CAPITULO I

INTRODUCCIÓN

1.1 ÁMBITO

Laboratorios Clínico “G&Z” pertenece a la unidad educativa “Juan Pablo Segundo”; se encuentra ubicada en Chillogallo sector sur; la cual se dedica por más de 12 años a realizar exámenes clínicos de laboratorio periódicamente a los estudiantes de dicha institución y también a personas particulares; Además entrega estadísticas de los exámenes realizados.

El proceso actual para realizar un examen y registrar información es manual; la enfermera toma los datos personales del paciente; las muestras dependiendo del tipo de examen son registrados en un cuaderno, y se asigna un número dependiendo del tipo al que corresponde, estos pedidos son entregados al médico a quien corresponda realizar determinado examen.

Los resultados son registrados en un cuaderno para ser digitados a un documento y entregarlos a cada paciente.

La forma de pago se lo puede hacer por abonos el mismo que debe ser cancelado hasta la entrega del resultado.
Se realiza descuentos a los estudiantes de la institución y a personas de bajos recursos.

1.1.1 PLANTEAMIENTO DEL PROBLEMA

En la actualidad Laboratorios Clínico “G&Z” maneja manualmente la administración y control de exámenes médicos practicados sobre sus pacientes, lo cual genera algunos problemas como:

- La información se registra manualmente en cuadernos generando una gran cantidad de información redundante; con la consiguiente pérdida de fiabilidad.
- Para generar estadísticas y reportes mensuales: por edad, género, examen entre otras se tiene una gran confusión de datos siendo los resultados incompletos y poco confiables.
- No se lleva un control de ingresos.
- No se controla los abonos realizados a los pedidos.
- No poseen un formato estándar para la entrega de resultados.

1.1.2 JUSTIFICACIÓN

Con el desarrollo de un sistema informático se conseguirá apoyar el manejo de los procesos y se brindará mejor información a pacientes, administradores y personal médico.

1.2 OBJETIVOS

1.2.1 OBJETIVO GENERAL

Apoyar la administración del Laboratorios Clínico “ G&Z” , mediante el desarrollo e implementación de un sistema informático.

1.2.2 OBJETIVOS ESPECÍFICOS

- Administrar el registro de pacientes.

- Registrar y controlar el servicio prestado a los pacientes.
- Administrar el registro del tipo de exámenes que se realizan en Laboratorios Clínico “ G&Z”
- Administrar el registro de médicos
- Registrar los resultados obtenidos de los exámenes realizados a los pacientes.
- Generar las estadísticas y reportes mensuales.
- Controlar pagos
- Realizar facturación.
- Controlar permisos de acceso a los usuarios.

1.3 ALCANCE Y LIMITACIONES

1.3.1 ALCANCE

El Sistema de Administración de Laboratorios Clínicos (SIALAB.C), es un software de alto rendimiento, creado para el apoyo operacional y el control de Laboratorios Clínico “G&Z”; es capaz de procesar información del laboratorio de manera confiable.

SIALAB.C, realizará una administración de: registro de pacientes, registro de médicos, registro de catálogo de exámenes, recepción de pedidos, control de pedidos pendientes, descuentos, ingreso y emisión de resultados, registro de cobros, facturación, generación de reportes y estadísticas, control de usuarios y permisos de acceso.

SIALAB.C, generará un formulario para la entrega de resultados correspondiente a cada examen.

1.3.2 LIMITACIONES

- SIALAB.C, no realiza el control de reactivos que son utilizados en cada examen que se realiza en el laboratorio clínico.

- No incluye la contabilidad del laboratorio.

1.4 PRESUPUESTO

CUADRO 1.1

COSTO APROXIMADO PARA EL DESARROLLO DEL SISTEMA

DESCRIPCIÓN			**VALOR**	
		Número Horas	**Valor Horas**	**Costo Total**
Recurso humano :	Analista Desarrolladores	500 350	$ 4.00 $ 5.00	$ 2000.00 $ 1750.00
Software:	1 Lic. Rational Rose 2000 1 Lic. SQL Server 2000 1 Lic. Visual Basic 6.0			$ 180.00 $ 200.00 $ 180.00
Hardware:	Computadora Intel PIV 3.0			$ 750.00
Gastos Varios:	Hojas de papel Bonn Flash Memory Cartuchos de Impresora			$ 200.00
Costo aproximado del proyecto				**$ 5260.00**

CAPITULO II

MARCO TEÓRICO

2.1 INGENIERÍA DEL SOFTWARE

La Ingeniería de Software es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software, se considera que es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costos-efectivos (eficaces en costo o económicas) a los problemas de desarrollo de software, es decir, permite elaborar consistentemente productos correctos, utilizables y costos-efectivos. (Pressman,18)

El proceso de ingeniería de software se define como "un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad". El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo. (Pressman,18)

El proceso de desarrollo de software requiere por un lado un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama el ciclo de vida del software que comprende cuatro grandes fases: concepción, elaboración, construcción y transición. La concepción define el alcance del proyecto y desarrolla un caso de negocio. La elaboración define un

plan del proyecto, especifica las características y fundamenta la arquitectura. La construcción crea el producto y la transición transfiere el producto a los usuarios. . (Pressman,25)

2.1.1 Diseño Conceptual

El diseño conceptual se considera como un análisis de actividades y consiste en la solución de negocios para el usuario y se expresa con los casos de uso. El diseño lógico es la solución del equipo de proyecto del negocio y consiste de las siguientes tareas:

- Identificar los usuarios y sus roles
- Obtener datos de los usuarios
- Evaluar la información
- Documentar los escenarios de uso
- Validar con los usuarios
- Validar contra la arquitectura de la empresa.

Una forma de obtener estos requerimientos es construir una matriz usuarios-actividades de negocios, realizar entrevistas, encuestas y/o visitas a los usuarios, de tal manera que se obtenga quién, qué, cuándo, dónde y por qué de la solución.

2.1.2 Diseño Lógico

El diseño lógico traduce los escenarios de uso creados en el diseño conceptual en un conjunto de objetos de negocio y sus servicios. El diseño lógico se convierte en parte en la especificación funcional que se usa en el diseño físico. El diseño lógico es independiente de la tecnología. El diseño lógico refina, organiza y detalla la solución de negocios y define formalmente las reglas y políticas específicas de negocios. (Pressman,38)

Un objeto de negocios es la encapsulación de un servicio que abstrae las cualidades esenciales de algo de interés.

Un servicio es una unidad con capacidad de cómputo. Un servicio debe satisfacer lo siguiente:

- Ser seguro, lo que equivale a un uso correcto y con autorización
- Ser válido, qué tareas o reglas se pueden aplicar
- Manejar excepciones, informando al cliente
- Contar con un catálogo de servicios que constituye un repositorio de servicios.

