

**INSTITUTO TECNOLÓGICO DE COSTA RICA**

**ESCUELA DE BIOLOGÍA**



**INFORME DE TRABAJO FINAL DE GRADUACIÓN**

**Diseño, implementación y evaluación de un software para la captura,  
administración y análisis de información en laboratorios clínicos**

**Francisco G. Marín Hernández**

**CARTAGO, 2007**

**INSTITUTO TECNOLÓGICO DE COSTA RICA**

**ESCUELA DE BIOLOGÍA**



**Asesores en Tecnologías Médicas**

**DISEÑO, IMPLEMENTACIÓN Y EVALUACIÓN DE UN SOFTWARE PARA LA  
CAPTURA, ADMINISTRACIÓN Y ANÁLISIS DE INFORMACIÓN EN  
LABORATORIOS CLÍNICOS**

**Informe de Proyecto de Graduación para optar por el grado de  
Bachiller en Ingeniería en Biotecnología**

**Francisco G. Marín Hernández**

**CARTAGO, 2007**

# DISEÑO, IMPLEMENTACIÓN Y EVALUACIÓN DE UN SOFTWARE PARA LA CAPTURA, ADMINISTRACIÓN Y ANÁLISIS DE INFORMACIÓN EN LABORATORIOS CLÍNICOS

Francisco G. Marín Hernández\*

## RESUMEN

*La gran cantidad de información generada en laboratorios implica la necesidad de sistemas de software que faciliten la recolección, administración y análisis de los mismos.*

*El desarrollo de una nueva versión del sistema Labsys mediante la asistencia del profesional en biotecnología como intermediario entre personal de tecnologías de información y de laboratorios es una manera de brindar una solución.*

*Se desarrolló un sistema orientado a laboratorios pequeños basado en un modelo de dos capas sobre un motor de base de datos de SQL Server 2005 y tecnología de Microsoft .NET en el cliente. El software fue diseñado como un primer paso en el planeamiento de un sistema SOA.*

*Los resultados son satisfactorios y los principales puntos a considerar son la abstracción de los datos, capacitación de usuarios y la estandarización de nombres de pruebas y nomenclatura de resultados.*

**Palabras clave:** Labsys, biotecnología, laboratorios clínicos, sistemas información

\*INFORME DE TRABAJO FINAL DE GRADUACIÓN, Ingeniería en Biotecnología, Instituto Tecnológico de Costa Rica, Cartago, Costa Rica. 2007.

**DISEÑO, IMPLEMENTACIÓN Y EVALUACIÓN DE UN SOFTWARE PARA LA  
CAPTURA, ADMINISTRACIÓN Y ANÁLISIS DE INFORMACIÓN EN  
LABORATORIOS CLÍNICOS**

**Informe presentado a la Escuela de Biología del  
Instituto Tecnológico de Costa Rica como requisito parcial  
para optar al título de Bachiller en Ingeniería en Biotecnología.**

**Miembros del Tribunal**

---

**MSc. Jhonny Peraza,  
Profesor Asesor-ITCR**

---

**Ing. Rodolfo Araya,  
Asesor- Empresa**

---

**MSc. Maritza Guerrero,  
Lector**

## INDICE GENERAL

1.INTRODUCCION .....	11
2.REVISION DE LITERATURA.....	14
3.OBJETIVOS .....	22
3.1 Objetivo General.....	22
3.2 Objetivos específicos.....	22
4.MÉTODOS .....	23
4.1 Etapa I: Análisis de sistemas anteriores .....	24
4.2 Etapa II: Diseño de la arquitectura de la aplicación .....	25
4.3 Etapa III: Diseño de la bases de datos .....	26
4.4 Etapa IV: Elaboración de Procedimientos Almacenados.....	27
4.5 Etapa V: Creación de interfaz y ensamblado de prototipo.....	28
4.6 Etapa VI: Desarrollo de módulos de mantenimientos .....	28
4.7 Etapa VII: Desarrollo de módulos de procesos.....	29
4.8 Etapa VIII: Ensamblaje de la aplicación.....	29
5.RESULTADOS.....	30
5.1 Análisis de sistemas Labsys 3.0 y Labsys 4.0 .....	30
5.2 Arquitectura .....	35
5.3 Diseño de base de datos .....	36
5.4 Procedimientos almacenados.....	43
5.5 Interfaz con Usuario.....	45
5.6 Mantenimientos y procesos .....	46
5.7 Ensamblado de la aplicación .....	55

6.DISCUSION .....	62
6.1 Diseño del sistema .....	62
6.2 Diseño de base de datos relacional.....	65
6.3 Procedimientos almacenados.....	68
6.4 Funcionamiento del software .....	69
6.5 Aprovechamiento de la Información .....	71
7.1 Conclusiones.....	74
7.2 Recomendaciones .....	75

## INDICE DE CUADROS

<b>Núm.</b>	<b>Título</b>	<b>Pág.</b>
	Cuadro 1: Soluciones a problemas de diseño en el sistema Labsys .....	31
	Cuadro 2: Datos principales en el funcionamiento del sistema Labsys.....	36

## INDICE DE FIGURAS

<b>Núm.</b>	<b>Título</b>	<b>Pág.</b>
	Figura 1: Arquitectura del sistema Labsys 5.0 .....	35
	Figura 2: Entidades que componen el núcleo de la base datos del sistema .....	41
	Figura 3: Diagrama de relación de entidades en manejo de resultados .....	42
	Figura 5: Puntos comunes de la interfaz con usuario de Labsys 5.0. ....	45
	Figura 6: Mantenimiento de Catálogo Técnico en Labsys 5.0. ....	46
	Figura 8: Flujo de trabajo para la creación de solicitudes .....	49
	Figura 9: Análisis de información en el Laboratorio .....	50
	Figura 10: Evaluación de expedientes con muestras .....	51
	Figura 11: Evaluación de Resultados.....	53
	Figura 12: Pantalla de Manejo de Expedientes Individuales .....	55
	Figura 13: Pantalla de Manejo de Expedientes con Muestras .....	56
	Figura 14: Pantalla de Solicitud Individual .....	57
	Figura 15: Pantalla de solicitudes por Lotes .....	58
	Figura 16: Pantalla de solicitudes con muestras .....	59
	Figura 17: Pantalla para manejo de resultados de solicitudes individuales .....	60
	Figura 18: Pantalla para manejo de solicitudes con muestras .....	61

## INDICE DE LISTADOS

<b>Núm.</b>	<b>Título</b>	<b>Pág.</b>
	Listado 1: Ejemplo de un procedimiento almacenado.....	43
	Listado 2: Ejemplo de código C# para el manejo de datos.....	47

## Índice de Anexos

<b>Núm.</b>	<b>Título</b>	<b>Pág.</b>
	Anexo I: Vista general del diseño de base de datos aplicado en caracterización de orígenes de muestra y solicitudes de pruebas del software Labsys .....	76
	Anexo II: Diseño de base de datos en el registro de resultados de espectros de masa .....	77
	Anexo III: Diseño de base de datos en el sistema PLAN .....	78

## **1.INTRODUCCION**

---

Labsys se ha desarrollado como una herramienta de vital importancia para el funcionamiento de laboratorios clínicos privados y más recientemente ha sido incorporado al laboratorio del Hospital México.

El Labsys es un registro que considera la estructura general de datos de los laboratorios clínicos, permite la creación de un sistema de exámenes y pruebas definido por el usuario. Administra el flujo de trabajo dentro del laboratorio, revisa los resultados promedio para emitir alertas sobre anomalías en exámenes, se enlaza a equipo de diagnóstico electrónico y permite realizar leves análisis para determinar cuando los equipos pierden sensibilidad ante la prueba.

El paquete Labsys consiste en una aplicación que funciona en el sistema operativo Windows, creado sobre el lenguaje de programación Visual Basic 6.0 (VB 6.0) y conectado una base de datos creada sobre el motor SQL Server 2000, ambos productos de Microsoft. Dot Net (.NET) es una plataforma estándar de desarrollo a nivel de Windows que impulsa Microsoft desde hace varios años, consiste en un ambiente controlado que facilita el acceso a todo tipo de servicios (Microsoft Corporation 2002).

El sistema Labsys permite manipular la información de las pruebas realizadas en un laboratorio y el reporte de las mismas en pocos pasos (cuatro pantallas para el trabajo diario). Por lo que, para responder al competitivo mercado de tecnologías clínicas es necesario seguir la tendencia de transformar Labsys en una plataforma que permita la expansión del producto hasta ser una poderosa herramienta capaz de aprovechar los avances en la Tecnologías de Información (TI) en función de aumentar la productividad, disminuir costos y estimular la inversión en el desarrollo de productos, servicios e investigación (Kline *et al* 2004).

Para la creación de una versión del Labsys en la plataforma .NET es necesario realizar un análisis de código de VB 6.0 y SQL Server 2000, de tal manera que sea posible realizar la transferencia del código a una versión C# 2.0. Con el fin de

completar este proceso de cambio de lenguaje y ambiente de programación es necesario ejecutar un análisis detenido del flujo de trabajo dentro del laboratorio en busca de comprender la información que debe almacenarse y la forma en la cual varía. Así, se realizaron visitas a laboratorios clínicos para analizar problemas en el manejo y procesamiento de información, formas de automatización y consulta de datos. Una vez que se cuente con la información del sistema anterior y la obtenida de la investigación en laboratorio, se procede con el diseño arquitectónico de la aplicación, o sea, diagramar y planificar los componentes, servicios e interacciones del sistema. De esta manera, se puede no solo cambiar el lenguaje de la aplicación, sino que es posible optimizar el funcionamiento de la misma y expandir sus herramientas.

Así, el sistema fue reestructurado en la plataforma .NET de Microsoft, generando un centro de almacenamiento de datos. Además, Labsys 5.0 se convirtió en la base de un software previsto para la integración de bases de datos de distintos laboratorios a nivel privado o estatal en Costa Rica u otros países. La estructura del Labsys 5.0 permitiría establecer redes de extracción de datos en los lugares donde se implemente, con lo cual podrían realizar grandes proyectos de análisis de datos en busca de patrones o predicción de tendencias. Estos datos podrían ser punto de partida en estudios epidemiológicos a partir de los cuales se pueden desarrollar estrategias para combatir de manera localizada ciertas afecciones de la población.

El empleo de la plataforma .NET de Microsoft permite generar una serie de componentes comunes en los procesos a nivel de laboratorios clínicos, los cuales funcionarían como pilar en el desarrollo de un marco de trabajo (*Framework*) para la sistematización de la producción de soluciones especializadas (administración, caracterización de poblaciones, entre otras) a bajo costo. El *framework* desarrollado será a su vez el origen de otro especializado en crear aplicaciones a nivel de laboratorios orientados a otros sectores pero con una estructura muy similar en el proceso de captura e interpretación de resultados. De esta manera, como arranque

de un plan de expansión, la reconstrucción del sistema Labsys es el paso fundamental en la construcción de aplicaciones más complejas.

La importancia del proyecto reside en brindar una plataforma funcional que aumente la productividad, disminuya costos y sirva como base en el desarrollo de herramientas de investigación a nivel de tecnologías de información en el área de laboratorios clínicos. El análisis de la antigua plataforma es necesario para establecer una base a partir de la cual realizar mejoras y agregar nuevas funcionalidades.

En este proceso de recreación del software en particular, es necesaria la presencia de una persona con experiencia en el flujo de trabajo del laboratorio. Los conocimientos en química, microbiología, bioquímica y biología molecular son necesarios para una correcta interpretación de los procedimientos y resultados obtenidos en un laboratorio. Los profesionales que comprendan el principio de operación de los dispositivos empleados en el laboratorio son fundamentales en el desarrollo de herramientas implicadas en registrar la incertidumbre y la sensibilidad de equipos.

Finalmente, el enlace entre el trabajo en laboratorio y la labor de creación de software alcanza su máxima importancia cuando se planea crear componentes que optimicen tareas o protocolos mediante la asistencia automatizada. Por ello, la presencia de personal capacitado en distintas áreas de las ciencias naturales en el desarrollo de programas afines a esas aéreas es fundamental en el manejo de material biológico, necesaria en la abstracción y optimización de procesos de laboratorio en el desarrollo de un software clínico.

## **2.REVISION DE LITERATURA**

---

Los laboratorios de análisis clínicos generan gran cantidad de datos que deben ser procesados y organizados cuidadosamente con el fin de reportar resultados de manera confiable y en corto tiempo, la ausencia de un sistema que organice la administración adecuada de los datos es una desventaja a nivel técnico y de competitividad (Roos *et al* 2003).

Los laboratorios se ven presionados a automatizar sus procesos para hacerlos menos dependientes del factor humano y aumentar la eficacia a niveles que han alcanzado otros sectores productivos. Para ello, es necesario examinar los procesos de laboratorios que conducen a un resultado para identificar los puntos donde es posible reducir costos, tiempos de espera y mejorar la calidad del servicio (Clinical and Laboratory Standards Institute 2006). Los sistemas de manejo de información de laboratorios pueden permitir un mejor rastreo de las etapas del proceso, lo cual lleva a controlar de forma detalla las actividades. Si se representa adecuadamente el flujo de trabajo del laboratorio será posible obtener beneficios económicos y técnicos, la capacidad de toma de decisiones basadas en información obtenida del análisis de datos se volverá una ventaja competitiva muy importante para quien implemente correctamente un sistema de este tipo (Marocco & Winstanley 2004).

La validación de resultados se vuelve un tema difícil por la necesidad de contar con profesionales con conocimientos específicos. La logística de una organización depende de una correcta distribución de funciones para lograr una alta productividad por un costo minimizado. De esta forma, la recolección, análisis, distribución y validación de información se convierte en un factor clave en el éxito de cualquier sistema productivo (Owen *et al* 2004).

El beneficio de contar con un manejo de información organizada mediante software se ha reflejado en muchos latitudes, en África por ejemplo, la implantación de un sistema de archivos médicos para registrar expedientes y resultados de pruebas de laboratorio conlleva el manejo de un alto volumen de datos. La información de los

pacientes fue analizada en busca de caracterizar el tiempo de procesamiento de muestras y resultados con el fin de identificar cuáles etapas de los procesos pueden ser optimizadas. Los datos del análisis de información de pacientes y resultados permitieron realizar correcciones en las metodologías empleadas en atención y análisis de muestras, disminuyendo en un 22% la estancia de los pacientes al ser atendidos por el personal, duraban 58% menos tiempo con los proveedores de medicamentos y 38% menos en tiempo de espera. A su vez, el personal clínico interactuó un 50% menos con los pacientes y dos terceras partes menos entre sí (Rotich *et al* 2003), con ello, los costos a nivel administrativo fueron reducidos de manera importante. Además, se ha reportado en otros casos que el tiempo de respuesta ante resultados críticos de pruebas de laboratorio cuando se automatizan ciertos protocolos disminuye desde 22% hasta cifras de 51%, disminuyendo también el tiempo de reporte a médicos y pacientes mediante redes de sistemas de información (Kuperman *et al* 1999). Esto repercute directamente en la eficiencia en la toma de decisiones en situaciones delicadas donde cada segundo es importante. Los datos recolectados a partir de pruebas han permitido realizar estudios de caracterización de poblaciones, de manera totalmente anónima al emplear procedimientos que aíslan la extracción de datos clínicos de los personales (Pace *et al* 2003). En la epidemiología es sumamente importante determinar de manera muy exacta el aumento de la frecuencia de ciertos padecimientos para prever posibles brotes de enfermedades (Madigan *et al* 1999). De esta forma, la planificación de estrategias para combatir focos de padecimientos importantes es beneficiada mediante la monitorización de laboratorios clínicos (Anderson *et al* 2002). Con esta digitalización de la información, es posible estudiar estadísticamente en busca de patrones y determinar comportamientos futuros. Por ejemplo, se ha logrado establecer modelos de predicción para la tasa de mortalidad ante infecciones bacterianas en hospitales, el modelo se basa en la aplicación de técnicas de inteligencia artificial, concretamente: redes neuronales. La búsqueda de patrones por los modelos de redes neuronales se perfilan como herramientas capaces de asistir en la toma de decisiones durante el establecimiento de asepsia en los hospitales al

describir comportamientos difíciles de detectar en una gran cantidad de datos sobre infecciones, tipos de pacientes, épocas del año, infraestructura, entre otras (Jaimes *et al* 2005). Las estructuras complejas de datos deben ser analizadas mediante soluciones de software específicas para ese tipo de información, en el caso de datos de laboratorios e historiales de pacientes es posible concentrar esfuerzos en la detección de sectores de la población sensibles a diversas patologías en un momento y lugar dado, lo cual beneficiaría de manera importante la distribución de los recursos para cuidar la salud de las personas al señalar dónde es necesario realizar inversiones en prevención de enfermedades (Carlsson *et al* 2006).

Un caso en el área médica es el trabajo de Fine *et al* 2007, que ejemplifica el uso de datos provenientes de laboratorios y otras fuentes para alimentar un programa capaz de realizar predicciones de sensibilidad a meningitis bacteriana basado en datos epidemiológicos de casos registrados. Se tomó la información de 696 individuos de una edad que varía de 29 días a 19 años, registrados en un periodo de ocho años en un centro médico y con un diagnóstico confirmado de meningitis, a este conjunto de datos se aplicó un análisis retrospectivo produciendo un modelo logístico multivariado de la regresión. Así se obtuvieron 540 posibles reglas para la predicción de sensibilidad a un tipo de meningitis basándose en el historial almacenado en un sistema electrónico (datos sociales, de pruebas de laboratorio, entre otros). Un trabajo de ese calibre es un prometedor avance en el aprovechamiento de la gran cantidad de datos que se pueden captar, no obstante aun presenta muchas limitaciones pues es necesario contar con información actualizada para crear un modelo que se ajuste a la realidad en un momento dado. Otro punto importante es que esta clase de proyectos es la necesidad de una infraestructura de hardware y software que implemente tecnologías de inteligencia de negocios evitando entorpecer el funcionamiento normal de las instituciones, sin embargo, implica a su vez una concientización de las organizaciones de salud y gobiernos para dictar normativas capaces de respaldar estos proyectos de dimensiones e importancia colosales (Fine *et al* 2007).

