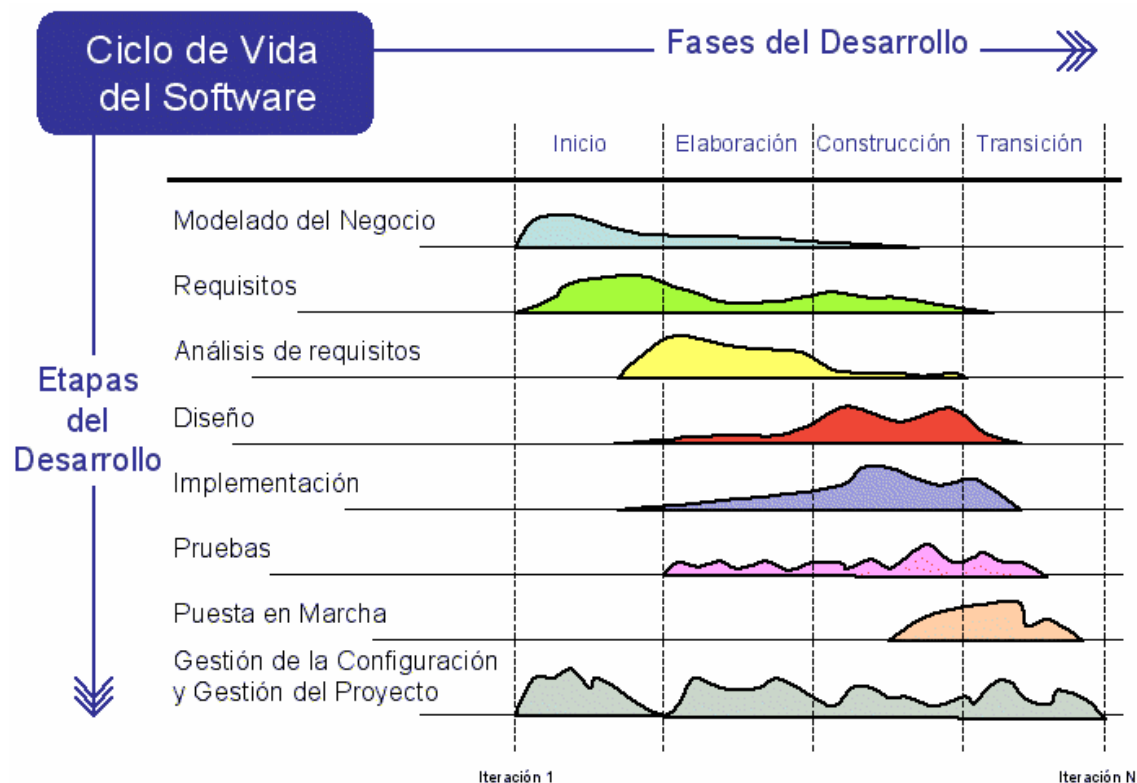


Figura. 1.1 Ciclo de vida del software

Además la Ingeniería de Software comprende todos los aspectos de la producción de software, y en ello podemos ver dos frases claves:

1. “La disciplina de la ingeniería”. Los ingenieros hacen que las cosas funcionen. Aplican teorías, métodos y herramientas donde sean convenientes, pero las utilizan de forma selectiva y siempre tratando de descubrir soluciones a los problemas, aun cuando no existan teorías y métodos aplicables para resolverlos.
2. “Todos los aspectos de producción de software”. La ingeniería de software comprende también las actividades, como la administración de proyectos de software y el desarrollo de herramientas, métodos y teorías de apoyo a la producción de software.

1.1.1 SOFTWARE

El software no son solo programas, sino todos los documentos asociados y la configuración de datos que se necesitan para hacer que estos programas operen de manera correcta. Un sistema de software consiste en diversos programas independientes.

Los ingenieros de software se concentran en el software que se vende a un cliente. Existen dos tipos de productos de software:

1. Productos genéricos; Son sistemas que se venden al mercado abierto a cualquier cliente, se denominan software empaquetado. Por ejemplo: bases de datos, los procesadores de texto, los paquetes de dibujo y las herramientas de administración de proyectos.
2. Productos personalizados; Son sistemas requeridos por un cliente en particular. Por ejemplo: sistemas de control para instrumentos eléctricos, sistemas de desarrollo para llevar a cabo procesos de negocios específicos y sistemas de control de tráfico aéreo.

DIFERENCIAS ENTRE INGENIERÍA DE SOFTWARE Y CIENCIAS DE LA COMPUTACIÓN

La Ciencia de la Computación se refiere a las teorías y métodos subyacentes a las computadoras y los sistemas de software, mientras que la Ingeniería de Software se refiere a los problemas prácticos de producir software¹.

¹ Sommerville Ian, *Madrid (2005), 7ª Edición, Ingeniería de Software, Editorial Pearson Educación S.A.*

1.1.2 PROCESO DE SOFTWARE

Es un conjunto de actividades y resultados asociados que producen un producto de software. Existen cuatro actividades fundamentales del proceso que son comunes para todos los procesos del software y son:

1. Especificación del software. La funcionalidad del software y las restricciones sobre su operación deben quedar definidas.
2. Desarrollo del software. Debe producirse software que cumpla con la especificación.
3. Validación del software. El software debe validarse para asegurar que es lo que el cliente quiere.
4. Evolución del software. El software debe evolucionar para cumplir con los cambios requeridos por el cliente.

1.1.3 MODELO DE PROCESOS DEL SOFTWARE

Los modelos son simplificaciones, por lo tanto un modelo de procesos del software es una abstracción del proceso real.

Los modelos de procesos que se discuten aquí son:

1. El modelo de cascada. Toma las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución, y los representa como fases separadas del proceso, como especificación de requerimientos, diseño de software, implementación, pruebas, etc.
2. Desarrollo evolutivo. Este enfoque entrelaza las actividades de especificación, desarrollo y validación. Un sistema inicial se desarrolla rápidamente a partir de especificaciones muy abstractas. Éste se refina basándose en las peticiones del cliente para producir un sistema que satisfaga las necesidades de dicho cliente.
3. Desarrollo formal del sistema. Este enfoque se basa en la producción de la especificación matemática formal del sistema y la transformación de esta especificación a un programa utilizando métodos matemáticos. La verificación de los componentes del sistema se lleva a cabo asiendo argumentos matemáticos acordes a su especificación.
4. Desarrollo basado en la reutilización. Este enfoque se basa en la existencia de un número significativo de componentes reutilizables. El proceso de desarrollo del

sistema se enfoca en integrar estos componentes en el sistema más que en desarrollarlos desde cero.

Aunque existan muchos procesos diferentes de software, tienen actividades fundamentales que son comunes para todos ellos. Éstas son:

- Especificación del software. Se debe definir la funcionalidad del software y las restricciones en sus operaciones.
- Diseño e implementación del software. Se debe producir software que cumpla su especificación.
- Validación del software. Se debe validar el software para asegurar que hace lo que el cliente desea.
- Evolución del software. El software debe evolucionar para cumplir los cambios en la necesidad del usuario.

1.1.4 ITERACIÓN DE PROCESOS

Los modelos anteriores tienen sus ventajas y desventajas. Para sistemas demasiado grandes, existe la necesidad de utilizar diferentes enfoques para distintas partes del sistema, por lo que se debe utilizar un modelo híbrido.

Los siguientes modelos híbridos ayudan a diferentes enfoques de desarrollo los cuales se diseñaron explícitamente para apoyar la iteración de procesos, y éstos son:

1. El desarrollo incremental en el que especificación, el diseño y la implementación del software se dividen en una serie de incrementos los cuales se desarrollan uno a uno.
2. El desarrollo en espiral en el que el desarrollo en espiral hacia fuera, empezando con un esbozo inicial y terminando con el desarrollo final del sistema.

1.1.5 MODELO DE CASCADA

Es el enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de forma tal que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior (Figura 1.2).

1. Análisis y definición de requerimientos. Los servicios, restricciones y metas del sistema se definen a partir de las consultas con los usuarios. Entonces, se definen en detalle y sirven como una especificación del sistema.
2. Diseño de sistema y de software. El proceso de diseño de sistemas divide los requerimientos en sistemas de hardware o de software. Establece una

arquitectura completa del sistema. El diseño de software identifica y describe las abstracciones fundamentales del sistema de software y sus relaciones.

3. Implementación y prueba de unidades. Durante esta etapa, el diseño de software se lleva a cabo como un conjunto o unidades de programas. La prueba de unidades implica verificar que cada una cumpla su especificación.
4. Integración y prueba del sistema. Los programas o las unidades individuales de programas se integran y prueban como un sistema completo para asegurar que se cumplan los requerimientos del software. Después de las pruebas, el sistema de software se entrega al usuario.
5. Operación y mantenimiento. Por lo general, esta es la fase más grande del ciclo de vida. El sistema se instala y pone en uso práctico. El mantenimiento implica corregir errores no descubiertos en las etapas anteriores del ciclo de vida, mejorar la implementación de las unidades del sistema y resaltar los servicios del sistema una vez que se descubran nuevos requerimientos.

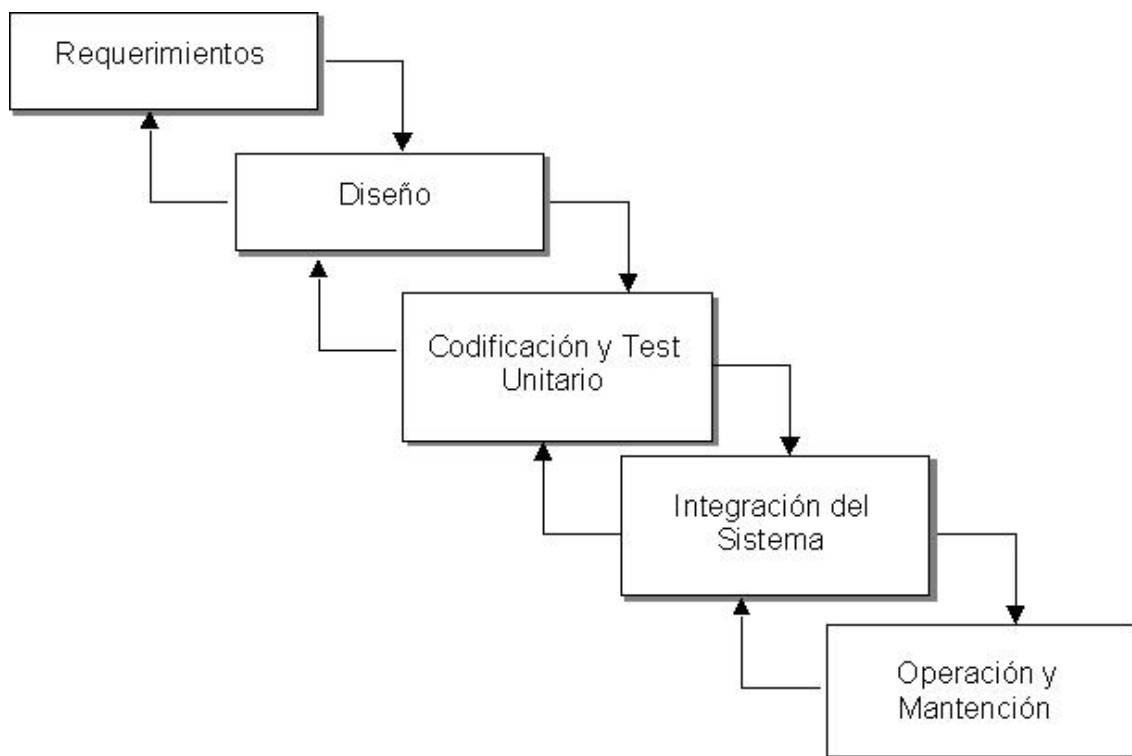


Figura. 1.2 Diagrama del modelo de cascada

VENTAJAS DEL MODELO DE CASCADA

- Su planificación es sencilla.
- La calidad del producto resultante es alta.
- Permite trabajar con personal poco calificado.

INCONVENIENTES DEL MODELO DE CASCADA

- Lo peor es la necesidad de tener todos los requisitos al principio. Mas sin en cambio lo normal es que el cliente no tiene perfectamente definidas las especificaciones, o por otro lado que surjan necesidades imprevistas.
- Si se han cometido errores en una fase es difícil volver atrás.
- El producto se tiene hasta el final.
- Es comparativamente más lento que los demás.