Los objetos de negocio deben verificarse y probarse de tal manera que asegure que los módulos operen como unidades completas de trabajo. Las tareas de verificación incluyen:

- Una verificación independiente:
- Pre y post condiciones
- Lógica y funcionalidad individual
- Una verificación dependiente:
- Verificación de dependencias
- Que operan como una unidad específica de trabajo

El diseño lógico comprende las siguientes tareas:

- Identificar y definir los objetos de negocio y sus servicios
- Definir las interfaces
- Identificar las dependencias entre objetos
- Validar contra los escenarios de uso
- Comparar con la arquitectura de la empresa
- Revisar y refinar tanto como sea necesario

Para definir los objetos de negocios y sus servicios se puede usar la técnica de análisis nombre-verbo de los escenarios de uso. También se puede emplear la técnica sujeto-verbo-objeto directo. En estas técnicas los sujetos y el objeto

directo son los candidatos a objetos de negocio y los verbos activos son los candidatos a servicios.

Una interfaz tiene las siguientes partes:

- Nombre
- Precondiciones, lo que debe estar presente antes de ejecutarse
- Postcondiciones, estado final
- Capacidad o funcionalidad (SQL, pseudocódigofunción matemática)
- Dependencias

La tarea de identificar las dependencias entre objetos permite identificar eventos, sucesos o condiciones que permitan la realización de tareas de negocios coordinadamente o transaccionalmente. Para ello se debe considerar lo siguiente:

- Identificar los eventos disparadores (triggers)
- Determinar cualquier dependencia (existencial o funcional)
- Determinar cualquier problema de consistencia o secuencia
- Identificar cualquier regulación de tiempo crítica
- Considerar algún problema organizacional (transacciones)
- Identificar y auditar los requerimientos de control
- Determinar lugares y dependencias a través de la ubicación
- Determinar cuando el servicio que controla la transacción es dependiente de los servicios contenidos en otros objetos de negocio.

La validación del modelo lógico debe ser tal que éste sea:

- Completo .-debe representar todos los escenarios de uso,
- Correcto.- el comportamiento lógico debe corresponder con el comportamiento conceptual.
- Los objetos de negocio y servicios no deben ser ambiguos
- En el diseño lógico conceptualmente se divide en tres niveles de servicios con el fin de que la aplicación resulte flexible ante los cambios de requerimientos y/o de tecnología cambiando únicamente la capa o capas necesarias. Los tres niveles son: servicios de usuario, servicios de negocio y servicios de datos.

***Los servicios de usuario*.**

Controlan la interacción. Un servicio de usuario son personas, aplicaciones, otros servicios o la combinación de éstos. Generalmente involucra una interfase gráfica de usuario (GUI) o pude ser no visual (mensajes o funciones), maneja todos los aspectos de la interacción con la aplicación. El objetivo central es minimizar el esfuerzo de conocimiento requerido para interpretar la información. Un servicio de usuario incluye un contenido (qué se necesita comunicar al usuario) y una forma (cómo se comunica el contenido) cuando es necesaria la comunicación.

Los servicios de negocio

Convierten datos recibidos de los servicios de datos y de usuario en información (datos + regla de negocio) y pueden usar otros servicios de negocio para completar su tarea.

Las tareas de los servicios de negocio son:

- Dar formato a los datos
- Obtener y mover datos desde y hasta los servicios de datos
- Transformar los datos en información
- Validar los datos inmediatamente en el contexto o en forma diferida una vez terminada la transacción.

Los servicios de datos.

Son los servicios de bajo nivel que apoyan los servicios de negocio y son de una amplia gama de categorías como las siguientes:

- Declaración del esquema y su evolución (estructuras de datos, tipos, acceso indexado, SQL, APIs)
- Respaldo y recuperación (recuperación de datos si un evento falla)
- Búsqueda y Lectura (búsquedas, compilación, optimización y ejecución de solicitudes, formación de un conjunto de resultados)
- Inserción, actualización y borrado (procesar modificaciones consistentemente transaccional). Una transacción es atómica (ocurre o no), consistente

(preserva integridad), aislada (otras transacciones ocurren antes o después) y durable (una vez completada, ésta sobrevive).

- Bloqueo (permite al acceso concurrente a los datos)
- Validación de datos (verifica la integridad del dominio, triggers y gateways para verificar el estado de los datos antes de aceptarlos, manejo de errores)
- Seguridad (acceso seguro a los objetos, operaciones, permisos a usuario y grupos y servicios)
- Administración de la conexión (mecanismos básicos para establecer una sesión de los servicios de datos). Establecer una conexión involucra: una identificación, la colocación y provisión de datos, tiempo de sesión, el tipo de interacción (conversacional, transaccional, multiusuario, monousuario).
- Distribución de datos (Distribuye información, a múltiples unidades de recuperación, bases de datos heterogéneas, según la topologías de la red).

2.1.3 Diseño físico

El diseño físico traduce el diseño lógico en una solución implementable y costo-efectiva o económica. (Pressman,48)

El componente es la unidad de construcción elemental del diseño físico. Las características de un componente son:

- Se define según cómo interactúa con otros.
- Encapsula sus funciones y sus datos.
- Es reusable a través de las aplicaciones.
- Puede verse como una caja negra.
- Puede contener otros componentes.

En el diseño físico se debe cuidar el nivel de granularidad (un componente puede ser tan grande o tan pequeño según su funcionalidad, es decir, del tamaño tal que pueda proveer de una funcionalidad compleja pero de control genérico) y la agregación y contención (un componente puede reusar utilizando técnicas de agregación y contención, sin duplicar código).

El diseño físico debe involucrar:

- El diseño para distribución – debe minimizarse la cantidad de datos que pasan como parámetros entre los componentes y éstos deben enviarse de manera segura por la red.
- El diseño para multitarea – debe diseñarse en términos de la administración concurrente de dos o más tareas distintas por una computadora y múltiples hilos de un mismo proceso)
- El diseño para uso concurrente – el desempeño de un componente remoto depende de si está corriendo mientras recibe una solicitud.
- El diseño con el manejo de errores y prueba de eventos:
- Validando los parámetros- a la entrada antes de continuar con cualquier proceso.
- Protegiendo recursos críticos –manejar excepciones para evitar la falla o terminación sin cerrar archivos, liberar objetos sincronizados o memoria.
- Protegiendo datos importantes – contar con una excepción a la mitad de la actuación en las bases de datos.
- Crear una versión para limpiar errores.

El diseño físico comprende las siguientes tareas:

- Definir los componentes
- Refinar el empaquetamiento y distribución de componentes
- Especificar las interfases de los componentes
- Distribuir los componentes en la red
- Distribuir los repositorios físicos de datos
- Examinar la tolerancia a fallas y la recuperación de errores
- Validar el diseño físico

De las tareas anteriores la más importante es la distribución de los datos que pueden ser centralizados, una partición, un extracto o una réplica.
Los datos centralizados equivalen a una base de datos maestra ubicada en un lugar central. No hay copias de los datos.

2.2 ARQUITECTURA CLIENTE /SERVIDOR

La arquitectura cliente / servidor es un modelo para el desarrollo de sistemas de información, en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el dialogo o solicita los recursos y servidor, al proceso que responde a las solicitudes. (http://dominiopublico.com/intranets/cliente_servidor.php)

En este modelo las aplicaciones se dividen de forma que el servidor contiene la parte que debe ser compartida por varios usuarios, y en el cliente permanece sólo lo particular de cada usuario.

