



**UNIVERSIDAD DE LAS CIENCIAS INFORMATICAS**

**FACULTAD # 7**

**ANÁLISIS Y DISEÑO PARA UN SISTEMA DE INFORMACIÓN DE  
LABORATORIOS (LIS)**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE  
INGENIERO EN CIENCIAS INFORMÁTICAS**

**AUTORES:** Alieny García Martínez  
Abel Guzmán Sánchez

**TUTOR:** Ing. William Soñora Cruz  
**ASESOR:** Lic. Ernesto Sarduy Alonso

Ciudad de La Habana, julio del 2007

## **DECLARACIÓN DE AUTORÍA**

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los 5 días del mes de julio del año 2007.

Alieny García Martínez    Abel Guzmán Sánchez

\_\_\_\_\_  
Firma del Autor

\_\_\_\_\_  
Firma del Autor

William Soñora Cruz

\_\_\_\_\_  
Firma del Tutor

## **Agradecimientos**

**De Abel**

*A mi madre por el amor, cariño, dedicación, respeto y confianza que siempre tuvo en mí, por traerme al camino correcto cuando lo necesité y por darme dos hermanos.*

*A mis hermanos por existir.*

*A Ale por ser mi verdadero padre siempre, y a su familia por tratarme como tal.*

*Al resto de mi familia, en especial mis abuelos, a tios, primos y demás, que me harian la tesis de mil páginas si los menciono pero que tengo presentes por ayudarme en lo que han podido. Ojalá siempre los tenga.*

*A mis amigos de la UCI por estar ahí, haciéndonos las vidas más agradables y más pasajeras estos 5 años y ayudarme en lo que pudieron, de todo corazón gracias, sin ustedes no lo habría logrado.*

*A mis vecinos y amigos.*

*A Sorangel y el resto de las personas del laboratorio del hospital de la UCI.*

*A la Mayor Adela, Jefa del laboratorio del Hospital Militar Carlos J. Finlay.*

*A el Guille, por se el primero que me ayudó.*

*Al tutor William*

## **Agradecimientos**

### **De Alieny**

*A mi mamá, por todo el amor, cariño y apoyo que siempre me ha dado.*

*A mi papá, por su confianza y a mi hermana, por ser tan buena conmigo.*

*A mi abuela, por su preocupación en todos estos años, a mi tío Manuel y a mi tía Berta, por toda su ayuda.*

*A mi tía Beki por quererme tanto, a Marta, Belén, Mañe e Isidro por estar siempre.*

*A mis primos*

*A Anet, por soportarme, por quererme y por estar siempre a mi lado, a la Dalex, por ser siempre mi amiga*

*y a Lien, por su comprensión, su apoyo y su cariño.*

*A mi amigo Reinel.*

*A mis amigos Adís, Karen, Yanitza, Yele, Ernest, Annia por su compañía y por todos los momentos que hemos pasado juntos.*

*A Enrique, por su paciencia, su cariño y todo lo que ha hecho por mí.*

*A todos los profesores que me han ayudado en el desarrollo de este trabajo, a mi tutor William.*

*A mi novio.*

*A todos, muchas gracias.*

## **Resumen**

El trabajo que se presenta a continuación es el Análisis y Diseño de un Sistema de Información de Laboratorios para los diferentes centros de atención médica.

Actualmente la gestión de la información relacionada con los pacientes que acuden a los laboratorios clínicos se hace de forma manual, lo que provoca que el flujo de esta sea más lento y que en ocasiones hayan pérdidas de la misma, además de alargarse su período de actualización. También trae como consecuencia que el trabajo sea más complejo para el personal que labora en estos laboratorios. Es por eso, que se hace necesario el diseño de un sistema que gestione y controle dicha información.

Para el diseño del sistema se utiliza la metodología RUP (Proceso Unificado de Desarrollo), el lenguaje de modelado visual UML, y Rational Rose como herramienta CASE para la representación de los diagramas y artefactos generados por la metodología.

Con la implementación de este sistema se espera que se logre el incremento de la capacidad organizativa y mejoras en el control y almacenamiento de la información de los laboratorios clínicos del país, además de una mayor facilidad y organización en el trabajo así como un ahorro de tiempo y recursos.

## Tabla de Contenido.

<b>Introducción .....</b>	<b>1</b>
<b>Capítulo 1. Fundamentación teórica.....</b>	<b>5</b>
1.1 EL SISTEMA NACIONAL DE SALUD .....	5
1.1.1 Los Laboratorios Clínicos.....	6
1.1.2 Informatización del Sistema de Salud. ....	6
1.2 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA. ....	7
Sistema de Información de Laboratorios (SIL). ....	7
Análisis Clínico.....	8
1.3 SISTEMAS EXISTENTES VINCULADOS AL PROBLEMA .....	9
GALEN LAB .....	9
COYA Laboratorios.....	9
CONLAB LIS - Laboratorio .....	10
Soarian MedSuite.....	10
Sistema de Información para Laboratorios Clínicos DLabo. ....	11
1.4 TENDENCIAS, TECNOLOGÍAS Y METODOLOGÍAS ACTUALES. ....	12
Internet .....	12
Aplicaciones Web. ....	13
AJAX.....	14
Plataforma .NET.....	14
1.4.1 Sistema Gestor de Bases de Datos (SGBD).....	15
1.4.2 Lenguaje de programación.....	15
1.4.3 Metodología de desarrollo de software.....	17
1.4.4 UML (Lenguaje de Modelación Unificado).....	20
1.4.5 Herramientas CASE .....	21
1.4.6 Patrones de Diseño.....	23
1.4.7 Arquitectura .....	26
1.4.8 Tecnologías y Herramientas a utilizar. ....	26
<b>Capítulo 2. Características del sistema. ....</b>	<b>28</b>
2.1 OBJETO DE ESTUDIO .....	28
2.1.1 Problema y situación problemática: .....	28
2.1.4 Propuesta de sistema. ....	29
2.1.5 Modelo del Negocio.....	29
2.2.6 Especificación de los requisitos de software. ....	38
2.2.7 Definición de los casos de uso. ....	40
<b>Capítulo 3. Análisis y diseño del sistema. ....</b>	<b>55</b>
3.1 ANÁLISIS DEL SISTEMA .....	55
3.1.1 Modelo de análisis.....	55
3.2 DISEÑO.....	61
3.2.1 Diagramas de interacción .....	61
3.2.2 Diagrama de clases.....	61
3.2.3 Descripción de las clases.....	69
3.2.4 Definiciones de diseño que se aplican .....	69
3.2.5 Concepción de la ayuda.....	72

<b>Conclusiones .....</b>	<b>73</b>
<b>Recomendaciones.....</b>	<b>74</b>
<b>Referencias Bibliográficas .....</b>	<b>75</b>
<b>Bibliografía.....</b>	<b>78</b>
<b>Anexos.....</b>	<b>80</b>
ANEXO #1. DIAGRAMAS DE INTERACCIÓN .....	80
ANEXO #2 DESCRIPCIÓN DE LAS CLASES. ....	84
<b>Glosario de Términos .....</b>	<b>91</b>



## Introducción

En el mundo de hoy, es cada vez más evidente que las Nuevas Tecnologías de la Información y las Comunicaciones cubren las diferentes esferas de la sociedad, contribuyendo al desarrollo de las mismas, y el dominarlas, se convierte en una impostergable necesidad. El país no se encuentra ajeno a este desarrollo, enfrentando el reto de automatizar la sociedad con vistas a integrarse plenamente a la infraestructura global de la información haciendo uso de la informática y las comunicaciones.

Por esta razón, el estado cubano se encuentra inmerso en lo que se ha llamado la “Informatización de la Sociedad Cubana”, este proyecto, mediante el cual se aplican las Nuevas Tecnologías de la Información y las Comunicaciones a las diferentes esferas y sectores de la sociedad para lograr como resultado una mayor eficiencia y eficacia con la optimización de los recursos y el logro de una mayor productividad y competitividad en dichas esferas y sectores. [1]

La salud pública es uno de los sectores que se ha visto favorecido por los impactos de las Nuevas Tecnologías de la Información y las Comunicaciones, pues actualmente se proyecta un proceso aún mayor de informatización del Sistema Nacional de la Salud.

Este proceso de automatización tiene como objetivo la optimización de los Servicios de Salud que se brindan a la población; mayor productividad y competencia en el desempeño de sus profesionales y técnicos, control en la administración de sus recursos, eficacia, y otros aspectos importantes.

El uso de la Informática en la Medicina es una de las aplicaciones más comunes e importantes desde hace varias décadas y actualmente ha permitido al sector de la salud, contar con técnicas novedosas y eficaces en consultas, hospitales y centros de investigación biomédica, no sólo para la gestión de información administrativa, sino que posibilita mejor organización, control y estadísticas en este sector, mejor organización y funcionamiento de los centros hospitalarios, simulación de modelos concretos de investigación, diagnóstico automatizado y diferencial, pronóstico y selección de tratamientos y control de procesos. [1]

Este proceso de informatización de la salud tiene como objetivo abarcar todas las ramas de los diferentes niveles de atención médica del Sistema Nacional de Salud. Una de estas ramas, es el área de los laboratorios clínicos.



## INTRODUCCIÓN

El laboratorio clínico es el lugar donde se realizan análisis clínicos que contribuyen al estudio, prevención, diagnóstico y tratamiento de los problemas de salud de los pacientes. También se le conoce como Laboratorio de Patología Clínica.

Actualmente, ha tomado auge la implantación de sistemas para la automatización de los servicios en los sistemas de laboratorios de varios hospitales del mundo. Estos sistemas mejoran la gestión, el procesado y archivo de toda la información que en ellos se manipula. Entre ellos se encuentran Soarian MedSuite, Coya Laboratorios, CONLAB-LIS, etc.

A nivel nacional sólo se ha implementado un Sistema, GALEN LAB. El mismo está diseñado para ser utilizado por los técnicos, médicos, enfermeras y personal administrativo de Medios de diagnósticos independientes o pertenecientes a un centro hospitalario, que necesitan del sistema para optimizar su trabajo y elevar su eficiencia.

Aunque se han logrado grandes avances en la informatización de esta rama de la salud, no se ha logrado una informatización general que abarque la gran mayoría de los laboratorios. A pesar que se han automatizado varios servicios en muchos hospitales, aun falta mucho por lograr en este sentido, ya que en la mayoría de los centros hospitalarios existentes en el país se está realizando el trabajo manualmente y se hace muy difícil llevar un control de toda la información que hay en movimiento, lo que provoca que el flujo de esta información sea mas lento e incluso haya hasta pérdida de la misma.

Los laboratorios trabajan con diferentes reactivos y productos de los cuales es necesario tener un control estricto, este proceso también se lleva a cabo de forma manual, lo que dificulta el trabajo estadístico sobre la planificación e historiales de consumo por la pérdida de tiempo y esfuerzo del personal implicado en la tarea. También es necesario llevar una serie de estadísticas sobre los pacientes que se tratan, sus enfermedades, análisis y todos los procesos en general del laboratorio, estas estadísticas se dificultan a niveles extremos por la cantidad de pacientes y muestras de análisis con los que se trabajan.

Teniendo en cuenta la gran cantidad de información que se gestiona hoy en día en cualquier servicio de laboratorios, y lo complejo que se hace el trabajo en los mismos, se hace necesario implementar una solución de software, que permita realizar todas las funciones de cualquier laboratorio clínico que permita de manera inmediata, haciendo uso de las tecnologías disponibles, el control y manejo de la información que se mueve en los laboratorios y que garantice la calidad e integridad de la misma.

Ante esta situación se identificó el siguiente problema:



### **Problema Científico**

¿Cómo automatizar el proceso de gestión de información de los laboratorios clínicos?

### **Objeto de Estudio**

El Objeto de Estudio se centra en los procesos de gestión de información vinculados al hospital.

### **Campo de Acción**

El Campo de Acción apunta al proceso gestión de la información en los Laboratorios de un Hospital.