TIPOS DE PROYECTOS PARA LOS QUE ES ADECUADO EL MODELO DE CASCADA

- Aquellos para los que se dispone de todas las especificaciones desde el principio.
- El desarrollo de un producto que no es novedoso.
- Proyectos que aunque complejos se comprendan bien desde el principio.

VARIANTES DEL MODELO DE CASCADA

Como el modelo de cascada ha sido muy popular, pues al paso de los años ha generado algunas variantes como son:

- Ciclo de vida en V. Propuesto por Alan Davis. Una fase además de utilizarse como entrada para la siguiente, sirve para validar o verificar otras fases posteriores.
- Ciclo de vida Sashimi. Se permite un solapamiento entre fases, es decir no necesariamente debe terminar una fase antes de comenzar la otra.
- Ciclo de vida en cascada incremental. Se va creando el sistema añadiendo pequeñas funcionalidades. Cada uno de los pequeños incrementos es parecido a lo que ocurre dentro de la fase de mantenimiento.
- Ciclo de vida en cascada con reducción de riesgos. Realiza un desarrollo iterativo durante las fases de análisis y diseño, esto es para descubrir el mal entendimiento de requisitos antes de entregar el producto.

1.2 BASES DE DATOS

Son muy numerosas las definiciones de bases de datos, y si se analiza detenidamente, se observa en todas ellas coincidencias en ciertos elementos. En el (cuadro1.1) se reproducen distintas definiciones de bases de datos.

- “Colección de datos interrelacionados almacenados en conjunto sin redundancia perjudiciales o innecesarias; su finalidad es servir a una aplicación o más, de la mejor manera posible; los datos se almacenan de modo que resulten independientes de los programas que los usan; se emplean métodos bien determinados para incluir nuevos datos y para modificar o extraer los datos almacenados.”
(Martin, 1975)
- “Colección o deposito de datos, donde los datos están lógicamente relacionados entre si, tienen una definición y descripción común y están estructurados de una forma particular. Una base de datos es también un modelo del mundo real y, como tal, debe poder servir para toda una gama de usos y aplicaciones.”
(Conference des Statisticiens Européens, 1977)
- “Conjunto de datos de la empresa memorizado en un ordenador que es utilizado por numerosas personas y cuya organización está regida por un modelo de datos.”
(Flory, 1982)
- “Conjunto estructurado de datos registrados sobre soportes accesibles por ordenador para satisfacer simultáneamente a varios usuarios de forma selectiva y en tiempo oportuno”
(Delobel, 1982)
- “Colección no redundante de datos que son compartidos por diferentes sistemas de aplicación.”
(Howe, 1983)
- “Colección integrada y generalizada de datos, estructurada atendiendo a las relaciones naturales de modo que suministre todos los caminos de acceso necesarios a cada unidad de datos con objeto de poder atender todas las necesidades de los diferentes usuarios.”
(Deen, 1985)
- “Conjunto de ficheros maestros, organizados y administrados de una manera flexible de modo que los ficheros puedan ser fácilmente adoptados a nuevas tareas imprevisibles.”
(Frank, 1988)
- “Colección de datos interrelacionados.”
(Elsmari y navathe, 1989)

Cuadro1.1 Diferentes definiciones de Base de Datos

1.2.1 INTRODUCCIÓN A LAS BASES DE DATOS

El termino Bases de Datos no apareció hasta mediados de los años sesenta, época en la cual la información era representada haciendo uso de un conjunto de archivos, generalmente planos.

El desarrollo sistemático de bases de datos comenzó con los modelos jerárquicos y de redes, que gozaron de popularidad en las décadas de 1960 y 1970. El modelo relacional es el preferido en la actualidad, mientras que los modelos orientados a objetos y deductivos marcan la tendencia así el futuro. Cualquiera que sea el modelo fundamental una base de datos especifica administra información proveniente de alguna aplicación real².

La independencia de la información con respecto a los procedimientos que la maneja debe satisfacerse a dos niveles de abstracción para que sea efectiva:

Independencia lógica de los datos, es la capacidad de modificar el esquema conceptual sin tener que alterar los esquemas externos no los programas de aplicación.

Independencia física de los datos, es la capacidad de modificar el esquema interno por la necesidad de reorganizar ciertos archivos físicos a fin de mejorar el rendimiento de las operaciones de obtención o actualización.

1.2.2 CLASIFICACIÓN DE LOS MODELOS DE DATOS

Un modelo de datos es un conjunto de conceptos que permiten describir a distintos niveles de abstracción. Es básicamente una descripción de algo conocido como contenedor de datos así como de los métodos para almacenar y recuperar información de esos contenedores.

BASE DE DATOS JERÁRQUICA

Las bases de datos jerárquicas fueron concebidas en los años 1960. Uno de sus pioneros más importante fue Charles Bachman.

² Piattini, M., *Fundamentos y modelos de BASES de DATOS*. Ed. Alfaomega.

Conocido comúnmente como el sistema IMS de IBM. Establece una jerarquía de fichas, de manera que cada ficha puede contener a su vez listas de otras fichas y así sucesivamente. De este modo podemos indicar que su organización es en forma similar a la de un árbol (visto al revés).

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables.

BASE DE DATOS DE RED

Este modelo lo podemos considerar como una potencia media entre el jerárquico y el relacional, aunque su estructura es parecida al jerárquico pero es más compleja, puesto que permite que un mismo nodo tenga varios padres lo cual no lo hace el jerárquico y con ello se consigue evitar el problema de la redundancia de datos.

BASE DE DATOS RELACIONAL

Este modelo intenta representar la base de datos como un conjunto de tablas.

Para manipular la información utilizamos un lenguaje relacional, actualmente se cuenta con dos lenguajes formales el Álgebra relacional (permite describir la forma de realizar una consulta) y el Cálculo relacional (indica lo que se desea resolver).

BASE DE DATOS ORIENTADAS A OBJETOS

En una base de datos Orientada a Objetos, la información se presenta mediante objetos como los presentes en la programación Orientada a Objetos. Cuando se integra las características de una base de datos con las de un lenguaje de programación orientado a objetos, el resultado es un Sistema Gestor de Base de Datos Orientada a Objetos (ODBMS).

Los ODBMS son una buena elección para aquellos sistemas que necesitan un buen rendimiento en la manipulación de tipos de datos complejos.

Los ODBMS proporcionan los costes de desarrollo más bajos y el mejor rendimiento cuando se usan objetos.

1.2.3 MODELO ENTIDAD - RELACIÓN

El modelo de datos relacional, presentado por el Dr. E.F. Codd en 1970, en su célebre artículo de ACM titulado “Un modelo de datos relacional para grandes bancos de datos compartidos”.

El modelo relacional propone como principal objetivo, aislar al usuario de las estructuras físicas de los datos, consiguiendo así la independencia de las aplicaciones respecto de los datos, una finalidad que fue perseguida desde el principio de los datos.

Los avances más importantes que el modelo de datos relacional incorpora respecto a los modelos de datos anteriores son:

- Sencillez y uniformidad: Los usuarios ven la base de datos relacional como una colección de tablas, y al ser la tabla la estructura fundamental del modelo, éste goza de una gran uniformidad.
- Sólida fundamentación teórica: Al ser definido con rigor matemático, el diseño y la evaluación del mismo puede realizarse por métodos sistemáticos basados en abstracción.
- Independencia de la interfaz del usuario: Al manipular conjuntos de registros, proporcionan una gran independencia respecto a la forma en la que los datos están almacenados.

En un SGBD relacional pueden existir varios tipos de relaciones, aunque no todos manejan todos los tipos:

- Relaciones base. Son relaciones reales que tienen nombre y forman parte directa de la base de datos almacenada (son autónomas).
- Vistas. También denominadas relaciones virtuales son relaciones con nombre y derivadas: se representan mediante su definición en términos de otras relaciones con nombre, no poseen datos almacenados propios.
- Instantáneas. Son relaciones con nombre y derivadas. Pero a diferencia de las vistas, son reales, no virtuales: están representadas no solo por su definición en términos de otras relaciones con nombre, sino también por sus propios datos almacenados. Son relaciones de solo lectura y se refrescan periódicamente.
- Resultados de consultas. Son las relaciones consultantes de alguna consulta específica. Pueden o no tener nombre y no persisten en la base de datos.
- Resultados intermedios. Son las relaciones que contienen los resultados de las subconsultas. Normalmente no tienen nombre y tampoco persisten en la base de datos.
- Resultados temporales. Son relaciones con nombre, similares a las relaciones base o a las instantáneas, pero la diferencia es que se destruyen automáticamente en algún momento apropiado.

DEFINICIONES INFORMALES

Una relación es una tabla con columnas y filas. Un SGBD solo necesita que el usuario pueda percibir la base de datos como un conjunto de tablas. Esta percepción solo se aplica a la estructura lógica de la base de datos. No se aplica a la estructura física de la base de datos, que se puede implementar con distintas estructuras de almacenamiento.

Un atributo es el nombre de una columna de una relación. En el modelo relacional, las relaciones se utilizan para almacenar información sobre los objetos que se representan en la base de datos. Una relación se representa gráficamente como una tabla bidimensional en la que las filas corresponden a registros individuales y las columnas corresponden a los campos o atributos de esos registros. Los atributos pueden aparecer en la relación en cualquier orden.

Un dominio es el conjunto de valores legales de uno o varios atributos. Los dominios constituyen una poderosa característica del modelo relacional. Cada atributo de una base de datos relacional se define sobre un dominio, pudiendo haber varios atributos definidos sobre el mismo dominio.

El concepto de dominio es importante porque permite que el usuario defina, el significado y la fuente de los valores que los atributos pueden tomar.

Una tupla es una fila de una relación. Los elementos de una relación son las tuplas o filas de la tabla. Las tuplas de una relación no siguen ningún orden.

El grado de una relación es el número de atributos que contiene. El grado de una relación no cambia con frecuencia.

La cardinalidad de una relación es el número de tuplas que contiene. Ya que en las relaciones se van insertando y borrando tuplas a menudo, la cardinalidad de la misma varía constantemente.

Claves de relaciones. Una clave es una serie de atributos, cuyo valor identifica un renglón único en una relación. Así, dos renglones o n tuplos no pueden tener el mismo valor de clave, lo que a veces se le llama propiedad de unicidad.

Superclaves. Una superclave puede incluir cualquier número de atributos que posean la propiedad de unicidad.

Claves de relación. Una clave de relación se define formalmente como un conjunto de atributos de relación encadenados para que las siguientes tres propiedades se mantengan en todo momento y para cualquier caso de relación:

1. **Unicidad:** El conjunto de atributos tiene un valor único para cada tuplo en la relación.
2. **No redundancia:** Si un atributo se elimina de la serie, los demás no tendrán la propiedad de unicidad.
3. **Validez:** No puede ser nulo ningún valor de atributo clave.

Atributos primos y no primos. Un atributo primo es parte de por lo menos una clave de relación y un atributo no primo no es un integrante de ninguna clave de relación.

Claves extranjeras. El termino claves extranjeras se usa en relación con la integridad de la base de datos. Una clave extranjera en la relación R1 es un conjunto de 1 o más atributos que están en una clave de relación en otra relación, R2; pero no una clave de relación R1.

Una base de datos relacional es un conjunto de relaciones normalizadas.

EL DICCIONARIO DE DATOS

Es un conjunto de archivos que contienen información acerca de los datos que pueden ser almacenados en la base de datos. Se trata de una *metabase de datos*.