Los clientes realizan generalmente funciones como:

- Manejo de la interfaz de usuario.
- Captura y validación de los datos de entrada.
- Generación de consultas e informes sobre las bases de datos.
- Por su parte los servidores realizan, entre otras, las siguientes funciones:
- Gestión de periféricos compartidos.
- Control de accesos concurrentes a bases de datos compartidas.
- Enlaces de comunicaciones con otras redes de área local o extensa.

Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y éste le responde proporcionándolo. Normalmente, pero no necesariamente, el cliente y el servidor están ubicados en distintos procesadores. Los clientes se suelen situar en ordenadores personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo.
(http://dominiopublico.com/intranets/cliente_servidor.php)

2.2.1 CARACTERÍSTICAS DE LA ARQUITECTURA CLIENTE / SERVIDOR

Entre las principales características de la arquitectura cliente/servidor se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

(http://dominiopublico.com/intranets/cliente_servidor.php)

2.2.2 ELEMENTOS DE LA ARQUITECTURA CLIENTE / SERVIDOR

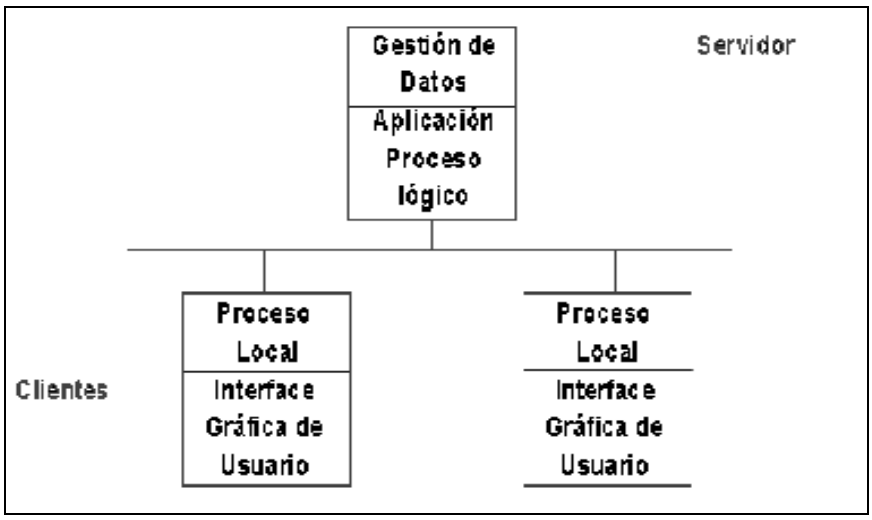
Entre los principales componentes que forman dicha arquitectura está caracterizada por tres componentes básicos:

(http://www.uacam.mx/dgsc.nsf/pages/cadmpower)

- Presentación/Captación de Información
- Procesos
- Almacenamiento de la Información

Los cuales se suelen distribuir tal como se presenta en la figura

GRÁFICO 2.1
ARQUITECTURA CLIENTE /SERVIDOR



Se integran en una arquitectura Cliente/Servidor en base a los elementos que caracterizan dicha arquitectura, es decir:

- Puestos de Trabajo
- Comunicaciones
- Servidores

De estos elementos debemos destacar:

2.2.2.1 El Puesto de Trabajo o Cliente.

Una Estación de trabajo o microcomputador (PC: Computador Personal) conectado a una red, que le permite acceder y gestionar una serie de recursos, el cual se perfila como un puesto de trabajo universal. Nos referimos a un microcomputador conectado al sistema de información y en el que se realiza una parte mayoritaria de los procesos. (http://www.uacam.mx/dgsc.nsf/pages/cadmpower)

2.2.2.2 *Las Comunicaciones*

En sus dos vertientes:

- **Infraestructura de redes.** Componentes Hardware y Software que garantizan la conexión física y la transferencia de datos entre los distintos equipos de la red.

- **Infraestructura de comunicaciones** Componentes Hardware y Software que permiten la comunicación y su gestión, entre los clientes y los servidores.

La arquitectura Cliente/Servidor es el resultado de la integración de dos culturas. Por un lado, la del Mainframe que aporta capacidad de almacenamiento, integridad y acceso a la información y, por el otro, la del computador que aporta facilidad de uso (cultura de PC), bajo costo, presentación atractiva (aspecto lúdico) y una amplia oferta en productos y aplicaciones.
(http://www.uacam.mx/dgsc.nsf/pages/cadmpower)

2.2.2.3 *Los Servidores o Back-end.*

Una máquina que suministra una serie de servicios como Bases de Datos, Archivos, Comunicaciones.

Los Servidores, según la especialización y los requerimientos de los servicios que debe suministrar pueden ser:

- Mainframes
- Mini ordenadores
- Especializados (Dispositivos de Red, Imagen, etc.)

Una característica a considerar es que los diferentes servicios, según el caso, pueden ser suministrados por un único Servidor o por varios Servidores especializados.

2.3 HERRAMIENTAS DE SOPORTE.

2.3.1 RATIONAL ROSE 2000

- Para el análisis y el diseño del Sistema se usará RATIONAL ROSE 2000, este modelo tiene clases, casos de uso, paquetes lógicos, operaciones, componentes empaquetados, procesos, proyectos y la relación entre ellos, los mismos que nos servirán para diseñar los diagramas. (Ayudas de Rational Rose).

2.3.2 SQL SERVER 2000

La base de datos se soportará en SQL SERVER 2000, por ser una plataforma actualizada y difundida para el manejo de datos; es un lenguaje de sintaxis simple y muy potente. Mediante él se puede recorrer, modificar o borrar registros de las tablas de datos; además cuenta con las seguridades adecuadas para el manejo de información. Mediante la adopción del estándar ANSI SQL-92 y ODBC, se ha logrado hacer los datos accesibles a cualquier aplicación cliente.
(http://www.microsoft.com/latam/sql/evaluation/features/default.asp)

SQL SERVER 2000, es una base de datos de relacional cliente- servidor , admite operaciones que abarcan un amplio conjunto de plataformas: desde sistemas personales, como equipos de escritorio y portátiles, hasta grandes servidores multiproceso simétrico con 8 a 16 procesadores, vario gigabytes de memoria y un terabyte o para más espacio de almacenamiento en el disco.

El lenguaje de consulta estructurado (SQL) es un lenguaje de base de datos normalizado, utilizado por el motor de base de datos de Microsoft Jet. SQL se utiliza para crear objetos QueryDef, como el argumento de origen del método OpenRecordSet y como la propiedad RecordSource del control de datos. (http://www.microsoft.com/latam/sql/evaluation/features/default.asp)

2.3.2.1 Componentes de SQL

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

2.3.3 VISUAL BASIC V 6.0

Lo que se refiere a la codificación del cliente se utilizará como lenguaje de programación, Visual Basic Versión 6.0, por tener una interfase amigable tanto para el programador como para el usuario.