### **Objetivos Generales de la Investigación**

Como Objetivos Generales de la investigación se propone desarrollar el análisis y diseño de un producto informático, para la gestión y procesamiento de la información de los laboratorios clínicos.

### **Objetivos Específicos de la Investigación**

Entre los Objetivos específicos de la investigación se tienen:

- Realizar el diseño de un sistema para automatizar la gestión de la información relacionada con los servicios que se brindan a los pacientes en los laboratorios.
- Definir un *prototipo no funcional*.

### **Tareas de la Investigación**

Con el objetivo de guiar y controlar la investigación se definieron las siguientes Tareas de la Investigación

- Analizar el problema de los laboratorios y la situación actual.
- Realizar un estudio profundo sobre los aspectos con los que debe contar el sistema, los requisitos que debe cumplir.
- Analizar sistemas previos y antecedentes en esta esfera.
- Realizar una ingeniería de software viable para el sistema.

Este trabajo, significará un avance dentro del sistema de los laboratorios cubanos; los especialistas tendrán el trabajo más organizado y accesible, se ahorrarán más recursos y tiempo que de manera manual. Además se espera lograr el incremento de la capacidad organizativa y mejoras en el control y almacenamiento de la información en los laboratorios clínicos del país.

El contenido de este trabajo se estructura en tres capítulos:



## INTRODUCCIÓN

El Capítulo I. Fundamentación teórica: donde se realiza un estudio del estado del arte, o sea un análisis de las principales tecnologías y metodologías existentes que son consideradas para la elaboración de la aplicación. Además, se describen los principales aspectos de las herramientas a utilizar para la implementación de la aplicación y para la realización del análisis y diseño de la misma.

El Capítulo II. Modelado del Negocio y Requerimientos: Se plantea el modelo del negocio, los requisitos y casos de uso del sistema, algunos de los diagramas desarrollados, así como la expansión de los casos de uso.

El Capítulo III. Análisis y diseño del sistema: aquí se muestra el diagrama de clases del análisis. En la parte del diseño se exponen los detalles relacionados con el diseño del sistema propuesto. Se utilizan para su modelado los diagramas de interacción.

Cada capítulo es iniciado por una breve introducción donde se dan a conocer los temas que se desarrollarán durante el mismo. Finaliza con las conclusiones, en las que se plantean los resultados obtenidos.

## **Capitulo 1. Fundamentación teórica.**

En el presente capítulo se muestra una visión de los aspectos relacionados con el Sistema Nacional de Salud, su estructura organizativa y cómo se está llevando a cabo su proceso de informatización. Además se hace referencia a varias tecnologías que pueden ser adecuadas para el buen funcionamiento del sistema, así como a metodologías de desarrollo de software.

Se da una breve explicación del lenguaje de programación y del Sistema de Gestión de Bases de Datos (SGBD) y otras herramientas a utilizar en el desarrollo del sistema. Se hace referencia a sistemas automatizados existentes, mencionando varias características de los mismos, además se abordan conceptos asociados al tema tratado.

### **1.1 El Sistema Nacional de Salud**

El Ministerio de Salud Pública (MINSAP) es el Organismo rector del Sistema Nacional de Salud. Encargado de dirigir, ejecutar y controlar la aplicación de la política del Estado y del Gobierno en cuanto a la Salud Pública, el desarrollo de las Ciencias Médicas y la Industria Médico Farmacéutica.

El Sistema Nacional de Salud (SNS), brinda atención gratuita a toda la población cubana y viene realizando importantes reformas desde los años 60, como parte fundamental de las transformaciones del período revolucionario. [2]

Estas reformas, más que un hecho eventual han sido un proceso permanente, ya que desde los inicios del proceso revolucionario comenzó la creación del Servicio Médico Rural llevando la atención médica a zonas apartadas de la geografía nacional, se dan los primeros pasos para el fortalecimiento de la atención primaria; surgen los policlínicos integrales como una unidad asistencial creada para brindar servicios y resolver los principales problemas existentes en los primeros años de la revolución.

En la década del 80 surge el Programa del Médico y la Enfermera de la Familia, sentando precedentes en la salud pública internacional por su carácter novedoso y futurista, especialmente con la implantación y desarrollo del modelo de atención de Medicina Familiar a partir de 1984.

En 1996, el SNS adoptó desde el punto de vista organizativo, estrategias fundamentales y priorizó cuatro programas básicos para continuar perfeccionándose: Programa de Atención Materno Infantil (PAMI), el de control de enfermedades transmisibles, el de control de enfermedades crónicas no transmisibles, y el de



atención al adulto mayor, todos los que han sido monitorizados, controlados y evaluados de acuerdo a la metodología establecida. [3]

### **Los Laboratorios Clínicos.**

Los adelantos en la salud se han visto reflejados en el laboratorio desde sus mismos inicios. Especialidad surgida en los primeros años de la segunda mitad del siglo XIX, participan médicos por primera vez en 1878, mencionándose el “Laboratorio Clínico” como una profesión médica en 1948, condición ratificada por el Ministro de Salubridad y Asistencia Social y por la Facultad de Medicina de la Universidad de la Habana.

En el año 1962 se crea la especialidad de “Laboratorio Clínico”, por el gobierno revolucionario y se comienza la introducción de nuevos equipos y tecnología, siendo esta especialidad un pilar importante para el progreso y correcto diagnóstico de otras especialidades médicas, comienza un desarrollo vertiginoso de la especialidad en correspondencia con los adelantos científico-técnicos y la industria tecnológica, todo esto gracias a la revolución.

Proporcionando estos adelantos, modernos métodos de diagnósticos en beneficio de la calidad de vida del hombre, por solo mencionar algunos se tiene, por ejemplo en las embarazadas la realización del alfa feto-proteína en el diagnóstico precoz de enfermedades neurológicas en el feto, estudios para el diagnóstico temprano de enfermedades coronarias cardíacas, monitoreo de sustancias corporales fundamentalmente medicamentos, estudios virológicos, creación de nuevos métodos en el diagnóstico de enfermedades parasitarias, aplicación de equipos de microtécnicas computarizadas para la dosificación de diferentes parámetros bioquímicos y hematológicos en el laboratorio. [4]

Actualmente se sigue trabajando en el perfeccionamiento del Sistema Nacional de Salud con el fin de optimizar los servicios que se prestan a la población, elevar los indicadores de salud en el país y satisfacer las necesidades de la población cubana.

### **Informatización del Sistema de Salud.**

Actualmente el Ministerio de Salud Pública (MINSAP) ha definido a la informatización como una de sus prioridades y ha convocando para ello a un grupo de instituciones propias de sector, del Ministerio de Informática y Comunicaciones y de otros organismos de la administración central del estado, para definir

de conjunto la estrategia a desarrollar. En algunos casos se ha tomado como punto de partida sistemas ya desarrollados en el país en el marco de aquella primera estrategia de desarrollo. [5]

La informatización del SNS tiene como objetivo acercar eficientemente y con calidad la prestación de los servicios de salud a la población, incorporando progresiva y sistemática las Nuevas Tecnologías de la Información y las Comunicaciones (NTIC) en función de la adquisición y gestión del conocimiento y los servicios de salud.

Los proyectos que se definan permitirán alcanzar por etapas la informatización de la salud pública cubana, al contar con la integración de los datos generados en los distintos niveles de salud donde puede ser atendido un paciente. Lo anterior permitirá perfeccionar la calidad asistencial ofrecida a la sociedad, facilitar las funciones del personal de la salud y colaborar con la gestión administrativa, asistencial, docente y de investigación. Hay que destacar que estos proyectos se han concebido de forma integrada, y es esta integración la que permite hablar de informatización de la salud pública, no de proyectos aislados.

De acuerdo con la Política de Informatización de la Sociedad Cubana, en estos momentos se trabaja en el desarrollo de un grupo de aplicaciones básicas para la informatización del sector de la salud, como el establecimiento de las redes nacionales de Bancos de Sangre, Nefrología, e Imágenes Médicas. Asimismo, se estructura el Registro Informatizado de Salud, el Sistema de Gestión Hospitalaria, de Atención Primaria de Salud, de Gestión Académica, los Proyectos de Genética Médica, y de Software Educativo.

En el desarrollo e implementación de estos sistemas participan diferentes empresas del Ministerio de la Informática y Comunicaciones como Softel, la Universidad de Ciencias Informáticas (UCI), INFOMED, CEDISAP y las Direcciones Nacionales del Ministerio de Salud Pública implicadas directamente en los primeros productos.

## **1.2 Conceptos asociados al dominio del problema.**

### **Sistema de Información de Laboratorios (SIL).**

Es el sistema informático de los servicios de los laboratorios clínicos que recoge, controla, analiza y gestiona todos los datos e información correspondiente a estos servicios.

El SIL ha de mantener la autonomía necesaria para atender sus propias necesidades. Al hablar de SIL, normalmente se refiere al manejo de la información derivada de todo el proceso analítico, desde que el

médico formula una solicitud de análisis hasta que el laboratorio emite un informe. Actualmente, ha sido necesario ampliar el campo del SIL, para ello se divide en los siguientes subsistemas:

- El sistema de información asistencial abarca desde la formulación de una petición analítica por parte del médico hasta la edición del informe final.
- El sistema de información de la calidad ha de permitir la toma de decisiones relacionadas con la gestión de la calidad del laboratorio.
- El sistema de información de la actividad recoge y agrupa los datos de actividad del SIL para su posterior estudio y explotación.
- El sistema de información económica integrará datos procedentes de diferentes áreas de la institución con los generados por el propio laboratorio.
- El sistema de información de suministros abarcará todos los aspectos relacionados con la gestión de suministros.
- El sistema de información para la gestión ha de permitir interrelacionar los datos procedentes de los diferentes subsistemas con objeto de facilitar el manejo de toda la información, que en última instancia servirá para la toma de decisiones. [6]

De manera general se puede decir que es la herramienta informática que permite realizar los procesos de gestión de información de un laboratorio clínico.

### **Análisis Clínico**

Análisis clínico se le llama comúnmente a la prueba solicitada al laboratorio clínico por un médico para confirmar o descartar un diagnóstico. Forma parte del proceso de atención a la salud que se apoya en el estudio de distintas muestras biológicas mediante su análisis en laboratorio y que brinda un resultado objetivo que puede ser tanto cuantitativo (un número) o cualitativo (positivo o negativo). [7]

### **1.3 Sistemas existentes vinculados al problema**

#### **GALEN LAB**

Es un sistema desarrollado por la empresa SOFTEL, es una aplicación de escritorio, está diseñado para ser utilizado por los técnicos, médicos, enfermeras y personal administrativo de Medios de diagnósticos independientes o pertenecientes a un centro hospitalario, que necesitan del sistema para optimizar su trabajo y elevar su eficiencia.

Cuenta con un sistema de ayuda en línea que brinda al usuario toda la información que necesite sobre el proceso que está efectuando y posee un estricto control de acceso que permite a cada técnico visualizar solamente la información y opciones del sistema relacionadas con su actividad. No permite el acceso a ninguno de los módulos e informaciones a partir de puntos no autorizados.

Está desarrollado en Delphi, para una plataforma Windows de 32 bits (Windows NT o superior y Windows 98 o superior) con una configuración Cliente/Servidor y el uso del gestor de bases de datos relacional SQL Server como reservorio de la información.

El personal que usa GALEN LAB, mantiene una relación continua y estable con el sistema y no requiere de conocimientos profundos del software de soporte para trabajar con el mismo.

El sistema está conformado por cuatro módulos funcionales:

- Configuración
- Solicitudes
- Resultados
- Autoanalizadores [8]

#### **Ámbito Internacional.**

En el mercado del software a nivel internacional existen varios ejemplos de sistemas para hospitales y organizaciones sanitarias.