Al estudiar las bases de datos de sistemas de laboratorios y médicos un punto destacable es la localización de inconsistencias en el uso de unidades o formas de reportar datos o nombrar pruebas (Khan *et al* 2006). Al analizar los datos de los laboratorios pertenecientes a cinco centros de salud de la India se encontró un alto número de incongruencias en los datos reportados, este análisis se realizó mediante un software orientado a la identificación de nombres y códigos de pruebas (Khan *et al* 2006). Este estudio evidenció la necesidad de contemplar el establecimiento de unidades en los sistemas de registro de datos junto a convenciones estrictas en cuanto a la representación del resultado y la clasificación y estructuración de pruebas en un modelo de software (Khan *et al* 2006). Joseph & Frassica 2005 muestran la variación de la misma información dentro un recinto de salud, en pruebas realizadas sobre los 85 257 resultados de laboratorio en un periodo de 390 días de operación de un centro de cuidados y entrenamiento, la búsqueda de patrones en información permitió identificar divergencia de esquemas de nombramiento de pruebas y de reporte de resultados, estas diferencias se originaban en la ausencia de nomenclaturas obligatorias para el personal encargado de reportar datos o ingresar pruebas a los sistemas empleados.

Un ejemplo de la necesidad de crear fuentes digitales de información de resultados de laboratorios es el caso del VIH en países en vías de desarrollo. La Organización Mundial de la Salud señala la importancia de sistemas automatizados que provean un acceso inmediato y preciso a información de salud de la población necesaria para planear, implementar y evaluar planes que combatan enfermedades antes de convertirse en verdaderas pandemias (Tierney *et al* 2006). Esta recolección de información debe realizarse en todas las fuentes de datos posibles, especialmente en los laboratorios donde se realizan pruebas que permiten un diagnóstico, pues sería posible localizar sujetos enfermos y con base en el registro de pacientes comprender las condiciones en las cuales viven las personas enfermas y de esta manera llegar a comprender cómo, dónde y cuando se enfermaron. Según Tierney *et al* 2006, en un estudio realizado sobre el VIH basándose en archivos médicos y de laboratorio, se determinó que el primer paso es establecer un conjunto mínimo de datos a

considerar en el análisis de la población por lo cual es necesario desarrollar un sistema escalable para el almacenamiento de datos demográficos específicos y variables de una nación a otra. Estos modelos son difíciles de implementar en los países en vías de desarrollo donde existe poca o ninguna experiencia en sistemas de información en el área de la salud, es imprescindible proveer un sistema accesible y flexible en la automatización del manejo de la información (Tierney *et al* 2006).

Las enfermedades crónicas representan un 75% del total de la inversión en salud, alrededor de 100 millones de personas poseen un padecimiento de este tipo y están en tratamiento activo, otras 30 millones desconocen su enfermedad (Dorr *et al* 2007). En un intento por medir la efectividad de los sistemas de información aplicados al manejo de información en enfermedades crónicas se analizaron los resultados de 109 artículos sobre la utilización software médico. El 65 por ciento de los artículos con experimentos controlados y certificados por otros investigadores describía resultados positivos en cuanto a facilidades de almacenamiento, manejo, búsqueda, reporte y análisis de información, el 94 por ciento de los artículos sin verificación de otros investigadores reportaron datos positivos en las aéreas anteriormente citadas (Dorr *et al* 2007). Ante esta perspectiva, se concluyó que el software dedicado al almacenamiento, procesamiento, administración y reporte de datos facilita las investigaciones y trabajo diario en el área de enfermedades crónicas (Dorr *et al* 2007).

El razonamiento basado en casos (**RBC**) es un ejemplo de la utilidad de la información almacenada en software médico y de laboratorios, un sistema de **RBC** puede analizar resultados y de acuerdo a las características que se le permite correlacionar es capaz de dar diagnósticos sobre la normalidad o no del dato basándose en el historial de poblaciones o del paciente (Frithsche *et al* 2002). Los intervalos normales empleados en el análisis son producto de la información ingresada inicialmente al sistema enfocada en la población en general, cuando un experto en medicina u otra área considera que el resultado es normal si se consideran la situación particular del individuo de donde viene la muestra, el experto

le indica al sistema RBC que debe corregir el diagnóstico y el software será capaz de aprender esa flexibilización en la evaluación y aplicarla en casos semejantes (Frithsche *et al* 2002). La experiencia de los profesionales puede ser almacenada en parte en un software adecuadamente diseñado con el fin de aprovecharla y preservarla como parte de la infraestructura tecnológica de la institución o empresa (Frithsche *et al* 2002). Frithsche *et al* (2002) desarrollaron un algoritmo de reconocimiento basado en casos con el fin de crear un sistema automatizado de análisis y alertas de resultados anormales en software aplicado a laboratorios clínicos, el algoritmo fue probado en el reconocimiento de los niveles de creatinina asociados con el rechazo agudo de órganos en trasplantes, los resultados arrojados por el algoritmo se compararon a su vez con los diagnósticos realizados por médicos. Según Frithsche *et al* (2002), el algoritmo de RBC presentó una exactitud del  $78 \pm 2\%$  mientras que los profesionales médicos alcanzaron un  $(69 \pm 5.3\%)$  ( $p < 0.001$ ), por lo cual se consideran resultados positivos en el desarrollo de un sistema confiable de alertas. Este tipo de estudios son posibles al superar la barrera que plantean Khan y colaboradores (2006) con respecto a la unificación de datos. Es por ello que el diseño de un sistema que maneje información en laboratorios de cualquier tipo debe ser pensado desde el punto de vista del usuario, con la experiencia de este en poder determinar las pautas de estandarización de datos entre uno o varios centros de salud.

En California (Estados Unidos de América) el sistema de servicios de salud no lucrativos conocido como Intermountain Healthcare cuenta con un amplio flujo de datos desde distintos medios electrónicos pues muchas pruebas de laboratorio son remitidas a otros centros. Igualmente los servicios se ofrecen a las mismas personas en centros diferentes y la consistencia del expediente conlleva una actualización constante de la información para responder de manera adecuada a las necesidades de las personas. Para establecer un canal de comunicación y unificación de datos adecuados fue diseñado un software para el manejo de resultados de laboratorio empleando conceptos de diseño basados en servicios de conversión de datos en tiempo real, con el fin de disminuir tiempos de espera en el paso de información y

reducir significativamente la inconsistencia de datos de una misma persona en diferentes centros (Staes *et al* 2006). La información se centralizó en una base de datos y mediante acceso web se realiza la inserción, modificación, consulta y eliminación de los datos. Al incorporar un sistema de alertas basado en resultados de laboratorio y contexto del paciente se mejoraron las respuestas de seguimiento en casos críticos como los trasplantes de órganos. El problema asociado a este sistema fue un inconveniente común en casi todos los nuevos paquetes que se implementan en procesos realizados a mano: la aceptación del usuario. Un paso muy importante es el desarrollo de conexiones a equipos para incorporar los resultados al sistema inmediatamente que están listos, de esta manera se evita el ingreso manual de los mismos pues generalmente es una tarea que consume mucho tiempo cuando se manejan volúmenes altos de muestras. El valor de una conexión del software con un equipo de análisis se estima es la parte más costosa de este proyecto de software de laboratorio, con un costo en Estados Unidos de América de aproximadamente \$15000 y hasta el momento del artículo eran necesarias 85 conexiones diferentes (Staes *et al* 2006). Finalmente, uno de los principales problemas es la falta de estandarización de nombres de pruebas, resultados y protocolos de comunicación entre sistemas para almacenamiento de información de expedientes médicos y resultados de laboratorio (Staes *et al* 2006).

Al desarrollar sistemas para administración y análisis de datos en laboratorios es necesario tomar en cuenta entonces a los usuarios meta, los tipos de prueba realizados, el análisis de resultados, la distribución de los mismos, la estandarización de nombres de pruebas, resultados y cualquier información que deba introducirse al software. Además es necesario tener en mente las necesidades de los laboratorios para evitar un conflicto con las capacidades de desarrollo y lograr crear un sistema que agilice procesos en lugar de entorpecerlos. Así, sea un sistema clínico o industrial es necesario considerar este conjunto de factores para desarrollar un paquete útil.

La caracterización de poblaciones, el manejo de información y su análisis es un brazo importante para alcanzar los objetivos de una entidad pública o privada. Es necesario brindar un sistema sólido que supla las necesidades de una industria creciente, de tal manera que se disminuyan errores y se creen nuevas expectativas en la investigación en productos y servicios que atiendan a las necesidades de salud de la población en área de diagnóstico (inmunológico, molecular, entre otros) y tratamiento al paciente (Roos *et al* 2003).

## **3.OBJETIVOS**

---

### **3.1 Objetivo General**

- Desarrollar un software que facilite el manejo y análisis de información en laboratorios clínicos e industriales.

### **3.2 Objetivos específicos**

- Analizar los sistemas existentes de Labsys y su desempeño en laboratorios clínicos.
- Diseñar la arquitectura de la aplicación desde la perspectiva del trabajo de laboratorio en pruebas de detección, captación y comprensión de resultados.
- Construir bases de datos en Microsoft SQL Server 2005 mediante relaciones de datos ajustadas al flujo de datos real dentro de laboratorios clínicos.
- Construir procedimientos almacenados con T-SQL 2005 adecuándolos a las tareas comunes de manipulación de información de laboratorios.
- Crear módulos de mantenimiento y procesos en la plataforma .NET 2.0 tomando como fundamento la aplicabilidad en laboratorios clínicos.
- Realizar pruebas de implementación del software en laboratorios clínicos para corregir errores mediante la organización de información y análisis de resultados de exámenes químicos, microbiológicos e inmunológicos.

## **4.MÉTODOS**

---

La aplicación Labsys se desarrolló bajo un proceso de diseño e implementación basado en estudio de necesidades de almacenamiento y manejo de información, usuarios meta y versiones anteriores de Labsys.

Es necesario definir algunos términos de uso común en el sistema:

- **Pruebas:** Unidad de trabajo que en el laboratorio corresponde al procesamiento de una muestra y la obtención de un resultado.
- **División:** Agrupamiento opcional de pruebas.
- **Examen:** Agrupamiento de divisiones y/o pruebas, es la unidad solicitada empleada en la solicitud de pruebas de laboratorio.
- **Sección:** Agrupamiento de exámenes de laboratorio, es la unidad máxima de categorización de pruebas.
- **Expediente:** Objeto de base de datos que almacena la información sobre el origen de un muestra (en laboratorios clínicos por ejemplo representa a un paciente).
- **Muestra:** Objeto de base de datos correspondiente a una muestra de material orgánico o inorgánico enviada a un laboratorio.
- **Repetición:** Objeto de base de datos que representa la repetición de una prueba de una solicitud en una muestra.
- **Solicitud:** Objeto de base de datos representando la petición de varios exámenes a una muestra.
- **Solicitud Individual:** Objeto de base de datos que registra las peticiones de exámenes a una muestra asociada a un expediente, trabajando una muestra por petición.
- **Solicitud con Muestras:** Objeto de base de datos que almacena la petición de exámenes para un conjunto de muestras asociadas a un expediente.
- **Solicitud por Lotes:** Objeto de base de datos que representa la petición de exámenes a una muestra sin caracterizar su origen.

- **Valores Normales (referencia):** Magnitudes consideradas normales como resultados de una prueba según la literatura empleada en el laboratorio que los inserte, corresponden a un ámbito que permite evaluar la existencia de anormalidades en el origen de la muestra.
- **Valores de Linealidad:** Magnitudes consideradas anormales en los resultados de una prueba según la literatura empleada en el laboratorio que los inserte, corresponden a un ámbito que permite evaluar la existencia de anormalidades en el protocolo o equipo empleado al realizar la prueba, generalmente corresponden a valores muy altos o bajos, poco probables de registrar en un origen de muestra determinado.

#### ***4.1 Etapa I: Análisis de sistemas anteriores***

Esta etapa consiste en realizar un estudio de los sistemas Labsys 3.0 y 4.0. A partir de la información recabada se seleccionan las funcionalidades que mejor se ajusten para analizar los datos del laboratorio.

De esta forma se caracteriza el software existente, se reestructura de acuerdo al trabajo realizado en laboratorio, para hacerlo más práctico y orientado hacia una funcionalidad generalizada en un modelo capaz de ofrecer posibilidades de especificación para agilizar tareas. En esta etapa se realiza un modelo abstracto del flujo de trabajo en un laboratorio con el fin de tener la lógica de trabajo necesaria para crear una representación a nivel de software.

La principal consigna en esta fase es tener una visión clara del trabajo de laboratorio, funcionamiento de equipos, tratamiento de muestras, principios de las técnicas empleadas. Ese conocimiento es fundamental para realizar una representación acertada de los procesos, así el software puede amoldarse a distintos procesos muy semejantes y se evita el problema de generar una herramienta que obliga a los laboratorios a realizar un cambio en su flujo de trabajo.

#### ***4.2 Etapa II: Diseño de la arquitectura de la aplicación***

La Arquitectura de Software consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software en un sistema determinado.

En este punto es necesario retomar la información recolectada en la fase anterior sobre la abstracción de procesos en laboratorios y funcionalidades deseadas del software. A partir de esos datos se diseña un esquema del sistema, sus subsistemas, componentes, métodos de comunicación externos e internos, tecnologías empleadas, posibilidades de acoplamiento con otros sistemas y finalmente, interacción con usuarios. De esta forma se genera un diseño a partir del cual se crean unidades de código específicas y se establece la manera en la cual deben unirse. Al establecer este modelo es posible crear los planes de desarrollo, asignar labores y tener una predicción de costos, tiempo y beneficios.

Sin embargo, además de diseñar un correcto flujo de trabajo es necesario realizar una constante inspección de la arquitectura del sistema, pues una errada estructura o selección de tecnologías pueden limitar la aplicabilidad del software. Por ello es necesario cotejar cada diseño de componentes y comunicación contra el trabajo de laboratorio. Si los arquitectos de software desconocen completamente los detalles de los procesos a nivel de laboratorio es muy probable que cometan errores de diseño que son generalmente detectados en la fase de prueba del sistema. Ello implica realizar importantes correcciones al sistema e incluso puede significar que gran parte del mismo llegue a ser inaplicable.

### **4.3 Etapa III: Diseño de la bases de datos**

En esta etapa se crea el diseño de la base de datos, es un punto clave para corregir los errores a nivel de abstracción. Una base de datos es un objeto de software capaz de contener otros objetos que almacenan información, realizan operaciones sobre los datos o sobre objetos en la base de datos.

La base de datos se diseña con base en un modelo relacional, el cual permite ordenar la información en una determinada jerarquía de relaciones, los datos pueden derivarse de otros datos “*heredando*” el contenido de los datos padre (Dewson 2006). Por ejemplo, un conjunto de variables que represente las características comunes de las pruebas inmunológicas se organiza en un cuadro (referido de ahora en adelante como tabla, por la denominación que dan los expertos en el área según Dewson 2006) que será el punto de partida para derivar tablas de pruebas específicas de inmunología. Cada tabla hija tendrá sus variables específicas pero todas tendrán variables comunes. Estas relaciones condicionan todos los procesos de manejo de datos. Una incorrecta relación, asignación de tipos (número, texto, entre otros) o determinación de valores requeridos puede significar un desastre al probar el sistema bajo una condición que los diseñadores no hayan previsto.

Para realizar tal diseño según las prácticas recomendadas por autores (Davidson, Kline y Windisch 2006) es necesario:

- Definir el tipo de sistema de la base de datos (OLTP u OLAP).
- Determinar los datos a almacenar y como deben ser almacenados (tipo de información).
- Diseñar tablas.
- Construir relaciones.
- Normalizar la estructura de la base de datos empleando las formas de normalización (1NF, 2NF, 3NF).
- Definir esquema de seguridad (logins, roles y schemes).
- Implementar el diseño en el motor de base de datos Microsoft SQL Server 2005.

Una vez más es imperativa la revisión del diseño de las bases de datos al cotejarse con el flujo de información en un laboratorio a nivel de los resultados generados, incertidumbres, procesos de diluciones, entre otros. El diseño de la base de datos debe acoplarse al trabajo de laboratorio y para desarrollador de software muchos detalles pueden pasar desapercibidos si no posee experiencia en manipulación de material biológico.

#### ***4.4 Etapa IV: Elaboración de Procedimientos Almacenados***

Existen muchas formas de manejar la información en bases de datos, una de ellas consiste en crear Procedimientos Almacenados, los cuales son un conjunto de comandos compilados de T-SQL (lenguaje empleado por el motor de base de datos) directamente accedidos por SQL Server 2005. Estas entidades son cápsulas de funcionalidades que se alojan en la base de datos, se diseñan de acuerdo a necesidades específicas (Dewson 2006).

En esta etapa se crean los procedimientos requeridos para obtener la información que los distintos componentes del software necesita y además, para recibir la información entrante. Estas unidades de código permiten separar el manejo de datos del resto del software.

Cada procedimiento almacenado debe ser fiscalizado empleando como referencia el flujo de trabajo en laboratorios, pues un programador sin ninguna experiencia en laboratorio puede crear procedimientos que no se ajustan a todos los posibles casos y el error se llega a detectar en la fase de prueba, cuando el producto está completo. Los procedimientos almacenados se crean siguiendo este patrón (Dewson 2006):

- Identificar operación (es) a realizar.
- Determinar datos a manejar y sus tipos.
- Establecer cuáles son los permisos de ejecución.
- Crear los procedimientos almacenados.

#### ***4.5 Etapa V: Creación de interfaz y ensamblado de prototipo***

Esta fase implica la creación de las pantallas diseñadas en la segunda etapa, permiten que el usuario se comunique con la aplicación, son la parte visible de la misma y son conocidas comúnmente como interfaz con el usuario.

Una vez más es necesaria una última revisión para evitar errores de diseño. A nivel de interfaz de usuario la implementación inadecuada del flujo de trabajo dentro de un laboratorio se traduce generalmente en dificultades para el usuario, al consumir mucho tiempo realizando tareas comunes o mostrando información de manera poco práctica, por ejemplo. Estas situaciones que no imposibilitan el desarrollo del proyecto, tienden de predisponer al usuario a no utilizar la aplicación.