En el diccionario de datos se almacenan todas las definiciones realizadas por el DDL sobre el problema que va a ser tratado por el SGBD y, algunas de las realizadas por el DML. Así, en el diccionario de datos se encuentra almacenado:

- El esquema lógico de la base de datos
- El esquema físico de la base de datos
- Los subesquemas de la base de datos

NORMALIZACIÓN

Es evidente que una base de datos no puede estar constituida por una única relación con todos los atributos, ya que ello daría lugar a una gran cantidad de redundancias provocando las siguientes anomalías:

- Incapacidad para almacenar ciertos hechos
- Redundancias y, por tanto, posibilidad de inconsistencia
- Ambigüedad
- Pérdida de información
- Pérdida de dependencias funcionales, es decir, de ciertas restricciones de integridad que dan lugar a independencias entre los datos.
- Existencia de valores nulos
- Aparición, en la base de datos, de estados que no son válidos en el mundo real

Por ello se trata, de buscar un conjunto de esquemas que sean equivalentes al original pero que sea mejor. Para esto son tres las propiedades que ha de cumplir el esquema de relación para ser equivalente y mejor; y estas propiedades son:

- a) Conservación de la información.
- b) Conservación de las dependencias.
- c) Mínima redundancia de los datos (Normalización de las relaciones)

Si se cumplen las dos primeras propiedades se dice entonces que la relación obtenida $\{R_i\}$ y la inicial R son equivalentes:

$\{R_i\}$ es **equivalente** a R

Y si las relaciones del esquema relacional resultante $\{R_i\}$ están en forma normales más avanzadas que el esquema origen R , se dice que en este sentido de nivel de normalización y, por tanto, de eliminación de redundancia de datos:

$\{R_i\}$ es **mejor** que R

Así pues el conjunto R_i de esquemas resultantes en un proceso debe alcanzar un nivel de normalización a fin de eliminar las posibles redundancias y, por tanto, las anomalías de actualización.

Cuanto mas alta sea la forma normal en la que se encuentra el esquema de relación, menores serán los problemas que aparecen en el mantenimiento de la base de datos.

Codd propuso inicialmente tres formas normales basadas en las dependencias funcionales: primera forma normal (1FN), segunda forma normal (2FN) y tercera forma normal (3FN).

Debido a que aun persistían los problemas en relaciones en 3FN, Codd, en 1974, introdujo una definición mas restrictiva de la tercera forma normal, que se denominó Forma Normal de Bobice-Codd (FNBC).

Fagin en 1977 introduce la cuarta forma normal (4FN) y posteriormente en 1979 la quinta (5FN), basadas en otro tipo de dependencias distintas de las funcionales: las dependencias multivaluadas y las dependencias de proyección-combinación, respectivamente.

1.2.4 FORMAS NORMALES

Se encargan de reducir los problemas que puedan mostrar las tablas, con una serie de reglas establecidas. Contamos con 6 formas normales, pero es considerado que tan solo con cumplir con las primeras tres formas normales es mas que suficiente para corregir problemas que se presentan en una base de datos.

PRIMERA FORMA NORMAL FN1

Una relación R satisface la primera forma normal (FN1) si, y solo si, todos los dominios subyacentes de la relación R contienen valores atómicos, es decir si todos los dominios son simples.

Se considera que una relación se encuentra en FN1 cuando cumple lo siguiente:

1. Las celdas de las tablas poseen valores simples y no se permiten grupos ni arreglos repetidos como valores, es decir, conteniendo un solo valor por cada celda.
2. Todos los ingresos en cualquier columna (atributo) deben ser del mismo tipo.
3. Cada columna debe tener un nombre único, el orden de las columnas en la tabla no es importante.

4. Dos filas o renglones de una misma tabla no deben ser idénticas, aunque el orden de las filas no es importante.

Por lo general la mayoría de las relaciones cumplen con estas características, así que podemos decir que la mayoría de las relaciones se encuentran en la primera forma normal.

SEGUNDA FORMA NORMAL FN2

Es una dependencia funcional: consiste en edificar que atributos dependen de otros atributos.

Una relación R satisface la segunda forma normal (FN2) si, y solo si, satisface la primera forma normal y cada atributo de la relación depende funcionalmente de forma completa de la clave primaria de esa relación. De acuerdo con esta definición, cada tabla que contiene un atributo único como clave, esta en segunda forma normal.

TERCERA FORMA NORMAL FN3

Una relación R satisface la tercera forma normal (FN3) si, y solo si, satisface la segunda forma normal y los atributos no primos dependen no transitivamente de la llave primaria.

Consiste en eliminar la dependencia transitiva que queda en una segunda forma normal, en pocas palabras una relación esta en tercera forma normal si no existen dependencias transitivas entre los atributos, se refiere a dependencias transitiva cuando existe mas de una forma de llegar a referencias a un atributo de una relación.

LA FORMA NORMAL DE ÓBICE – CODD FNBC

Las relaciones que satisfacen la restricción impuesta por la forma normal FNBC satisfacen la FN2 y FN3. La FNBC se basa en el concepto de Determinante funcional, y está soportada en las características de las claves candidatas de las relaciones³.

Se denomina determinante funcional a uno o un conjunto de atributos de una relación R del cual depende funcionalmente de forma completa algún otro atributo de la misma relación.

Se puede expresar la forma normal de Óbice – Codd (FNBC) si, y sólo si, se encuentra en FN1, y cada determinante funcional es una clave candidata de la relación R.

³ Rivero, E. (1988) Bases de Datos Relacionales, Ed. Paraninfo.

CUARTA FORMA NORMAL FN4

La regla de normalización FN4 está basada en la eliminación de las relaciones de un tipo especial de dependencias denominadas Dependencias Multivaluadas (DMV), las cuales fueron consideradas por primera vez por Fagin y Zamiolo en 1977, y se pueden definir de la forma siguiente:

Dada una relación R, se dice que el atributo R.y \in R depende de forma multivaluada de otro atributo R.x \in R, o que R.x multidetermina a R.y, y se expresa de la forma R.x \rightarrow R.y si y solo si, cada valor de R.x tiene asignado un conjunto bien definido de valores de R.y y este conjunto es independiente de cualquier valor que tome otro atributo R.z \in R, el cual depende del valor de R.x

QUINTA FORMA NORMAL FN5

Un esquema de relaciones R esta en 5FN con respecto a un conjunto D de dependencias funcionales, de valores múltiples y de producto, si para todas las dependencias de productos en D se cumple por lo menos una de las siguientes condiciones:

- (R1, R2, R3, ..., Rn) es una dependencia de producto trivial.
- Toda Ri es una superllave de R.

La quinta forma normal se refiere a dependencias que son extrañas. Tiene que ver con tablas que pueden dividirse en subtablas, pero que no pueden reconstruirse.

1.2.5 DIAGRAMA

El diagrama es la manera más fácil de representar un sistema para indicarle al usuario las distintas etapas y condiciones con que cumple o debe cumplir dicho sistema.

DIAGRAMAS DE LA ENTIDAD – RELACIÓN

Una aplicación relacional utiliza dos notaciones distintas para describir tablas, relaciones y restricciones. La primera notación sirve a las necesidades de los diseñadores de bases de datos.

La segunda notación es un guión que la maquina puede leer y que presenta el diseño final al sistema de administración de bases de datos.

ENTIDADES Y SUS ATRIBUTOS EN DIAGRAMAS E-R

Un diagrama de entidad – relación (ER) muestra una entidad de aplicación como rectángulo, con los atributos en óvalo circundantes, cada uno conectado al rectángulo con una línea. El rectángulo contiene el nombre de la entidad.

Un atributo subrayado es un atributo de clave.

Un óvalo de línea interrumpida indica un atributo derivado, es decir, aquel que no está explícitamente almacenado porque puede ser calculado a partir de los valores de otros atributos. Un atributo de valores múltiples aparece en un doble óvalo, y un atributo compuesto exhibe sub óvalos para sus componentes (Figura 1.3).

Los diagramas ER no están asociados con ningún modelo de bases de datos en particular. Una vez que un diagrama ER capta la aplicación, el diseñador lo convierte en un esquema de bases de datos para el sistema disponible⁴.

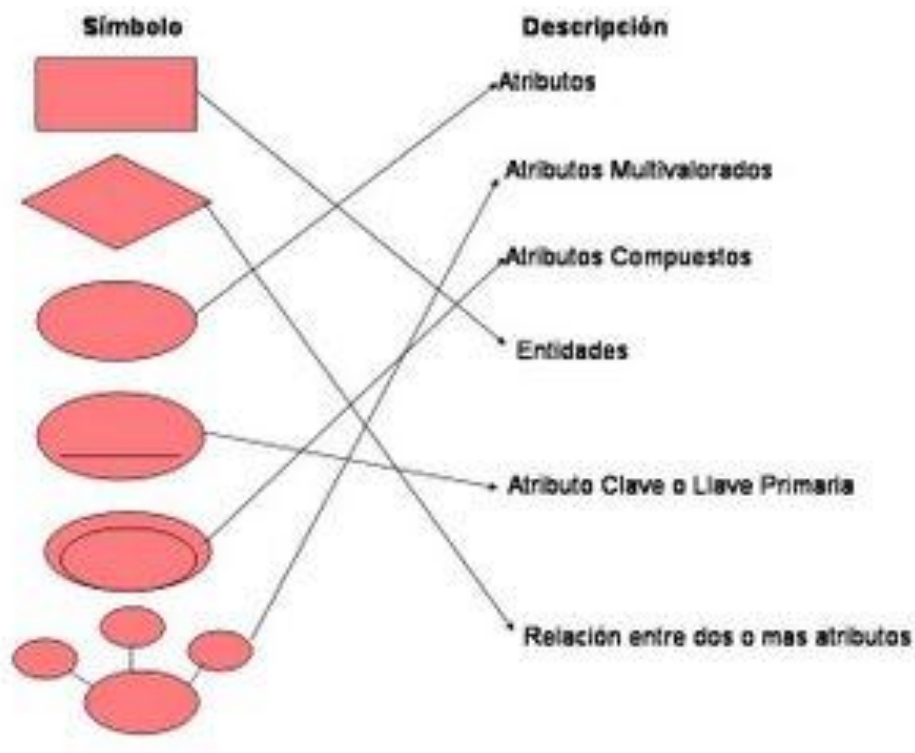


Figura.1.3 Simbología diagrama relacional

Otro formato más conocido. Un solo rectángulo contiene el nombre de la entidad y sus atributos, y dispositivos rotacionales distinguen los diversos tipos de atributos. La convención de subrayar marca la llave, y un asterisco indica un atributo derivado. Los

⁴ Abraham Silberschatz, Henry F. Korth, S. Sudarshan.,(2002), *Fundamentos de Bases de Datos*. Mc Graw Hill.

puntos suspensivos denotan atributos de valores múltiples, y el esquema de sangría elabora atributos compuestos (Figura 1.4).

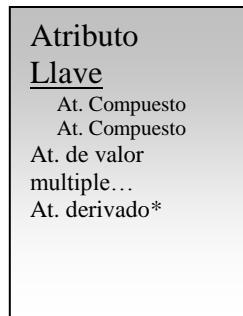


Figura 1.4 Indicado la llave en el diagrama

* Una llave primaria es aquel atributo el cual consideramos clave para la identificación de los demás atributos que describa la entidad.

Claro que puede haber más de un atributo que pueda identificarse como llave primaria en este caso se selecciona la que consideramos más importante, los demás atributos son denominados llaves secundarias.

RELACIONES BINARIAS EN DIAGRAMAS ER

En un diagrama de entidad - relación (ER), aparece una relación como un diamante que contiene el nombre de la relación. Unas líneas enlazan el diamante de relación en las entidades participantes. Normalmente, las líneas emanan de los vértices del diamante, que parece que limita la notación a relaciones de grado 4 o menos; pero ninguna regla expresa que las líneas deben originarse en un vértice y, por tanto, las relaciones de grado 5 o mayor son posibles.

La mayor parte de las relaciones son binarias; ocasionalmente aparece una relación ternaria. Debido a que las relaciones de grado más alto son raras, los cuatro vértices del diamante suelen ser más que suficientes.

La liga con la flecha la utilizamos cuando se desea aclarar la relación de uno a muchos donde la punta de la flecha indica la relación dominante, y el otro extremo de la ausencia de la flecha, indica la parte no dominante.

Un 1 en ambos lados indica relación uno a uno, y las letras M, N representan la relación de muchos a muchos. Las designaciones pueden ser 1:1, 1:N y M:N(Figura 1.5).