#### **COYA Laboratorios**

COYA Laboratorios es un nuevo y moderno sistema aplicativo para laboratorios. Derivado de los productos de línea alta hospitalaria, COYA Laboratorios integra todas las necesidades del laboratorio con las ventajas de su entorno Microsoft Windows, y la integración con todas sus aplicaciones Office. Permite

el ingreso de datos multimediales: fotos, gráficos, videos, o cualquier tipo de objeto insertable, como una hoja de cálculo o documento de texto.

Se integra a Internet, logrando enviar todos los informes y protocolos a través del correo electrónico. COYA Laboratorios es compatible con Microsoft Office, así se puede migrar datos estadísticos o informes a Excel o Word, obteniendo un valor agregado en el análisis de la información. Además ahora se agregó la posibilidad de exportar a formato PDF los informes. [9]

Este software es propietario. Es una aplicación de escritorio.

### **CONLAB LIS - Laboratorio**

CONLAB LIS - Laboratorio es un sistema de información y manejo administrativo para Laboratorios Clínicos, de fácil manejo para el usuario final debido a su sistema autodidáctico. Está escrito con productos de Microsoft. Cumple con las necesidades de la vida diaria de los Laboratorios Clínicos de Centro, Sur América y Europa, pues ha sido diseñado luego de un estudio de las necesidades de la región.

Sus principales características son:

- Plataforma en Visual Basic, Access, SQL y Windows NT/2000
- Crecimiento Ilimitado
- Ingreso de pacientes desde sitios remotos como Laboratorios satélites, y oficinas de Médicos
- Hojas de flebotomía y recibo de las muestras tomadas o llegadas al Laboratorio.
- Múltiples listados de trabajo, configurables por el usuario. Hojas de trabajo por: Instrumento, sección, departamento, operario, o por prueba.
- Captura de resultados de igual forma a los listados de trabajo.
- Interfaces listas para su uso, con mas de 400 equipos
- Verificación de los resultados anormales por filtro automático. [10]

Este software es propietario. Es una aplicación de escritorio.

### **Soarian MedSuite**

Es un sistema integral de gestión hospitalaria desarrollado en Argentina con varias funcionalidades que incluye varios módulos entre ellos un Sistema Integral de Laboratorios (LIS)

LIS es una herramienta diseñada para brindar soporte en todas las actividades de un equipo de laboratorio. Sus funcionalidades se adaptan tanto a laboratorios de Análisis clínicos como Microbiológicos. Con este Módulo de Soarian Medsuite el personal del Laboratorio puede ordenar y procesar las pruebas de laboratorio cuyos resultados son automáticamente registrados en la Historia Clínica Digital del paciente. El módulo también permite definir las pruebas de cada una de las secciones del laboratorio especificando el tipo de resultado, muestra, recipiente y otras características necesarias para la determinación automática de las muestras que se deben obtener de cada paciente. LIS provee funcionalidades que facilitan al profesional del Laboratorio el análisis de la evolución de los resultados de pruebas particulares en un paciente.

El módulo permite especificar las pruebas que serán procesadas por algún instrumento del laboratorio, y por otro lado, el Módulo Interfaces con Instrumentos de Soarian Medsuite provee las interfaces con diversos instrumentos de laboratorio.

El Módulo Interfaces con Instrumentos de Soarian Medsuite permite a LIS recibir automáticamente resultados a través de vínculos electrónicos a instrumentos de Laboratorio. También puede enviar órdenes hacia los instrumentos para la realización de pruebas [11].

Este software es propietario. Es una aplicación de escritorio.

### **Sistema de Información para Laboratorios Clínicos DLabo.**

Equipos y Sistemas Computarizados ESICOM, S.A. se consolida como una compañía costarricense, para el desarrollo de sistemas para Laboratorios Clínicos en Centroamérica. Además brinda consultorías en el área de sistemas de información para empresas.

DLabo es un sistema diseñado por esta compañía para facilitar la labor de los operadores, la administración eficiente y el reporte de los resultados de las pruebas realizadas en un laboratorio clínico moderno. Es un sistema sumamente robusto, probado, fácil de usar y sobre todo confiable. También tiene la capacidad de enviar resultados por fax, por correo electrónicos o que mediante autenticación por medio de una clave de acceso los doctores y pacientes puedan ver sus resultados en un sitio de Web. (Estos son módulos opcionales)

El sistema DLabo está desarrollado en Linux, lo que permite contar con las características de un sistema operativo multiusuario, multitarea, escalable, con gran seguridad y como beneficio adicional el no tener el

costo asociado a licencias de sistema operativo. El sistema cuenta con un subsistema que chequea los valores normales de acuerdo con la edad y el sexo de los pacientes y además previene la impresión de resultados cuando estos se encuentran fuera de rangos posibles de las mismas, todos estos parámetros pueden ser definidos por el personal autorizado del laboratorio sin tener que recurrir a los programadores del sistema. [12]

Este software es propietario. Es una aplicación de escritorio.

Como se ha podido observar, en el mundo existen muchos sistemas para la informatización de los servicios en los laboratorios clínicos, sistemas que cuentan con las funcionalidades necesarias para agilizar los servicios de laboratorios y mejorarlos, pero por su elevado precio de venta no pueden ser adquiridos, y existen otros que no cumplen con los requisitos necesarios para darle cumplimiento a los procesos que se desarrollan a diario en los laboratorios clínicos del país.

En Cuba, el único sistema existente es GalenLab, que hasta el momento no cuenta con una versión Web.

A partir del estudio detallado de los diferentes sistemas mencionados anteriormente y del sistema de laboratorios en Cuba, se decide desarrollar una aplicación Web funcional, que se adapte a las necesidades del sistema de salud cubano, y que permita a los clientes finales un entorno de trabajo flexible, entendible y con toda la información necesaria para su trabajo.

#### **1.4 Tendencias, Tecnologías y Metodologías actuales.**

En la actualidad, para la realización de un software se tienen en cuenta una serie de tendencias y tecnologías informáticas, con el objetivo de buscar la más adecuada para el desarrollo del mismo.

Las tecnologías de la Información no son más que aquellas herramientas computacionales e informáticas que procesan, almacenan, sintetizan, recuperan y presentan información representada de la más variada forma. Es un conjunto de herramientas, soportes y canales para el tratamiento y acceso a la información. Constituyen nuevos soportes y canales para dar forma, registrar, almacenar y difundir contenidos informacionales. Algunos ejemplos de estas tecnologías son la pizarra digital (ordenador personal + proyector multimedia) y la Web. [13]

#### **Internet**

Internet es una red mundial de computadoras con un conjunto de protocolos, el más destacado, el TCP/IP, que comparten información unas con otras por medio de páginas o sitios. Para que todos estas



computadoras puedan coexistir y comunicarse efectivamente entre sí, debe existir un camino físico que las una (líneas telefónicas, conmutadas, redes digitales, enlaces satelitales, fibra óptica, cable coaxial, etc.)

Algunos de los servicios disponibles en Internet aparte de la Web son el acceso remoto a otras máquinas (SSH y telnet), transferencia de archivos (FTP), correo electrónico (SMTP), boletines electrónicos, conversaciones en línea (IRC y Chat), mensajería instantánea, transmisión de archivos (P2P, P2M, Descarga Directa), etc. [14]

### **Aplicaciones Web.**

Una aplicación Web es un sistema informático que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una Intranet. Las aplicaciones Web son populares debido a la practicidad del navegador Web como cliente ligero. La habilidad para actualizar y mantener aplicaciones Web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad.

Las aplicaciones Web generan dinámicamente una serie de páginas en un formato estándar, soportado por navegadores Web comunes como HTML o XHTML. La mayoría de los usuarios potenciales tienen familiaridad en navegar por Internet y están acostumbrados con la apariencia de las páginas Web. También se cuenta con los beneficios de una instalación centralizada en un sólo lugar, así como las actualizaciones sin tener que distribuir programa alguno. En vez de tener que repartir la aplicación a cientos ó miles de máquinas, únicamente se coloca el nuevo software en un solo sitio y de inmediato todos los usuarios acceden a éste. Lo mismo ocurre cuando se requiere hacer alguna actualización.

Las interfaces Web tienen ciertas limitantes en la funcionalidad del cliente. Métodos comunes en las aplicaciones de escritorio como dibujar en la pantalla o arrastrar-y-soltar no están soportadas por las tecnologías Web estándar. Los desarrolladores Web comúnmente utilizan lenguajes interpretados del lado del cliente para añadir más funcionalidad, especialmente para crear una experiencia interactiva que no requiera recargar la página cada vez.

Ventajas de las Aplicaciones Web:

- No se requieren complicadas combinaciones de Hardware/Software para utilizar estas aplicaciones. Solo un computador con un buen navegador Web.

- Se facilita el trabajo a distancia. Se puede trabajar desde cualquier PC o computador portátil con conexión a Internet.
- Actualizar o hacer cambios en el Software es sencillo y sin riesgos de incompatibilidades. Existe solo una versión en el servidor lo que implica que no hay que distribuirla entre los demás computadores. El proceso es rápido y limpio.
- No requieren instalación, pues usan tecnología Web, lo cual permite el aprovechamiento de todas las características del Internet.
- Son fáciles de usar (no requieren conocimientos avanzados de computación). [15]

## **AJAX**

El AJAX, acrónimo de JavaScript y XML asincrónicos, es una metodología de desarrollo Web para crear partes de aplicaciones más interactivas, que corran en el cliente es decir en el navegador del usuario, manteniendo comunicación asincrónica con el servidor. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma. AJAX es una combinación de tres tecnologías ya existentes.

La primera XHTML o HTML y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información, la segunda Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMA Script como Java Script y Jscript , para mostrar e interactuar dinámicamente con la información presentada, y el objeto XML http Request para intercambiar datos asincrónicamente con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XML http Request para realizar dichos intercambios. [16]

## **Plataforma .NET**

.NET es una plataforma estándar compatible con una serie de lenguajes, así como con aplicaciones orientadas a Internet, que permite programar en cualquier lenguaje, ya sea Visual Basic, XML o ASP, entre otros, y que la aplicación desarrollada funcione en un sistema operativo independientemente de la versión utilizada. Funciona con todas las versiones de Windows.

El objetivo de la plataforma .NET es simplificar el desarrollo de aplicaciones Web. Esta plataforma además soporta los estándares sobre los cuales se basan los servicios Web.

Ventajas de la Plataforma .NET

- Modelo de programación consistente y sencillo, completamente orientado a objetos.

- Eliminación del temido problema de compatibilidad entre DLL.
- Ejecución multiplataforma.
- Aislamiento de memoria entre procesos y comprobaciones automáticas de seguridad de tipos en las conversiones.
- Soporte multihilo.
- Gestión del acceso a objetos remotos que permite el desarrollo de aplicaciones distribuidas de manera transparente a la ubicación real de cada uno de los objetos utilizados en las mismas.
- Interoperabilidad con código preexistente, de manera que es posible utilizar con facilidad cualquier librería de funciones u objetos creados con anterioridad a la aparición de la plataforma .NET. [17]

#### **1.4.1 Sistema Gestor de Bases de Datos (SGBD)**

Los SGBD son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. El objetivo fundamental de un SGBD consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, o sea, de forma que no le sea necesario conocer el modo de almacenamiento de los datos en la computadora, ni el método de acceso empleado.

#### **POSTGRESQL.**

PostgreSQL es un motor de base de datos. Es servidor de base de datos relacional libre. Los usuarios pueden crear sus propios tipos de datos. Corre en la mayoría de los Sistemas Operativos más utilizados incluyendo, Linux, varias versiones de UNIX, BeOS y Windows. Tiene soporte completo para llaves foráneas, joins (unión), vistas, subconsultas, disparadores (triggers), y procedimientos almacenados (en varios lenguajes) También soporta almacenamiento de objetos grandes (imágenes, sonido y video) Brinda la posibilidad de herencia de tablas y acceso concurrente multiversión (no se bloquean las tablas, ni siquiera las filas, cuando un procesador escribe).

Tiene la desventaja que consume bastante recursos.