Las pantallas se ensamblan para navegar entre ellas, ahora sólo es necesario conectar los componentes que tienen las funciones del sistema (también conocidas como lógica de negocios). Con este prototipo aun inoperante de la aplicación es necesario tomar el tiempo para navegar a través de los procesos desde el punto de vista del usuario de un laboratorio, así se realiza la revisión final en esta etapa.

#### ***4.6 Etapa VI: Desarrollo de módulos de mantenimientos***

Los módulos de mantenimientos son las secciones del software donde el usuario define la configuración específica de su sistema. De esta forma por ejemplo, crea el catálogo de exámenes y pruebas de su laboratorio, administración de pacientes, manejo de reactivos, determinación de umbrales de tolerancia de valores, linealidad de pruebas, entre otros.

Se desarrolla entonces la lógica de la aplicación que corresponde a la especificación del esquema de flujo de trabajo. El software contiene un esquema básico que es expandido por el usuario, este agrega los datos que necesita y configura los procesos a realizar. En esta sección es imperativa la vigilancia de la implementación del diseño, pues es aquí donde el usuario comienza a encontrar problemas al haber confrontaciones de los datos que exige el sistema comparados con los que el usuario necesita ingresar.

Una vez más, hay que probar el sistema desde el punto de vista del personal de los laboratorios y si es necesario entrevistarse con ellos para determinar correcciones a tiempo.

#### ***4.7 Etapa VII: Desarrollo de módulos de procesos***

Estos módulos corresponden a las tareas principales del sistema, que implican el uso de los datos ingresados en los módulos de mantenimientos. Los procesos implican la solicitud de exámenes, planificación, captura de resultados (manual o desde equipos automatizados), análisis, validación, almacenamiento y distribución en informes.

Se crea la lógica de negocios para los procesos citados anteriormente con la inevitable necesidad de una participación activa como desarrollador y inspector en errores de implementación del diseño, cuestionando siempre la utilidad práctica de los módulos en laboratorio bajo distintas situaciones.

#### ***4.8 Etapa VIII: Ensamblaje de la aplicación***

Una vez que se cuente con esta versión completa para pruebas (versión beta) se instalará en varios laboratorios para detectar posibles errores.

Aquí se observa el desempeño del software, se encuentran fallas, se comprende cuál es su origen desde el punto de vista de código fuente y de la naturaleza del proceso de laboratorio que se intenta completar. Esta es la última etapa para detectar errores críticos, es fundamental establecer la vigilancia donde puede haber situaciones de manejo de información que se escapen al criterio de los desarrolladores de software. Básicamente, se realizarán distintos exámenes clínicos y se analizará la información que muestra el sistema para comprobar manualmente la veracidad de los mismos.

## **5.RESULTADOS**

---

### **5.1 Análisis de sistemas Labsys 3.0 y Labsys 4.0**

Se detectaron los siguientes puntos positivos:

- Estructura del catálogo de pruebas en cuatro niveles definidos por el usuario (Secciones, Exámenes, Divisiones, Pruebas).
- Manejo de valores normales para cada prueba.
- Trabajo de laboratorio rastreable en unidades llamadas solicitudes.
- Orígenes de muestra caracterizados mediante tres niveles y campos fijos en la base de datos.
- Definición de valores normales y catálogo por parte del usuario.
- Sistema de seguridad por perfiles personalizables.
- Bitácora de operaciones sobre el sistema.

Se detectaron los siguientes puntos negativos:

- Duplicación de información en el diseño de la base de datos.
- Ausencia de separación entre la lógica de acceso y manejo de datos de la interfaz de usuario.
- Consultas con código T-SQL alojadas en el código de VB 6.0 que compone la interfaz de usuario.
- Las solicitudes pueden manejarse a nivel de muestras pero no es posible solicitar pruebas como repeticiones de una muestra sin crear solicitudes nuevas.
- La interfaz de usuario no representa de manera sencilla la abstracción de la información, como la creación del catálogo técnico que en la versión Labsys 3.0 involucra emplear siete pantallas.
- Los orígenes de muestras poseen características fijas, por lo cual cuando es necesario incluir una nueva característica es necesaria la modificación de la base de datos y código de la aplicación.

Después de realizar un análisis de los procesos en software de laboratorios desde la perspectiva de un usuario final se identificaron tres fuentes principales de datos:

- La información de la muestra en la cual se solicita exámenes.
- El conjunto de pruebas que conforma cada examen.
- Los resultados de las pruebas.

Con base en el análisis anterior se formuló un listado de problemas específicos a tratar en el software y se presentaron las soluciones correspondientes (Cuadro 1).

**Cuadro 1: Soluciones a problemas de diseño en el sistema Labsys**

<b>Necesidad</b>	<b>Solución</b>
<b>Una flexible caracterización de muestras</b>	<ul style="list-style-type: none"> <li>• Clasificación de áreas de trabajo en laboratorios y división por géneros.</li> <li>• Categorías de características dependientes de categorías de Trabajo.</li> <li>• Características asociadas a cada expediente.</li> <li>• Número de características limitadas por capacidad de hardware.</li> </ul>

**Cuadro 1: Soluciones a problemas de diseño en el sistema Labsys (Continuación)**

<p><b>Un robusto manejo de información de pacientes, muestras y repeticiones.</b></p>	<ul style="list-style-type: none"> <li>• Creación de expedientes agrupando solicitudes.</li> <li>• Un expediente no tiene un número limitado de solicitudes.</li> <li>• Solicitudes son exámenes realizados a muestras</li> <li>• Las muestras poseen un identificador único lo que permite manejar varias muestras en una solicitud, relacionadas por el expediente</li> <li>• Una muestra es una repetición con identificador de repetición único, lo que permite manejar cualquier número de repeticiones por muestra.</li> <li>• Número de elementos creados para un expediente limitado únicamente por la capacidad del hardware.</li> </ul>
<p><b>Análisis de datos específico y eficiente</b></p>	<ul style="list-style-type: none"> <li>• Creación de valores de referencia asociados a categorías de trabajo, géneros, pruebas y metodologías de obtención de datos.</li> <li>• Reporte preventivo para el análisis de profesionales</li> </ul>

<b>Detección de problemas en equipos y protocolos</b>	<ul style="list-style-type: none"> <li>• Creación de umbrales para valores alterados en los resultados de las pruebas.</li> <li>• Asociados a valores por referencia (valores normales) reportados para las pruebas de acuerdo al origen de muestra.</li> </ul>
---	---

**Cuadro 1: Soluciones a problemas de diseño en el sistema Labsys (Continuación)**

<b>Control del proceso de análisis</b>	<ul style="list-style-type: none"> <li>• Identificación de estado del análisis de muestras</li> <li>• Establecimiento de cuatro etapas en la base de datos: recepción de solicitud, obtención de resultados, validación y reporte de resultados.</li> <li>• Almacenamiento de fecha y hora en cada etapa para controlar el tiempo de procesamiento de cada resultado en cada prueba de cada muestra y repetición.</li> </ul>
<b>Creación de catálogo de pruebas del laboratorio</b>	<ul style="list-style-type: none"> <li>• Representación jerarquizada de datos.</li> <li>• Unificación de pantallas para facilitar la comprensión de los conceptos de agrupamiento de pruebas clínicas.</li> <li>• Asociación de valores por referencia y umbrales con pruebas clínicas durante su creación.</li> <li>• Identificación de valores posibles en pruebas clínicas.</li> </ul>

**Cuadro 1: Soluciones a problemas de diseño en el sistema Labsys (Continuación)**

<b>Disminución de tiempos de digitado para el trabajador de laboratorios</b>	<ul style="list-style-type: none"><li>• Creación de listas de pruebas para inserción simultánea, realizada por el usuario.</li><li>• Abreviaturas reemplazadas por palabras o frases, el usuario crea las abreviaturas y su significado.</li><li>• Sistema de auto digitación de palabras en búsquedas.</li><li>• Asistentes de búsqueda sencillos, bajo un mismo patrón y con filtros en resultados de una búsqueda ya filtrada por otros criterios.</li></ul>
--	---

## 5.2 Arquitectura

El sistema de diseño de la siguiente manera: el motor de base de datos Microsoft SQL Server 2005 se encuentra en una computadora denominada "Server". La base de datos incorpora una subcapa de manejo de datos conformada por los procedimientos almacenados y las estaciones de trabajo se conectan al servidor mediante la red local (Figura 1).

Los clientes son computadoras de menor capacidad de procesamiento que realizan tareas mínimas de interpretación de datos, presentan la información al usuario y este envía los cambios que crea pertinentes. Los datos se procesan mediante clases basadas en ADO.NET. Los requerimientos recomendados del servidor de laboratorios pequeños son máquinas con 1 GB RAM DDR, procesador Pentium 4 HT superior a 2.0 Ghz, 10 MB disco duro (la base de datos aumenta su tamaño con el ingreso de información).

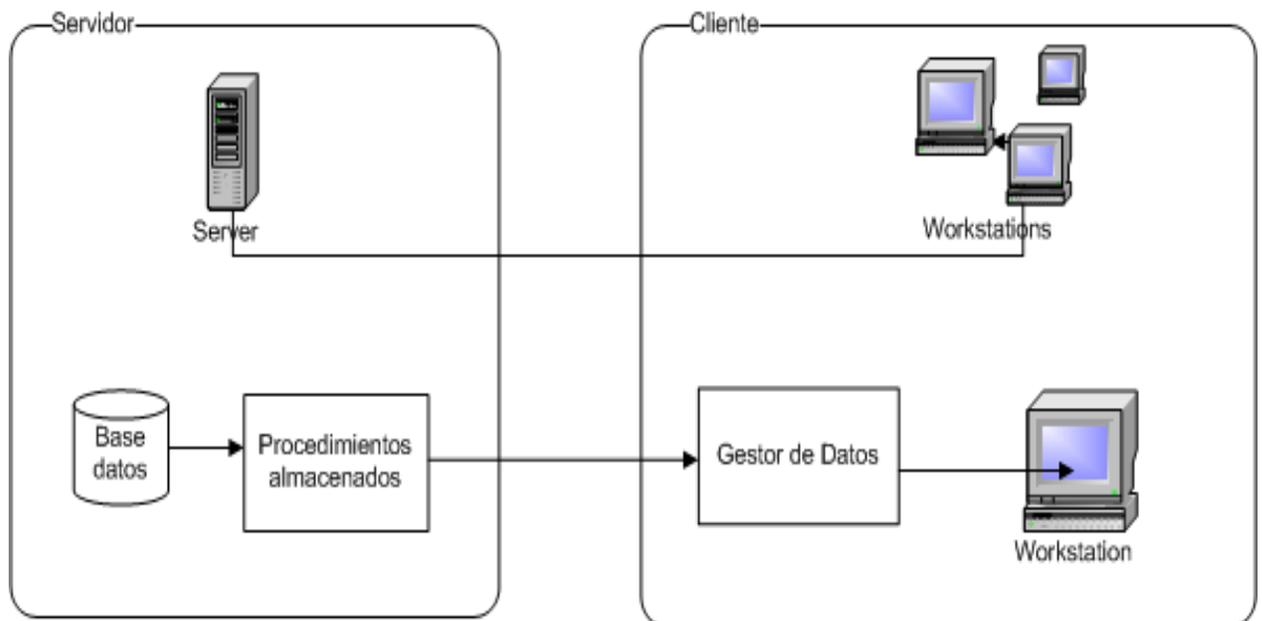


Figura 1: Arquitectura del sistema Labsys 5.0

### 5.3 Diseño de base de datos

Se implementó un sistema de base de datos OLTP con un modelo relacional y el conjunto de datos a almacenar pueden enumerarse de manera general:

- Orígenes de muestra.
- Catálogo Técnico.
- Solicitudes.
- Resultados de pruebas.
- Perfiles y listas de exámenes predeterminados.
- Abreviaturas.
- Personal responsable.
- Usuarios.
- Bitácora.

Al desglosar la información mínima requerida a almacenar en la base de datos se realizó un boceto de los principales datos a almacenarse, la forma en que se agrupan y los códigos que identifican cada elemento (Cuadro 2). Cada elemento en el boceto de diseño consta de identificadores simples (un campo) o complejos (varios identificadores correspondiendo a las relaciones jerárquicas con otros objetos).

**Cuadro 2: Datos principales en el funcionamiento del sistema Labsys**

<b>Conjuntos de datos</b>	<b>Campos</b>
Categorías	<ul style="list-style-type: none"><li>• Id Categoría</li><li>• Nombre categoría</li></ul>
Géneros	<ul style="list-style-type: none"><li>• Id Géneros</li><li>• Id Categoría</li><li>• Nombre Género</li></ul>
Nivel 1	<ul style="list-style-type: none"><li>• Id Nivel 1</li><li>• Nombre Nivel 1</li></ul>
Nivel 2	<ul style="list-style-type: none"><li>• Id Nivel 1 e Id Nivel 2</li><li>• Nombre Nivel 2</li></ul>

**Cuadro 2: Datos principales en el funcionamiento del sistema Labsys (continuación)**

Nivel 3	<ul style="list-style-type: none"> <li>• Id Nivel 1, Id Nivel 2 e Id Nivel 3</li> <li>• Nombre Nivel 3</li> </ul>
Expediente	<ul style="list-style-type: none"> <li>• Id Expediente</li> <li>• Nombre expediente</li> <li>• Id género</li> <li>• Id Nivel 1, Id Nivel 2 e Id Nivel 3</li> </ul>
Categorías demográficas	<ul style="list-style-type: none"> <li>• Id Categoría</li> <li>• Id Categoría Demográfica</li> <li>• Nombre</li> </ul>
Características Demográficas	<ul style="list-style-type: none"> <li>• Id Categoría</li> <li>• Id Categoría Demográfica</li> <li>• Id Característica Demográfica</li> <li>• Nombre</li> </ul>
Resultados Características Demográficas	<ul style="list-style-type: none"> <li>• Id Categoría</li> <li>• Id Categoría Demográfica</li> <li>• Id Característica Demográfica</li> <li>• Resultado</li> </ul>
Listado de profesionales de laboratorio	<ul style="list-style-type: none"> <li>• Id Profesional</li> <li>• Nombre</li> </ul>
Listado de referencias	<ul style="list-style-type: none"> <li>• Id referencia</li> <li>• Nombre</li> </ul>
Secciones	<ul style="list-style-type: none"> <li>• Id Sección</li> <li>• Nombre</li> </ul>
Exámenes	<ul style="list-style-type: none"> <li>• Id Sección</li> <li>• Id Examen</li> </ul>

	<ul style="list-style-type: none"> <li>• Nombre</li> </ul>
--	--

**Cuadro 2: Datos principales en el funcionamiento del sistema Labsys (continuación)**

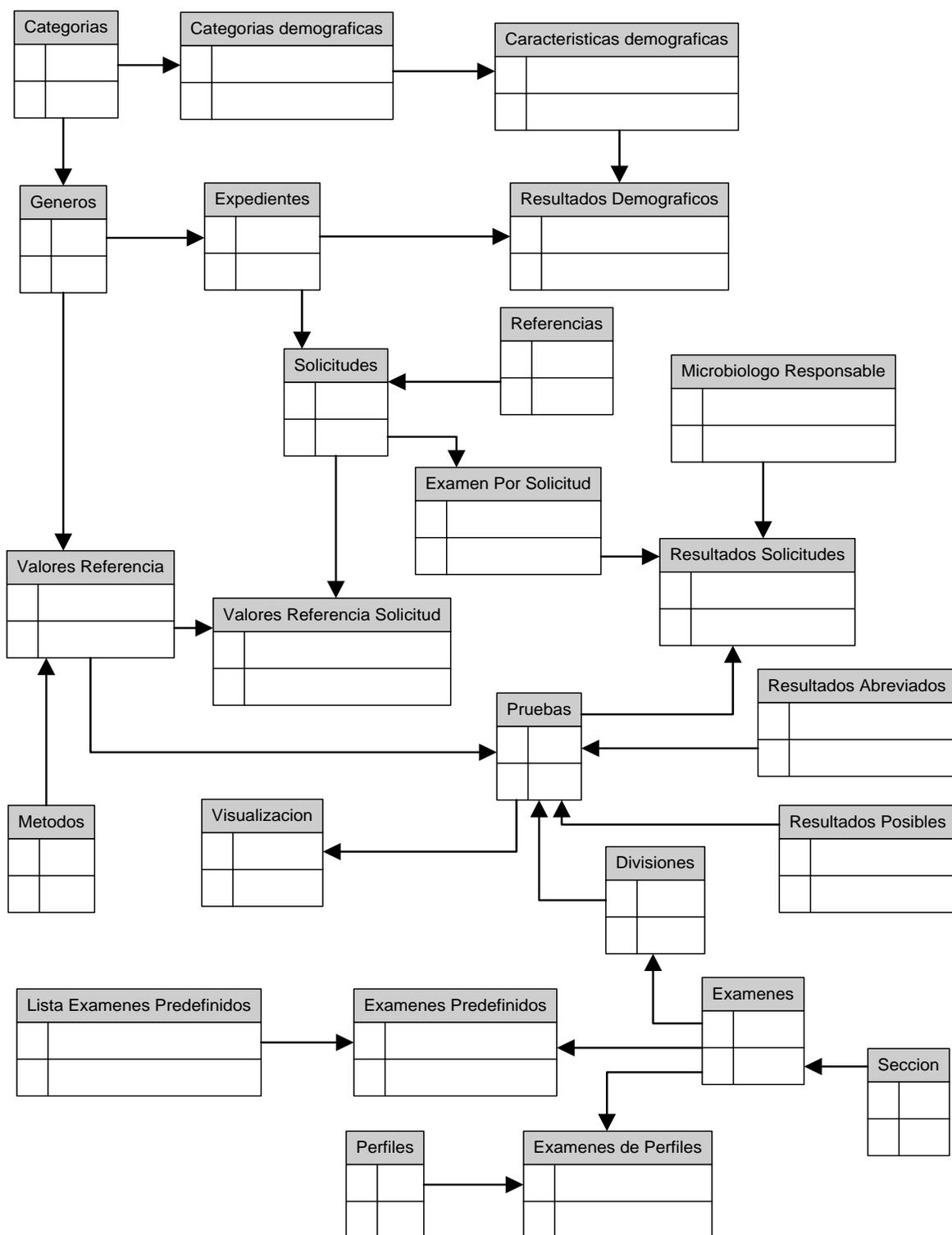
Divisiones	<ul style="list-style-type: none"> <li>• Id Sección</li> <li>• Id Examen</li> <li>• Id División</li> <li>• Nombre</li> </ul>
Pruebas	<ul style="list-style-type: none"> <li>• Id Sección</li> <li>• Id Examen</li> <li>• Id División</li> <li>• Id Pruebas</li> <li>• Nombre</li> </ul>
Métodos	<ul style="list-style-type: none"> <li>• Id Método</li> <li>• Nombre</li> </ul>
Valores normales	<ul style="list-style-type: none"> <li>• ID Género</li> <li>• ID Método</li> <li>• Id Sección</li> <li>• Id Examen</li> <li>• Id División</li> <li>• Id Pruebas</li> <li>• Valor mínimo</li> <li>• Valor Máximo</li> </ul>
Valores linealidad	<ul style="list-style-type: none"> <li>• ID Género</li> <li>• ID Método</li> <li>• Id Sección</li> <li>• Id Examen</li> <li>• Id División</li> <li>• Id Pruebas</li> <li>• Valor mínimo</li> </ul>