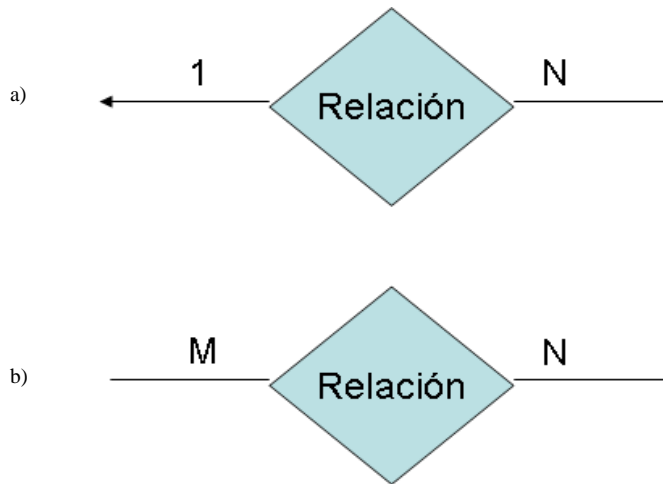


Figura 1.5 a) Relación uno a muchos, b) Relación muchos a muchos

Se sitúan límites de participación cerca de una línea de conexión donde enlace una entidad. Un límite de participación contiene dos números (x,y), donde x da el número mínimo de pares en que un objeto de entidad determinado puede participar y donde y da el número máximo. La letra n significa “muchos”, cuando el número exacto no es importante. Es decir, n significa 0, 1 o más (Figura 1.6).



Figura 1.6 La relación puede ser desde 0, 1 o mas

1.3 PLATAFORMA CLIENTE SERVIDOR

Esta es una red LAN, la cual contiene un computador especial que llamamos un servidor, y los computadores personales son los clientes, que solicitan al servidor diferentes servicios.

El servidor recupera y actualiza los datos.

1.3.1 SERVIDOR DE BASES DE DATOS

Es un programa que corre en un computador servidor para dar servicios de bases de datos a las maquinas clientes.

Inicialmente los servidores se instalaron para controlar la impresión y el acceso a los archivos. Estos constituyeron los servidores de impresión y los servidores de archivos (Figura 1.7).

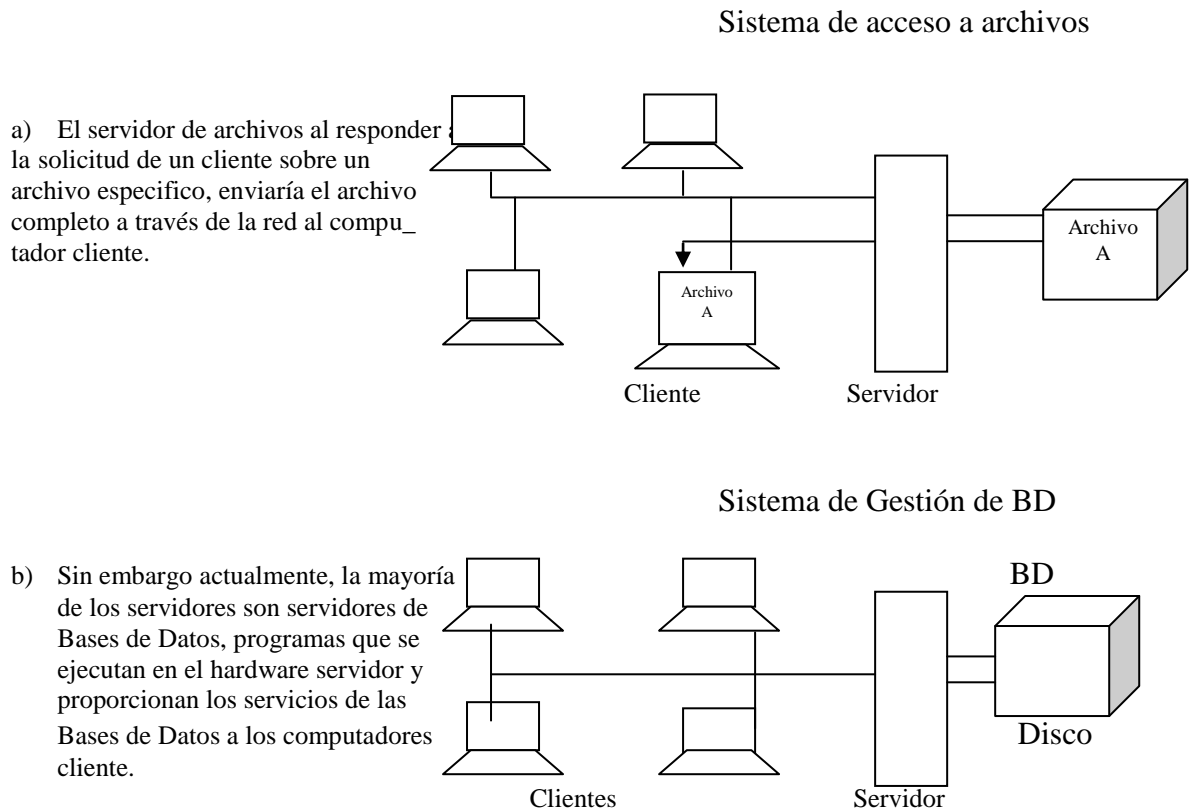


Figura 1.7 a) Sistema de acceso a archivos, b) Sistema de Gestión de BD

De esta manera, un cliente que está ejecutando un proceso de una aplicación y necesita una interrogación (consulta) a una Base de Datos, envía la petición al servidor de la B.D. y este le devolverá los datos solicitados. El programa de aplicación puede enviar también datos al servidor con la solicitud de actualizar la B.D. El servidor efectuaría esta actualización.

La potencia de la plataforma cliente/servidor descansa en el concepto de división de funciones. El cliente es el computador frontal que tiene interfaz directamente con el usuario. Manipula la Interfaz Gráfica de Usuario (GUI) y realiza los cálculos y otros programas de interés para usuario final. El servidor es el que gestiona

fundamentalmente el acceso a los datos atendiendo las diferentes peticiones de los clientes, también realiza funciones de control y seguridad de acceso a los datos.

Las plataformas cliente/servidor son parte del concepto de sistemas abiertos, en el cual todo tipo de computadores, *sistemas operativos*, protocolos de redes y otros *software* y *hardware* pueden interconectarse y trabajar coordinadamente para lograr los objetivos del usuario.

El objetivo de los sistemas abiertos consiste en lograr la interoperabilidad, que es el estado de dos o más sistemas heterogéneos comunicándose y contribuyendo cada uno a alguna parte del trabajo que corresponde a una tarea común.

CAPITULO DOS

ANÁLISIS DEL Sistema de Ayuda Médica (SAMÉD)

Se realiza un análisis con respecto a las necesidades del usuario, al cual se le va indicando paso a paso, si el sistema que se está planeando, cumple con lo que el requiere.

Toda la documentación se trata de representar de manera narrada, para que el usuario comprenda a la perfección, sin necesidad de complicarse con términos computacionales. Del mismo modo se muestra el cuestionario que se le aplicó al nutriólogo, para tratar de obtener la mayor información posible con respecto al deseo del sistema.

2.1 PLANTEAMIENTO DEL PROBLEMA

La computación brindado apoyo hacia muchas áreas en la medicina ha sido solo muy poco lo que le han permitido aportar, tan solo como base de datos pero hacia el manejo de información administrativa, puesto que crea un temor de seguridad y fiabilidad en cuanto al apoyo de diagnósticos médicos.

Por ello el sistema SAMÉD pretende inmiscuirse en el diagnóstico médico, mas sin en cambio asesorado y supervisado 100% por el médico encargado de la atención del paciente.

2.2 ESTUDIO DE VIABILIDAD

Realizamos la investigación para que SAMÉD cumpla con los requerimientos especificados por nuestro cliente y llevar dicho sistema a su total funcionalidad.

2.2.1 VIABILIDAD TECNOLÓGICA

Se cuenta con la tecnología adecuada disponible para poder dar un soporte al desarrollo de la Base de Datos, esto es porque el sistema es de alguna manera amigable, fácil y sencillo de elaborar.

2.2.2 VIABILIDAD OPERACIONAL

El personal quien dará uso del sistema (SAMED) será directamente el médico Nutriólogo, quien comprenderá perfectamente al sistema por ser enfocado y enfocado a su área.

No se requiere de grandes conocimientos de computación, puesto que (SAMED) es de consulta y lo mas básico posible, tan fácil de utilizar como guiarse de libros ó de las tablas de peso colocadas en los consultorios.

2.2.3 VIABILIDAD ECONÓMICA

Los beneficios que mostrara SAMED serán cuestión del tiempo de la consulta, puesto que realizara una opinión en cuestión a las enfermedades en muy corto tiempo, y el análisis se realizara como si se tratara del apoyo de un grupo de nutriólogos en un mismo tiempo. Por lo tanto dicho beneficio puede en realidad costear al sistema puesto que al reducir el tiempo de consulta, podrán aumentar el número de pacientes consultados al día. Además de realizar un diagnóstico mas preciso y mas rápido sin necesidad de posponer o retardar el diagnóstico por alguna duda posible. Esto dará mayor confianza al paciente para una próxima cita médica.

Por tales motivos, se podrá en un tiempo determinado realizar una evaluación y comparación grafica, para verificar el beneficio y el costo. Indicando así una mejora o en el peor de los casos una perdida.

Además de lo expuesto el cliente mostró algunas inquietudes:

Como el saber si el sistema esta actualmente siendo manejado por la competencia. De lo cual se informo que actualmente hay sistemas similares pero ninguno enfocado de esta manera a la nutrición y menos como apoyo médico

2.3 ESPECIFICACIÓN DE REQUERIMIENTOS

Realizaremos una descripción del comportamiento del sistema que se está desarrollando. Con ello se tendrá una fácil comprensión para todas las partes involucradas en el sistema.

2.3.1 DOCUMENTO DE REQUERIMIENTOS DE USUARIO

Por principio se debe tener en el sistema una captura de solo algunos datos básicos para control del paciente. Esto consta de una tabla que contiene el Nombre del paciente, sexo, edad, teléfono, e-mail y un historial clínico en donde se realiza el reporte de la consulta hacia dicho paciente.

También se utilizan tablas con respecto al manejo de nutrimentos, donde es necesario saber el nutrimento que se maneja y a que alimento pertenece.

Por otro lado se contara con el manejo de diagnóstico de enfermedades, donde la tabla cuenta con los síntomas que presenta el paciente y la falta de nutriente a que se atribuye dicho síntoma.

Y por ultimo se contendrán dos tipos de tablas:

1 En donde se realiza un Análisis Clínico. En este se deben tomar en cuenta los atributos de: Edad, sexo, Peso mínimo, Peso máximo, Estatura mínima y estatura máxima.

2 Chequeo Masa Corporal. Solo manejaremos 4 atributos como son: La edad, sexo, masa corporal mínima y masa corporal máxima.

El encargado de operar el sistema, deberá tomarlo únicamente como lo que es un sistema de ayuda, una herramienta de salud. El doctor deberá aplicar sus conocimientos para el control proteínico necesario para el paciente, y para señalar si los síntomas presentados deben ser atendidos por el o por algún otro especialista.

El sistema tendrá acceso restringido por un usuario y contraseña, esto para que de ninguna manera alguna persona ajena al área, realice intentos por utilizar dicho sistema sin autorización alguna.

El sistema se ubicara en red para poder ser utilizado por un Médico desde cualquier lugar que tenga acceso a una computadora e Internet.