#### **1.4.2 Lenguaje de programación**

Los lenguajes de programación son herramientas que permiten crear programas y software. Consiste en un conjunto de reglas sintácticas y semánticas que definen un lenguaje informático.



La programación Web permite la creación de sitios dinámicos en Internet. Los lenguajes de programación Web se dividen en lenguajes del lado del cliente, que no son más que aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pretratamiento, como son el HTML, el Java y el JavaScript, los cuales son simplemente incluidos en el código HTML, y los lenguajes del lado del servidor que son aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Entre estos se encuentran PERL, ASP, PHP, JSP.

### **JavaScript**

JavaScript es un lenguaje interpretado, es decir, que no requiere compilación. Introducido por primera vez en 1995 en la versión 2 de Netscape, JavaScript representa, actualmente, el estándar no oficial de referencia para el uso de script en documentos hipertextuales. Está diseñado para controlar la apariencia y manipular los eventos dentro de la ventana del navegador Web. Este se integra directamente en páginas HTML y la ventaja que presenta sobre el HTML es que permite crear páginas Web más dinámicas, lo que las hace más atractivas para el usuario. [18]

### **ASP.NET**

ASP.NET es un conjunto de tecnologías de desarrollo de aplicaciones Web comercializado por Microsoft. Es usado por programadores para construir sitios Web, aplicaciones Web y servicios XML. Forma parte de la plataforma .NET de Microsoft y es la tecnología sucesora de la tecnología Active Server Pages (ASP).

ASP.NET introduce el concepto del code-behind, por el que una misma página se compone de dos ficheros: el de la interfaz de usuario y el de código. Con ello se facilita la programación de aplicaciones en múltiples capas, lo que en definitiva se traduce en la total separación entre lo que el usuario ve y lo que la base de datos tiene almacenado. Por tanto, cualquier cambio drástico de especificaciones minimiza los cambios en la aplicación y maximiza la facilidad de mantenimiento. Asimismo, ASP.NET sirve tanto para Web sencillas como para grandes aplicaciones. [19]

Brinda la posibilidad de escribir código en diferentes lenguajes. ASP.NET soporta la programación en lenguajes potentes como, VisualBasic.Net (VB) y C#.

## **C Sharp (C#)**

C# es un lenguaje creado por Microsoft para la plataforma .NET, un descendiente de C++ orientado específicamente al trabajo con componentes. Tiene lo mejor de C++, Java y los lenguajes de entornos RAD como Visual Basic y Delphi.

En la actualidad existen los siguientes compiladores para el lenguaje C#:

- Microsoft .NET framework SDK incluye un compilador y librerías reutilizables, pero no un IDE.
- Microsoft Visual C#, IDE por excelencia de este lenguaje, versión 2002, 2003 y 2005.
- #develop, es un IDE libre bajo licencia LGPL, muy similar a Microsoft Visual C#.
- Mono, es una implementación GPL de todo el entorno .NET. Como parte de esta implementación se incluye un compilador de C#.
- DotGNU Portable.NET, de la Free Software Foundation. [20]

### **1.4.3 Metodología de desarrollo de software.**

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. Están clasificadas en metodologías tradicionales y metodologías ágiles. Las metodologías tradicionales se caracterizan por exponer procesos basados en planeación exhaustiva. Van indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

Por otra parte están las metodologías ágiles, estas ponen más énfasis en la adaptabilidad que en la previsibilidad.

### **Extreme Programing (XP)**

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes y simplicidad en las soluciones implementadas. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. [21]

La metodología se basa en:

- Pruebas Unitarias: se basa en las pruebas realizadas a los principales procesos, de tal manera que si se adelanta en algo hacia el futuro, se puedan hacer pruebas de las fallas que pudieran ocurrir. Es como adelantarse a obtener los posibles errores.
- Refabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

Lo fundamental en este tipo de metodología es la comunicación, entre los usuarios y los desarrolladores, la simplicidad, al desarrollar y codificar los módulos del sistema y la retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales. [22]

### **Crystal Methodologies**

Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependen del tamaño del equipo, estableciéndose una clasificación por colores. [23]

La familia Crystal dispone un código de color para marcar la complejidad de una metodología: cuanto más oscuro un color, más “pesado” es el método. Cuanto más crítico es un sistema, más rigor se requiere. Los métodos se llaman Crystal evocando las facetas de una gema: cada faceta es otra versión del proceso, y todas se sitúan en torno a un núcleo idéntico. Hay cuatro variantes de metodologías: Crystal Clear (“Claro como el cristal”) para equipos de 8 o menos integrantes; Amarillo, para 8 a 20; Naranja, para 20 a 50; Rojo, para 50 a 100. La más exhaustivamente documentada es Crystal Clear (CC), CC puede ser



usado en proyectos pequeños de categoría D6, aunque con alguna extensión se aplica también a niveles E8 a D10. El otro método elaborado en profundidad es el Naranja, apto para proyectos de duración estimada en 2 años. Los otros dos aún se están desarrollando. Como casi todos los otros métodos, CC consiste en valores, técnicas y procesos.

Los métodos Crystal no prescriben las prácticas de desarrollo, las herramientas o los productos que pueden usarse, pudiendo combinarse con otros métodos como Scrum, XP y Microsoft Solutions Framework. [24]

### **Scrum**

SCRUM es una metodología ágil de gestión de proyectos cuyo objetivo primordial es elevar al máximo la productividad de un equipo. Reduce al máximo la burocracia y actividades no orientadas a producir software que funcione y produce resultados en periodos muy breves de tiempo (cada 30 días). Sólo abarca prácticas de gestión sin entrar en las prácticas de desarrollo como puede hacer XP. Más bien delega completamente en el equipo la responsabilidad de decidir la mejor manera de trabajar para ser lo más productivos posibles. Está especialmente indicada para proyectos con un rápido cambio de requisitos. [25]

Una de las características más importantes es que es muy fácil de explicar y de entender, lo que ayuda mucho a su implantación.

Por otra parte SCRUM puede ser aplicado a distintos modelos de calidad (como podría ser CMMI). Esta metodología se caracteriza también por las reuniones a lo largo proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. [26]

### **RUP**

El Proceso Unificado (RUP, el original inglés Rational Unified Process) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es en realidad un refinamiento realizado por Rational Software del más genérico Proceso Unificado. La metodología RUP, divide en 4 fases el desarrollo del software:

- Inicio: El Objetivo en esta etapa es determinar la visión del proyecto.
- Elaboración: En esta etapa el objetivo es determinar la arquitectura óptima.

- **Construcción:** En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- **Transmisión:** El objetivo es llegar a obtener el release del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los Objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

Es recomendable que a cada una de estas iteraciones se les clasifique y ordene según su prioridad, y que cada una se convierte luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. [27]

Es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos, aunque es mas apropiada para proyectos grandes.

Los aspectos definitorios del proceso unificado se resumen en tres fases claves: dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental.

El Proceso Unificado está basado en componentes. Utiliza el lenguaje unificado de modelado (UML) para preparar todos los esquemas de un sistema de software. De hecho, UML es una parte esencial de RUP.

#### **1.4.4 UML (Lenguaje de Modelación Unificado)**

UML (Unified Modeling Language) o Lenguaje de Modelación Unificado es un lenguaje gráfico para especificar, construir, visualizar y documentar las partes o artefactos (información que se utiliza o produce mediante un proceso de software). Pueden ser artefactos: un modelo, una descripción que comprende el desarrollo de software que se basen en el enfoque Orientado a Objetos, utilizándose también en el diseño Web. UML usa procesos de otras metodologías, aprovechando la experiencia de sus creadores, eliminó los componentes que resultaban de poca utilidad práctica y añadió nuevos elementos.

UML es un lenguaje más expresivo, claro y uniforme que los anteriores definidos para el diseño Orientado a Objetos, que no garantiza el éxito de los proyectos pero si mejora sustancialmente el desarrollo de los mismos, al permitir una nueva y fuerte integración entre las herramientas, los procesos y los dominios.

De forma general las principales características son:

- Lenguaje unificado para la modelación de sistemas
- Tecnología orientada a objetos
- El cliente participa en todas las etapas del proyecto
- Corrección de errores viables en todas las etapas
- Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente/servidor

UML es desde finales de 1997, un lenguaje de modelado orientado a objetos estándar, de acuerdo con el Object Management Group, siendo utilizado diariamente por grandes organizaciones como: Microsoft, Oracle, Rational. [28]

#### **1.4.5 Herramientas CASE**

Las Herramientas CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, calculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

#### **Rational Rose**

Existen herramientas Case de trabajo visuales como el Analise, el Designe y el Rational Rose, que permiten realizar el modelado del desarrollo de los proyectos, en la actualidad la mejor y más utilizada en el mercado mundial es Rational Rose.

Es una herramienta software para el Modelado Visual mediante UML de sistemas software, con plataforma independiente que ayuda a la comunicación entre los miembros de equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Una de las grandes ventajas de Rose es que

utiliza la notación estándar en la arquitectura de software (UML), o sea que permite a todo el equipo de desarrollo comunicarse con un lenguaje y una herramienta.

Rational Rose le permite visualizar, entender, y refinar sus requerimientos y arquitectura antes de enfrentarse al código. Esto le permite evitar esfuerzos desperdiciados en el ciclo de desarrollo. [29]

Esta herramienta permite completar una gran parte de las disciplinas (flujos fundamentales) del proceso unificado de Rational (RUP):

- Modelado del negocio
- Captura de requisitos
- Análisis y diseño
- Implementación
- Control de cambios y gestión de configuración.

### **CASE Studio**

Este software ha sido especialmente creado para desarrolladores, diseñadores de bases de datos y toda aquella persona que desee mantener una estructura eficiente en su base de datos.

El programa permite generar rápidamente diagramas gráficos de las bases de datos relacionales con las que se está trabajando, simplificando mucho el trabajo del programador. Su principal característica, es su potente sistema de ingeniería inversa, que permite identificar y estructurar bases de datos ya existentes para poder trabajar con ellas sin problemas. CASE Studio 2 soporta Oracle, MySQL, MS SQL, MaxDB, Firebird, PostgreSQL y otros sistemas.

Características:

- Diagramas entidad relación.
- Soporta variados tipos de bases de datos.
- Genera scripts SQL (dml).
- Diagrama de flujo.
- Genera documentación detallada de HTML y RTF, y mucho más. [30]

### 1.4.6 Patrones de Diseño

Los Patrones de Diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Un patrón de diseño es una solución a un problema de diseño no trivial que es efectiva (ya se resolvió el problema satisfactoriamente en ocasiones anteriores) y reusable (se puede aplicar a diferentes problemas de diseño en distintas circunstancias).

#### Patrón Experto

¿Cómo asignar responsabilidades, de la forma más eficiente?

Asignar una responsabilidad al experto en la información, la clase que cuenta con la información necesaria para cumplir la responsabilidad.

Se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto provee un bajo nivel de acoplamiento.

#### Patrón Creador

¿Quién debería ser responsable de crear una nueva instancia de alguna clase?

Las responsabilidades de crear una instancia de la **clase A** se le dará a aquella **clase B**, en los siguientes casos:

- B agrega los objetos de A
- B contiene los objetos de A
- B registra las instancias de los objetos de A.
- B tiene los datos de inicialización que serán enviados a A cuando este objeto sea creado.

El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento.

#### Patrón Controlador

¿Quién debería encargarse de atender un evento del sistema?

El controlador es un intermediario entre la interfaz de usuario y el núcleo de las clases donde reside la lógica de la aplicación.

El controlador no realiza mucho trabajo por si mismo; más bien coordina la actividad de otros objetos.

Asignar la responsabilidad del manejo de mensajes de los eventos del sistema a una clase que represente alguna de las siguientes opciones:

- El sistema global.
- La empresa u organización global.
- Algo activo en el mundo real que pueda participar en la tarea.
- Un manejador artificial de todos los eventos del sistema de un caso de uso (controlador de casos de uso)

### **Patrón Bajo Acoplamiento**

¿Cómo dar soporte a una mínima dependencia y a un aumento de la reutilización?