	<ul style="list-style-type: none"> <li>• Valor Máximo</li> </ul>
Resultados Abreviados	<ul style="list-style-type: none"> <li>• Abreviatura</li> <li>• Equivalencia</li> </ul>

**Cuadro 2: Datos principales en el funcionamiento del sistema Labsys (continuación)**

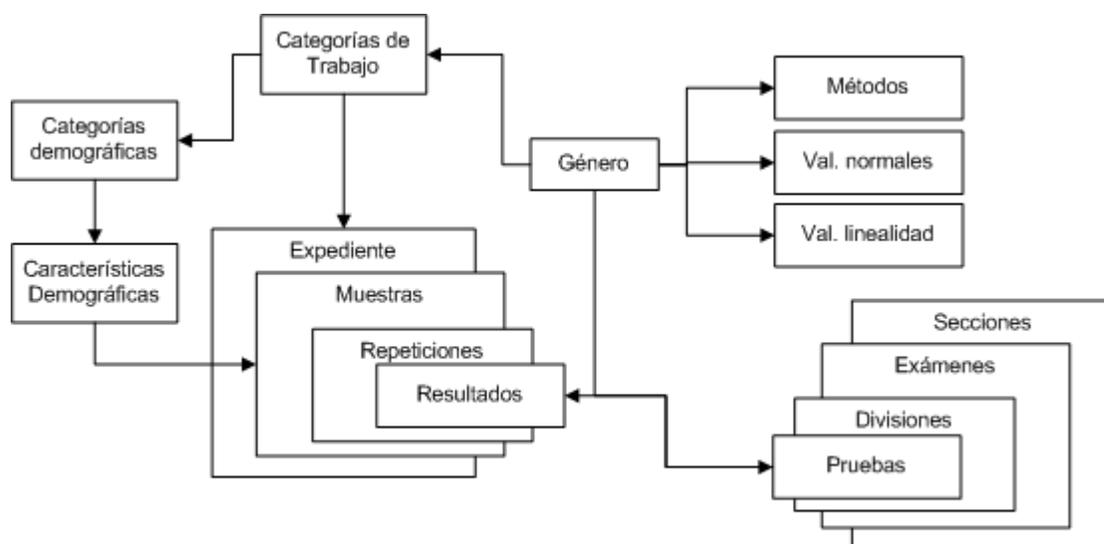
Perfiles	<ul style="list-style-type: none"> <li>• Id Lista</li> <li>• Nombre</li> </ul>
Datos del perfil	<ul style="list-style-type: none"> <li>• Id lista</li> <li>• Elementos integrantes</li> </ul>
Exámenes predefinidos	<ul style="list-style-type: none"> <li>• Id lista</li> <li>• Nombre</li> </ul>
Datos de exámenes predefinidos	<ul style="list-style-type: none"> <li>• Id lista</li> <li>• Elementos integrantes</li> </ul>
Solicitudes	<ul style="list-style-type: none"> <li>• Id Solicitud</li> <li>• Id Expediente</li> <li>• Pruebas</li> <li>• Fecha y hora solicitud</li> <li>• Fecha y hora de resultados</li> <li>• Resultados</li> <li>• Fecha y hora de validación</li> <li>• Fecha y hora de impresión</li> <li>• Usuario responsable de cada etapa</li> </ul>
Usuarios	<ul style="list-style-type: none"> <li>• Id Usuario</li> <li>• Id Perfil</li> <li>• Nombre</li> </ul>
Bitácora	<ul style="list-style-type: none"> <li>• Id Usuario</li> <li>• Id Evento</li> <li>• Fecha y hora de evento</li> </ul>

Al aplicar un diseño de modelo relacional sobre la totalidad de los datos se separaron algunos objetos del boceto del diseño en varias tablas con el fin de vincular la mayor cantidad de elementos a un objeto padre, como el caso de las solicitudes, donde la lista de exámenes destinados a cada solicitud corresponde a una lista en una tabla separada de la definición forma de la solicitud, y los resultados a las pruebas que componen cada examen están su vez en otra tabla. Se realizó un proceso de normalización sobre la estructura y el resultado final es una base de datos compuesta por 36 tablas (excluyendo las correspondientes a seguridad), implicando una disminución si se compara con las 57 tablas de las versiones 3.0 y 4.0 del software Labsys (Figura 2).



**Figura 2: Entidades que componen el núcleo de la base datos del sistema**

La información fue organizada en entidades denominadas categorías, las cuales envuelven tipos de géneros, orígenes de muestra y las características asociadas. Los expedientes pueden contener muestras que al ser procesadas pueden tratarse como repeticiones pendientes de resultados. Las pruebas están contenidas en una estructura de cuatro niveles, donde el nivel padre corresponde a las secciones, seguido por exámenes, divisiones y finalmente las pruebas. Estas se encuentran asociadas a valores normales y de linealidad, los cuales a su vez están ligados a métodos y géneros (Figura 3).



**Figura 3: Diagrama de relación de entidades en manejo de resultados**

De esta manera un usuario puede definir una categoría llamada “Laboratorio Clínico”, luego es capaz de insertar los géneros “Femenino”, “Masculino” e “Indefinido”. Además, define una categoría de características demográficas pertenecientes a “Laboratorio Clínico” denominada: “Datos Paciente”. El usuario puede crear las características demográficas denominadas “Masa Corporal”, “Estatura”, “Edad”, “Estado Civil”, entre otros. Empleando esta metodología es posible para el usuario insertar un número indefinido de categorías, características, géneros, pruebas, valores normales y de linealidad manteniendo siempre una relación capaz de facilitar el análisis sobre la información.

## 5.4 Procedimientos almacenados

Se creó una subcapa de acceso a datos alojada en la base de datos. Todas las operaciones sobre datos son realizadas a través de estos bloques de código.

Se crearon 394 procedimientos que sirven como intermediarios de la base de datos con el resto del software, los cuales contemplan las necesidades actuales y varias opciones para expansiones del sistema (Figura 4). Los procedimientos almacenados fueron diseñados con base en las necesidades de manejo de información detectadas al realizar el diseño de la base de datos tomando en cuenta los datos necesarios para beneficiar a un trabajador de laboratorios (Cuadro 1).

El desarrollo de los procedimientos almacenados implicó el trabajo en T-SQL sobre el motor de base de datos Microsoft SQL Server 2005 creando unidades de código con diversa longitud y finalidad. Están presentes procedimientos que realizan las operaciones CRUD (**C**reate, **R**ead, **U**ppdate y **D**elte) además de ciertos procedimientos destinados a actuar como parte de otros procedimientos almacenados.

```
set ANSI_NULLS ON
set QUOTED_IDENTIFIER ON
go

-- =====
-- Author:          fmarin & fsalazar
-- Create date:    06-10-06
-- Description:    Insertar una nueva solicitud
-- =====
CREATE PROCEDURE [dbo].[spl_InsertarSolicitud]
    @Valor_CodFechaSolicitud int, @Valor_Expediente int, @Valor_Muestra
int,
    @Valor_FechaSolicitud datetime, @Valor_Urgencia varchar(2),
@Valor_Observacion varchar(150),
    @Valor_CodReferencia int, @Valor_Imp varchar(2),@Valor int OUTPUT
AS
declare @sol int,
        @cant int
set @sol = 1
BEGIN

DECLARE @Valor_ConsecSolicitud AS INT
SELECT @Valor_ConsecSolicitud = Valor_Consec FROM
```

**Listado 1: Ejemplo de un procedimiento almacenado.**

```

dbo.Tbl_OpcionesGenerales

if (@Valor_ConsecSolicitud <> 1)
begin
    SELECT @cant = count(Cod_ConsecSolicitud)
    FROM dbo.Tbl_Solicitud

    if (@cant <> 0)
    begin
        SELECT top 1 @sol = Cod_ConsecSolicitud + 1
        FROM dbo.Tbl_Solicitud
        WHERE Cod_FechaSolicitud = @Valor_CodFechaSolicitud
        ORDER BY Cod_ConsecSolicitud desc

        INSERT INTO dbo.Tbl_Solicitud (Cod_FechaSolicitud,
Cod_ConsecSolicitud, Cod_Expediente, Cod_Muestra,
Fecha_Solicitud, Urgencia, Obs_Solicitud, Cod_Referencia,
Imp_Solicitud)
VALUES (@Valor_CodFechaSolicitud, @sol, @Valor_Expediente,
@Valor_Muestra,
@Valor_FechaSolicitud, @Valor_Urgencia, @Valor_Observacion,
@Valor_CodReferencia, @Valor_Imp)

        SELECT @Valor = @sol
    end
    ELSE
    BEGIN
        INSERT INTO dbo.Tbl_Solicitud (Cod_FechaSolicitud,
Cod_ConsecSolicitud, Cod_Expediente, Cod_Muestra,
Fecha_Solicitud, Urgencia, Obs_Solicitud, Cod_Referencia,
Imp_Solicitud)
VALUES (@Valor_CodFechaSolicitud, 1, @Valor_Expediente,
@Valor_Muestra,
@Valor_FechaSolicitud, @Valor_Urgencia, @Valor_Observacion,
@Valor_CodReferencia, @Valor_Imp)
        SELECT @Valor = 1
    END
end
else
begin
    INSERT INTO dbo.Tbl_Solicitud (Cod_FechaSolicitud,
Cod_ConsecSolicitud, Cod_Expediente, Cod_Muestra,
Fecha_Solicitud, Urgencia, Obs_Solicitud, Cod_Referencia,
Imp_Solicitud)
VALUES (@Valor_CodFechaSolicitud, @Valor_ConsecSolicitud,
@Valor_Expediente, @Valor_Muestra,
@Valor_FechaSolicitud, @Valor_Urgencia, @Valor_Observacion,
@Valor_CodReferencia, @Valor_Imp)
    update dbo.Tbl_OpcionesGenerales
    set Valor_Consec = 0

    SELECT @Valor = @Valor_ConsecSolicitud
end
end

```

**Listado 1: Ejemplo de un procedimiento almacenado (Continuación).**

END

```
set ANSI_NULLS ON
```

```
set QUOTED_IDENTIFIER ON
```

**Listado 1: Ejemplo de un procedimiento almacenado (Continuación).**

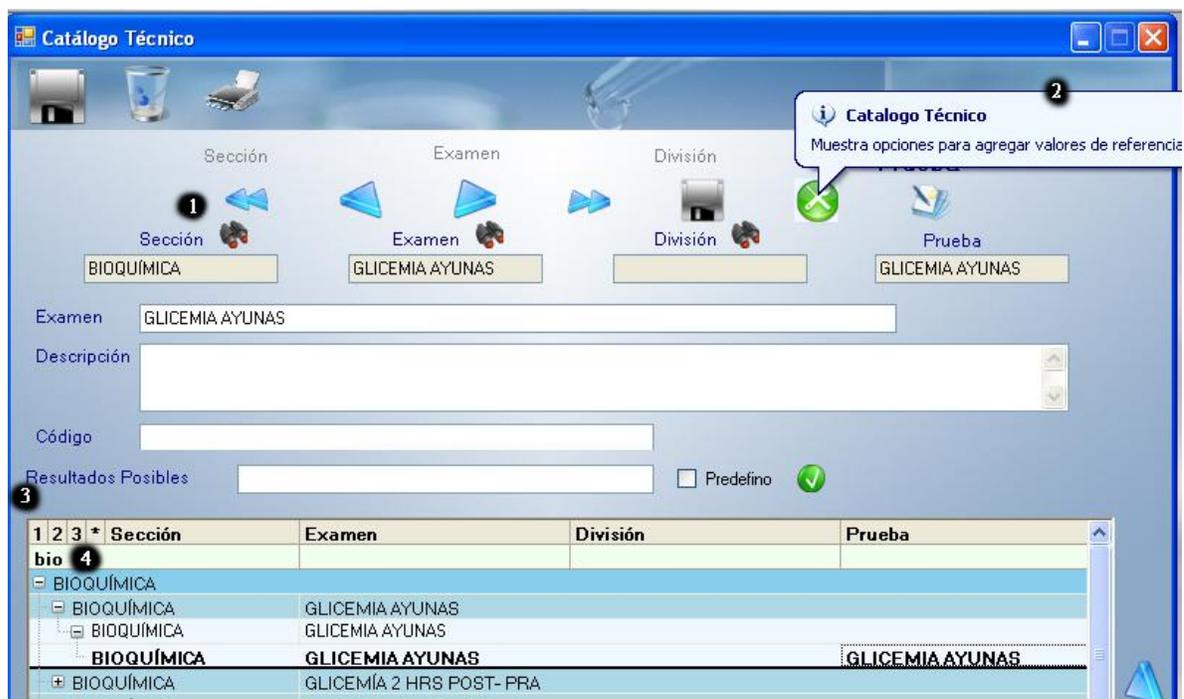
## 5.5 Interfaz con Usuario

Se desarrolló una interfaz simple, basada en iconos ilustrativos, tonalidades azules, texto informativo en botones (Tooltips) y se mantuvo un esquema constante con el fin de facilitar al usuario la capacidad de intuir funcionalidad en pantallas que no conozca (Figura 5).



**Figura 4: Puntos comunes de la interfaz con usuario de Labsys 5.0: 1) Barra de acciones. 2) Control de filtro autocompletable. 3) Botón que invoca al asistente de búsquedas. 4) Rejilla de datos.**

Se incluyeron filtros con la finalidad de discriminar información proveniente de consultas ya filtradas. De esta manera, cuando los criterios de búsqueda coinciden con muchos datos es posible establecer aun más criterios de filtración para evitar largas búsquedas manuales (Figura 6). Otro punto muy importante es la visualización jerarquizada aplicada en el catálogo técnico, la cual permite conceptualizar más fácilmente las relaciones de la información en el caso de las pruebas (Figura 6).



**Figura 5: Mantenimiento de Catálogo Técnico en Labsys 5.0.** 1) Barra para inserción de datos en el catálogo. 2)Tooltip. 3) Botones de acceso rápido para navegar a través de jerarquías. 4) Filtro de búsqueda.

## 5.6 Mantenimientos y procesos

Se implementó la lógica de acceso a datos, validación, análisis y flujo de trabajo. Primeramente, los mantenimientos contienen toda la lógica para la administración de la información necesaria en el trabajo normal del sistema. Los procesos fueron compuestos de cuatro áreas principales:

- Expedientes.
- Solicitudes.
- Resultados.
- Validación.

El proceso de validación no se implementó en la lógica, sin embargo se diseñó en la base de datos con el fin de crearlo en una versión posterior del sistema. El trabajo implicó el desarrollo de unidades de código en C# capaces de comunicarse con el motor de bases de datos (Figura 7).