Visual Basic 6.0 es uno de los lenguajes de programación que más entusiasmo despiertan entre los programadores de PCs, tanto expertos como novatos. En el caso de los programadores expertos por la facilidad con la que desarrollan aplicaciones complejas en poquísimo tiempo (comparado con lo que cuesta programar en Visual C++, por ejemplo). En el caso de los programadores novatos por el hecho de ver de lo que son capaces a los pocos minutos de empezar su aprendizaje.
(http://www.microsoft.com/catalog/display.asp?subid=45&site=730)

Visual Basic 6.0 es un lenguaje de programación visual, también llamado lenguaje de 4ª generación. Esto quiere decir que un gran número de tareas se realizan sin escribir código, simplemente con operaciones gráficas realizadas con el ratón sobre la pantalla.

Visual Basic 6.0 es también un programa basado en objetos, aunque no orientado a objetos como C++ o Java. La diferencia está en que Visual Basic 6.0 utiliza objetos con propiedades y métodos, pero carece de los mecanismos de herencia y polimorfismo propios de los verdaderos lenguajes orientados a objetos como Java y C++.
(http://www.microsoft.com/catalog/display.asp?subid=45&site=730)

Visual Basic, es un potente entorno de desarrollo de aplicaciones de alta productividad, rapidez, eficiencia y proporciona herramientas con amplias funcionalidades, basadas en estándares para controlar u entregar datos de negocio, además , tiene un conjunto completo de componentes para GUI que llevan rápidamente las aplicaciones de datos desde el prototipo a la producción, también, integra fácilmente la información corporativa dentro de sus aplicaciones con los drivers de bases de datos de alta velocidad para SQL SERVER que será el servidor de base de datos que usaremos para este sistema.

CAPITULO III

MARCO CONCEPTUAL

3.1 INTRODUCCIÓN A LABORATORIO CLÍNICO

El Laboratorio Clínico es una herramienta primordial para el área médica, ya que por medio de este se diagnostican diferentes patologías y además se realizan estudios para establecer el tipo de tratamiento que se debe administrar al paciente, al igual que el seguimiento del mismo.

El paciente o usuario llega al Laboratorio para realizarse sus exámenes clínicos, del laboratorista y del Auxiliar depende que este usuario reciba el servicio adecuado en todo sentido, ya sea científico o humano, el profesional de la salud debe estar en condiciones de proporcionar una ayuda integral.

3.2 SERVICIOS DEL LABORATORIO CLINICO

Cada examen de laboratorio clínico debe ser realizado a los pacientes de forma individual, guiándose siempre por los parámetros profesionales y éticos. Básicamente, el trabajo en el laboratorio clínico se clasifica en tres grandes grupos temáticos:

- Toma de muestras.
- Análisis de las muestras.
- Entrega de resultados.

En cada uno de estos temas, se requiere de numerosas medidas de atención y cuidado, con el fin de minimizar al máximo los errores factibles de ser cometidos en la práctica diaria. Se debe enfatizar que el trabajo en el laboratorio clínico, como cualquier tipo de trabajo, es realizado por seres humanos y no se esta exento de cometer equivocaciones. Pero estas equivocaciones pueden ser erradicadas de los laboratorios clínicos, si se mantienen eficientes actitudes éticas, profesionales y de procedimiento.

3.3 RAZONES PARA UTILIZAR LOS SERVICIOS DEL LABORATORIO CLÍNICO

- Descubrir enfermedades en etapas subclínicas
- Ratificar un diagnostico sospechado clínicamente.
- Obtener información sobre el pronóstico de una enfermedad.
- Establecer un diagnóstico basado en una sospecha bien definida.
- Vigilar un tratamiento o conocer una determinada respuesta terapéutica.
- Precisar factores de riesgo.

3.4 TIPOS DE EXÁMENES

- **EXÁMEN FÍSICO:**

  Depende de dos factores: el tiempo de exposición y la temperatura. Todos los microorganismos son susceptibles, en distinto grado, a la acción del calor. El calor provoca desnaturalización de proteínas, fusión y desorganización de las membranas y/o procesos oxidantes irreversibles en los microorganismos.

- **EXAMEN QUÍMICO:**

  Los resultados dependen de una combinación de químicos o reactivos los mismos que presentan diferentes reacciones y cada uno de ellos determinan el resultado.

- **EXAMEN MICROSCÓPICO:**

  El examen microscópico sólo se puede observar utilizando microscopios ópticos o electrónicos.

3.5 TERMINOLOGÍA UTILIZADO EN LABORATORIO CLINICO.

ESTERILIZACIÓN: Proceso mediante el cual se elimina todas las formas de vida de los microorganismos de un objeto o de una sustancia para evitar su reproducción.

ASEPSIA: Libre de microorganismos.

MICROORGANISMOS: Ser vivo que sólo se puede observar utilizando microscopios ópticos o electrónicos.

BACTERIAS: Bacteria (del griego, bakteria, ‘bastón’), nombre que reciben los organismos unicelulares y microscópicos, que carecen de núcleo diferenciado y se reproducen por división celular sencilla. Las bacterias son muy pequeñas, entre 1 y 10 micrómetros (µm) de longitud, y muy variables en cuanto al modo de obtener la energía y el alimento.

VIRUS: Virus (en latín, ‘veneno’), entidades orgánicas compuestas tan sólo de material genético, rodeado por una envoltura protectora Los cientos de virus conocidos son causa de muchas enfermedades distintas en los seres humanos, animales, bacterias y plantas. Los virus son parásitos intracelulares submicroscópicos, compuestos por ARN o por ácido desoxirribonucleico (ADN) nunca ambos.

PARÁSITOS. Parásito, cualquier organismo que vive sobre o dentro de otro organismo vivo, del que obtiene parte o todos sus nutrientes, sin dar ninguna compensación a cambio al hospedador.

ENDOPARÁSITOS: Parásitos que están dentro del huésped.

NEMÁTODOS: Gusano redondo. Se aloja en el intestino y a veces se abre camino hasta otras partes del cuerpo, son de color blanquecino, rosa y cafe. Sus huevos se desarrollan en el agua o en tierra húmeda, y es probable que sus embriones entren en el cuerpo por ingestión directa. Los gusanos pueden expulsarse por medio de purgantes.

FILARIAS: del gusano causante de la loaiasis Loa loa, el del gusano de Guinea Dracunculus medinensis y el de la filaria que causa la oncocercosis, Onchocerca volvulus.

CÉSTODOS: Gusano plano platelminto, nombre común de un grupo de animales de cuerpo blando. Son los animales más sencillos entre los que poseen cabeza. Presentan simetría bilateral y son un tanto aplanados dorsoventral mente. La mayoría son alargados. El filo al que pertenecen los gusanos planos o platelmintos comprende: las tenias, que en su fase adulta son parásitos del tracto digestivo de los animales; Existen dos especies que pueden dar lugar a la enfermedad, la tenia porcina o Taenia solium y la tenia del ganado vacuno o Taenia saginata. las duelas, que parasitan diversos órganos de distintos animales; y los gusanos planos de vida libre.

TREMÁTODOS: Gusanos planos. El nombre científico de la duela del hígado del cordero es Fasciola hepatica. Las duelas conocidas como duelas de la sangre pertenecen al género Schistosoma. Tanto la duela del hígado del cordero como las duelas de la sangre pertenecen a la clase Trematodos.