Una clase con bajo acoplamiento no depende de “muchas otras” clases. Las clases con alto acoplamiento recurren a muchas clases y no es conveniente.

Son más difíciles de mantener, entender y reutilizar.

### **Patrón Alta Cohesión**

Es la meta principal que ha de buscarse en todo momento. Es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño.

Una clase con mucha cohesión es útil porque es bastante fácil darle mantenimiento, entenderla y reutilizarla. Su alto grado de funcionalidad, combinada con una reducida cantidad de operaciones, también simplifica el mantenimiento y los mejoramientos. La ventaja que significa una gran funcionalidad también soporta un aumento de la capacidad de reutilización. [31]

### **Fábrica Abstracta**

El problema que intenta solucionar este patrón es el de crear diferentes familias de objetos. Uno de los usos más comunes es el de creación de interfaces gráficas de distinto tipo.

El patrón Fábrica Abstracta está aconsejado cuando se prevé la inclusión de nuevas familias de productos, pero puede resultar contraproducente cuando se añaden nuevos productos o cambian los existentes. Cuando se usa en conjunción con Singleton se crea una única fábrica, de modo que no habrá usos inconsistentes de los productos en el programa.

## **Fábrica Pura**

En diseño de software, el patrón de diseño Fabrica Pura consiste en utilizar una clase constructora (al estilo del Fábrica Abstracta) abstracta con unos cuantos métodos definidos y otro(s) abstracto(s): el dedicado a la construcción de objetos de un subtipo de un tipo determinado. Es una simplificación de la Fábrica Abstracta, en la que la clase abstracta tiene métodos concretos que usan algunos de los abstractos; según sean usados una u otra hija de esta clase abstracta, se tendrá uno u otro comportamiento.

Las clases principales en este patrón son el *creador* y el *producto*. El creador necesita crear instancias de productos, pero el tipo concreto de producto no debe ser forzado en las subclases del creador, porque entonces las posibles subclases del creador deben poder especificar subclases del producto para utilizar.

La solución para esto es hacer un método abstracto (el método de la fábrica) que se define en el creador. Este método abstracto se define para que devuelva un producto. Las subclases del creador pueden sobrescribir este método para devolver subclases apropiadas del producto.

## **Patrón Singleton**

El patrón de diseño Singleton está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto.

Su intención consiste en garantizar que una clase sólo tenga una instancia y proporcionar un punto de acceso global a ella.

El patrón Singleton se implementa creando en la clase un método que crea una instancia del objeto sólo si todavía no existe alguna. Para asegurar que la clase no puede ser instanciada nuevamente se regula el alcance del constructor (con atributos como protegido o privado).

Las situaciones más habituales de aplicación de este patrón son aquellas en las que dicha clase controla el acceso a un recurso físico único (como puede ser el ratón o un archivo abierto en modo exclusivo) o cuando cierto tipo de datos debe estar disponible para todos los demás objetos de la aplicación. [32]

### **1.4.7 Arquitectura**

#### **Modelo Cliente-Servidor en tres capas**

El sistema a desarrollar se centra en una arquitectura de tres capas. Estas capas tienen diferentes propósitos e interactúan entre sí pero de forma independiente, de forma tal que al hacer cambios en una no se afectan las otras dos. Las capas son las siguientes:

La **Capa de Presentación**, que está formada por los Componentes de Interfaz de Usuario. Estos componentes pueden ser vistos como la parte con la cual interactuar el usuario.

La **Capa del Negocio**, donde van implementadas todas las reglas de negocio. Esta capa es la encargada de recibir el pedido del cliente, extrae los datos al interactuar con la capa de datos y le da una respuesta al cliente tal y como él la pidió

La **Capa de Acceso a Datos**, donde están guardados los datos que serán utilizados por la aplicación.

### **1.4.8 Tecnologías y Herramientas a utilizar.**

Se desarrollará una aplicación Web. La metodología usada para el desarrollo del proyecto fue RUP. Este es un proceso que garantiza la elaboración de todas las fases de un producto de software orientado a objeto. RUP utiliza UML, que es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. Además ofrece comodidades y facilidad de entendimiento de la ingeniería del software desde el punto de vista organizativo.

Se utilizara Rational Rose Enterprise Edition 2003 como herramienta CASE, para sustentar la documentación, como modelador visual de la notación UML (Unified Modeling Language), para la confección de los diagramas que se ilustran en este documento. Esta herramienta es muy completa y ofrece amplias potencialidades. Como gestor de bases de datos para el trabajo se utilizará Postgree SQL, ya que para sistemas con grandes bases de datos, es más rápido y eficiente usarlo. Posee estabilidad y confiabilidad, nunca ha presentado caídas en varios años de operación de alta actividad. Tiene ciertas características orientadas a objetos, aunque consume bastantes recursos y carga más el sistema.

El lenguaje de programación del lado del servidor que se utilizará para la implementación del sistema será ASP, a través de la tecnología ASP.NET y el lenguaje de programación C#. El lenguaje ASP tendrá embebido código Java Script (Controles de AjaxToolkit) que corre del lado del cliente.

### **Conclusiones**

En este capítulo se profundizó en el conocimiento de algunos sistemas existentes semejantes al que se va realizar. Además se realizó un análisis de las tecnologías que serán utilizadas a lo largo del desarrollo del sistema propuesto, fundamentándose las elecciones del lenguaje, el sistema gestor de bases de datos, y la metodología a utilizar, con el objetivo de profundizar en el conocimiento de estos temas y fundamentar la necesidad de dar una solución informática a la situación existente en los laboratorios clínicos de los hospitales del país.



## **Capitulo 2. Características del sistema.**

En el presente capítulo se aborda el tema relacionado con la modelación del negocio que tiene lugar en los laboratorios clínicos. Este es el primer flujo de trabajo que se realiza durante las fases de desarrollo de un sistema informático y tiene el objetivo de que se pueda comprender la estructura y la dinámica de la organización, los problemas actuales de la misma, que se puedan identificar las mejoras potenciales y obtener los requerimientos del sistema que va a soportar la organización.

Se definen y se describen además los actores y trabajadores del negocio, se describe el flujo actual de los procesos así como los procesos que serán objeto de automatización.

### **2.1 Objeto de estudio**

#### **2.1.1 Problema y situación problémica:**

El laboratorio clínico de un hospital, es el lugar donde se realizan análisis clínicos que contribuyen al estudio, prevención, diagnóstico y tratamiento de los problemas de salud de los pacientes. A estos acuden diariamente un gran número de pacientes para ser atendidos, o sea que el flujo de información diario en ellos es muy grande. En el país existe un sistema informático para el control de esta información, pero dicho sistema esta instalado solo en algunos laboratorios clínicos. En el resto de los laboratorios, todos los procesos que en ellos se realizan, toda la gestión de la información se realiza de forma manual, lo que dificulta un poco el trabajo de los compañeros que allí laboran.

#### **2.1.2 Descripción de los procesos a automatizar.**

Se desea automatizar los procesos de emisión de una solicitud de análisis en una consulta, para lo que primeramente es necesario hacer una búsqueda del paciente al que se le va a realizar el análisis, seleccionar los exámenes y establecer una prioridad. Otro proceso que se desea automatizar es la recepción de la solicitud de análisis, emitidas desde la consulta. En este caso ya se recibirían las solicitudes de análisis al ser emitidas por el doctor en su consulta y solamente se le asigna un encargado de tomar las muestras.

Cuando el paciente se presenta en el área indicada, el encargado de muestras toma datos relativos a la muestra; envase, estado y algún juicio aclaratorio sobre el estado de muestra que serán informatizados al igual que los datos si es necesario almacenar la muestra estos parámetros son temperatura, reactivo a utilizar y fotosensibilidad. Este es otro proceso que requiere ser automatizado.



Luego para realizar el análisis la muestra pasa al laboratorio donde el técnico realiza el análisis y emite un resultado, este proceso también se automatizará al igual que la liberación del mismo, proceso en el que se decide si el análisis esta listo para que sus resultados sean vistos por los interesados en el mismo.

### **2.1.4 Propuesta de sistema.**

Para solucionar los problemas que se están presentando en los laboratorios clínicos, se propone realizar una aplicación. La misma permitirá automatizar el proceso de gestión de la información en los laboratorios de los pacientes que acudan a los mismos y el procesamiento, manipulación, almacenamiento de los datos generados por los laboratorios.

También se encargará de la gestión de los procesos que se desarrollan en los laboratorios de un hospital, este tendrá la capacidad de emitir una solicitud de análisis de los pacientes que se presentan, así como recepcionar dicha solicitud cuando sea emitida por el médico.

El sistema debe permitir la devolución de resultado de un análisis y la búsqueda de un análisis específico y de una solicitud, para ello debe facilitar toda la información necesaria relacionada con los pacientes y los diferentes análisis se realicen.

También permitirá generar informes de estadísticas, control de los reactivos y medios. Los datos necesarios son almacenados en una base de datos, esta información estará disponible durante el tiempo que se crea conveniente para su uso posterior.

### **2.1.5 Modelo del Negocio**

El modelo del negocio describe los procesos de un negocio y su interacción con elementos internos, tales como socios y clientes, es decir, describe las funciones que el negocio pretende realizar. Comprende la descripción de los actores y trabajadores que intervienen en el negocio. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados y los trabajadores son los que actúan directamente dentro del negocio.

Los procesos que se llevan a cabo son representados por los casos de uso. Las entidades de negocio, representan a los objetos que los trabajadores toman, inspeccionan, manipulan, producen o utilizan durante la realización de los casos de uso de negocio. Para una mejor comprensión se describe a continuación los procesos del negocio del sistema de información de laboratorios.



### **Descripción general de los procesos del negocio**

Los procesos del negocio se refieren al flujo de información en los laboratorios clínicos de los hospitales del país.

El Proceso comienza cuando el doctor en su consulta decide hacer un examen de laboratorio. Este realiza una solicitud de análisis en la que incluye datos del Paciente (nombre, HC, etc.) luego escoge los exámenes a realizar, y de ser necesario puede que también el tipo de muestra a tomar, la fecha y hora, su firma y tal vez algún otro comentario adicional sobre la solicitud en general o algún análisis en particular.

El paciente se presenta en el laboratorio para la toma de muestra, la recepcionista recoge la fecha y hora en la que se presenta el paciente, los datos del mismo, si el doctor no escogió el tipo de la muestra ella se encarga de escogerla, luego se procede a tomar la muestra, este proceso puede ser de dos formas: el paciente trae la muestra ya tomada o que haya que tomársela. En todo caso los datos que se toman de la muestra son los mismos.

Si la muestra no va a ser procesada al instante, decir la razón, el tiempo que estará guardada, y especificar las condiciones en que se guardará en caso de ser necesario. Luego de esto la muestra pasa al técnico, que se encarga de analizarla; el técnico recibe una serie de solicitudes ordenadas y con un número que las identifica. Dichas solicitudes serán realizadas en dependencia de la prioridad de las mismas pero teniendo en cuenta la cola de solicitudes que se tengan.

El técnico llega a un resultado del análisis, lo emite, o sea lo deja plasmado en la solicitud de análisis y luego el responsable de área lo libera, en caso de laboratorios pequeños el propio técnico puede liberarlos. El Proceso de liberación consiste en lo siguiente: si ya se terminó de analizar la muestra definitivamente liberarlo y eliminar la muestra en caso de que no se le vaya a realizar otro análisis, si luego de tener el resultado el responsable de liberarlos considera que es necesario realizar otros análisis adicionales los escoge, procede a asignarlos a la persona correspondiente. Si para realizar estos análisis es necesario que se le tomen muestras al paciente nuevamente se procede a citarlo.

En caso de que el responsable considere que la muestra no se debe liberar por alguna razón; o porque hubo malos procedimientos con la misma, decide realizar el análisis nuevamente y se procede a volver a tomar la muestra al paciente. Si le parece que el resultado del examen no es correcto procede a realizar el análisis nuevamente (con la misma muestra de ser posible si no se solicita una nueva muestra).