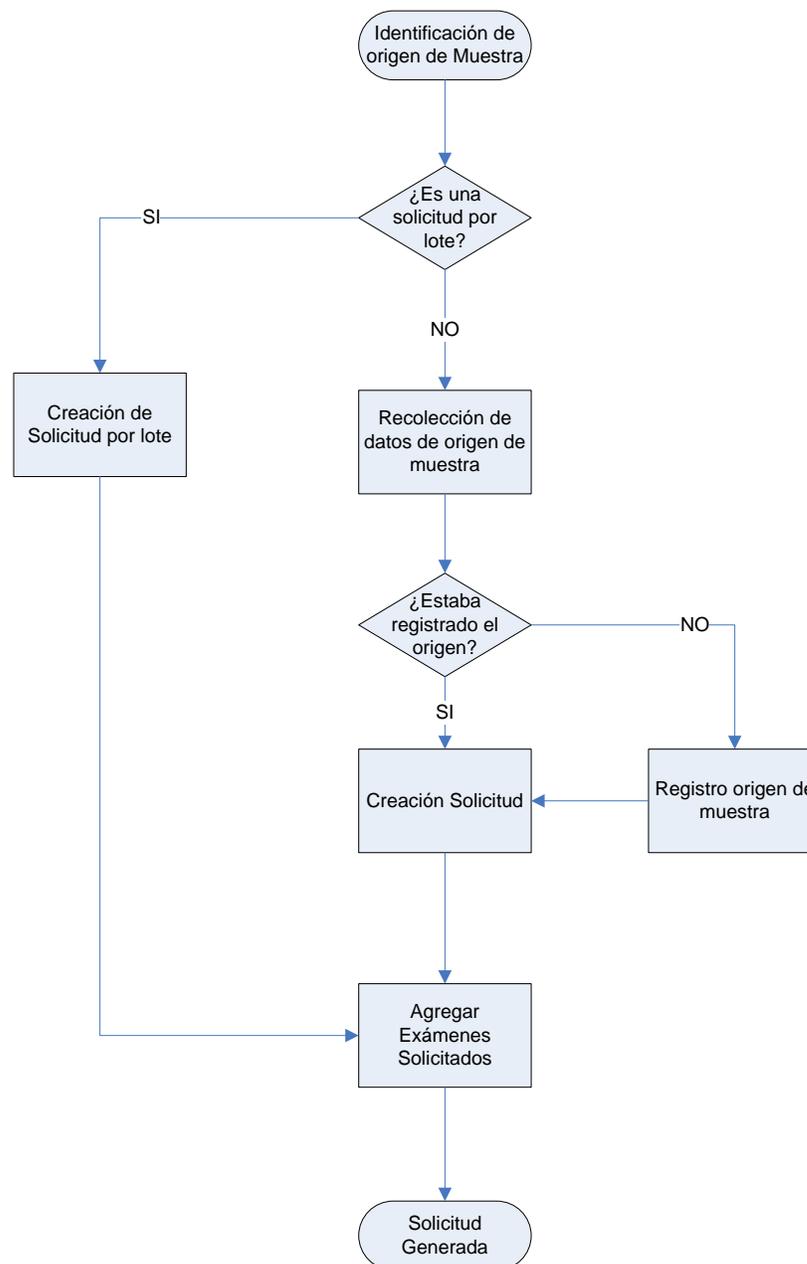
```

DateTime tiempo = DateTime.Parse(DateTime.Now.ToShortDateString());
ObtenerFecha(ref tiempo);
SqlCommand cmdSalvar = new SqlCommand();
cmdSalvar.CommandText = "spl_ModificarResultadoSol";
cmdSalvar.CommandType = CommandType.StoredProcedure;
cmdSalvar.Connection = ut.AbrirConn();
int f;
for (int i = 0; i < (modificados.Count); i++)
{
    //reemplazaResultadosAbrev(i);
    //modificados
    f = modificados[i];
    cmdSalvar.Parameters.Add(new SqlParameter("@Valor_CodFechaSol", SqlDbType.Tiempo));
    cmdSalvar.Parameters.Add(new SqlParameter("@Valor_ConsecSol", SqlDbType.Int));
    cmdSalvar.Parameters.Add(new SqlParameter("@Valor_Seccion", SqlDbType.VarChar));
    cmdSalvar.Parameters.Add(new SqlParameter("@Valor_Examen", SqlDbType.Int));
    (local variable) SqlCommand cmdSalvar (new SqlParameter("@Valor_Division", SqlDbType.VarChar));
    cmdSalvar.Parameters.Add(new SqlParameter("@Valor_Prueba", SqlDbType.Int));
    cmdSalvar.Parameters.Add(new SqlParameter("@Valor_Muestra", SqlDbType.VarChar));
    cmdSalvar.Parameters.Add(new SqlParameter("@Valor_Repeticion", SqlDbType.Int));
    cmdSalvar.Parameters.Add(new SqlParameter("@Valor_Expediente", SqlDbType.VarChar));
    cmdSalvar.Parameters.Add(new SqlParameter("@Valor_Resultado", SqlDbType.VarChar));
    cmdSalvar.Parameters.Add(new SqlParameter("@Valor_FechaRes", SqlDbType.DateTime));
    //MessageBox.Show(tiempo.Date.ToShortDateString());
    cmdSalvar.Parameters.Add(new SqlParameter("@Valor_HoraRes", SqlDbType.Int));
    //MessageBox.Show(tiempo.ToShortTimeString());
    cmdSalvar.Parameters.Add(new SqlParameter("@flagValor", SqlDbType.Bit));
    cmdSalvar.Parameters.Add(new SqlParameter("@flaglineal", SqlDbType.Bit));
    if (Txt_Microb.Text == "")
    {
        cmdSalvar.Parameters.Add(new SqlParameter("@Valor_Microbiologo", SqlDbType.VarChar));
    }
    else
    {
        cmdSalvar.Parameters.Add(new SqlParameter("@Valor_Microbiologo", SqlDbType.VarChar));
    }
    try
    {
        cmdSalvar.ExecuteNonQuery();
    }
}

```

Listado 2: Ejemplo de código C# para el manejo de datos

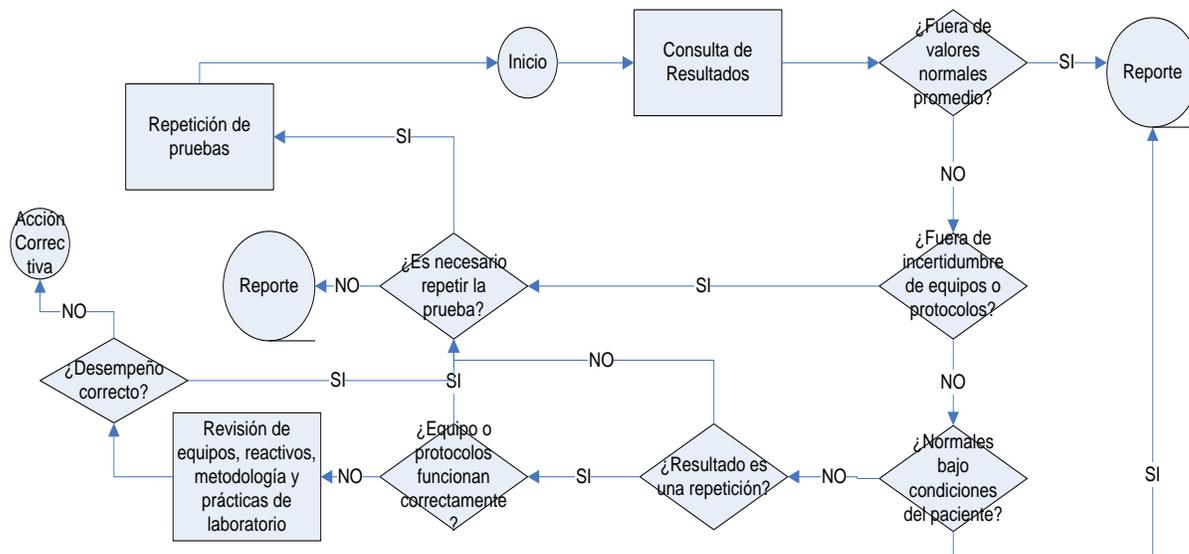
El proceso creación de solicitudes individuales de pruebas comienza con la identificación del origen de muestra, se define el tipo de solicitud: normal o por lotes (Figura 8). Las solicitudes por lotes no requieren asociar la prueba a un expediente, se omite con el fin de ganar tiempo en solicitudes donde es necesario comenzar el análisis de laboratorio inmediatamente y la información del origen puede ingresarse luego. Cuando es una solicitud individual normal se asocia la muestra a un expediente (datos de origen de muestra), si no existe un expediente se crea uno, se adhieren los exámenes y finalmente se almacenan los datos (Figura 8). El proceso de creación de solicitud para expedientes con muestras es muy similar, la única diferencia es que el flujo completo descrito en las solicitudes individuales se repite en cada muestra, cuyas solicitudes pueden ser iguales o diferentes en cuanto a los exámenes seleccionados.



**Figura 6: Flujo de trabajo para la creación de solicitudes**

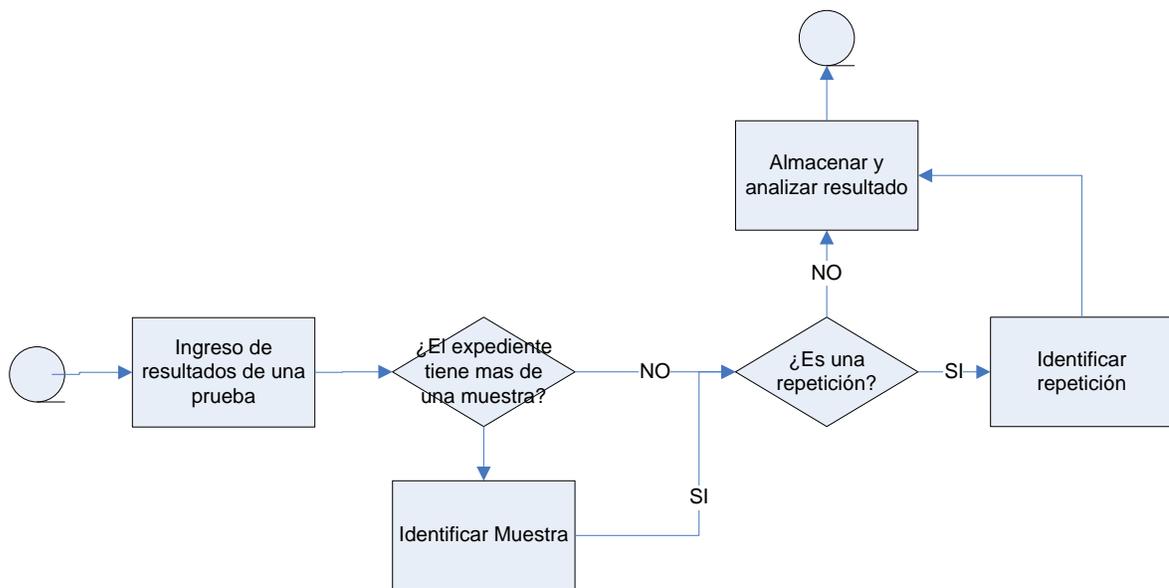
Diagramando el proceso realizado manualmente en el ingreso de resultados se comienza con la obtención de los datos, los cuales son evaluados determinando si están fuera de los ámbitos normales o de linealidad (empleando literatura), el profesional determina si es necesario repetir la prueba o no y si los resultados deben reportarse con los valores obtenidos aunque difieran de los ámbitos de normalidad o

linealidad, lo cual puede implicar la revisión de protocolos, equipos, reactivos o prácticas de laboratorio (Figura 9)



**Figura 7: Análisis de información en el Laboratorio**

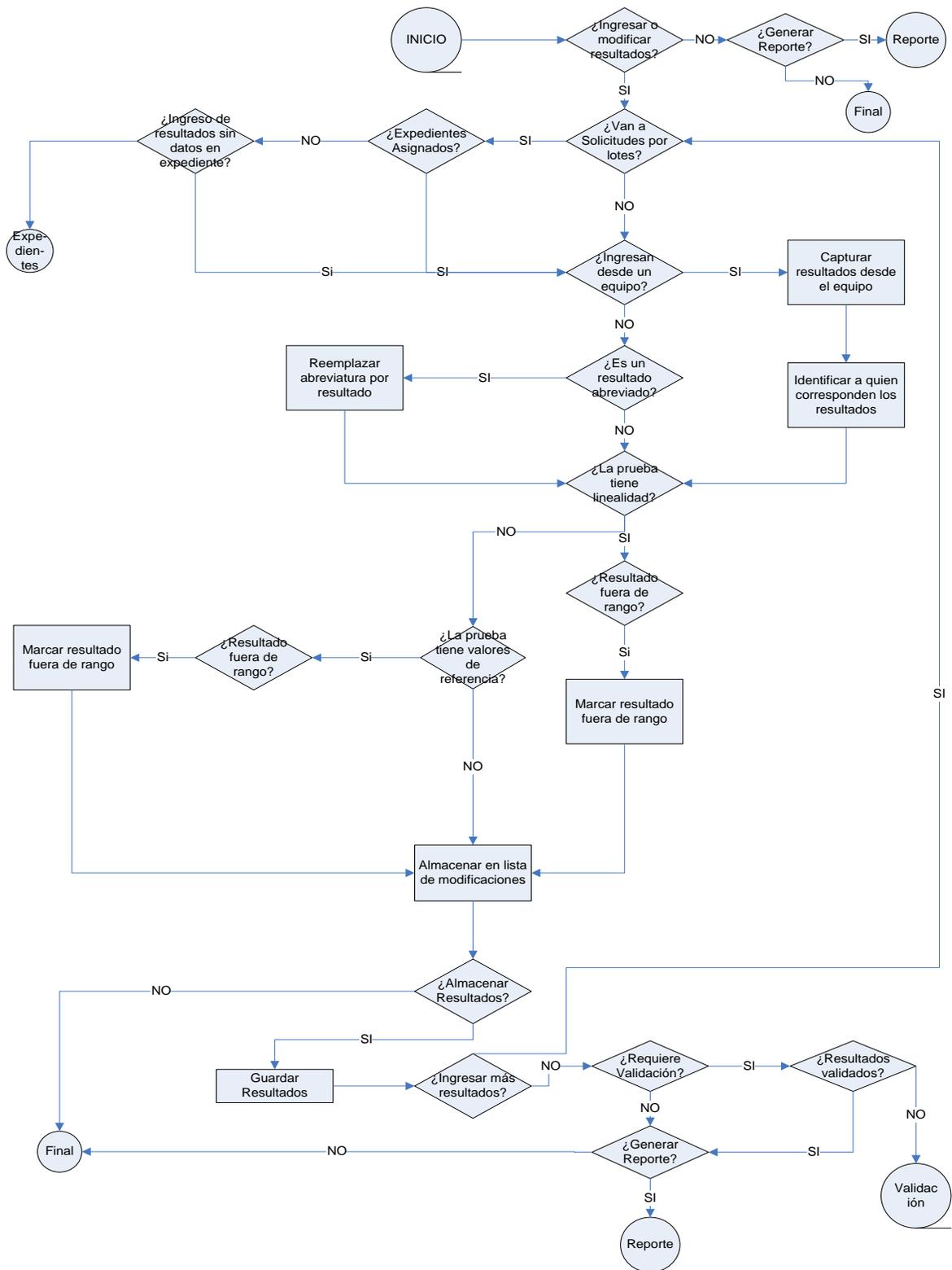
La forma de manejar el ingreso de resultados en el software Labsys comienza por determinar la cantidad de muestras contenidas en el expediente (este proceso lo realiza el programa), si es un expediente individual se pasa a la siguiente fase, de lo contrario la siguiente fase se repite en cada muestra del expediente (Figura 10). Si el resultado es una repetición se identifica cada una y la siguiente fase se realiza en cada repetición de la muestra, de esta manera la mínima unidad referida a resultados puede ahora ser procesada y al descartar los tipos de solicitudes se emplea el código optimizado para cada caso.



**Figura 8: Evaluación de expedientes con muestras**

El almacenamiento y análisis de un resultado implica procesar los datos obtenidos de cada prueba en cada repetición de cada muestra de cada expediente. Al comenzar este proceso el sistema identifica si los resultados pueden o no ser modificados, de no poder serlo permite generar un reporte con lo cual se finaliza un sesión de consulta sobre esos resultados (Figura 11). De ser posible el ingreso o modificación de resultados, se revisa el estado de la solicitud, si es una solicitud por lotes se determina si tiene un expediente asignado, de no ser así el usuario tiene la oportunidad de asignarle uno lo cual lo lleva a la pantalla de expediente, si se desea continuar el proceso con un expediente anónimo no se evaluarán los ámbitos de normalidad y linealidad de manera automática. Si los datos ingresan desde un equipo

automatizado se captura la información de este y se identifica a qué muestra y repetición pertenece (Figura 11). Si los resultados son ingresados manualmente se analiza el texto escrito contra la tabla de abreviaturas que el usuario puede definir en el sistema, de corresponder a una el sistema realiza el reemplazo de la abreviatura por el valor asociado con el fin de ahorrar tiempo al digitar y homogenizar los resultados reportados (Figura 11). En la siguiente fase se consulta si existen valores de linealidad en la prueba evaluada, de existir se obtienen los valores correspondientes a las características descritas en el expediente y si el resultado resulta estar fuera del ámbito se marca como tal y se almacena temporalmente. Si el resultado no está fuera del valor de linealidad o no existen valores de linealidad se buscan valores de normalidad que se ajusten a los descritos para las características del expediente. Si existen los valores normales (de referencia) se evalúa el resultado con respecto a estos, si los sobrepasa son marcados y almacenados temporalmente. A continuación el usuario puede realizar cambios si lo cree conveniente y de considerar válidos los resultados son almacenados definitivamente en la base de datos. Antes de poder generar un reporte de resultados el sistema revisa si pueden ser imprimidos sin validación, de estar activada la validación será imposible reportar los resultados de la solicitud hasta validar los datos (Figura 11).



**Figura 9: Evaluación de Resultados**

La determinación de si un valor se encuentra fuera de un ámbito de linealidad o normalidad debe considerar la existencia de dos tipos de resultados que deben considerarse por separado: los resultados cualitativos y los cuantitativos. Los resultados cualitativos hacen referencia a valores producto de la opinión de los expertos, pueden ser un texto amplio incluyendo opinión profesional, bibliografía o magnitudes. Los resultados cualitativos pueden ser desde un simple color hasta las razones tras un diagnóstico originado en los datos de un conjunto de pruebas. Los resultados cuantitativos pueden ser obtenidos desde equipos o ingresados manualmente, son magnitudes expresadas mediante números y existen intervalos de validez para los mismos, por lo cual es posible comparar los resultados cuantitativos con lo reportado en la literatura y así determinar la validez del resultado. Es común obtener resultados que indican padecimientos en el paciente, sin embargo, a veces esos resultados tienen una magnitud fuera de toda proporción concordante con esa prueba.

El manejo de valores cualitativos implica que el sistema debe reconocer cuando existe un valor cuantitativo y cuando no para decidir si debe o no aplicar la evaluación de límites. Una vez que los resultados se encuentran listos para ser almacenados es necesario determinar si requieren la validación de un experto, este proceso diverge de acuerdo al volumen de procesamiento de muestras en un laboratorio. En alto volumen existe un experto que realiza una exhaustiva valoración de los datos y la realiza en conjunto con el sistema en otra pantalla pendiente de elaboración para versiones avanzadas del sistema. En el caso de los bajos volúmenes de procesamiento de muestras, la valoración se realiza directamente sobre la entrada de resultados.

## 5.7 Ensamblado de la aplicación

Los diferentes bloques de código fueron ensamblados, completando la aplicación y de esta forma es posible comenzar a modelar la información de un laboratorio mediante las pantallas de mantenimientos para luego iniciar el proceso de captura de datos, análisis y reportes.

A nivel de procesos se cuenta con las pantallas de Expedientes Individuales (un solo expediente y una sola muestra con n repeticiones) y Expedientes con Muestras (un expediente, n muestras y n repeticiones por muestra). La pantalla de Expedientes Individuales cuenta con campos donde se ingresa nombre, apellidos, genero, categoría y otros datos fijos, además posee un área donde se ingresa la información de las características demográficas asociadas a la categoría empleada (Figura 12).