PROTOZOARIOS – PROTISTA: Ameba o Amiba, organismo unicelular perteneciente al filo Sarcodinos (Sarcodina) y al reino Protistas.

ECTOPARASITOS: Parásitos que están encima del huésped. Como las pulgas, piojos y garrapatas.

SANGRE: sustancia líquida que circula por las arterias y las venas del organismo. La sangre es roja brillante o escarlata cuando ha sido oxigenada en los pulmones y pasa a las arterias; adquiere una tonalidad más azulada cuando ha cedido su oxígeno para nutrir los tejidos del organismo y regresa a los pulmones a través de las venas y de los pequeños vasos denominados capilares. El cuerpo humano posee cinco litros de sangre en su totalidad.

COMPOSICIÓN DE LA SANGRE: En una persona normal sana, el 45% del volumen de su sangre son células, glóbulos rojos (la mayoría), glóbulos blancos y plaquetas. Un fluido claro y amarillento, llamado plasma, constituye el resto de la sangre. El plasma, del cual el 95% es agua, contiene también nutrientes como

glucosa, grasas, proteínas, vitaminas, minerales y los aminoácidos necesarios para la síntesis de proteínas.

GLÓBULOS ROJOS, ERITROCITOS O HEMATÍES: Son células de forma discoidea y bicóncava con un diámetro promedio de 7,5 µm y un espesor que llega a 2 µm en sus bordes y que no alcanza 1 µm en el centro y constituyen el 99% del total de células en la sangre.

GLÓBULOS BLANCOS O LEUCOCITOS: Los glóbulos blancos son una vital fuerza de defensa contra organismos extraños. También funcionan como nuestro "aseo urbano" ya que limpian y eliminan células muertas y desechos tisulares que de otra manera se acumularían. Los leucocitos son células de forma redondeada mientras circulan en la sangre y adoptan formas muy variadas cuando salen de los vasos sanguíneos y su diámetro oscila entre 6 y 18 µm.

NEUTROFILO: La principal función de los neutrófilos es la de detener o retardar la acción de agentes infecciosos o materiales extraños. Su propiedad más importante es la fagocitosis y son capaces de ingerir bacterias y pequeñas partículas. Los neutrófilos, además de defender el organismo contra las infecciones, pueden ser dañinos también, al liberar los componentes de sus gránulos tóxicos en diversos tejidos.

EOSINÓFILOS: Tienen una igual actividad motriz que los neutrófilos y aunque poseen propiedades fagocíticas, participan menos en la ingestión y muerte de las bacterias. Un aumento en su número frecuentemente acompaña a reacciones alérgicas o procesos inmunológicos, al igual que presencia de parásitos.

BASÓFILOS: Son los que tienen menos movilidad y menor capacidad fagocítica. Participan en reacciones de hipersensibilidad (picaduras).

LINFOCITOS: El linfocito es una de las células más intrigantes de la sangre humana y bajo ese nombre se engloban varios tipos diferentes de células

linfoides, que encierran diferencias estructurales y funcionales aún no bien esclarecidas.

MONOCITOS: Son los grandes fagocitos mononucleares de la sangre periférica. Son un sistema de células fagocíticas producidas en la médula ósea, que viajan como tales por la sangre, para luego emigrar a diferentes tejidos como hígado, bazo, pulmones, ganglios linfáticos, hueso, cavidades serosas, etc., para convertirse en esos tejidos en macrófagos libres o fijos, cuyas funciones se corresponden con lo que se conoce como sistema mononuclear-fagocitario.

LAS PLAQUETAS O TROMBOCITOS: Las plaquetas son framentos de citoplasma de megacariocitos, que circulan como pequeños discos en la sangre periférica. En promedio, tienen un diámetro entre 1 a 4 µm, su citoplasma se tiñe azul claro a púrpura y es muy granular. No tienen núcleo y su concentración normal en sangre periférica es entre 150.000 y 450.000/µl. Su duración en circulación es de 8 a 11 días, interviene en el proceso de coagulación de la sangre.

ANTICOAGULANTES: Los anticoagulantes son sustancias que previenen la formación de coágulos. Existen diferentes tipos de ellos en polvo o líquidos. Deben seleccionarse siempre el anticoagulante apropiado según el estudio que se quiera realizar.

CUADRO HEMÁTICO: Es uno de los exámenes de laboratorio que más se solicitan, comprende numerosas pruebas o parámetros, los cuales proporcionan individualmente o en conjunto un resultado de enorme valor para numerosas entidades clínicas.

HEMATOCRITO: Este mide el tanto por ciento del volumen total de una muestra de sangre venosa ocupado por los hematíes o expresado de otra manera es la relación entre el volumen de eritrocitos y el de la sangre total. Se expresa como porcentaje (%).

HEMOGLOBINA: Es el componente principal de los glóbulos rojos, es una proteína conjugada que sirve de vehículo para el transporte de O2 y CO2. Se aumenta en hemoconcentración, en estados de shock, quemaduras, por diarrea, vomito y poliglobulina primaria.
Se disminuye en casos de anemia.

RECUENTO DE PLAQUETAS: Este resultado es importante ya que desempeñan un papel vital en la hemostasis.
El método utilizado es un método directo en el que se utiliza Oxalato de amonio al 1% 1.98 ml y 0.02ml (20 landas) de sangre anticoagulada con EDTA, Se mezcla bien y se deja en reposo aproximadamente durante 10 minutos para permitir la lisis total del resto de las células.

RECUENTO DE RETICULOCITOS: Son eritrocitos no nucleados inmaduros, que contienen RNA y que continúan sintetizando hemoglobina después de la pérdida del núcleo.
Se mezcla en un tubo tres gotas de azul de cresil brillante y tres gotas de sangre, se incuban durante 15 minutos a 37 grados, se hacen dos extendidos en lámina y se miran con objetivo de 100 X. Se cuentan aproximadamente 1.000 hematies y se saca el promedio de reticulocitos.

PH: Es el reflejo de la capacidad del riñón para mantener la concentración normal de hidrogeniones. El pH normal va de 5.5 - 6.5. Influyendo el régimen dietético el cada paciente.

DENSIDAD: Esta varia en razón directa a la cantidad de sólidos, principalmente cloruros, urea, sulfatos, la densidad normal va de 1.015 - 1.025.