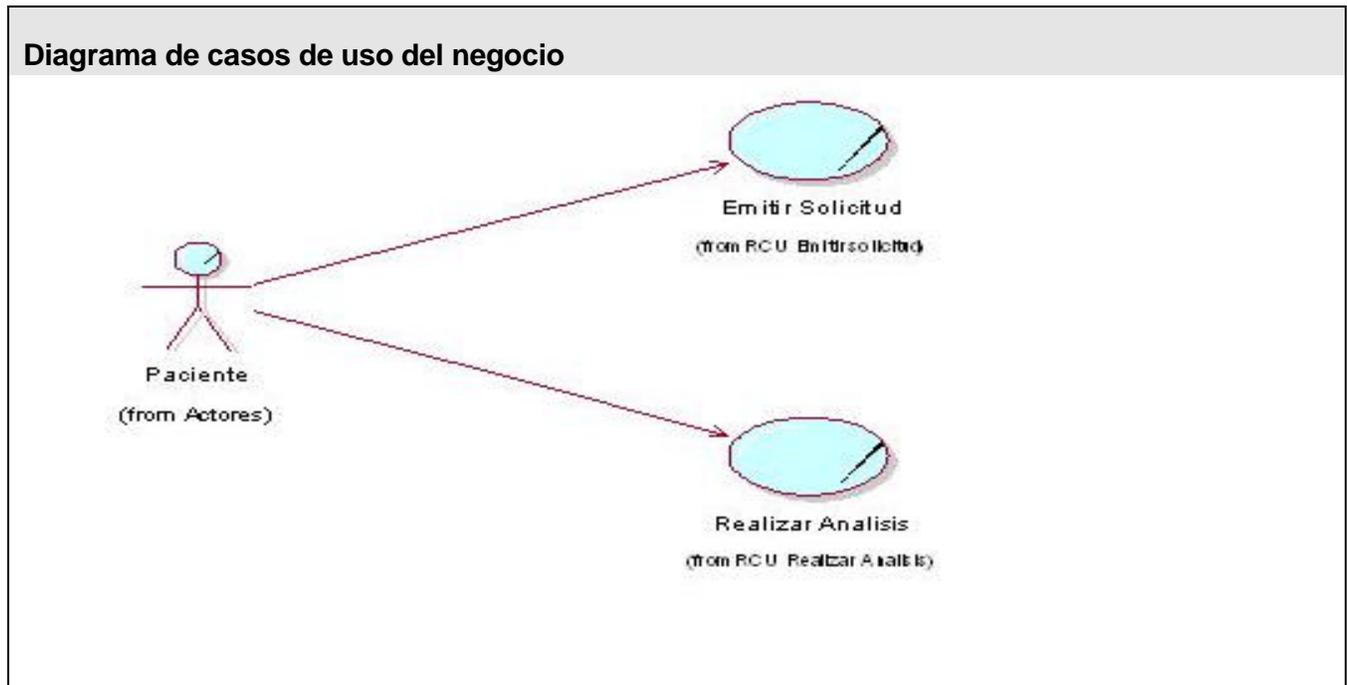
**Actores del Negocio**

<b>Actores del negocio</b>	<b>Justificación</b>
Paciente	Interviene en la emisión de la solicitud y en la remisión. Se presenta con el médico y luego en la recepción cuando el médico le orienta exámenes. Es el que recibe los servicios que se prestan en los laboratorios.

**Trabajadores del Negocio**

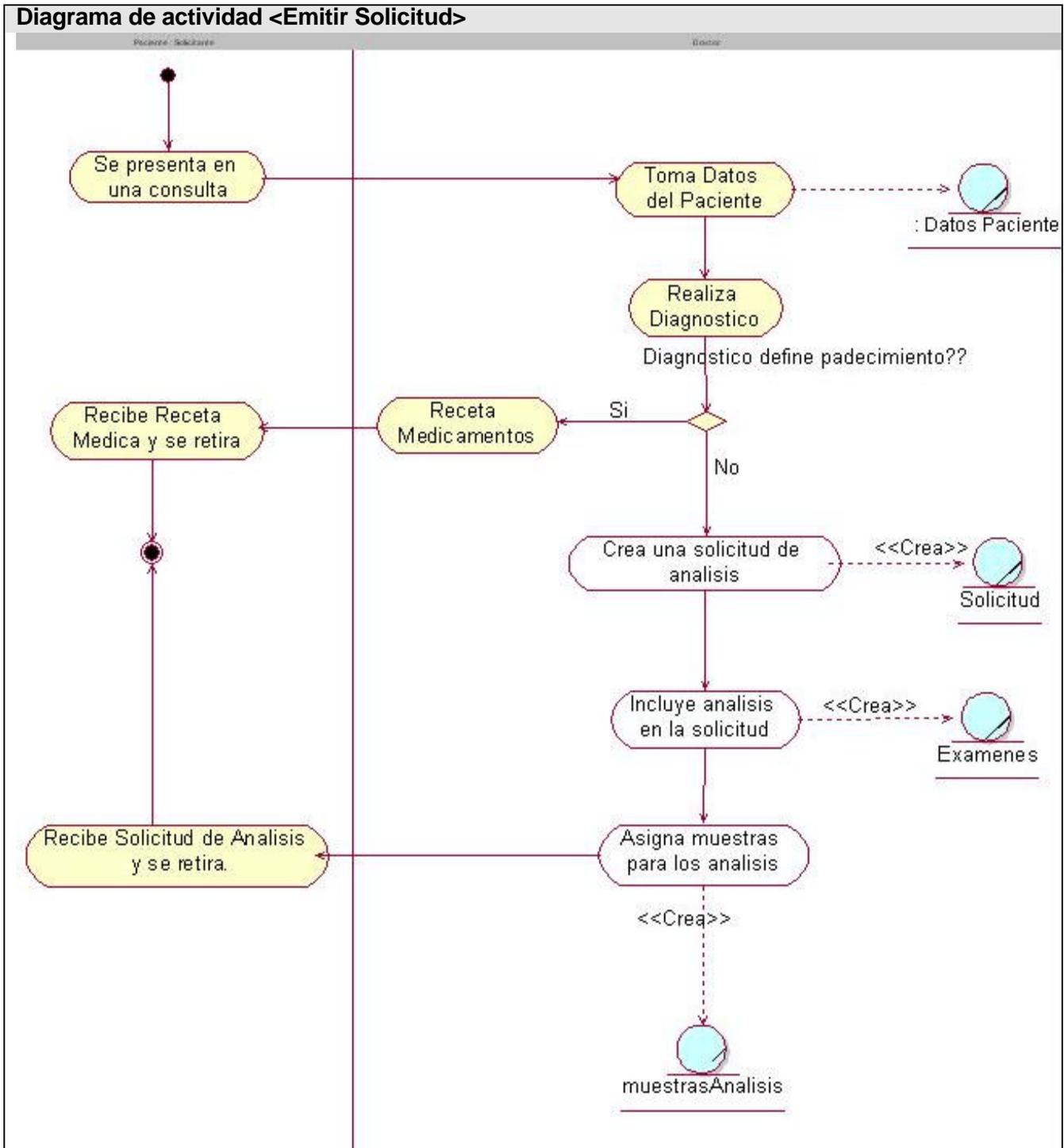
<b>Trabajadores del negocio</b>	<b>Justificación</b>
Técnico	Analiza las muestras de análisis y emite los resultados de los mismos.
Recepcionista	Es el encargado de recepcionar la solicitud de análisis del paciente que se presente al laboratorio.
Jefe de área	Libera los resultados e incluye más análisis en una solicitud.
Doctor	Es el encargado de emitir una solicitud de análisis.
Encargado de muestras	Se encarga de tomar las muestras y del almacenaje en caso que sea necesario.

## Diagrama de Casos de Uso del Negocio.



## Diagramas de Actividades del Negocio.

El diagrama de actividades es un artefacto generado en el modelo del negocio y ayudan a describir el detalle de que es lo que pasa dentro del negocio, además ayuda a identificar qué funciones deberá asumir el producto de software.





**Descripción textual de los Casos de Uso del Negocio.**

<b>Caso de uso del negocio</b>		<b>Emitir Solicitud de Análisis</b>
Actores del negocio		Paciente(inicia)
Propósito		Obtener orden de análisis.
Resumen		El Caso de Uso inicia cuando el Paciente se presenta en la consulta del Médico. El Médico registra los datos generales del Paciente, realiza el diagnóstico del mismo y de ser necesario entrega orden de análisis. El caso de uso termina cuando el Paciente se retira.
<b>Acción del actor</b>		<b>Respuesta del Sistema</b>
1- El Paciente se presenta en la consulta. 3-El Paciente informa sus datos generales. 7- El Paciente recibe orden de análisis y se retira.		2 - El Médico solicita datos del Paciente. 4 - El Médico anota datos del Paciente en el registro. 5- El Médico realiza diagnostico al paciente. 6- El Médico en dependencia de los síntomas y el diagnostico entrega la orden de análisis.
<b>Cursos Alternos de los eventos.</b>		
<b>Acción del actor</b>		<b>Respuesta del Sistema</b>
7. El paciente recibe receta y se retira.		6.1 - El Médico receta medicamentos
<b>Prioridad</b>	Crítica.	
<b>Mejoras</b>	Se agiliza la gestión de la información de un Paciente cuando acude a una consulta.	



<b>Caso de uso del negocio</b>	<b>Realizar Análisis</b>
Actores del negocio	Paciente(inicia)
Propósito	Realizar el análisis correspondiente.
Resumen	El Caso de Uso inicia cuando el Paciente se presenta en el laboratorio para que se le tome la muestra. El encargado de muestra solicita la orden de análisis, el paciente la entrega y el encargado de muestra procede a tomar la muestra. Luego, si la muestra necesita incubación le pone los datos de incubación, sino la envía al laboratorio para su posterior análisis. El técnico la analiza y emite un resultado.
<b>Acción del actor</b>	<b>Respuesta del proceso de negocio</b>
1- El Paciente se presenta en el laboratorio.	2 – La recepcionista toma la solicitud de análisis y verifica que este bien formulada. 3- Ordena tomar muestra al paciente. 4 - El encargado de muestra toma la muestra y anota los datos de la misma. 5- El encargado de muestra pasa la muestra al laboratorio. 6- El técnico analiza la muestra y pasa los resultados el jefe de área. 7- El jefe de área recibe a analiza los resultados. 8- El jefe de área libera el resultado del análisis.
<b>Cursos Alternos de los eventos.</b>	
<b>Acción del actor</b>	<b>Respuesta del proceso de negocio</b>
4- El paciente toma la solicitud y se	2.1 – La recepcionista devuelve la solicitud al paciente porque no esta bien formulada. 3- Indica al paciente como proceder.