The screenshot shows a software window titled 'Expedientes'. It contains two main sections: 'Información General' and 'Características'. The 'Información General' section has several input fields: 'Nombre' (LEONARDO), 'Apellidos' (MARIN), 'Identificación' (empty), 'Categoría' (EXPEDIENTE MEDICO), 'Género' (MASCULINO), 'Nivel1' (NIVEL PRUEBA), 'Nivel2' (empty), and 'Nivel3' (empty). Below these is an 'Observaciones' text area. The 'Características' section contains a table with two columns: 'Característica' and 'Resultado'. The table has four rows: 'EDAD', 'HIPERTENSION', 'EDAD', and 'CARDIACOS'. The 'EDAD' row is highlighted in blue.

Característica	Resultado
EDAD	
HIPERTENSION	
EDAD	
CARDIACOS	

Figura 10: Pantalla de Manejo de Expedientes Individuales

La pantalla de manejo de Expedientes con Muestras cuenta con un campo de descripción general que asigna un identificador al expediente (puede emplearse

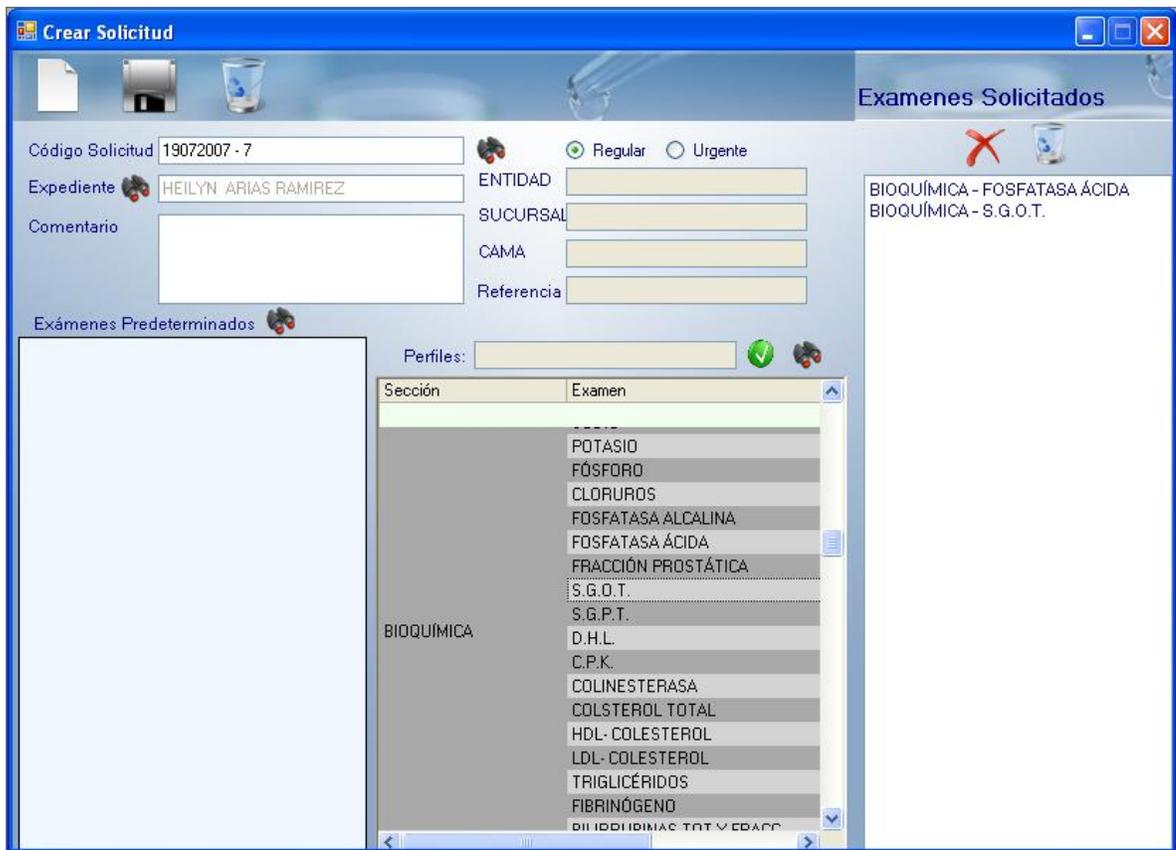
como el nombre si se aplica a una persona), posee también un área donde se inserta un identificador y comentario en cada muestra, una región de despliegue de características demográficas y otra donde se visualizan y seleccionan las muestras contenidas en el expediente (Figura 13).

The screenshot shows a software window titled "Expediente Muestra". At the top, there is a toolbar with icons for file operations and a red close button. Below the toolbar, the window is divided into several sections:

- Información General Muestra:** This section contains several input fields: "Descripción", "Observaciones" (with a scrollable area), "Categoría", and "Género". To the right of these fields are three "Nivel" (Level 1, 2, 3) fields, each accompanied by a small icon of a cluster of grapes.
- Descripción de Muestras:** This section contains two input fields: "Código" and "Detalle" (with a scrollable area).
- Características de Muestras:** This section contains a large, empty rectangular area for displaying or selecting sample characteristics.
- Muestras:** This section contains a table with two columns: "Código" and "Descripción". The table is currently empty.

**Figura 11: Pantalla de Manejo de Expedientes con Muestras**

En el proceso de solicitudes se creó una interfaz con el usuario capaz de permitir el acceso a exámenes y pruebas de laboratorio de una manera sencilla y veloz para quien crea la solicitud. Las distintas necesidades de los posibles usuarios conllevaron a la creación de tres pantallas de solicitudes, la solicitud individual se emplea con los expedientes individuales, para realizar la petición de exámenes (Figura 14).



**Figura 12: Pantalla de Solicitud Individual**

La Solicitud por Lotes consiste en la generación de expedientes anónimos, que por el momento corresponden a expedientes individuales y se muestra con campos donde se indica la cantidad de solicitudes a generar, por lo demás la pantalla posee las mismas áreas de selección de exámenes (Figura 15). Finalmente, la Solicitud para Muestras es muy semejante a la pantalla de solicitud individual, la diferencia radica en un área donde es posible seleccionar la muestra a la cual será realizada una petición de exámenes (Figura 16). De esta manera es posible asignar a cada muestra diferentes exámenes.

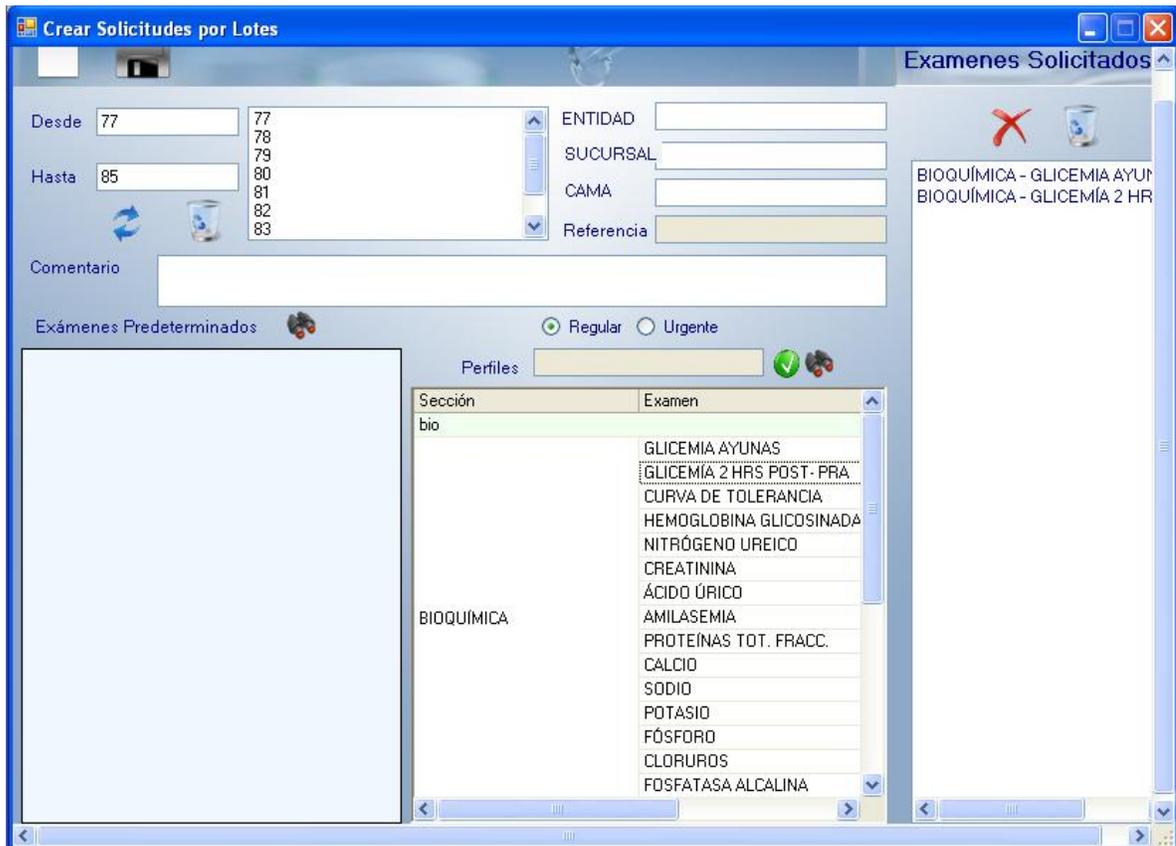
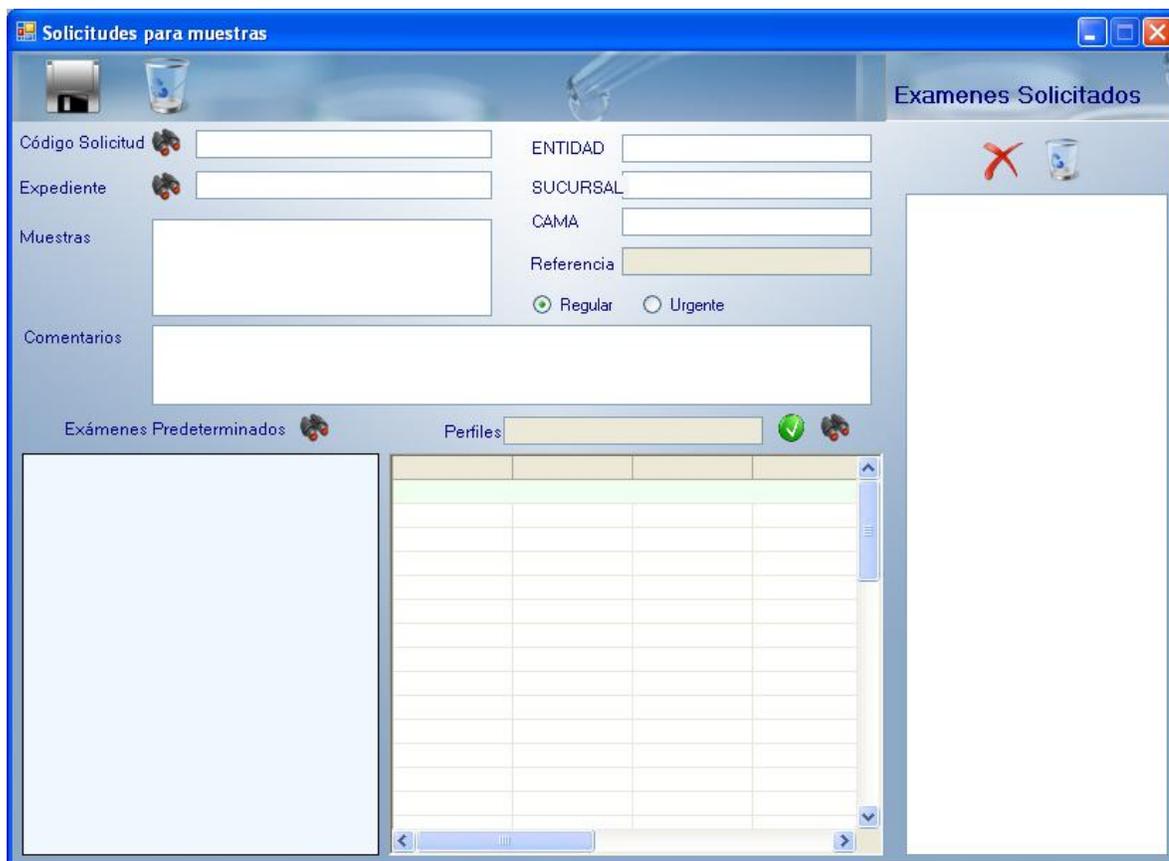


Figura 13: Pantalla de solicitudes por Lotes



**Figura 14: Pantalla de solicitudes con muestras**

En la inserción de resultados es posible crear repeticiones para las pruebas e ingresar los resultados de las mismas, se crearon dos pantallas: entrada de Resultados a Solicitud Individual y a Solicitud con Muestras. La entrada de resultados a solicitud Individual despliega información de la solicitud y el expediente, mediante una pantalla asistente coloca información sobre solicitudes anteriores o sobre el expediente. Esta pantalla realiza una validación automática de los resultados (alterable por el usuario) donde alerta sobre resultados salidos de la linealidad reportada con un color verde en el resultado o un color rojo si se trata de valores fuera del intervalo normal (Figura 17).

Entrada Resultados

Solicitud: 20122006-1 Expediente: CRISTINA GARCIA

Fecha: Responsable Médico:

Seccion	Examen	Division	Prueba	Resultado
ULTRASO	ULTRA SI		DIAGNÓSTICO:	
			IMPRESIÓN DIAGNÓSTICA:	
			ANÁLISIS DE LA ONDA ESPE	
	ULTRA SI		DIAGNÓSTICO:	
			IMPRESIÓN DIAGNÓSTICA:	
			SISTEMA VENOSO:	
	ULTRASO		DIAGNÓSTICO:	
			IMPRESIÓN DIAGNÓSTICA:	
			AORTA Y RETROPERITONEI	
	ULTRASO		BAZO:	
			ECDESTRUCTURA RENAL:	
			GLÁNDULAS SUPRARENAL	
ULTRASO		HIGADO:		
		IMPRESIÓN DIAGNÓSTICA:		
		PÁNCREAS:		
		PÉLVICO:		
		QUISTES:		
		RIÑÓN DERECHO:		
	RIÑÓN IZQUIERDO:			
	ÚTERO:			

Display Entrada Resultados

**Información de Resultados Anteriores**

Fecha	Consec	Prueba	Resultado

**Información Solicitudes Anteriores**

Código	Consec	Fecha
20122006	1	20/12/2006...

**Información de Repeticiones**

Prueba	Repeticion	Resultado

**Información del Paciente**

Categoría: CLÍNICA MÉDICA  
 Género: Femenino  
 Identificación:  
 Nivel 1  
 Nivel 2  
 Nivel 3

Figura 15: Pantalla para manejo de resultados de solicitudes individuales

La entrada de resultados en solicitudes con muestras es muy semejante a la de solicitudes individuales, la diferencia radica en un área donde es posible seleccionar la muestra a revisar (Figura 18), al seleccionar una de esas muestras la información de esa muestra es cargada en la pantalla y si se realiza algún cambio antes de cargar otra información, el programa alerta al usuario si no se ha almacenado la información.

**Figura 16: Pantalla para manejo de solicitudes con muestras**

Finalmente, es posible reportar la información generada a nivel de documentos mediante los distintos tipos de reportes: resultados de pruebas, hojas de trabajo, catálogo técnico, entre otros.

## **6.DISCUSION**

El software Labsys 5.0 se diseñó con una estructura de almacenamiento de datos capaz de brindar a los usuarios la posibilidad de definir la información a almacenar en lugar de restringir la cantidad de datos a campos definidos por los creadores del sistema y el diseño de la base de datos se desarrolló con el fin de proveer información organizada que facilite análisis complejos.

### **6.1 Diseño del sistema**

La información de sistemas consolidados es vital para el desarrollo de nuevos paquetes capaces de realizar análisis complejos sobre datos históricos (Fine *et al*, 2007). La existencia de estos sistemas implica la comprensión de sus estructura para enriquecer el desarrollo de nuevos sistemas y brindar soporte a versiones anteriores (Microsoft Corporation 2002).

El sistema Labsys se construyó basado en una arquitectura de dos capas, hospedando el motor de base de datos en un ordenador accedido por equipos clientes albergando la interfaz con el usuario. Microsoft Corporation (2002) indica que la arquitectura de dos capas se contraponen a una práctica extendida entre muchos desarrolladores de software que emplean una arquitectura basada en tres o más capas fundamentadas en servicios. Jakobovits, Rosse y Brinkley (2002) desarrollaron un conjunto de herramientas de software destinado a la creación de aplicaciones web biomédicas (denominado WIRM) empleando una arquitectura compuesta por una capa de base de datos, otra de acceso a recursos, una de servicios donde se exponen las funcionalidades y una de desarrollo donde se implementa el manejo sobre las capas anteriores para exponerse en otra capa hacia nuevas capas, sistemas o una interfaz con el usuario. He *et al* (2007) crearon un sistema de automatización de alto rendimiento en búsquedas de BLAST y para el manejo de resultados de minería de datos (denominado PLAN) implementando una capa de almacenamiento de datos, capa de acceso a información, capa de procesamiento de información y capa de presentación de datos. En los proyectos PLAN y WIRM se

emplearon arquitecturas de tres capas o más pues son muchas sus ventajas, por ejemplo, según Microsoft Corporation (2002) es posible centralizar en el área de servidores la mayoría del trabajo de procesamiento de datos, mediante el uso de capas de presentación basadas en web la aplicación puede accederse a través de cualquier navegador compatible con las tecnologías empleadas, lo cual implica que cualquier máquina capaz de manejar un navegador compatible puede utilizarse sin importar el sistema operativo o capacidad de procesamiento, las aplicaciones diseñadas correctamente son capaces de manejar gran cantidad de información, son más escalables y es posible llamar más fácilmente sus componentes desde otros sistemas. He *et al* (2007) señalan que es factible alojar las capas en diferentes servidores, por ejemplo, si el motor de base de datos soporta particiones de tablas la información puede almacenarse en diferentes servidores. Las capas de accesos a datos pueden residir en otro servidor, al igual que capas de procesamiento de datos o presentación de los mismos, esto permite añadir poder de procesamiento a aplicaciones accedidas por cantidades importantes de usuarios o que manejan volúmenes de información importantes (He *et al* 2007).