2.3.2 DESCRIPCIÓN DETALLADA DEL SISTEMA

A continuación se narrara de manera sencilla el sistema, se ara una descripción de todas sus entidades:

- Historial: Es en donde el doctor lleva un control de las visitas del paciente, indicando sus síntomas, enfermedad y los signos vitales, además de poder hacer anotaciones que considere importantes para un apoyo a futuras consultas.
Y sus atributos son: el id del historial, la estatura del paciente, el peso del paciente, su masa corporal (la cual se obtendrá automáticamente por la computadora con el peso y la estatura), comentarios del doctor con respecto a la consulta que ha realizado, la fecha en que se realiza la consulta (la que obtendrá automáticamente el sistema con respecto a la fecha de la computadora) y el nombre del doctor como referencia para futuras consultas.
- Paciente: Esta entidad se encarga de un control de los pacientes que son atendidos, esto sirve para llevar un historial clínico ya que en la etapa preescolar (0 años a 5 años) es primordial para plantar los cimientos de las enfermedades que tenga el infante, complicaciones en caso de haberlas y algunas anotaciones extras, que sirvan de ayuda para cualquier otro médico que lo llegue a atender posteriormente.
Los atributos que en esta entidad manejaremos son: id_paciente (para darle un número de registro único), Nombre del paciente, sexo, edad, además de contener el número telefónico del padre y su e-mail estos últimos para poder dar alguna indicación sobre el paciente.
- Enfermedad: Esta entidad trata de todas las enfermedades relacionadas hacia sus síntomas y los nutrientes que pueden aliviarla.
Los atributos que manejaremos en la entidad enfermedad son: id_ enfermedad (para identificarla dentro del sistema de manera única) y el nombre de la enfermedad.
- Función del Nutriente: Es una explicación de todo lo q puede realizar cada nutriente.
Los atributos manejados son Id_funcion del nutriente y la función de cada nutriente.
- Fuentes: Son los alimentos de donde podemos obtener cada uno de los nutrientes pues no olvidemos que un solo alimento puede contener más de una vitamina
Los atributos con que cuenta son su id y el nombre del alimento.
- Nutriente: Esta tabla llevara un control para conocer cada uno de los nutrientes y su relación que llevan tanto con las enfermedades como con las funciones que cumplen cada uno de ellos.
Los atributos que manejamos son: id_nutriente y el nombre del nutriente.

ENCUESTA AL CLIENTE DEL SISTEMA

P ¿Tiene usted alguna tabla relacionada con los síntomas de las enfermedades?

R= Dentro de la licenciatura, hay literatura de apoyo en la cual se dan a conocer las enfermedades y es así como nosotros tenemos conocimientos de ellas en la vida profesional, en algún caso de tener alguna duda, se acude a la literatura para tomarla de apoyo aunque en la mayoría de las ocasiones no se tiene el tiempo ni el espacio para contar con los libros necesarios.

P ¿Los síntomas pueden variar dependiendo de la persona?

R= En la formación que tenemos tanto teórica como practica, se tiene una especie de lema: “no hay enfermedades, si no enfermos.” Ya que una misma enfermedad puede variar, en algunas ocasiones no de una manera considerable, hablando de pacientes, ya sea crónicos cuya enfermedad tiene un curso idéntico que la mayoría que tienen la

misma patología, o bien personas cuyas enfermedades son simples y se consideran sanos en los demás aspectos fisiológicos.

P ¿Qué tan necesario es el chequeo nutrimental par un niño?

R= elemental en los primeros cinco años de su vida

P ¿Necesita tener acceso restringido al sistema?

R= Es restringido, ya que se presentan fenómenos con mucha frecuencia de automedicación y eso es causado por los aparentes conocimientos que ellos creen tener de sus enfermedades, en el caso de el equipo de enfermería, debido a su capacitación como auxiliares de médicos, no se les permite dar consultas ya que no reciben los conocimientos requeridos para interpretar los síntomas, y su experiencia no es lo suficientemente valida.

P ¿Qué se debe realizar cuando la enfermedad se encuentra fuera de sus conocimientos para darle el seguimiento preciso?

R= Se le realiza el expediente correspondiente, indicado el manejo que se le a brindado desde el comienzo de su enfermedad y se le canaliza con los estudios correspondientes a el médico especialista que cuente con el equipo y los conocimientos necesarios para el manejo del paciente.

P Si las enfermedades compaginan con alguna otra conocida fuera de sus dominios ¿deberá realizar análisis más profundos?

R= Exámenes, de gabinete, y pruebas auxiliares para diagnóstico de enfermedades, que según el estado del paciente tenga riesgo de padecer.

.

P ¿Antes de diagnosticar a la persona necesita revisar su historial?

R= Se tiene la obligación de revisar el expediente de un paciente ya que si, por ejemplo, el paciente tiene antecedentes de alergia a penicilina y el paciente no lo menciona puede haber un gran problema si se le administra el medicamento, o es un paciente diabético se le tiene que tomar en cuenta sus llamados antecedentes para tomar una decisión en el presente.

P ¿Desea que se muestre en el programa solo el nutriente deficiente de la persona, o todos los posibles aproximados con cierta prioridad?

R= Que se presenten los posibles nutrientes ya que debo revisar la cantidad en que se encuentre bajo de nutriente, y con respecto a ello podré indicar el mas necesario.

2.3.3 DOCUMENTO DE ESPECIFICACIÓN DEL SISTEMA

En el área de la salud es sabido que hay gran cantidad de enfermedades, y síntomas que se asemejan en muchas de ellas, por ello en algunas ocasiones es difícil para un médico el poder diagnosticar con rapidez y precisión el problema que pueda tener dicho paciente.

Del mismo modo sabemos que actualmente nos encontramos con problemas nutrimentarios por sobre peso, el cual se ha convertido en un problema grave. Y esto lo podemos solucionar si desde la niñez lo atendemos.

Por ello esta herramienta facilitara y agilizara ese trabajo al Doctor.

Doctores

- Obtienen los síntomas del paciente (“el mayor número de síntomas posibles”).
- Realizara una búsqueda y comparación de dichos síntomas.
- Al ubicar la enfermedad aplicando también su criterio.
 - o Buscara sus posibles soluciones
- Realizar un chequeo clínico ó de masa corporal del paciente para saber su estado físico de nutrición.
- Verificar una tabla de nutrientes, como apoyo para la elaboración de una dieta para cada paciente.
- Realizara una búsqueda del paciente por:
 - o Id_paciente
 - o Nombre
 - o Apellidos
 - o O cualquier dato personal que sirva para encontrarlo en caso de no conocer las especificaciones del mismo.
- Actualiza el historial del paciente

2.3.4 IDENTIFICAR LAS ENTIDADES

A continuación se ara mención acerca de las posibles entidades que se manejaran dentro del sistema. Estas se presentan abstractamente acorde a las necesidades que ha mencionado el cliente:

- o Enfermedad
- o Paciente
- o Nutriente
- o Fuentes
- o Funciones del Nutriente
- o Historial
- o Síntomas

2.4 DESCRIPCIÓN DE LA INFORMACIÓN

CASOS DE USO

El modelado de casos de uso es la técnica más efectiva y más simple para modelar los requisitos del sistema desde la perspectiva del usuario (figura 2.1).

El modelado de casos de uso consiste en actores y casos de uso. Los actores representan a los usuarios y a otros sistemas que interaccionan con el sistema. Se dibujan en estilo de “muñecos” de palo. Los casos de uso representan el comportamiento del sistema y se dibujan a manera de elipses.

2.4.1 IDENTIFICACION DE CASOS DE USO DEL SISTEMA

Casos de Uso	Iniciar Sesión
Actores	Usuario o Administrador
Descripción	El usuario o administrador escribirán su password para utilizar el sistema

Casos de Uso	Altas o bajas de Pacientes
Actores	Usuario o Administrador
Descripción	Realizaran un alta de los pacientes consultados.

Casos de Uso	Consultas o bajas de Pacientes
Actores	Usuario o Administrador
Descripción	Se podrá realizar una consulta del paciente por nombre, id, apellido o cualquier dato que sirva para identificarlo, además de poder dar de baja a dicho paciente si así se desea.

Casos de Uso	Actualizar Historial
Actores	Usuario
Descripción	El doctor tendrá que registrar los signos vitales, su nombre y algún comentario con respecto a la consulta del paciente.

Casos de Uso	Consulta de Historial
Actores	Usuario
Descripción	El doctor podrá consultar el historial clínico de los pacientes por medio del nombre del paciente.

Casos de Uso	Búsqueda de Enfermedad
Actores	Usuario
Descripción	El doctor ingresará los síntomas que muestre el paciente para poder obtener las posibles enfermedades que este contiene.

Casos de Uso	Búsqueda de Nutrientes
Actores	Usuario
Descripción	El doctor ingresará el nombre de algún nutriente del cual desee saber sus funciones y procedencias.

Casos de Uso	Chequeo de Peso
Actores	Usuario
Descripción	El doctor realizara una búsqueda por edad (en meses) con respecto al peso, estatura y sexo del paciente.

Casos de Uso	Alta o baja de Nutrientes
Actores	Administrador
Descripción	El administrador podrá dar de alta o baja algún nutriente del sistema

Casos de Uso	Alta o baja de Enfermedades
Actores	Administrador
Descripción	El administrador podrá dar de alta o baja alguna enfermedad del sistema

Casos de Uso	Alta o baja de Síntomas
Actores	Administrador
Descripción	El administrador podrá dar de alta o baja algún síntoma del sistema

Casos de Uso	Alta o baja de Fuentes
Actores	Administrador
Descripción	El administrador podrá dar de alta o baja alguna fuente de los nutrientes

Casos de Uso	Alta o baja de Funciones de los nutrientes
Actores	Administrador
Descripción	El administrador podrá dar de alta o baja de las funciones en que benefician los nutrientes

Identificación de casos de uso

DIAGRAMA DE CASOS DE USO

El diagrama se muestra como una segunda aproximación al sistema de forma general, en este diagrama se presentan los actores que van a interactuar directamente con el sistema y sus casos de uso sobre los cuales interactúa (Figura 2.1).

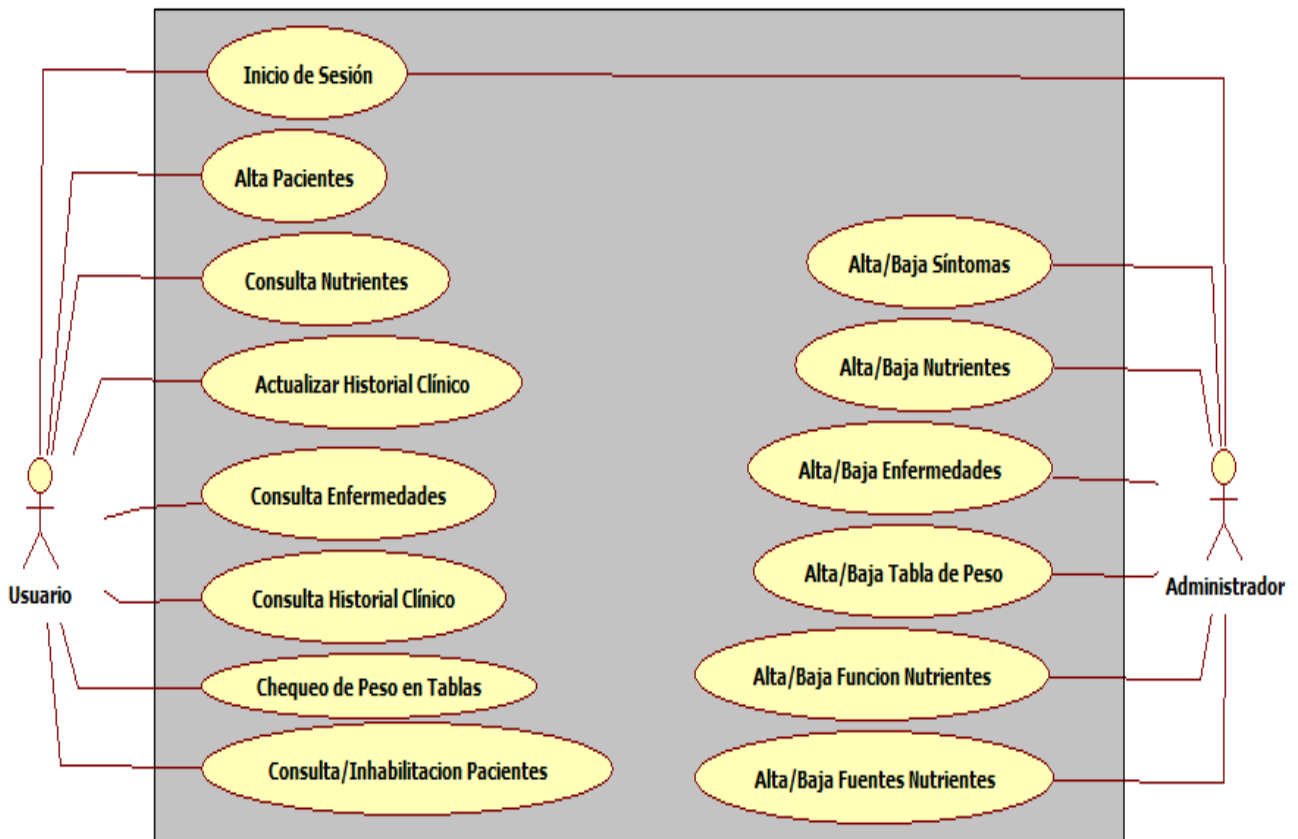


Figura 2.1 Diagrama de Casos de uso

2.4.2 IDENTIFICACIÓN DE RELACIONES

Aquí se toman las entidades y se concreta que asociaciones puede haber entre entidades. Estas asociaciones son consideradas en base al análisis previamente

realizado. Para ello se debe ser muy cuidadoso para no interpretar mal el funcionamiento del sistema.