CAPITULO IV

ASPECTOS METODOLÓGICOS

4.1 RESUMEN TABLA ASPECTOS METODOLÓGICOS

<table>
<tr><td>PARADIGMA
ESPIRAL
INCREMENTAL</td><td>METODOLOGIA OMT</td><td colspan="2">LENGUAJE UNIFICADO DE MODELADO</td></tr>
<tr><td rowspan="5">Análisis</td><td rowspan="2">Modelo Estático</td><td colspan="2">Identificación de Actores</td></tr>
<tr><td colspan="2">Diagrama de Casos de Uso</td></tr>
<tr><td rowspan="3">Modelo Dinámico</td><td colspan="2">Diagrama de Clases</td></tr>
<tr><td colspan="2">Diccionario de Clases</td></tr>
<tr><td colspan="2">Diagrama de Objetos</td></tr>
<tr><td rowspan="4">Diseño</td><td rowspan="2">Modelo Funcional</td><td rowspan="2">Diagrama de Interacción:</td><td>Secuencia</td></tr>
<tr><td>Colaboración</td></tr>
<tr><td rowspan="2"></td><td colspan="2">Diagrama de Actividades</td></tr>
<tr><td colspan="2">Diagrama de Estados</td></tr>
<tr><td>Construcción</td><td></td><td colspan="2">Racional Rose 2000
SQL Server 2000
Visual Basic 6.0
Cliente/Servidor</td></tr>
<tr><td>Pruebas</td><td>Pruebas funcionales</td><td colspan="2">Descripción de la prueba
Procedimiento de la prueba</td></tr>
<tr><td>Mantenimiento</td><td></td><td colspan="2"></td></tr>
</table>

4.2 PARADIGMA INCREMENTAL ESPIRAL

El paradigma del modelo incremental, se centra en la entrega de un producto operacional con cada incremental. Los incrementos son versiones no depuradas del producto final pero proporciona la capacidad de otorgar al usuario en la forma de evolución. (Presman Roger, 28)

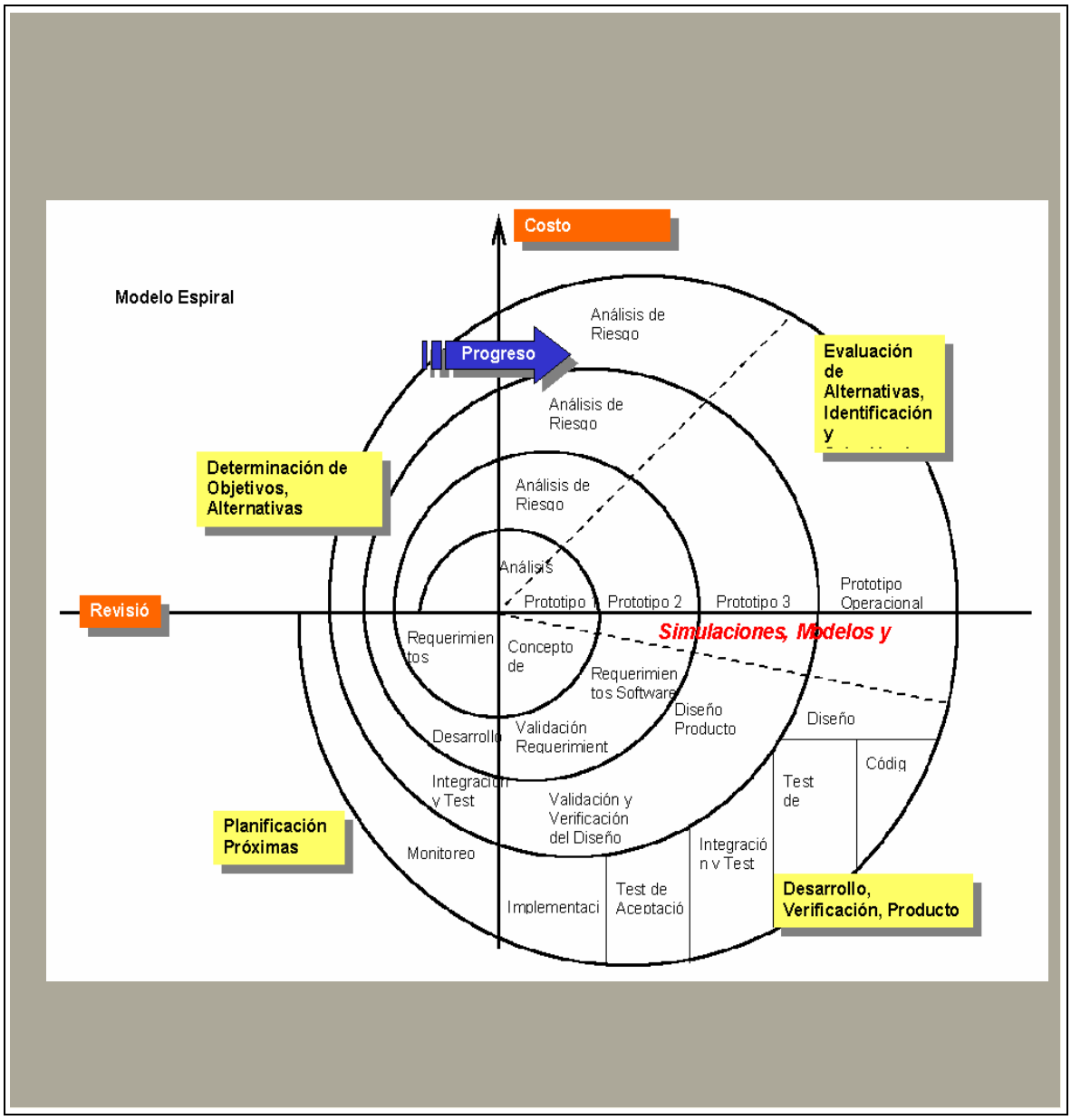
Se evitan proyecto largos y se entrega “algo de valor” a los usuarios con cierta frecuencia, el usuario se involucra más. [http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/Tema3_1x1.pdf]

El desarrollo incremental promueve gestionar riesgos técnicos. En el modelo incremental se ha combinado el desarrollo rápido de aplicaciones (RAD), con el uso de herramientas de generación de código.

Un aspecto intrigante del modelo en espiral se hace evidente cuando consideramos la dimensión radial representada en la Figura 3.1 Con cada iteración alrededor de la espiral (comenzando en el centro y siguiendo hacia el exterior), se construyen sucesivas versiones del software, cada vez más completas. Durante la primera vuelta alrededor de la espiral se definen los objetivos, las alternativas y las restricciones, y se analizan e identifican los riesgos. Si el análisis de riesgo indica que hay una incertidumbre en los requisitos, se puede usar la creación de prototipos en el cuadrante de ingeniería para dar asistencia tanto al encargado del desarrollo como al cliente. Se pueden usar simulaciones y otros modelos para definir más el problema y refinar los requisitos.