Según He *et al* (2007), Jakobovits *et al* (2002) y Microsoft Corporation (2002) el sistema Labsys es un candidato en la implementación de una arquitectura de tres capas o más, sin embargo, en el presente trabajo se analizaron las versiones anteriores de Labsys y la infraestructura empleada en los lugares donde el sistema se ha aplicado determinando que los laboratorios de poco procesamiento de muestras no cuentan con el equipo necesario aplicaciones con arquitecturas complejas destinadas a grandes volúmenes de datos. Li *et al* (2006) diseñaron un sistema sencillo aplicable a laboratorios con poco volumen de trabajo, ellos señalan que el diseño de la mayoría de los sistemas de manejo de información en laboratorios se orienta a complejos ambientes de servicios y componentes contrapuestos contra la realidad económica y tecnológica de laboratorios pequeños donde la solución necesaria es algo de poco costo y altamente personalizable como el caso del sistema Labsys, diseñado con el fin de que el usuario defina gran parte de la estructura de la información a almacenar. En la mayoría de los laboratorios

pequeños donde existe una versión anterior del Labsys un ordenador casero realiza las funciones del servidor y aloja la base de datos para que se conecten de una a ocho máquinas. El software Labsys desarrollado en el presente trabajo es una opción destinada a laboratorios pequeños-medianos con bajo costo y la funcionalidad necesaria en las tareas de almacenamiento y análisis de información, se limitó la arquitectura a una estructura de dos capas con el fin de no desarrollar un sistema cuyo costo sea muy superior al beneficio obtenido al colocarlo en clientes pequeños brindando un modelo que se adapte a las necesidades de administración de información. No obstante, considerando escalabilidad en una versión destinada a altos volúmenes de muestras se implementaron clases que pueden transformarse a una versión de arquitectura más compleja, basada en varias capas y aprovechando la estructura bajo la cual se desarrollo Labsys 5.0, las unidades de código divididas en gestores de manejo de información proveen una base para la separación de capas en el Labsys 5.0. De esta manera se adoptó un modelo que permite la expansión a un sistema más escalable empezando primero por brindar soluciones de tecnologías de información a laboratorios pequeños.

El desarrollo basado en formularios Windows se escogió debido a la flexibilidad en el diseño de la interfaz con usuarios y por la facilidad de acoplamiento de usuarios de las versiones anteriores. Sin embargo, esta interfaz es el primer paso en la migración a formularios web, para universalizar aun más al sistema y hacerlo dependiente de tecnologías específicas únicamente en las capas alojadas en un servidor. Posteriormente, el proceso de escalamiento implica desglosar al Labsys en un conjunto de componentes intercomunicados, capaces de ofrecer sus servicios vía intranet o web. Este modelo es conocido como SOA (Service-oriented architecture), es un modelo ideal para ofrecer servicios que manejen una gran concurrencia de usuarios e información siendo capaz de reutilizar sus componentes en sistemas implementados inclusive con tecnologías muy diferentes si se emplean métodos de comunicación estándar (vía servicios web u otras tecnologías) (Microsoft, 2002). Al acoplar a los laboratorios la versión 5.0 de Labsys es más fácil transferir datos a una

versión basada en una arquitectura más compleja si se compara con una transferencia desde la versión Labsys 3.0 por ejemplo.

## **6.2 Diseño de base de datos relacional**

Al ser el Labsys un sistema que involucra un manejo constante de datos y operaciones de reporte sobre los mismos se seleccionó un modelo relacional OLTP en busca de un mayor rendimiento en las tareas normales del software. Dewson (2005) define un sistema OLTP (OnLine Transaction Processing) como un motor de bases de datos que desarrolla un procesamiento de transacciones de datos a través de redes locales o externas, donde el número de transacciones es muy alto y requiere almacenar información constantemente; el modelo OLAP (On-Line Analytical Processing) representa una buena opción en el análisis de estructuras multidimensionales que impliquen grandes cantidades de datos debido a su capacidad de consultar información de manera muy eficaz, sin embargo, las operaciones de inserción, modificación y eliminación son mucho más eficientes en el sistema OLTP (Dewson 2005). Debido a la constante actividad en la base de datos escribiendo y leyendo información se seleccionó OLTP pues como Dewson (2005) señala, existe una deficiencia en la velocidad de consulta de información en el modelo OLTP con respecto al OLAP sin embargo la diferencia no va a ser notada por los usuarios de un sistema de almacenamiento de información y procesamiento como el Labsys versión 5.0 debido a que las operaciones realizadas en el software no implican análisis demasiado complejas sobre grandes cantidades de datos. Hancock y Toren (2006) señalan que si se desea implementar una solución OLAP en el estudio detallado de cantidades masivas de información ingresadas de manera intensiva, es necesario realizar un sistema OLTP destinado a la operación y colecta de datos, el cual proporcionaría la información a un sistema de apoyo basado en tecnologías OLAP, extrayendo la información del sistema OLTP y transformándola de acuerdo al análisis implementado. En este trabajo se diseñó Labsys como un sistema de ingreso y lectura intensiva con la intención de brindar una plataforma de datos a

partir de la cual sea posible desarrollar un sistema OLAP siguiendo el esquema recomendado por Hancock y Toren (2006).

En el sistema Labsys se seleccionó el modelo relacional de bases de datos pues, como señala Dewson (2005), se logra jerarquizar los datos de una manera más semejante a la realidad, los problemas pueden ser modelados en objetos que presentan relaciones de dependencia reflejadas en las operaciones de datos. Un objeto de base de datos conocido como Sección puede asociarse a muchos exámenes, pero un examen sólo puede asociarse a una sección. Este comportamiento se encuentra claramente documentado por Dewson (2005), las delimitaciones originadas en la asociación de objetos de manera relacional permiten restringir multiplicaciones erradas de la información por procesos anormales gracias a las referencias de la herencia entre tablas. Esta propiedad de anidamiento padre-hijo contribuye a una menor duplicación de la información, lo cual posibilita niveles de integridad referencial bastante altos, en el caso del sistema Labsys, cuando se cambia un dato sobre una sección, los exámenes hijos automáticamente reflejarán esos cambios si se consulta su sección padre pues dentro de la entidad examen existe una referencia señalando hacia la sección. Otra ventaja de la estructura relacional de la base de datos implementada en el sistema Labsys es indicada por Dewson (2005): sincroniza la actualización y eliminación de registros, evitando la existencia de datos no actualizados debido a errores fatales donde el procesamiento de una operación es interrumpida, por lo cual al consultar o analizar información se evitan colapsos al encontrar referencias inexistentes como ocurre más comúnmente en modelos no relaciones (las referencias en estos modelos deben emularse, aunque es muy difícil alcanzar el grado de restricciones del modelo relacional). La consulta y análisis se transforman en una navegación a través de entidades relacionadas capturando información desde diferentes puntos (Dewson 2005).

El diseño de base de datos del sistema Labsys se orientó a una estructura de personalizable por el usuario, con un origen de muestra asociado a características provenientes de una tabla categorizada mediante otra tabla (Anexo I). La

caracterización de muestras en otros sistemas varía mucho, Titulaer *et al* (2006) desarrollaron un software orientado a la administración y análisis de datos provenientes de mediciones de espectros de masa, en una tabla con cuatro campos se almacena las características de la muestra (Anexo II), lo cual implica una estructura simple con menor cantidad de consultas y menor complejidad de desarrollo con respecto a una estructura más personalizable, este tipo de aplicaciones suelen consumir menos recursos al ser muy específicas y se emplean en problemas que requieren análisis cuidadosos sobre ciertos datos. No obstante, Viksna *et al* (2007) en el sistema PLAN atacan el problema de la caracterización de muestras estableciendo una tabla denominada persona, capaz de tener n muestras asociadas y a su vez una muestra puede tener n alícuotas asociadas (Anexo III) y las características asociadas a cada muestra se almacenan puntualmente en campos fijos en la tabla de Personas y en otras tablas separadas, como la condición de almacenamiento o transporte (Anexo III). Titulaer *et al* (2006) y Viksna *et al* (2007) implementan en sus trabajos modelos grados de especificación orientados a solucionar problemas determinados considerando una alta velocidad de procesamiento de información, sin embargo en el presente trabajo, el Labsys 5.0 fue diseñado con una estructura genérica pues debe ser capaz de acoplarse a las diferentes políticas de registro de datos en los laboratorios, el producto final es un software comerciable cuya venta debe realizarse con el mínimo número de cambios aspirando a la nulidad completa de los mismos. La mayor complejidad en la constitución de las consultas del Labsys 5.0 repercute en mayores tiempos de búsqueda de información, pero este costo es aceptable en el Labsys 5.0 por la heterogeneidad de las necesidades de los usuarios.

### 6.3 Procedimientos almacenados

En el Labsys 5.0 los procedimientos almacenados son unidades de código alojados en el motor de base de datos y se convierten en una subcapa de acceso a datos a través de la cual es posible manejar operaciones sobre información.

Los procedimientos almacenados fueron diseñados en el sistema Labsys con el fin de que aunque la aplicación se encuentre en producción si algo falla a nivel de procedimientos almacenados puede bastar con alterar el código de los mismos sin variar el código C# de las capas de acceso a datos del cliente, si la consulta con código T-SQL estuviese alojada en la interfaz con el usuario, la corrección de la misma implicaría cambios en el ejecutable de Labsys 5 y por ello requería distribuir una nueva versión del mismo o de una librería de funciones. El caso se aplica igualmente en la optimización de operaciones, por ejemplo, si se desea mejorar la velocidad de una consulta basta con cambiar el código T-SQL, autores como Davidson *et al* (2006) recomiendan este modelo de acceso a datos basado en procedimientos almacenados debido a la facilidad de mantenimiento y optimización de operaciones. Davidson, Kline y Windisch (2006) señalan que los procedimientos almacenados se emplean mediante un plan de ejecución diseñado por el motor de base de datos gracias a la estructura de la consulta y de las bases de datos, así se ejecutan con mayor rapidez si se les compara con consultas dinámicas. Otro punto importante es que disminuyen las vulnerabilidades a fallos de seguridad como el SQL Injection (permite realizar operaciones sobre la base de datos desde interfaz de usuario con poca información de referencia), representan un nivel más de seguridad, es posible asignar permisos de ejecución separados, por ejemplo, un usuario de base de datos capaz de emplear solamente los procedimientos sin acceder a tablas u otros objetos, pueden incluirse dentro de esquemas de seguridad, los cuales son conjuntos de permisos asignables a objetos de la base de datos (Davidson *et al* 2006). Los procedimientos almacenados pueden ser reutilizados en otros sistemas que requieren manejar información del Labsys 5.0, si se emplea un software capaz de conectarse a la base de datos, es posible emplear los procedimientos

almacenados en lenguajes como c++ o java disminuyendo de esta forma los tiempos de desarrollo en un nuevo proyecto. Finalmente, los procedimientos almacenados son una práctica recomendada por los especialistas en Microsoft SQL Server 2005 (Microsoft Corporation 2002; Dewson *et al*, 2005).

Los beneficios de emplear procedimientos almacenados fueron el motivo para implementarlos, aunque el tiempo de desarrollo aumentó pues se requirió de una etapa de diseño y pruebas en estos bloques de código.

#### **6.4 Funcionamiento del software**

Labsys 5.0 implementa ADO.NET como base de acceso a SQL Server 2005, ADO .NET es un conjunto de librerías definidas en el framework de .NET y orientadas a la interacción con fuentes de datos locales y remotos.

En la implementación de la validación de datos, lógica de negocios y análisis del Labsys 5.0 se emplearon clases diseñadas en C# 2.0 basadas en clases de manejo de datos desarrolladas con un proveedor específico de SQL Server 2005, en estas últimas se encapsula la administración de datos en los mantenimientos y procesos. La validación, lógica de negocios y análisis fueron implementados con clases diseñadas en C# 2.0.

La definición de las etapas primordiales en el manejo de información no es un evento arbitrario, responde a necesidades básicas registradas por diversos autores en la creación de un software destinado a actividades de algún tipo en un laboratorio. Los pasos de recepción de muestras, registro de las mismas, asignación de pruebas, colecta de resultados, análisis y reporte de resultados son constantes en sistemas de laboratorio. Morisawa *et al* (2006) diseñaron un sistema de código abierto (“open source”) destinado al manejo de información de laboratorio en geles 2-D basados en electroforesis en el campo de los flujos de trabajo de proteómica, involucra una parte del software diseñada en la caracterización de muestras, ingreso de resultados y reporte de los mismos. Marzolf y Troisch (2006) desarrollaron el sistema SLIMarray (empleado en la captura de datos provenientes de microarrays), el software se

ejecuta registrando en una terminal la información del origen de la muestra, capturando los resultados de un equipo, organizándolos, almacenándolos y preparándolos para análisis y reportes. Li *et al* (2006) indican en su sistema de administración u manejo de información de laboratorios que los sistemas de este tipo deben contemplar la recepción, registro, captura de resultados, análisis y reporte como procesos fundamentales. Labsys 5.0 contempla que los resultados deben presentarse de manera eficiente, filtrando los datos innecesarios y orientando los reportes a estructuras de fácil comprensión. Por ello, los resultados anormales en el software Labsys son identificados con banderas que pueden detener el proceso hasta confirmar la validez de ese resultado. La pantalla de validación será una herramienta de consulta orientada a determinar si esos intervalos anormales realmente están fuera de un valor esperado. La estructura se diseñó con filtros de secciones, exámenes, fechas y códigos de petición; de esta manera los resultados anormales son detectados más fácilmente.

Es necesario realizar una tarea de acompañamiento con el fin de mostrarle al usuario el uso del Labsys 5.0 y la mejor manera de modelar la información de un laboratorio a la estructura de Categorías, Características Demográficas, Género, Expedientes, Solicitudes, Secciones, Exámenes, Divisiones y Pruebas. Si no se estructuran adecuadamente los datos, el análisis sobre conjuntos podría ser muy difícil, es más sencillo procesar de acuerdo a la Sección un conjunto de datos de un período determinado que analizar secciones compuestas de un solo examen. Por ello es fundamental la categorización de la información con el fin de crear una abstracción que facilite análisis de resultados de acuerdo a la estructura de pruebas. Este proceso de guía implica la mediación entre el área de ingeniería en sistemas y los trabajadores de laboratorios, por lo cual la experiencia del profesional en biotecnología involucrado en grupos multidisciplinarios de desarrollo de software es un aporte valioso en la adopción en laboratorios de tecnologías de información.

## **6.5 Aprovechamiento de la Información**

El sistema Labsys permite alimentar una base de datos con información de resultados de pruebas asociados a características del origen, esta información crece con el paso del tiempo y constituye un historial de resultados.

Labsys 5.0 presenta una base de datos OLTP donde se representa adecuadamente la situación actual de la información, si se desean realizar operaciones con grandes cantidades de datos (como las relaciones de las características demográficas con respecto a resultados) es recomendable complementar la aplicación con un sistema OLAP, almacenando la información en estructuras multidimensionales que faciliten los análisis (Hancock y Toren 2006). Hancock y Toren (2006) señalan que muchas veces la información es almacenada en una base de datos con el único objetivo de cumplir a cabalidad con los procesos de una organización, esa información puede llegar a convertirse en un simple archivo digital donde se almacena y reporta la última información ingresada. Los datos de una organización pueden aprovecharse, el área denominada “Business Intelligence” aplica una serie de conceptos, métodos y tecnologías diseñadas con el fin de transformar grandes cantidades de datos almacenados en información útil empleando sistemas OLAP, creando herramientas para tomar decisiones basadas en conocimiento del negocio o proceso que emplea un software (Hancock y Toren 2006). Con un sistema de este tipo es posible aplicar soluciones de minería de datos, las cuales permiten analizar la información, entender que pasa realmente y cómo será el comportamiento futuro, se buscan patrones de datos no detectados por analistas humanos para realizar predicciones, es posible además pensar en la implementación de soluciones personalizadas trabajando junto a las soluciones clásicas de minería de datos. Las desventajas de soluciones basadas en “Business Intelligence” son los costos de creación y mantenimiento pues es necesario desarrollar un software nuevo que amplía la funcionalidad del anterior, además implicaría destinar recursos (servidores, infraestructura) con el fin de brindar alojamiento a la aplicación en una máquina donde al realizar una exhaustiva consulta no se altere la funcionalidad del sistema OLTP. Hancock y Toren (2006)

recomiendan que debido a los costos económicos, en una organización debe valorarse que la inversión en el software de “Business Intelligence” permita obtener información nueva que contribuya a alcanzar objetivos importantes.

En el desarrollo del Labsys 5.0 se incluyó una funcionalidad para asistir al usuario en el establecimiento de estándares en el área de reporte de resultados, la creación de Abreviaturas permite reemplazar un texto por otro de manera automática. De esta forma pueden crearse las abreviaturas que serán reemplazadas por un valor que el usuario indique. Con el fin de obtener una mayor capacidad de aumentar la automatización de una nomenclatura es recomendable incluir en futuras versiones de Labsys por cada prueba un conjunto de reglas de validación definidas por el usuario y a su vez la posibilidad de incorporar conjuntos de reglas en la evaluación de resultados normales en pruebas cualitativas.

Viksna *et al* (2007) enfatizan que los usuarios deberían implementar una nomenclatura en su organización con el fin de facilitar análisis de datos de su organización o de información proveniente de diferentes bases de datos, lo anterior con el fin de evitar diferencias en el almacenamiento de un mismo dato, pues al analizar o reportar los datos pueden surgir incoherencias al examinar resultados de una o varias pruebas. Por ejemplo, en el caso del Labsys si se desea reportar como Negativo la ausencia de *E. coli* en un cultivo bacteriano el resultado podría ser reportado de la siguiente manera: NEGATIVO, -, NEG, entre otros, si no hay un proceso de crear una nomenclatura, al realizar un análisis de minería de datos, el sistema podría en el caso anterior, encontrar tres tipos de resultados en lugar de uno, incurriendo en un error muy significativo.

Viksna *et al* (2007) señalan la dificultad de estandarizar la manera de nombrar pruebas y reportar resultados, especialmente entre las compañías desarrolladoras de software, este problema está presente en cualquier sistema donde el usuario es capaz de ingresar o modificar los resultados de una prueba, sin embargo, existen casos especiales donde se ha logrado estandarizar el reporte de resultados, como el caso de ciertos tipos de análisis con microarreglos donde los resultados de diferentes

laboratorios pueden analizarse de la misma manera al contar con un alto grado de homogenización.