Una asociación es la acción que realiza una entidad para otra entidad, así se obtuvieron los siguientes:

- Paciente lleva Historial
- Enfermedad tiene Síntomas
- Nutrientes alivian Enfermedad
- Nutrientes están en Fuentes.
- Nutrientes sirven para Funciones Nutrientes

2.4.3 IDENTIFICACIÓN DE ATRIBUTOS

- 1 Paciente (Nombre del paciente, sexo, Fecha de Nacimiento (Edad), teléfono y e-mail).
- 2 Historial (estatura del paciente, peso del paciente, masa corporal, redacción del doctor, fecha de la consulta y nombre del doctor)
- 3 Enfermedad (nombre de la enfermedad).
- 4 Nutriente (nombre de los nutrientes).
- 5 Síntomas (definición del síntoma).
- 6 Fuentes (listado de alimentos donde encontramos a los nutrientes).
- 7 Funciones del nutriente (listado de todas las funciones que beneficia el nutriente).

2.4.4 IDENTIFICADORES DE LAS ENTIDADES

Para cada una de las entidades es necesario asignar una clave única bien conocida como identificador, quien distinguirá perfectamente las descripciones entre todos ellos y ayudara a no tener problemas de redundancias.

Nombre del identificador	Entidad	Tipo
Id_pac	Paciente	entero
Id_his	Historial	entero
Id_enf	Enfermedad	entero
Id_fuente	Fuentes	Entero
Id_fun	Funciones nutrientes	Entero
Id_nut	Nutrientes	entero
Id_sin	Síntomas	Entero

2.5 ESQUEMA ENTIDAD – RELACIÓN

El modelo entidad relación (Figura 2.2) es una técnica para el modelado de datos utilizando diagramas entidad relación. Es una representación conceptual de nuestro sistema para observar más gráficamente las relaciones entre nuestras tablas.

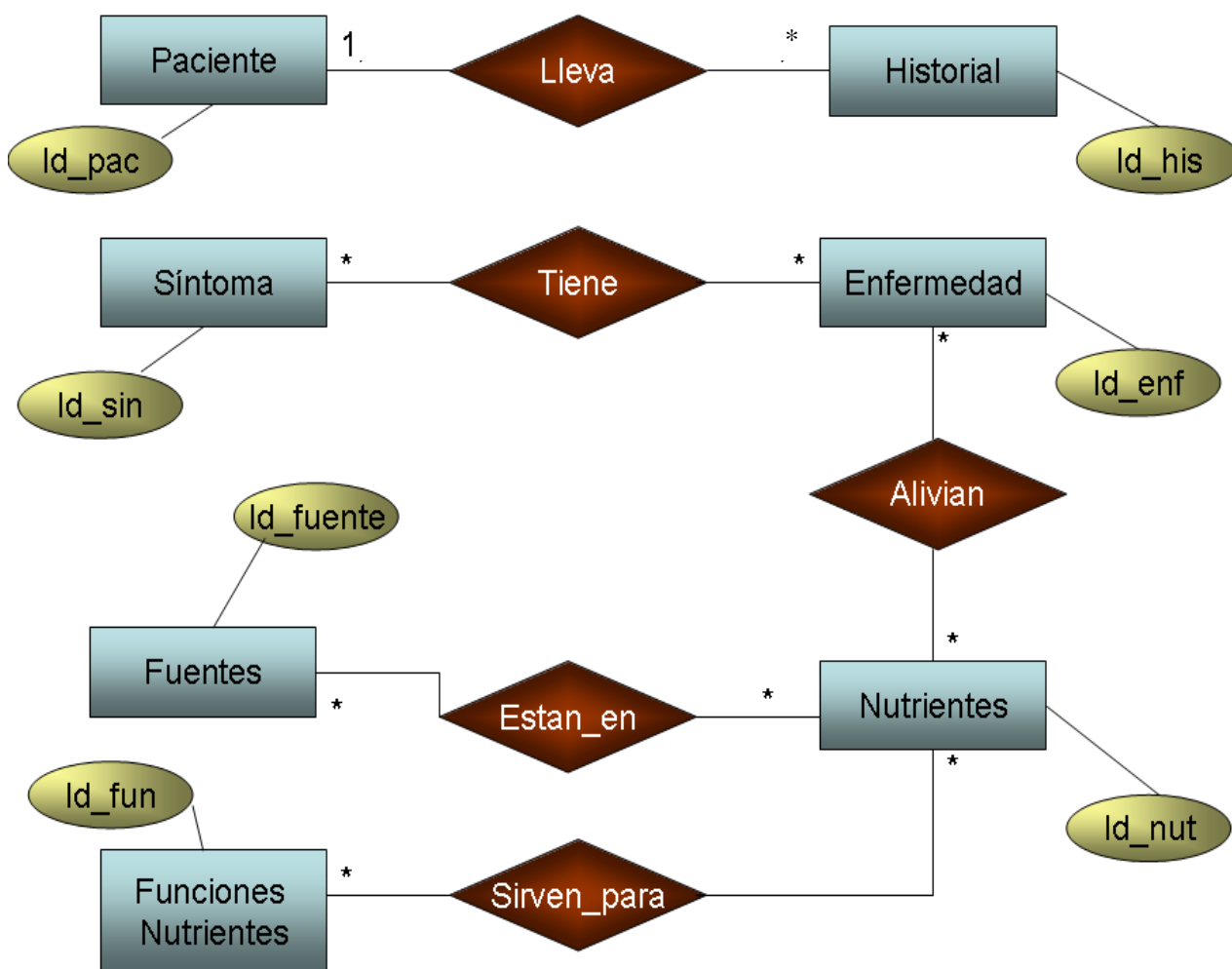


Figura 2.2 Diagrama entidad relación

2.5.1 REVISIÓN DEL ESQUEMA CON EL USUARIO

Se realizó una presentación del modelo conceptual con lo cual se obtuvieron resultados satisfactorios, pues hubo un entendimiento profundo con el usuario y una visión clara acerca de los resultados que se esperan.

El usuario hizo una revisión profunda y acepto que el sistema este enfocado en su totalidad al área medica y no al área administrativa como es usual.

Además remarco que es claro el sistema y simple apoyo o herramienta, ello para no tener problemas de auto consulta o automedicamento.

FUNCIONES BASICAS Y SECUNDARIAS

Las principales funciones a que se evoca el sistema son:

Apoyo de diagnóstico del paciente con respecto a las enfermedades

Y las funciones secundarias son:

Nutrientes

- Conocimiento de los nutrientes dentro de los alimentos.
- Chequeo del paciente con respecto a la tabla de peso ideal

Paciente

- Actualizar su historial medico.

2.6 DICcionario DE DATOS

Ya que se han conformado los atributos de las entidades, se probara hasta donde se puede llegar con la información de cada atributo o que información debe ser aceptada de una forma lógica y semántica.

1 PACIENTE

Nombre	Tipo
Id_Paciente(llave)	Numérico
Nombre Paciente (nom_pac)	Cadena
Sexo (sex)	Cadena
Edad (edad)	Date
Teléfono	Cadena
e-mail	Cadena

2 ENFERMEDAD

Nombre	Tipo
Id_Enfermedad(llave)	Numérico
Nombre de la enfermedad	Cadena

3 NUTRIENTE

Nombre	Tipo
Id_Nutriente(llave)	Numérico
Nombre del nutriente	Cadena

4 Historial

Nombre	Tipo
Id_Historial(llave)	Numérico
Estatura del paciente	Flotante
Peso del paciente	Flotante
Masa corporal	Flotante
Comentario médico	Texto
Fecha de la consulta	Date
Nombre del doctor	Cadena

5 FUENTES

Nombre	Tipo
Id_Fuente(llave)	Numérico
Fuente	Texto

6 FUNCIONES DEL NUTRIENTE

Nombre	Tipo
Id_Funcion(llave)	Numérico
Función	Texto

7 SÍNTOMAS

Nombre	Tipo
Síntoma	texto

8 CHEQUEO CLÍNICO

Nombre	Tipo
Id_Clinico (llave)	Numérico
Edad (edad)	Numérico
Peso mínimo varón	Numérico
Peso máximo varón	Numérico
Sobrepeso varón	Numérico
Desnutrición leve varón	Numérico
Estatura mínima varón	Numérico
Estatura máxima varón	Numérico
Peso mínimo mujer	Numérico
Peso máximo mujer	Numérico
Sobrepeso mujer	Numérico
Desnutrición leve mujer	Numérico
Estatura mínima mujer	Numérico
Estatura máxima mujer	Numérico

9 CHEQUEO DE MASA CORPORAL

Nombre	Tipo
Id_MC (llave)	Numérico
Edad (edad)	Numérico
Masa corporal mínima varón	Numérico
Masa corporal máxima varón	Numérico
Masa corporal mínima mujer	Numérico
Masa corporal máxima mujer	Numérico

CAPITULO TRES

DISEÑO DEL SAMED

Llevamos a cabo la normalización de nuestra base de datos para comenzar el diseño de la misma. Posteriormente se muestran las impresiones de pantalla con el sistema en funcionamiento.

3.1 DISEÑO LÓGICO

Es el proceso para construir un esquema de la información basándose en un modelo de bases de datos específico.

Aquí es donde se transforma el esquema conceptual en el esquema lógico que utilizara las estructuras de datos del modelo.

Para llevar a cabo la transformación del modelo entidad – relación al modelo relacional, se siguen las siguientes reglas:

- Toda entidad se convierte en una tabla.
- Toda relación M:N genera una nueva tabla que contenga las llaves primarias de ambas entidades.
- Toda relación 1:M incluye en la tabla M la llave primaria de la tabla 1.

3.2 REVISIÓN DE REQUERIMIENTOS

Al realizar un nuevo análisis de requerimientos, nos dimos cuenta de que ciertos detalles debieron refinarse, además de agregar algunos datos no considerados, y eliminar algunos más que no tenían mayor auge en el sistema solicitado, puesto que no cumplían con las condiciones solicitadas. Si mas adelante nos encontrásemos con más problemas, esto nos traerá gastos extras en todo el desarrollo del sistema.

Las principales preguntas que muestra el nutriólogo son las siguientes:

- ¿Cuántos Pacientes tengo?
- ¿Que alimentos contienen vitaminas y que tipo de vitaminas?
- ¿Que síntomas se asemejan en distintas enfermedades?
- ¿Cuándo catalogamos como sobrepeso o como obesidad?
- ¿Cuándo la desnutrición es leve y cuando es grave?

3.3 MODELO RELACIONAL

La estructura fundamental del modelo relacional es precisamente la de relacionar, es decir que una tabla bidimensional esta constituida por líneas (tuplas) y columnas (atributos). Las relaciones representan las entidades que se consideran interesantes en la base de datos.

Cada instancia de la entidad encontrara sitio en una tupla de la relación mientras que los atributos de la relación representan las propiedades de la entidad.

A continuación mostramos el modelo relacional de nuestro sistema:

PACIENTE (id_pac, nom_pac, sex_pac, edad, tel_pac, cor_pac)

HISTORIAL (id_his, est_pac, pes_pac, mc_pac, comentario, fecha_consulta, nom_doc)

FUENTES (id_fuente, fuente)

FUNCIONES NUTRIENTES (id_fun, funcion)

SÍNTOMA (id_sin, sintom)

ENFERMEDAD (id_enf, nom_enf)

NUTRIENTE (id_nut, nom_nut)

3.3.1 ELIMINACIÓN MUCHOS A MUCHOS N:M

Debemos asegurarnos de que nuestras tablas se relacionen entre si en la manera en que deben hacerlo. Para ello debemos seguir ciertas reglas, basadas en la interrelación en particular, para cada tabla.

Tenemos una regla que nos dice que si 2 tablas tienen una interrelación (*:*), entonces debe crearse una nueva tabla que tenga los campos claves de las 2 tablas.