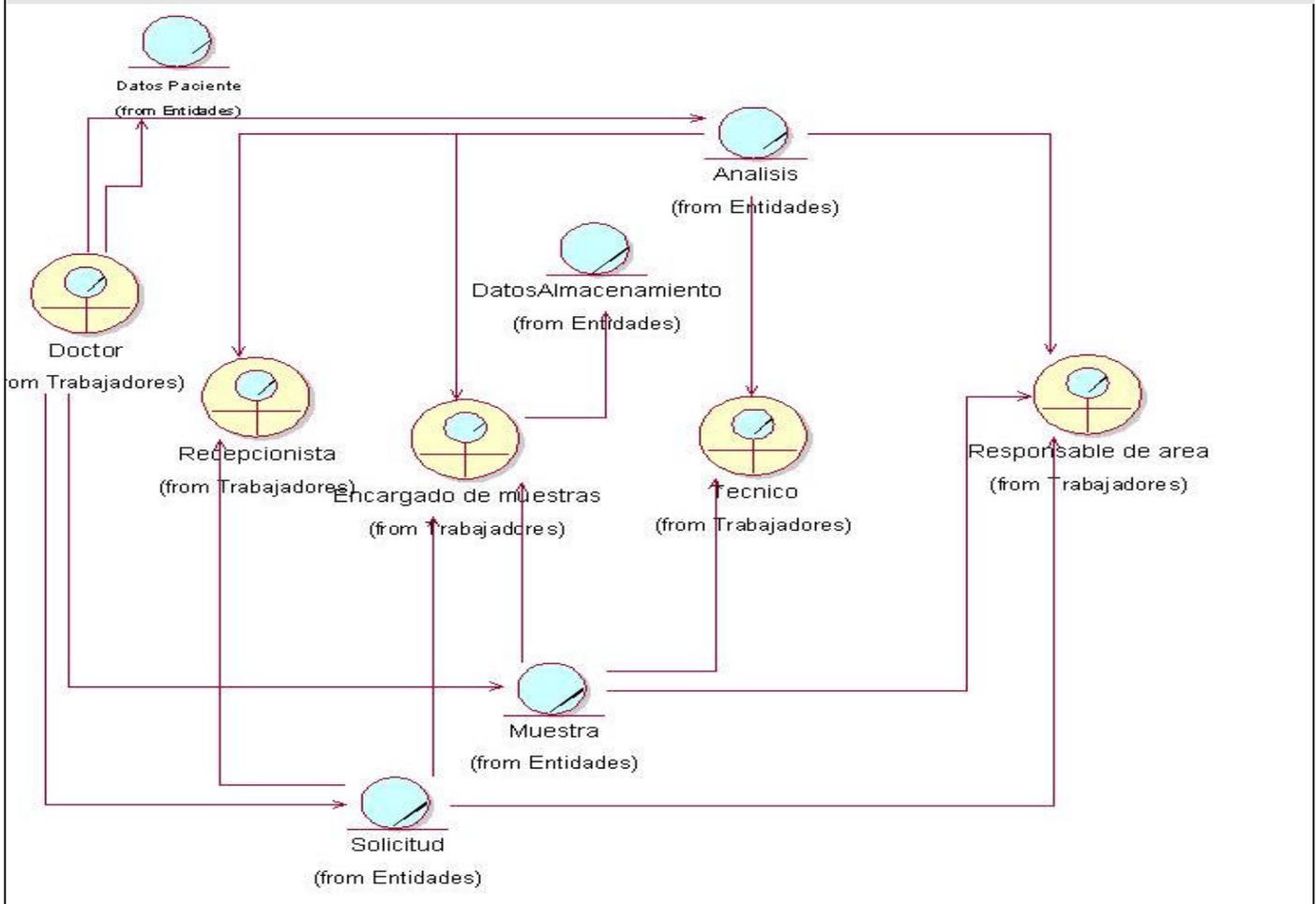


retira.	
<b>Cursos Alternos de los eventos.</b>	
<b>Acción del actor</b>	<b>Respuesta del proceso de negocio</b>
	5.1- Pone datos de almacenaje a la muestra. 6- Pasa la muestra al laboratorio.
<b>Cursos Alternos de los eventos.</b>	
<b>Acción del actor</b>	<b>Respuesta del proceso de negocio</b>
	8.1- El jefe de área agrega análisis a la solicitud. 9- El jefe de área libera el resultado del análisis.
<b>Cursos Alternos de los eventos.</b>	
<b>Acción del actor</b>	<b>Respuesta del proceso de negocio</b>
	8.2- El jefe de área devuelve el análisis porque no esta listo para ser liberado. 9-El jefe de área indica analizar la muestra nuevamente.
<b>Cursos Alternos de los eventos.</b>	
<b>Acción del actor</b>	<b>Respuesta del proceso de negocio</b>
	9.1- El jefe de área indica citar al paciente nuevamente.
<b>Prioridad</b>	<b>Crítico</b>
<b>Mejoras</b>	Se agiliza la gestión de la información de un Paciente a la hora de dar un resultado, se evita la posible pérdida de la emisión o la orden de análisis.

## Modelo de Objetos

El modelo de objetos describe como colaboran los trabajadores y entidades del negocio dentro del flujo de trabajo del proceso del negocio.

### GRAMA DE CLASES DEL MODELO DE OBJETOS



## 2.2.6 Especificación de los requisitos de software.

### Dependencias y Relaciones con otro software.

El módulo a implementar pertenece al sistema GeHos (Gestión Hospitalaria), y está integrado con otros módulos de este sistema como son el módulo de Configuración para gestionar los nomencladores que necesita el sistema. Con el módulo de Inscripción Admisión para gestionar los datos de los pacientes y



trabajadores. Los otros casos de integración que tiene con los demás módulos son servicios que se le prestan a estos, como por ejemplo el modulo de Cuerpo de Guardia, Banco de Sangre y Bloque Quirúrgico, además en un futuro se espera hacer pedidos al módulo de Farmacia.

### **Requisitos Funcionales:**

Los Requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir.

RF1 Emitir Solicitud

RF2 Recepcionar solicitud

RF3 Tomar Datos de Muestra

RF4 Poner datos de almacenaje.

RF5 Emitir resultados

RF6 Liberar Resultado

RF7 Incluir mas análisis en una solicitud.

RF8 Buscar Solicitud

RF9 Buscar Análisis

RF10 Emitir Reporte de Solicitud

RF11 Modificar Resultado

RF12 Control de materiales del laboratorio

### **Requisitos no Funcionales**

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Estas propiedades serian las características que hacen al producto atractivo, usable, rápido y confiable.

#### **Requerimientos de apariencia o interfaz externa**

El sistema debe tener un ambiente simple y entendible para los usuarios finales, de forma tal que no les sea muy complicado utilizar el software, pero a la vez se necesita tener todas las funcionalidades que brindan los laboratorios en la actualidad.

No debe utilizarse tecnología de *frames*.

Cada página no debe exceder los 500 Kb en imágenes.

#### **Requerimientos de usabilidad**

La aplicación debe cumplir con los principales principios de usabilidad, debe brindarse comodidad a la hora de acceder a las diferentes funcionalidades que brinda la aplicación mediante teclas de acceso



rápido, la navegabilidad no debe ser muy compleja, todas las funcionalidades deben ser rápidamente accesibles por el usuario.

### **Requerimientos de seguridad y privacidad**

La información debe transmitirse de manera segura, se debe garantizar la seguridad a todos los niveles (Interfaz, negocio y Acceso a datos) restringiendo las funcionalidades mediante roles de usuarios garantizando que la información sea accesible al usuario autorizado.

### **Requerimientos de ayudas y documentación en línea**

El sistema debe contar con una ayuda que explique las diferentes funcionalidades con que cuenta el sistema de manera rápida, además los manuales de usuario y toda la documentación actualizada de cada módulo de la aplicación.

### **Requerimientos de hardware**

Requerimientos para una estación de trabajo: 256Mb RAM (Recomendado 512Mb), 1GHz, 10Gb HDD.

Requerimientos para un servidor: 512Mb RAM (Recomendado 1Gb RAM o superior), 1GHz o superior, 60Gb HDD

### **Requerimientos de software**

El sistema debe correr en sistemas operativos Windows 98 o superior y sistemas Unix, Linux. Para sistemas Windows se deber tener instalado Microsoft Framework 2.0 y en sistemas Linux la plataforma Mono 1.2 o superior.

### **Restricciones en el diseño y la implementación**

El sistema será implementado utilizando como lenguaje de programación del lado del cliente java script auxiliándose de la metodología AJAX.

Se utilizará la plataforma de desarrollo Microsoft .NET que brinda una gama de facilidades en su entorno y que da la posibilidad de utilizar el lenguaje C# del lado del servidor.

El servidor debe contar con el Framework 2.0.

### **2.2.7 Definición de los casos de uso.**

**Definición de los actores.**

<b>Actores</b>	<b>Justificación</b>
Doctor	Interactúa con el sistema. Introduce los datos del paciente en el mismo para emitir una solicitud de análisis.
Recepcionista	Interactúa con el sistema. Se encarga de recepcionar la solicitud de análisis una vez emitida por el doctor. Gestiona la información de paciente cuando este se presenta en el laboratorio.
Encargado de muestra	Interactúa directamente con el sistema. Es el encargado de tomar los datos de la muestra y poner le datos de almacenaje en caso de ser necesario.
Técnicc	Interactúa directamente con el sistema. Se encarga de emitir un resultado una vez analizada la muestra. Además se encarga de buscar un análisis o modificar un resultado en caso que sea necesario.
Almacenero	Interactúa directamente con el sistema. Se encarga del control de materiales del laboratorio, el inventario de materias primas.
Usuario	Es una generalización de los actores Técnico, Jefe de Área, Doctor, Encargado de Muestra y Recepcionista, que realizan en común la búsqueda de un análisis, una solicitud de análisis, y el reporte de una solicitud.
Jefe de área	Interactúa directamente con el sistema. Se encarga de la liberar un análisis y de incluir mas análisis en una solicitud (de ser necesario) una vez emitido un resultado.

**Listado de casos de uso.**

<b>Caso de uso: Emitir Solicitud (CU-1)</b>
<b>Actores:</b> Doctor
<b>Descripción:</b> El caso de uso inicia cuando el paciente se presenta con el doctor y este decide emitir una solicitud de análisis. El doctor busca el paciente al cual le desea realizar los análisis, selecciona los exámenes a realizar y luego emite la solicitud de análisis.
<b>Referencias:</b> RF1

<b>Caso de uso: Recepcionar Solicitud de análisis (CU-2)</b>
<b>Actores:</b> Recepcionista
<b>Descripción:</b> El caso de uso inicia cuando el paciente se presenta en el laboratorio (primeramente en la recepción) para la realización del análisis emitida por el doctor. La recepcionista verifica que la solicitud haya sido emitida, o sea que busca esta solicitud y una vez encontrada, selecciona el encargado de muestra que debe atender al paciente y le envía la solicitud.
<b>Referencias:</b> RF2

<b>Caso de uso: Tomar Datos de Muestra. (CU-3)</b>
<b>Actores:</b> Encargado de Muestra
<b>Descripción:</b> El caso de uso inicia cuando el paciente, una vez llegado al laboratorio, se presenta con el encargado de muestra para que se le tome la muestra. El encargado de muestra anota los datos de dicha muestra y procede a tomársela.
<b>Referencias:</b> RF3

<b>Caso de uso: Registrar datos de almacenaje (CU-4)</b>
<b>Actores:</b> Encargado de muestra



<b>Descripción:</b> El caso de uso inicia cuando se le toma la muestra al paciente, si el análisis necesita incubación o almacenarlo de alguna otra manera, o sea que no pasara a ser analizado inmediatamente, el encargado de muestra le pone estos datos de almacenaje.
<b>Referencias:</b> RF4

<b>Caso de uso: Emitir resultados (CU-5)</b>
<b>Actores:</b> Técnico.
<b>Descripción:</b> El caso de uso inicia cuando se analiza la muestra del paciente. El técnico después de analizar la muestra emite un resultado. Dicho resultado se guarda y no es liberado hasta que se verifique que no es erróneo.
<b>Referencias:</b> RF5

<b>Caso de uso: Buscar Solicitud (CU-6)</b>
<b>Actores:</b> Usuario
<b>Descripción:</b> Este caso de uso inicia cuando el usuario necesita buscar una solicitud de análisis de un paciente por cualquier motivo. El caso de uso finaliza cuando se encuentra dicha solicitud.
<b>Referencias:</b> RF8

<b>Caso de uso: Buscar Análisis (CU-7)</b>
<b>Actores:</b> Usuario
<b>Descripción:</b> El caso de uso inicia cuando el usuario necesita buscar un análisis para emitir un resultado una vez analizada la muestra. El caso de uso finaliza cuando se encuentra dicho análisis.
<b>Referencias:</b> RF9