## **7.CONCLUSIONES Y RECOMENDACIONES**

---

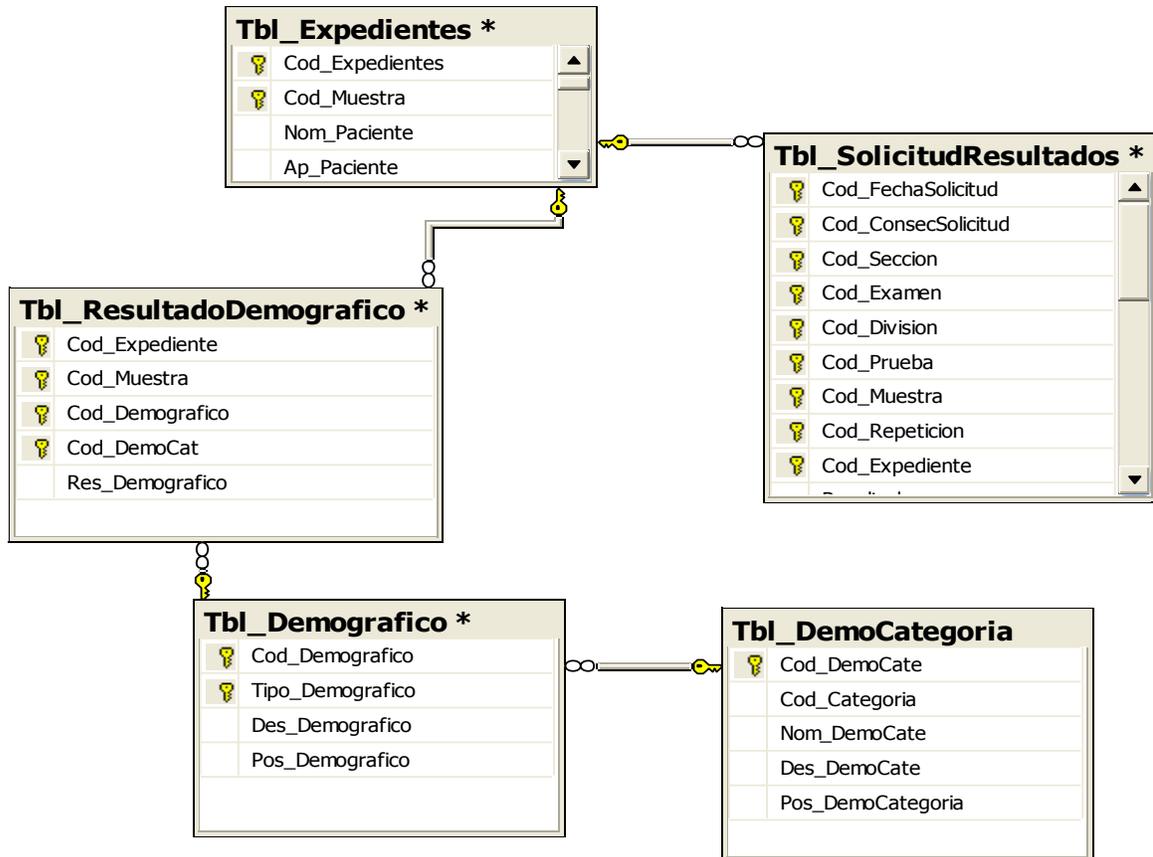
### **7.1 Conclusiones**

- Los modelos de datos no especializados pueden facilitar el almacenamiento de información si son diseñados con la posibilidad de que el usuario defina la estructura de datos a registrar.
- Al diseñar o adquirir un software aplicado a laboratorios es necesario definir el grado de especialización del mismo con el fin de poder balancear el rendimiento del programa con las necesidades del usuario y la implementación en diferentes organizaciones.
- El almacenamiento de información de muestras permite consolidar una fuente de datos para análisis complejos sobre la relación del ambiente en los resultados de pruebas de laboratorio realizados a individuos de una población.
- Los sistemas de software pueden brindar herramientas que faciliten el análisis de complejas estructuras de datos y permitan descubrir patrones difíciles de detectar por métodos manuales.
- El establecimiento de nomenclaturas de nombramiento de pruebas y reporte de resultados es un paso fundamental en la homogenización que será usada en procesos de análisis.

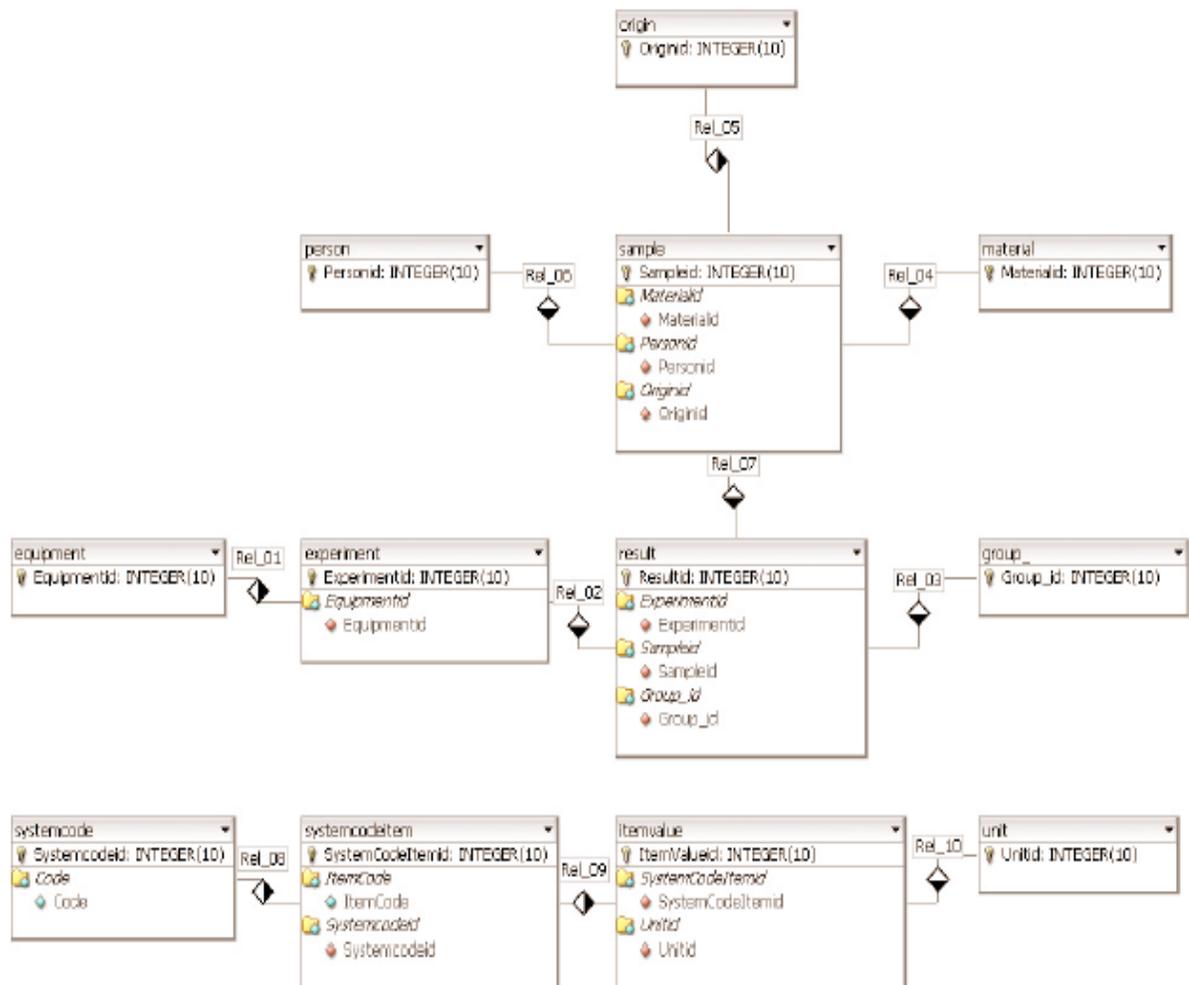
## **7.2 Recomendaciones**

- Analizar el flujo de datos dentro de un laboratorio con el fin de detectar si un determinado software realmente traerá beneficios al ser implementado en esa organización.
- Dedicar buena parte del tiempo de un proyecto de software aplicado a laboratorios a crear un diseño que además de suplir las necesidades actuales, permita establecer un punto de partida en la creación de complejas herramientas de almacenamiento y análisis de información.
- Implementar un modelo de reglas definidas por el usuario para la validación de resultados con el fin de aumentar la estandarización de reporte de datos y análisis de los mismos.

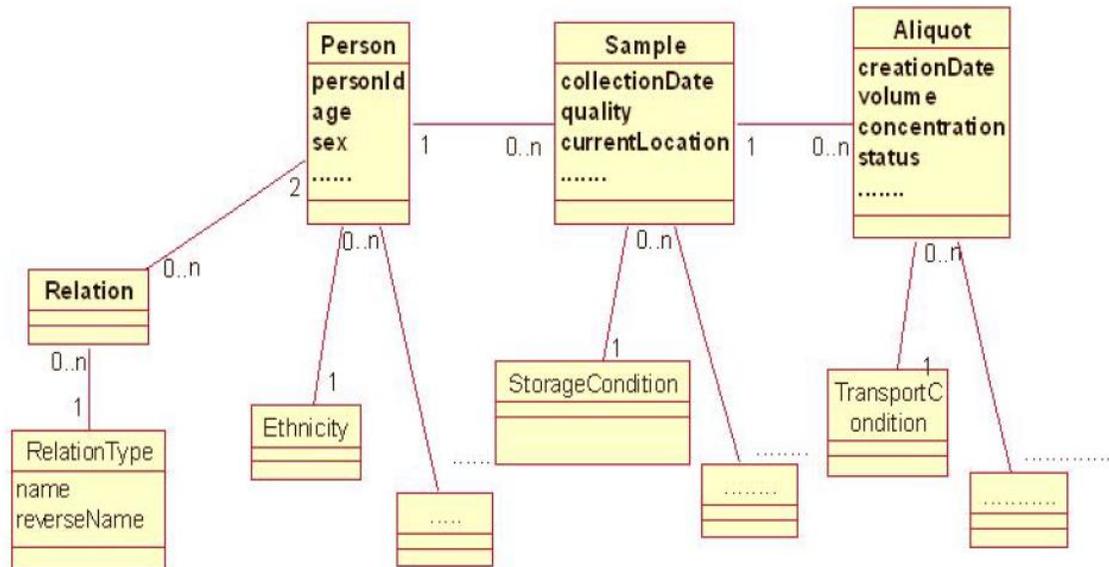
## Anexo I: Vista general del diseño de base de datos aplicado en caracterización de orígenes de muestra y solicitudes de pruebas del software Labsys



## Anexo II: Diseño de base de datos en el registro de resultados de espectros de masa (Tomado de Titulaer *et al*, 2006)



**Anexo III: Diseño de base de datos en el sistema PLAN (Tomado de Viksna et al, 2007)**



## **BIBLIOGRAFIA**

---

- Anderson, G. Jay, S. Anderson, M. & Hunt, T. 2002. Evaluating the Capability of Information Technology to Prevent Adverse Drug Events: A Computer Simulation Approach. *Journal of the American Medical Informatics Association* , 9 (5), 479-490.
- Carlsson, L. Strender, L. Fridh, G. Nilsson, G. 2006. Clinical categories of patients and encounter rates in primary health care – a three-year study in defined populations. *BMC Public Health* , 6 (35).
- Clinical and Laboratory Standards Institute (CLSI). 2006. Autoverification of Clinical Laboratory Test Results; Approved Guideline. CLSI document AUTO10-A (ISBN 1-56238-620-4). Clinical and Laboratory Standards Institute, 940 West Valley Road, Suite 1400, Wayne, Pennsylvania 19087-1898 USA.
- Davidson, L. Kline, K. Windisch, K. 2006. *Pro SQL Server 2005 Database Design and Optimization*. 1ª Edición. Apress. Estados Unidos. 672 p.
- Dewson, R. 2006. *Beginning SQL Server 2005 for Developers*. 1ª Edición. Apress. Estados Unidos. 537 p.
- Dorr, D. Bonner, L. Cohen, A. Shoai, R. Perrin, R. Chaney, E. Young, A. 2007. Informatics Systems to Promote Improved Care for Chronic Illness: A Literature Review. *Journal of the American Medical Informatics Association* , 14 (2), 156-163.
- Fine, A. Nigrovic, L. Reis, B. Cook, F. Mandl, K. 2007. Linking Surveillance to Action: Incorporation of Real-time Regional Data into a Medical Decision Rule. *Journal of the American Medical Informatics Association* , 14 (2), 206-211.
- Frithsche, L. Schlaefer, A. Budde, K. Schoeter, K. Neumayer, H. 2002. Recognition of Critical Situations from Time Series of Laboratory Results by Case-Based Reasoning. *Journal of American Medical Informatics Association* , 9 (5), 520-528.
- Hancock, J. Toren, R. 2006. *Practical Business Intelligence with SQL Server 2005*. 1ª Edición. Addison Wesley Professional. Estados Unidos. 432 p.

- He, J. Dai, X. Zhao, X. 2007. PLAN: a web platform for automating high-throughput BLAST searches and for managing and mining results. *BMC Bioinformatics*. 8 (53). 1-10.
- Jaimés, F. Farbiarz, J. Alvarez, D. Martínez, C. 2005. Comparison between logistic regression and neural networks to predict death in patients with suspected sepsis in the emergency room. *Critical Care* , 9 (2), R150-R156.
- Jakobovits, R. Rosse, C. Brinkley, J. 2002. WIRM: An Open Source Toolkit for Building Biomedical Web Applications. *Journal of the American Medical Informatics Association*. 9 (6) , 557-570.
- Joseph, J. & Frassica, M. 2005. Frequency of Laboratory Test Utilization in the Intensive Care Unit and Its Implications for Large-Scale Data Collection Efforts. *Journal of the American Medical Informatics Association* , 12 (2), 229-233.
- Khan, A. Griffith, S. Moore, C., Russell, D. Rosario, A. Bertolli, J. 2006. Standardizing Laboratory Data by Mapping to LOINC. *Journal of the American Medical Informatics Association* , 13 (3), 353-355.
- Kline, J. Johnson, C. Webb, W. Runyon, M. 2004. Prospective study of clinician-entered research data in the Emergency Department using an Internet-based system after the HIPAA Privacy Rule. *BMC Medical Informatics and Decision Making* , 4 (17), 1-13.
- Kuperman, G. J. Teich, J. M. Tanasijevic, M. Ma'luf, N. Rittenberg, E. Jha, A. Fiskio, J. Winkelman, J. Bates, D. 1999. Improving Response to Critical Laboratory Results with Automation: Results of a Randomized Controlled Trial. *Journal of the American Medical Informatics Association* , 6 (6), 512-522
- Li, H. Gennari, J. Brinkley, J. 2006. Model Driven Laboratory Information Management Systems. University of Washington, Seattle, WA, AMIA 2006 Symposium Proceedings Page. 484-488.
- Madigan, M. Matinko, J. Parker, J. 1999. *Brock Biología de los Microorganismos* (Vol. 8ª). Madrid, España: Prentice Hall.
- Marocco, S. & Winstanley, E. 2004. Developments and Trends in LIMS – A 'Coming of Age'? STARLIMS.

- Marzolf, B., Troisch, P. 2006. SLIMarray: Lightweight software for microarray facility management. Source Code for Biology and Medicine.
- Microsoft Corporation. 2002. Application Architecture for .NET: Designing Applications And Services. Microsoft Press.
- Morisawa, H. Hirota, M. Toda, T. 2006. Development of an open source laboratory information management system for 2-D gel electrophoresis-based proteomics workflow. BMC Bioinformatics. 7(430), 11p.
- Owen, R. Thrush, C. Cannon, D. Sloan, K. Curran, G. Hudson, T. 2004. Use of Electronic Medical Record Data for Quality Improvement in Schizophrenia Treatment. Journal of the American Medical Informatics Association , 11 (5), 351-357 pp.
- Pace, W. Staton, E. Higgins, G. Main, D. West, D. Harris, D. 2003. Database Design to Ensure Anonymous Study of Medical Errors: A Report from the ASIPS Collaborative. Journal of the American Medical Informatics Association , 10 (6), 531-540.
- Roos, L. Soodeen, R. Bond, R. Burchill, C. 2003. Working More Productively: Tools for Administrative Data. HSR (Health Services Research) , 38 (5), 1339-1358.
- Rotich, J. Hannan, T. Smith, F. Bii, J. Odero, W. Vu, N. Mamlin, B. Mamlin, J. Einterz, R. Tierney, W. 2003. Installing and Implementing a Computer-based Patient Record System in Sub-Saharan Africa: The Mosoriot Medical Record System. Journal of the American Medical Informatics Association , 10 (4), 295-303.
- Staes, C. Bennett, S. Evans, S. Narus, S. Huff, S. Sorensen, J. 2006. A Case for Manual Entry of Structured, Coded Laboratory Data from Multiple Sources into an Ambulatory Electronic Health Record. *Journal of the American Medical Informatics Association* , 13 (1), 12-15.
- Tierney, W. Beck, E. Gardner, R. Musick, B. Shields, M. Shiyonga, N. Spohr, M. 2006. Viewpoint: A Pragmatic Approach to Constructing a Minimum Data Set for Care of Patients with HIV in Developing Countries. *Journal of the American Medical Informatics Association*, 13 (3), 253-260.

- Titulaer, M. Siccama, Dekker, J. Van Rijswijk, Heeren, R, Sillevis, P. Luider, T. 2006. A database application for pre-processing, storage and comparison of mass spectra derived from patients and controls. *BMC Bioinformatics*. 7(403).
- Viksna, J. Celms, E. Opmanis, M. Podnieks, K. Rucevskis, P. Zarins, A. Barrett, A. Guha Neogi, S. Krestyaninova, M. McCarthy, M. Brazma, A. Sarkans, U. 2007. PASSIM – an open source software system for managing information in biomedical studies. *BMC Bioinformatics*. 8(52).