Esta situación la encontramos entre las tablas NUTRIENTES – FUENTES, NUTRIENTES – FUNCIONES NUTRIENTES y ENFERMEDAD-NUTRIENTE.



Figura 3.1 Relación Nutrientes - Enfermedad

La Figura 3.1 muestra la relación: Alivian (id_nut, id_enf)



Figura 3.2 Relación Nutrientes - Fuentes

La Figura 3.2 muestra la relación: Estan_en (id_nut, id_fuente)



Figura 3.3 Relación Nutrientes – Funciones Nutrientes

La Figura 3.3 muestra la relación: Sirven_para (id_fun, id_nut)

Del mismo modo sucede con la relación ENFERMEDAD – SÍNTOMA:



Figura 3.4 Relación Enfermedad - Sintoma

La Figura 3.4 muestra la relación: Tiene (id_enf, id_sin)

CREACIÓN DE UNA ENTIDAD PARA CADA RELACIÓN FUERTE

Entidades Fuertes.- Son entidades básicas del sistema SAMED

PACIENTE

| id_pac | nom_pac | sex_pac | edad | tel_pac | cor_pac |

HISTORIAL

| id_his | est_pac | pes_pac | mc_pac | comentario | fecha_consulta | nom_doc |

FUENTES

| id_fuente | fuente |

FUNCIONES NUTRIENTES

| id_fun | funcion |

SÍNTOMA

| id_sin | sintom |

ENFERMEDAD

| id_enf | nom_enf |

NUTRIENTE

| id_nut | nom_nut|

CREACIÓN DE UNA ENTIDAD PARA CADA RELACIÓN DÉBIL

Entidades débiles.- Son aquellas entidades que se obtienen de una relación de muchos a muchos N:M

TIENE

| id_tiene | id_sintoma | id_enfermedad |

ESTAN_EN

| id_estan_en | id_nut | id_fuente |

SIRVEN_PARA

| id_sirven_para | id_nut | id_fun |

ALIVIAN

| id_alivian | id_nut | id_enf |

CREACIÓN DE ENTIDADES DE JERARQUÍAS DE GENERALIZACIÓN

En las jerarquías, se denomina entidad padre a la entidad genérica y entidades hijo a las subentidades.

3.3.2 VALIDACIÓN DE ESQUEMAS MEDIANTE LA NORMALIZACIÓN

Como tercera parte se realizará una validación de cada esquema lógico local mediante la normalización, de la cual abarcaremos las tres primeras formas normales, ya que es suficiente para la solución de nuestro sistema y sus entidades.

3.3.2.1 PRIMERA FORMA NORMAL 1FN

Analizaremos que ninguna de las entidades contenga atributos Atómicos, es decir que deben ser simples e indivisibles.

ENTIDADES FUERTES

PACIENTE

| id_pac | **nom_pac** | sex_pac | edad | tel_pac | cor_pac |

| nom_pac | se descompuso de la siguiente manera:

| nom_pac | ape_pac |

HISTORIAL

| id_his | est_pac | pes_pac | mc_pac | comentario | fecha_consulta | nom_doc |

FUENTES

| id_fuente | fuente |

FUNCIONES NUTRIENTES

| id_fun | funcion |

SÍNTOMA

| id_sin | sintom |

ENFERMEDAD

| id_enf | nom_enf |

NUTRIENTE

| id_nut | nom_nut |

ENTIDADES DÉBILES

TIENE

| id_tiene | id_sintoma | id_enfermedad |

ESTAN_EN

| id_estan_en | id_nut | id_fuente |

SIRVEN_PARA

| id_sirven_para | id_nut | id_fun |

ALIVIAN

| id_alivian | id_nut | id_enf |

3.3.2.2 SEGUNDA FORMA NORMAL 2FN DEPENDENCIA PLENA

La mayoría de las entidades se manejaron de tal manera que no muestran dependencia errónea entre los atributos y la llave primaria.

Con excepción de la entidad enfermedad quien muestra problemas con los síntomas y los nutrientes puesto que no tienen dependencia total con el identificador de la enfermedad.

Por lo mismo es a quien debemos aplicarle la segunda forma normal quedándonos como se muestra a continuación.

ENTIDADES FUERTES

PACIENTE

| id_pac | nom_pac | ape_pac | | sex_pac | edad | tel_pac | cor_pac |

HISTORIAL

| id_his | est_pac | pes_pac | mc_pac | comentario | fecha_consulta | nom_doc |

FUENTES

| id_fuente | fuente |

FUNCIONES NUTRIENTES

| id_fun | funcion |

SÍNTOMA

| id_sin | sintom |

ENFERMEDAD

| id_enf | nom_enf |

NUTRIENTE

| id_nut | nom_nut|

ENTIDADES DÉBILES

TIENE

| id_tiene | id_sintoma | id_enfermedad |

ESTAN_EN

| id_estan_en | id_nut | id_fuente |

SIRVEN_PARA

| id_sirven_para | id_nut | id_fun |

ALIVIAN

| id_alivian | id_nut | id_enf |

3.3.2.3 TERCERA FORMA NORMAL 3FN DEPENDENCIAS TRANSITORIAS

No se encuentran con dependencias transitorias en las entidades que manejamos por tal motivo quedan de la misma manera que en la 2FN:

ENTIDADES FUERTES

PACIENTE

| id_pac | nom_pac | ape_pac | | sex_pac | edad | tel_pac | cor_pac |

HISTORIAL

| id_his | est_pac | pes_pac | mc_pac | comentario | fecha_consulta | nom_doc |

FUENTES

| id_fuente | fuente |

FUNCIONES NUTRIENTES

| id_fun | funcion |

SÍNTOMA

| id_sin | sintom |

ENFERMEDAD

| id_enf | nom_enf |

NUTRIENTE

| id_nut | nom_nut|

ENTIDADES DÉBILES

TIENE

| id_tiene | id_sintoma | id_enfermedad |

ESTAN_EN

| id_estan_en | id_nut | id_fuente |

SIRVEN_PARA

| id_sirven_para | id_nut | id_fun |

ALIVIAN

| id_alivian | id_nut | id_enf |

CAPITULO CUATRO

IMPLEMENTACIÓN Y PRUEBAS DEL SAMED

Se realiza la impresión de pantallas del sistema en su forma terminal indicando que realiza en cada apartado de dicho sistema.

4.1 IMPLEMENTACION DE LA BASE DE DATOS EN MySQL

Tabla paciente

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción					
<input type="checkbox"/>	<u>id_pac</u>	int(3)			No		auto_increment						
<input type="checkbox"/>	<u>nom_pac</u>	varchar(30)	utf8_general_ci		No								
<input type="checkbox"/>	<u>ape_pac</u>	varchar(50)	utf8_general_ci		No								
<input type="checkbox"/>	<u>eda_pac</u>	date			No								
<input type="checkbox"/>	<u>sex_pac</u>	varchar(1)	utf8_general_ci		No								
<input type="checkbox"/>	<u>tel_pac</u>	tinytext	utf8_general_ci		No								
<input type="checkbox"/>	<u>cor_pac</u>	varchar(40)	utf8_general_ci		No								

Tabla historial clínico

Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción					
<u>id_his</u>	int(11)			No		auto_increment						
<u>est_pac</u>	float			No								
<u>pes_pac</u>	float			No								
<u>mc_pac</u>	float			No								
<u>comentario</u>	text	utf8_general_ci		No								
<u>fec_consulta</u>	date			No								
<u>nom_doc</u>	varchar(40)	utf8_general_ci		No								
<u>id_pac</u>	int(11)			No								

Tabla enfermedad

Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción					
<u>id_enf</u>	int(11)			No		auto_increment						
<u>nom_enf</u>	varchar(50)	utf8_general_ci		No								

Tabla fuentes

Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción						
<u>id_fuente</u>	int(11)			No		auto_increment							
fuente	varchar(50)	utf8_general_ci		No									

Tabla funciones nutrientes

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción						
<input type="checkbox"/>	<u>id_fun</u>	int(11)			No		auto_increment							
<input type="checkbox"/>	funcion	varchar(50)	utf8_general_ci		No									

Tabla nutrientes

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción						
<input type="checkbox"/>	<u>id_nut</u>	int(11)			No		auto_increment							
<input type="checkbox"/>	<u>nom_nut</u>	varchar(20)	utf8_general_ci		No									

Tabla síntomas

	Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción						
<input type="checkbox"/>	<u>id_sin</u>	int(11)			No		auto_increment							
<input type="checkbox"/>	<u>sinto</u>	varchar(50)	utf8_general_ci		No									

4.2 INTERFAZ DEL SISTEMA

El diseño del sistema es elaborado con HTML. Las principales funciones del sistema son las siguientes:

1. Dietas prediseñadas por el nutriólogo solo para imprimir.
2. Consulta del chequeo clínico.
3. Consulta de la masa corporal.
4. Consulta de los nutrientes.
5. Diagnóstico del paciente.
6. Alta, baja, consulta o actualización del paciente.
7. Redacción y búsqueda del historial clínico.



Figura 3.5 Página principal

Consulta Chequeo Clínico

Elige la edad para la búsqueda

Edad: 0 meses

eda_pac	63.5
pes_minm	17.4908
pes_maxm	20.919
sobrepesom	22.9475
des_levm	16.2917
est_minm	107.587
est_maxm	113.999
pes_minf	16.9691
pes_maxf	20.6081
sobrepesof	22.9266
des_levf	15.7654
est_minf	106.483
est_maxf	113.067

Figura 3.6 Chequeo clínico

Al ingresar a la opción de chequeo clínico se podrá seleccionar la edad en meses del paciente, donde nos mostrara la tabla respectiva a dicha edad con su peso mínimo y máximo, estatura mínima y máxima, además de una desnutrición o sobrepeso tanto masculino como femenino (Figura 3.6).



Consulta Masa Corporal

Elige la edad para la búsqueda

Edad: 24.5 meses

eda_pac	63.5
mc_minm	14.6609
mc_maxm	16.2895
mc_minf	14.3646
mc_maxf	16.1644

Figura 3.7 Consulta de masa corporal

De igual manera que la consulta clínica se podrá seleccionar los meses y mostrara la tabla de masa corporal mínima y máxima correspondiente a cada sexo (Figura 3.7).

Consulta los Nutrientes

Escribe un nutriente (vitamina)

Nutriente:

Nutrientes	Fuentes	Funciones del Nutriente
zink	germen de trigo	biosíntesis de proteínas y ácidos nucleicos ;
zink	germen de trigo	respuesta inmune;
zink	germen de trigo	componente de diversas metaloenzimas;
zink	germen de trigo	antioxidante;
zink	levadura	biosíntesis de proteínas y ácidos nucleicos ;
zink	levadura	respuesta inmune;

Figura 3.8 Consulta de nutrientes

Se ingresa el nombre del nutriente y a continuación se muestra su proveniencia y las funciones para las que sirve dicho nutriente (Figura 3.8).

Consulta Enfermedad

Escribe un sintoma

Sintomas

Enfermedad	Sintomas	Nutrientes
piel	manchas de Bitot	ácidos grasos esenci;
piel	manchas de Bitot	biotina;
piel	manchas de Bitot	zink;
ojos	manchas de Bitot	vitamina A;

Figura 3.9 Búsqueda de enfermedades


Al ingresar a esta opción se tendrá que escribir los síntomas presentados por el paciente, mostrando así la posible enfermedad y los nutrientes que sirven tanto para su cura como para la prevención (Figura 3.9).