GRÁFICO 4.2

MODELO ESPIRAL INCREMENTAL



El cliente evalúa el trabajo de ingeniería (cuadrante de evaluación del cliente) y sugiere modificaciones. En base a los comentarios del cliente se produce la siguiente fase de planificación y de análisis de riesgo. En cada bucle alrededor de la espiral, la culminación del análisis de riesgo resulta en una decisión de “seguir

o no seguir". Si los riesgos son demasiado grandes, se puede dar por terminado el proyecto.
Sin embargo, en la mayoría de los casos, se sigue avanzando alrededor del camino de la espiral, y ese camino lleva a los desarrolladores hacia fuera, hacia un modelo más completo del sistema, y al final, al propio sistema operacional. Cada vuelta alrededor de la espiral requiere ingeniería (cuadrante inferior derecho), que se puede llevar a cabo mediante el enfoque del ciclo de vida clásico o de la creación de prototipos. Debe tenerse en cuenta que el número de actividades de desarrollo que ocurren en el cuadrante inferior derecho aumenta al alejarse del centro de la espiral. Lewis G(87)

El paradigma del modelo en espiral para la ingeniería del software es actualmente el enfoque más realista para el desarrollo de software y de sistemas a gran escala. Utiliza un enfoque "evolutivo" para la ingeniería del software, permitiendo al desarrollador y al cliente entender y reaccionar a los riesgos en reducción del riesgo, pero, lo que es más importante, permite a quien lo desarrolla aplicar el enfoque de creación de prototipos en cualquier etapa de la evolución del producto. Mantiene el enfoque sistemático correspondiente a los pasos sugeridos por el ciclo de vida clásico, pero incorporándola dentro de un marco de trabajo interactivo que refleja de forma más realista el mundo real. El modelo en espiral demanda una consideración directa de riesgos técnicos en todas las etapas del proyecto y, si se aplica adecuadamente, debe reducir técnicos en todas las etapas del proyecto y, si se aplica adecuadamente, debe reducir los riesgos antes de que se conviertan en problemáticos. Lewis G(117)

4.3 METODOLOGÍA

La metodología hace referencia a un conjunto de reglas y procedimientos que permiten ejecutar en forma lógica una serie de pasos para obtener un resultado determinado, indica qué tareas, en qué orden, cuándo ejecutarlas, cómo ejecutarlas, como controlar el avance de esas tareas y cuáles son las herramientas que deben utilizar en esa tarea.

Una metodología es una especificación de los pasos a seguir durante el ciclo de vida del desarrollo de sistemas que incluye:

- Tareas paso a paso por cada fase.
- Funciones individuales y en grupo desempeñadas en cada tarea.
- Productos resultantes.

Selección de la Metodología

De las diferentes metodologías hemos podido observar y determinar que OMT reúne las principales características de las metodologías estudiadas y se adapta a nuestras necesidades en el aspecto de poder realizar el análisis y diseño del sistema lo más conciso, así como plantear los problemas utilizando la técnica del Lenguaje Unificado de Modelado (UML) en los diferentes diagramas que abarca esta técnica y solucionarlos hasta terminar con el producto. (http://www.rational.com/uml)

4.3.1 METODOLOGÍA OMT

OMT fue creada por James Rumbaugh y Michael Blaha en 1991. Es una metodología de análisis y diseño orientado a objetos más popular y eficiente utilizado en mundo profesional.

OMT pone importancia en el uso de modelos para lograr una abstracción en el cual el análisis esta enfocado en el mundo real para un nivel de diseño.
La característica fundamental de OMT es tener en cuenta todos los principios del paradigma orientado a objetos y refuerza particularmente la necesidad de modelos consistentes de análisis y diseño del negocio como paso previo y fundamental al desarrollo de la programación y la definición de la base de datos.
(Roumbought James, 29)

Esta técnica de modelado de objetos se extiende desde el análisis en donde se abstrae los aspectos esenciales del dominio de la aplicación, es decir lo que debe hacer el sistema y no en la forma que lo hará.

OMT plantea una forma práctica y productiva de desarrollar software para diferentes aplicaciones independientemente del lenguaje de implementación acoplándose a todas las necesidades actuales y futuras de la ingeniera de software. (Roumbought James, 35)

4.3.1.1 *Etapas de la metodología OMT*

- **Análisis.**

Documento de análisis incluye:
Durante el análisis se construye un modelo de dominio de la aplicación sin tener en cuenta la implementación. Deberá incluir aquella información que sea significativa desde el punto de vista del mundo real, presentando el aspecto externo del sistema**.**

- Se establece la definición del problema.
- Se construye un modelo de objetos.
- Se desarrolla un modelo dinámico.
- Se construye un modelo funcional.
- Se verifican, integran y refinan los tres modelos.

*El modelo de objetos***:**

Es el modelo más importante, describe la estructura de los objetos de un sistema-identidad, relacionados con otros objetos, atributos y operaciones, el modelo de objetos proporciona el entorno esencial en el cual se pueden situar el modelo dinámico y el modelo funcional. (Roumbought James, 34)

- Identificar los objetos y las clases.
- Preparar un diccionario de datos.
- Identificar las asociaciones entre objetos.
- Identificar atributos de objetos y enlaces.
- Organizar y simplificar las clases de objetos empleando la herencia.

- Verificación de las vías de acceso necesarias para llevar a cabo las probables consultas.
- Realizar las interacciones necesarias para el refinamiento del modelo.
- Agrupar las clases en módulos.

El modelo dinámico:

Representa los aspectos temporales de comportamiento “de control” del sistema, mediante la secuencia de operaciones en el tiempo.

- Se preparan escenarios de secuencias típicas de interacción.
- Se identifican sucesos que actúen entre objetos.
- Se prepara un seguimiento de sucesos para cada escenario.
- Se construyen diagramas de estado
- Se comparan los sucesos intercambiados entre objetos para verificar la congruencia. (Roumbought James, 34)

El modelo funcional:

Representa los aspectos transformacionales “de función” del sistema, mediante la transformación de valores de los datos. Se representa mediante un diagrama de flujo.

- Identificación de los valores de entrada y de salida.
- Construcción de diagramas de flujo de datos que muestren las dependencias funcionales.
- Descripción de las funciones.
- Identificación de restricciones.
- Especificación de los criterios de optimización. (http://www.rational.com/uml)

- **Diseño del sistema**

Es la estrategia de alto nivel para resolver el problema y construir una solución, incluye decisiones acerca de la organización del sistema en

subsistemas, la asignación de subsistemas a componentes de hardware y software y decisiones fundamentales conceptuales y de política que constituyen el marco de trabajo para el diseño detallado.

- **Diseño de objetos**

Documento de diseño incluye versiones detalladas de los modelos de objetos dinámico y funcional.

En el diseño de objetos se toman las decisiones necesarias para construir un sistema sin descender a los detalles particulares de un lenguaje o sistema de base de datos.

- Obtención de las operaciones para el modelo de objetos a partir de los demás modelos.
- Diseño de algoritmos para la implementación de las operaciones.
- Optimización de las vías de acceso de datos.
- Implementar el control del software completando la aproximación seleccionada durante el diseño del sistema.
- Ajuste de la estructura de clases para incrementar la herencia.
- Diseño de la implementación de las asociaciones.
- Se determina la representación exacta de los atributos que son objetos.
- Empaquetamiento de las clases y asociaciones en módulos. (http://www.rational.com/uml)

- **Implementación**

- Diseño de base de datos
- Código

Durante la implementación se codifican, tanto las estructuras en el dominio de la solución. La base que la sustenta es el *diseño de objetos.*

El código puede ser una simple transición de las decisiones de diseño a las características propias de un lenguaje.

- **Pruebas**

Es una actividad para determinar si el sistema está siendo construido correctamente. Tanto la implementación como las pruebas son dos etapas que están involucradas durante el análisis y diseño.

4.3.2 Lenguaje Unificado de Modelado UML

El Lenguaje Unificado de Modelado (Unified Modeling Languaje, UML) es el lenguaje estándar para realizar el modelado de los sistemas de software y es independiente del lenguaje de programación utilizado. Jacobson (55)

En todo proceso de software donde se utilice una metodología orientada a objetos y la notación UML no pueden faltar los diagramas, para representar las diferentes vistas del producto final. (http://www.rational.com/uml)

Los diagramas de UML se pueden dividir en estáticos (aportan una visión estática del sistema) , dinámicos (aportan una visión dinámica del sistema) y funcional.

4.3.2.1 Diagramas estáticos:

Diagramas de casos de uso

Los Casos de Uso es una técnica para capturar información de cómo un sistema o negocio trabaja, o de cómo se desea que trabaje. No pertenece estrictamente al enfoque orientado a objetos, es una técnica para capturar requisitos. Jacobson (89)

- Los Casos de Uso describen bajo la forma de acciones y reacciones el comportamiento de un sistema

- Permiten definir los límites del sistema y las relaciones entre el sistema y el entorno.
- Los Casos de Uso son descripciones de la funcionalidad del sistema independientes de la implementación.
- Los Casos de Uso cubren la carencia existente en métodos previos (OMT, Booch) en cuanto a la determinación de requisitos.
- Los Casos de Uso particionan el conjunto de necesidades atendiendo a la categoría de usuarios que participan en el mismo.

- Están basados en el lenguaje natural, es decir, es accesible por los usuarios. (http://www.rational.com/uml)

Diagramas de Clases

El Diagrama de Clases es el diagrama principal para el análisis y diseño. Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. La definición de clase incluye definiciones para atributos y operaciones. El modelo de casos de uso aporta información para establecer las clases, objetos, atributos y operaciones.
El mundo real puede ser visto desde abstracciones diferentes (subjetividad).

(http://www.rational.com/uml)

4.3.2.2 **Diagramas dinámicos**:

Diagrama de secuencia

Los diagramas UML de secuencia se utilizan para modelar los aspectos dinámicos de un sistema.
Un diagrama de secuencia consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos. Los diagramas de secuencia destacan el orden temporal de los mensajes. El diagrama de secuencia muestra la interacción de un conjunto de objetos importante porque puede dar detalle a los casos de uso, aclarándolos al nivel de mensajes de los objetos existentes, como también muestra el uso de los mensajes de las clases diseñadas

en el contexto de una operación en una aplicación a través del tiempo. (http://www.rational.com/uml)

Es útil para observar la vida de los objetos en un sistema, identificar llamadas a realizar o posibles errores del modelado estático, que imposibiliten el flujo de información o de llamadas entre los componentes del sistema.

En el diagrama de secuencia se muestra el orden de las llamadas en el sistema. Es imposible representar en un solo diagrama de secuencia todas las secuencias posibles del sistema, por ello se escoge un punto de partida. (Addison,Rumbaugh,Jacabson;25)

Diagrama de colaboración

El Diagrama de Colaboración ofrece una mejor visión espacial mostrando los enlaces de comunicación entre objetos, muestra las relaciones entre objetos y son mejores para comprender todos los efectos que tiene un objeto y para el diseño de procedimientos. El diagrama de Colaboración puede obtenerse automáticamente a partir del correspondiente diagrama de Secuencia (o viceversa).

- Son útiles en la fase exploratoria para identificar objetos.
- La distribución de los objetos en el diagrama permite observar adecuadamente la interacción de un objeto con respecto de los demás.

La estructura estática viene dada por los enlaces; la dinámica por el envío de mensajes por los enlaces (Addison,Rumbaugh,Jacabson;509)

4.3.2.3 Diagramas funcionales:

- **Diagrama de Actividades**

  El Diagrama de Actividad es una especialización del Diagrama de Estado, organizado respecto de las acciones y usado para especificar:

- Un método

- Un caso de uso
- Un proceso de negocio (Workflow)

Un estado de actividad representa una actividad: un paso en el flujo de trabajo o la ejecución de una operación. Un grafo de actividades describe grupos secuenciales y concurrentes de actividades. Los grafos de actividades se muestran en diagramas de actividades.

Las actividades se enlazan por transiciones automáticas. Cuando una actividad termina se desencadena el paso a la siguiente actividad.

Un diagrama de actividades es provechoso para entender el comportamiento de alto nivel de la ejecución de un sistema, sin profundizar en los detalles internos de los mensajes. Los parámetros de entrada y salida de una acción se pueden mostrar usando las relaciones de flujo que conectan la acción y un estado de flujo de objeto. (http://www.rational.com/uml)

- **Diagrama de Estados**.

El diagrama de estados es un gráfico compuesto de los estados del sistema y sus transiciones.

Si se asocia a una clase describirá como una instancia de esta clase reacciona ante los eventos.

Si se asocia a un caso de uso describirá como funciona ese caso de uso con el sistema funcionando.

- Estos diagramas son muy útiles para mostrar el ciclo de vida de las clases con complejidad media o alta, pero no tiene mucho sentido para clases de una complejidad simple ni tampoco para clases con una complejidad bastante elevada. (http://www.rational.com/uml)

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- La aplicación de la metodología de OMT, nos permitió hacer uso de las herramientas adecuadas para el desarrollo del presente sistema.
- El conocimiento previo de las herramientas que se utilizan en la elaboración del sistema acorta el tiempo de desarrollo.
- Para el desarrollo del sistema se debe contar con un usuario que domine el tema y que participe activamente durante el proyecto.
- El paradigma espiral incremental permitió que el usuario evalúe el desarrollo del sistema en todas sus etapas.

5.2 RECOMENDACIONES

- Es necesario que el encargado de administrar la base de datos reciba capacitación para desempeñar dicha actividad.
- Para dar un mejor apoyo a la administración del laboratorio sería recomendable diseñar el soporte de contabilidad con interfaz de SIALAB.C.

5.3BIBLIOGRAFIA

- Avila C.; *Modelado con UML. Principios y Aplicaciones*; Perú, Primera Edición 2001.
- Cota A. 1994 "Ingeniería de Software". Soluciones Avanzadas. Julio de 1994.
- Ivar Jacobson, Grady Booch y James Rumbaugh ; *El Lenguaje Unificado de Modelado*; Editorial Pearson Educación; 2000 ISBN .
- Lewis G. 1994. "What is Software Engineering?" DataPro (4015). Feb 1994.
- Pressman Roger; *Ingeniería de Software*; Mc Graw-Hill/España, S.A , Cuarta Edición.

SITIOS WEB DE REFERENCIA.

- http://www.rational.com/uml
- http:www.medicentro.com.co"; Manual de Normas y Técnicas
- http://www .Monografias_com.htm ; " Manual Básico de Laboratorio Clínico
- http://dominiopublico.com/intranets/cliente_servidor.php; " Cliente / Servidor"
- http://www.microsoft.com/latam/sql/evaluation/features/default.asp; "Características de SQL Server 2000 "
- http://www.microsoft.com/catalog/display.asp?subid=45&site=730; " Microsoft Visual Basic 6.0"
- Ayudas de Rational Rose
- http://www.lawebdelprogramador/; "Códigos fuentes"