Introduce los datos del Paciente

Nombre(s):

Apellidos:

Sexo: Masculino
 Femenino

Fecha de nacimiento: 

Numero Telefonico:


E-mail:

id_pac	Nombre	Apellido	Fecha de Nacimiento	Sexo	Telefono	e-mail
75	luis	perez	1994-03-11		2545487	asasdw@asdaed
69	rosa	saldafia	0000-00-00	F	3652562	
68			2009-01-05	M	212121215	
74	luisito	garci	2002-07-03		202020220	asdsdf@asfdds
76	Juan	Rodriguez Carvente	2009-01-01	M	2816064	qase@asde.com

Nombre(s):

Apellidos:

Sexo: Masculino
 Femenino

Fecha de nacimiento: 

Numero Telefonico:

E-mail:

id_pac	Nombre	Apellido	Fecha de Nacimiento	Sexo	Telefono	e-mail
75	luis	perez	1994-03-11		2545487	asasdw@asdaed
69	rosa	saldafia	0000-00-00	F	3652562	
68			2009-01-05	M	212121215	
74	luisito	garci	2002-07-03		202020220	asdsdf@asfdds
76	Juan	Rodriguez Carvente	2009-01-01	M	2816064	qase@asde.com
77	Ashlei	Conde Garcia	2007-01-31	F		prue@hotmail.com

Figura 3.10 Alta del paciente

Al dar de alta a un paciente se llenaran todos sus datos, o pueden llenarse algunos y posteriormente realizar los cambios necesarios si así se desea, una vez ingresados se pulsa el botón DAR ALTA y se mostrara la base de datos actualizada donde se puede verificar que efectivamente se a ingresado a un nuevo paciente (Figura 3.10).

Introduce las palabras para la búsqueda

Buscar en: Palabra(s):

id_pac : 77
nom_pac : Ashlei
ape_pac : Conde Garcia
eda_pac : 2007-01-31
sex_pac : F
tel_pac :
cor_pac : prue@hotmail.com



[Home](#)

Figura 3.11 Búsqueda del paciente

El paciente se podrá buscar por su id, nombre, apellido, etc. Dependiendo del dato que se conozca de el. Una vez ingresado y buscado se podrá eliminar o modificar a dicho paciente (Figura 3.11).

Administrador

Opciones Administrador	
Chequeo Clínico	
Masa Corporal	
Fuentes	
Funciones	
Síntomas	
Enfermedad	
Nutrientes	

Relaciones	
Enfermedad - Nutriente	
Fuente - Nutriente	
Funcion - Nutriente	
Síntoma - Enfermedad	

Figura 3.12 Página principal del administrador del sistema SAMED

El administrador del sistema Contiene una serie de funciones primordiales para el enriquecimiento del sistema (Figura 3.12), como son:

1. El chequeo Clínico. Para indicar Altas o bajas de nuevos datos de chequeo clínico
2. Masa corporal. Del mismo modo que el chequeo clínico, puede agregarse para más edades si así se desea.
3. Fuentes. Altas o bajas de fuentes alimentarias
4. Funciones de los nutrientes.
5. Síntomas.
6. Enfermedades
7. Nutrientes.

Y por otra parte deberá realizar las relaciones entre las distintas tablas:

1. Enfermedad – Nutriente.
2. Fuente – Nutriente.
3. Función – Nutriente.
4. Síntoma – Enfermedad.

Inicio	Altas	Mostrar Lista y Dar de Baja
------------------------	-----------------------	---

Administrador Chequeo Clínico

Figura 3.13 Chequeo Clínico

Ventana donde se puede seleccionar altas, bajas y regresar al inicio (Figura 3.13).

Inicio	Altas	Mostrar Lista y Dar de Baja
------------------------	-----------------------	---

Alta de chequeo clínico

Edad paciente:	<input type="text"/>
Peso mínimo (masculino):	<input type="text"/>
Peso máximo (masculino):	<input type="text"/>
Sobrepeso (masculino):	<input type="text"/>
Desnutrición leve (masculino):	<input type="text"/>
Estatura mínima (masculino):	<input type="text"/>
Estatura máxima (masculino):	<input type="text"/>
Peso mínimo (femenino):	<input type="text"/>
Peso máximo (femenino):	<input type="text"/>
Sobrepeso (femenino):	<input type="text"/>
Desnutrición leve (femenino):	<input type="text"/>
Estatura mínima (femenino):	<input type="text"/>
Estatura máxima (femenino):	<input type="text"/>

Figura 3.14 Alta de chequeo clínico

Se deben llenar todos los campos y presionar el botón de alta (Figura 3.14). Y aunque no se muestra una imagen de impresión de pantalla del alta de masa corporal, su función es exactamente la misma.

Inicio	Altas	Mostrar Lista y Dar de Baja
------------------------	-----------------------	---

Lista Chequeo Clínico y Dar de Baja:

Edad	Masculino						Femenino						Opciones
	Peso Min	Peso Max	Sobrepeso	Desnutrición Leve	Estatura Min	Estatura Max	Peso Min	Peso Max	Sobrepeso	Desnutrición Leve	Estatura Min	Estatura Max	
0	3.15061	3.87907	4.17249	2.7738	43.2181	51.7427	3.06486	3.71751	3.99257	2.74722	47.7094	50.9905	Eliminar
0.5	3.59739	4.38742	4.71816	3.20951	51.0052	54.414	3.43762	4.14559	4.45012	3.10176	50.1214	53.3376	Eliminar
1.5	4.42387	5.32732	5.72815	4.02056	55.0024	58.3226	4.13899	4.94676	5.30563	3.77015	53.7138	56.9075	Eliminar
2.5	5.18337	6.17559	6.63857	4.75447	57.9963	61.3146	4.78482	5.68008	6.08764	4.38704	56.4536	59.7177	Eliminar
3.5	5.8658	6.94221	7.4607	5.4168	60.4555	63.8028	5.37914	6.35151	6.80277	4.95592	58.8253	62.1012	Eliminar
4.5	6.48496	7.63532	8.20219	6.01371	62.5748	65.9692	5.92588	6.96652	7.45711	5.48029	60.8655	64.2034	Eliminar
5.5	7.04362	8.26203	8.87138	6.55137	64.4559	67.9075	6.42882	7.53018	8.05633	5.96351	62.6573	66.1028	Eliminar
6.5	7.54334	8.82878	9.47546	7.03565	65.1592	69.6741	6.89153	8.04717	8.60563	6.40877	64.3714	67.8473	Eliminar
7.5	8.00439	9.34149	10.021	7.47202	67.7239	71.3058	7.31737	8.52187	9.10987	6.31912	65.9208	69.4686	Eliminar
8.5	8.41571	9.80559	10.5141	7.86553	69.177	72.828	7.70951	8.95832	9.57354	7.19741	67.3687	70.9891	Eliminar

Figura 3.15 Baja del chequeo clínico

Se muestra la tabla con todos los valores para que el administrador verifique si efectivamente son los datos que desea eliminar, y de ser así simplemente pulsara la opción eliminar que se ubica frente a cada una de las filas Figura (3.15).



Administrador Fuentes

Figura 3.16 Fuentes

Ventana para selección de altas o bajas de las fuentes (Figura 3.16).



Fuente a dar de Alta:

fFuente de nutrientes:

id_fuentes	fuentes
1	tortilla de nixtamal
2	leche
3	acociles
4	charales
5	mariscos
6	visceras
7	oleaginosas
8	leguminosas
9	cereales integrales
10	frutas secas

Figura 3.17 Alta de las fuentes

Solo se debe escribir el nombre de la fuente y pulsar DAR ALTA (Figura 3.17). Del mismo modo funciona para los nutrientes, síntomas, enfermedades y funciones del nutriente.

Inicio	Altas	Bajas
------------------------	-----------------------	-----------------------

Baja de Fuentes:

id_fuente	fuentes	Baja
1	tortilla de nixtamal	Baja
2	leche	Baja
3	acociles	Baja
4	charales	Baja
5	mariscos	Baja
6	vísceras	Baja
7	oleaginosas	Baja
8	leguminosas	Baja
9	cereales integrales	Baja
10	frutas secas	Baja
11	té	Baja
12	productos del mar	Baja
13	agua	Baja

Figura 3.18 Baja de las fuentes.

Se muestra la lista de las fuentes y en seguida la opción de baja para cada una de ellas (Figura 3.18).

RELACION FUENTES NUTRIENTES :

[Inicio](#)

Enfermedad Nutriente

Ingresar id de la Enfermedad a relacionar: Ingresar el id del nutriente a relacionar:

id_enf	id_nut
2	5
2	6
2	7
2	1
2	3
2	8
2	9

Figura 3.19 Relacione fuentes nutrientes

En este recuadro se mostraran las relaciones existentes, aun cuando no se comprendan, y en seguida se podrá realizar una búsqueda del id de la enfermedad y el id del nutriente para relacionarlos (Figura 3.19).

RELACION FUENTES NUTRIENTES :

[Inicio](#)

Enfermedad Nutriente

7	piel 2
1	vitamina A
2	vitamina C
3	vitamina D
4	vitamina K
15	vitamina B12
30	vitamina E

Ingresar id de la Enfermedad a relacionar: Ingresar el id del nutriente a relacionar:

Figura 3.20 Relación fuentes – Nutrientes

Al obtener el id de la enfermedad y el id del nutriente, se ingresarán respectivamente y se podrá pulsar el botón DAR ALTA (3.20).

CONCLUSIONES

El análisis, diseño e implementación de SAMED cumplió con los objetivos generales, que son los de dar el apoyo necesario en cuanto al ágil diagnóstico del paciente. La presentación de los nutrientes y la localización de los mismos.

Una de las limitaciones que podemos ver en el sistema es que en caso de encontrar nuevas enfermedades más complejas y con más síntomas que los previstos, se deberá mejorar la programación para filtrar los síntomas y encontrar las enfermedades.

La tecnología de la computación ha entrado hacia todas las áreas y el Internet ha logrado un crecimiento sorprendente para un rápido manejo de información desde cualquier parte del mundo. Por lo cual de alguna manera seria conveniente que el sistema creciera para todas las áreas de medicina, mejorando obviamente el sistema de filtrado de enfermedades.

Además como en la actualidad la tecnología lleva al uso de internet mediante dispositivos móviles, tales como el celular. Seria interesante realizar un sitio compatible con los dispositivos móviles para q se puedan realizar constancias desde cualquier sitio con acceso a internet.

El sistema se ira adaptando de manera cautelosa y periódica en nuevas especialidades, por tal motivo solo entonces se crearan diferentes sesiones con sus respectivas restricciones.

Se ligara a una base de datos con historial e información de los pacientes, para ir cargando de manera automática el doctor que atiende, la fecha los síntomas y su diagnóstico creando así una receta.

BIBLIOGRAFÍA

[4] Abraham Silberschatz, Henry F. Korth, S. Sudarshan., *Fundamentos de Bases de Datos*. Mc Graw Hill.

Buyens, J. *Aprenda. Desarrollo de Bases de Datos WEB Ya*. Ed. McGraw-Hill.

Casanueva Esther, Kaufer Martha, Arroyo Pedro, Berta Pérez Ana, *Nutriología Médica, Editorial Panamericana*.

Cervera P., Clapes J., Rigolfas R., *3ª edición, Alimentación y Dietoterapia, Editorial Mc Graw-Hill Interamericana*.

De Miguel, A y Plattini, M. (1997) *Diseño de Bases de Datos Relacionales*, Ed. Rama, Madrid

DuBois Paul, *MySQL*, Ed. Riders

Gardarin, G. (1987) *Bases de Datos, Madrid*: Ed. Paraninfo

Gil Rubio F., Tejedor Cerbel J.A. y otros. *Creación de sitios Web con PHP4*. Ed. Osborne McGraw-Hill.

Gutiérrez Abraham, Bravo Ginés. *Php 4 a través de ejemplos*, Ed. Alfaomega, Rama.

Krause Mendelson Marie, *9ª Edición, Nutrición y Dietoterapia de Krause, Editorial Mc Graw-Hill Interamericana*.

Lopez M. Josefina, *Nutrición y Salud para todos*, Ed. Trillas.

Luque Ruiz, Irene. *Desarrollo de Bases de Datos*. Ed. Alfaomega.

Martínez Martínez Ezequiel, *Nutrición y Salud*, Ed. Manuel Moderno.

Pérez Ana, *Dietas Normales y Terapéuticas, Editorial Científica la Prensa Médica Mexicana*.

Pérez César. *MySQL para Windows y Linux*, Ed. Alfaomega

[2] Piattini, M., *Fundamentos y modelos de BASES de DATOS*. Ed. Alfaomega.

[3] Rivero, E. (1988) *Bases de Datos Relacionales*, Ed. Paraninfo.

[1] Sommerville Ian, *Madrid (2005), 7ª Edición, Ingeniería de Software, Editorial Pearson Educación S.A.*

Ullman Larry, *MySQL*, Ed. Prentice Hall.

Valade, J., *PHP 5 para dummies*, St Editorial.