<b>Caso de uso: Liberar Resultado (CU-8)</b>
<b>Actores:</b> Jefe de área
<b>Descripción:</b> El caso de uso inicia cuando el jefe de área verifica que el resultado del análisis no sea erróneo. De no ser erróneo los libera para que puedan tener acceso a dicho resultado todo el personal



autorizado.
<b>Referencias:</b> RF6

<b>Caso de uso: Incluir mas análisis en la solicitud (CU-9)</b>
<b>Actores:</b> Jefe de área
<b>Descripción:</b> El caso de uso inicia cuando el jefe de área libera un resultado de análisis, si es necesario realizar otro análisis lo incluye especificando que análisis se va a realizar.
<b>Referencias:</b> RF7

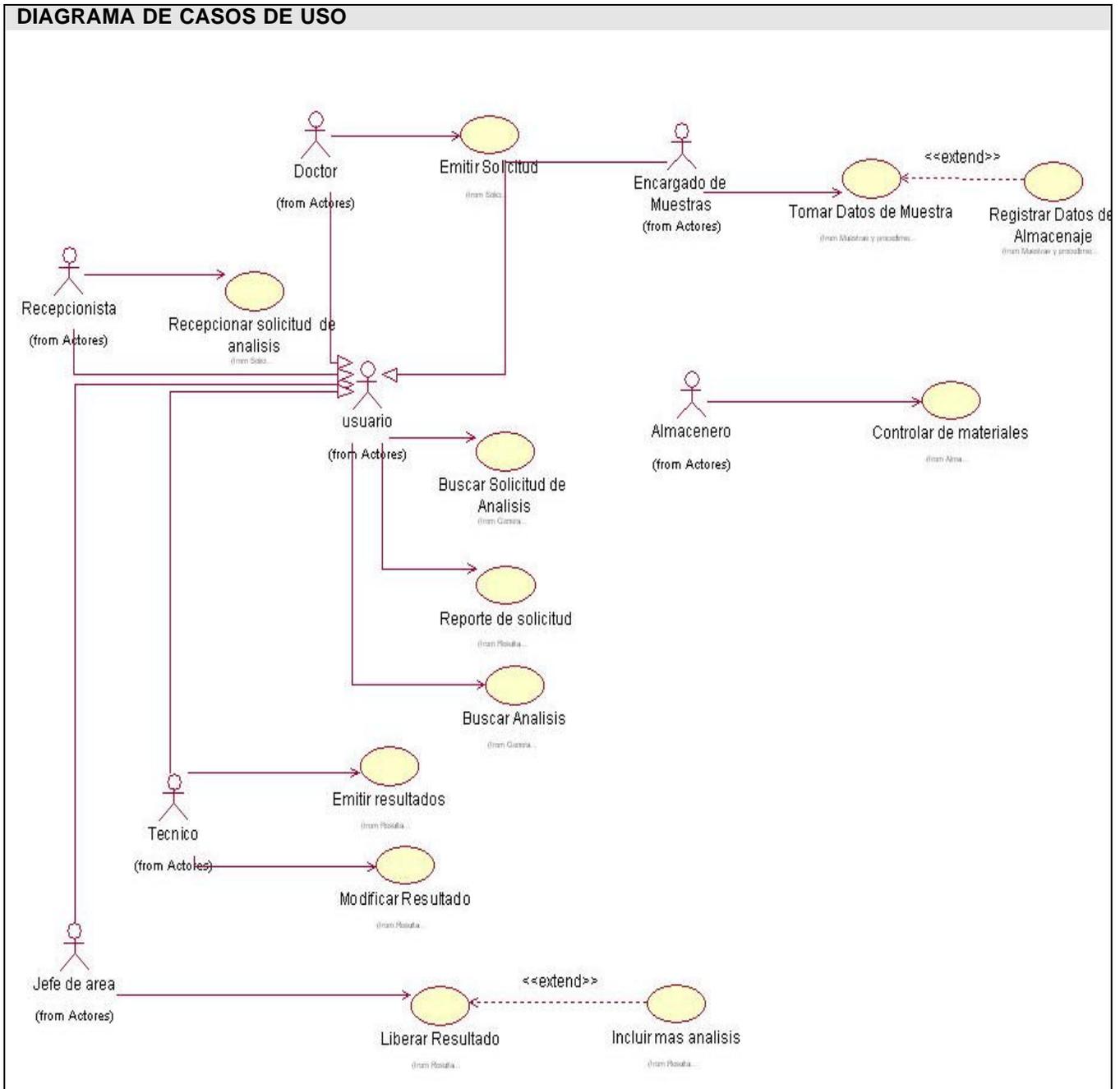
<b>Caso de uso: Reporte de Solicitud (CU-10)</b>
<b>Actores:</b> Usuario
<b>Descripción:</b> Este caso de uso inicia cuando el usuario necesita consultar los resultados de una solicitud de análisis de un paciente determinado.
<b>Referencias:</b> RF10

<b>Caso de uso: Modificar Resultado (CU-11)</b>
<b>Actores:</b> Técnico
<b>Descripción:</b> Este caso de uso inicia cuando el técnico necesita modificar un resultado.
<b>Referencias:</b> RF11

<b>Caso de uso: Controlar materiales del laboratorio. (CU-12)</b>
<b>Actores:</b> Almacenero
<b>Descripción:</b> El caso de uso se inicia cuando el almacenero necesita controlar los materiales que salen del almacén, para llevar un control estadístico sobre dichos materiales.
<b>Referencias:</b> RF12



### Diagrama de Casos de Uso.



**Casos de uso por ciclo.**

<b>Código</b>	<b>Nombre de caso de uso</b>	<b>Justificación de la selección.</b>
CU – 1	Emitir Solicitud	Este caso de uso es importante porque es donde se selecciona al paciente al que se le realizará el análisis, y el doctor emite la solicitud de análisis para que el paciente pueda recibir los servicios del laboratorio. Este caso de uso debe desarrollarse en el primer ciclo de desarrollo.
CU – 2	Recepcionar Solicitud de análisis	Luego de emitida la solicitud de análisis, pasa a la recepción del laboratorio. Cuando el paciente llega a este para ser atendido, la recepcionista busca la emisión de su solicitud de análisis, en caso de que haya faltado algún dato lo especifica y le indica al paciente como proceder y envía esta solicitud al encargado de muestra seleccionado para atender al paciente. Este caso de uso debe desarrollarse en el primer ciclo de desarrollo.
CU – 3	Tomar Datos de Muestra	Una vez recepcionada la solicitud de análisis del paciente, este se presenta con el encargado de muestra para que se le tome la muestra. El encargado de muestra anota los datos de necesarios, y procede a tomársela. Este caso de uso debe desarrollarse en el primer ciclo de desarrollo.
CU – 4	Registrar Datos de Almacenaje	Cuando se tiene la muestra del paciente, si la muestra se va a incubar o almacenar es necesario poner los datos de almacenaje. Este caso de uso debe desarrollarse en el primer ciclo de desarrollo.
CU – 5	Emitir Resultado	Después de tomada la muestra, se analiza para emitir un resultado. Este caso de uso debe desarrollarse en el primer ciclo de desarrollo.
CU – 6	Buscar Solicitud de Análisis	Este caso de uso es necesario para la realización del caso de uso Recepcionar Solicitud y Emitir Solicitud de análisis.



		Debido a esto se decidió desarrollarlo en el segundo ciclo de desarrollo.
CU – 7	Buscar Análisis	Este caso de uso es necesario para la realización del caso de uso Emitir Resultado y Liberar Análisis. Debido a esto se decidió desarrollarlo en el segundo ciclo de desarrollo.
CU – 8	Liberar Análisis	Después de emitir un resultado, es necesario liberarlo para que el doctor tenga esta información y pueda diagnosticar al paciente y en caso de ser necesario darle un tratamiento a seguir. Este caso de uso se desarrollara en el primer ciclo de desarrollo.
CU – 9	Incluir más Análisis en una solicitud.	Cuando se libera el análisis, en ocasiones es necesario volver a realizar el análisis para verificar el resultado o realizar algún dro. Este caso de uso debe desarrollarse en el segundo ciclo de desarrollo.
CU – 10	Reporte de Solicitud	Este caso de uso es necesario para consultar los resultados de los exámenes de una solicitud de análisis. Debe desarrollarse en el segundo ciclo de desarrollo.
CU – 10	Modificar Resultado	Este caso de uso es necesario para modificar un resultado. Debe desarrollarse en el segundo ciclo de desarrollo.
CU – 11	Control de Materiales.	Este caso de uso es necesario para tener un control estricto sobre los materiales que se mueven en el laboratorio, o sea para llevar una estadística de estos materiales. Se debe desarrollar en el segundo ciclo desarrollo.

**Casos de uso expandidos.**

Los casos de uso expandidos son de gran importancia, ya que permiten tener una adecuada visión de cómo deben realizarse dichos casos de uso, de esta forma se refleja el flujo de actividades que tiene lugar en cada uno de ellos.

<b>Caso de uso</b>	
CU-1	Emitir Solicitud
<b>Propósito</b>	Emitir una solicitud de análisis.
<b>Actores :</b> Doctor	
<b>Resumen:</b> El caso de uso inicia cuando el paciente se presenta con el doctor y este decide emitir una solicitud de análisis. El doctor le toma todos los datos necesarios y luego emite la solicitud de análisis.	
<b>Referencias</b>	RF1
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1- El doctor entra al sistema, específicamente a la sección de laboratorios clínicos.  3- El doctor selecciona los datos deseados.  4- Presiona el botón Aceptar.	2- El sistema muestra la página de emitir solicitud, donde el doctor debe buscar al paciente al que desea realizarle el análisis, además debe seleccionar los análisis a realizar.  5- El sistema verifica que los campos obligatorios no estén vacíos.  6- El sistema emite la solicitud.
<b>Flujo alternativo 1</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
6- El doctor llena los campos vacíos y presiona el botón Aceptar.	5.1- Si alguno de los campos obligatorios están vacíos, el sistema emite un mensaje para que llene dichos campos.
<b>Prioridad:</b> Crítico	

<b>Caso de uso</b>	
CU-2	Recepcionar Solicitud de análisis
<b>Propósito</b>	Recepcionar una solicitud de análisis, o sea cuando el paciente se presenta en el laboratorio para realizarse su análisis, la recepcionista verifica que se haya emitido su solicitud de análisis.
<b>Actores:</b> Recepcionista	



<b>Resumen:</b> El caso de uso inicia cuando el paciente se presenta en el laboratorio (primeramente en la recepción) para la realización del análisis emitida por el doctor. La recepcionista verifica que la solicitud haya sido emitida, o sea que busca esta solicitud y una vez encontrada enviaría esta solicitud al encargado de muestra que haya seleccionado.	
<b>Referencias</b>	RF2
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1- La recepcionista entra al sistema.  3- La recepcionista busca la emisión del paciente y presiona el botón aceptar.  5- La recepcionista selecciona un encargado de muestra y luego envía la solicitud de análisis al mismo.	2- El sistema muestra la bandeja de entrada de Recepcionar Solicitud, donde aparecen todos los análisis emitidos por el doctor.  4-El sistema muestra la solicitud de análisis del paciente seleccionado. 6- El sistema envía la solicitud.
<b>Flujo alternativo 1</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
3.1- La recepcionista busca la emisión del paciente, una vez encontrada recoge la muestra y selecciona la opción poner datos de muestra. 5- La recepcionista introduce los datos.	4.1- El sistema muestra una ventana con los datos que necesarios para la muestra.  6- El sistema guarda los datos
<b>Flujo alternativo 2</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
3.1- La recepcionista selecciona la opción Buscar Solicitud de Análisis.  5- La recepcionista anota los datos.	4- El sistema muestra una ventana con los datos necesarios para la búsqueda.  6- El sistema realiza la búsqueda.
<b>Prioridad:</b> Crítico.	

<b>Caso de uso</b>	
CU-3	Tomar Datos de Muestra
<b>Propósito</b>	Tomar los datos de la muestra como el nombre, el estado, si necesita ser almacenada.
<b>Actores:</b> Encargado de Muestra	



<b>Resumen:</b> El caso de uso inicia cuando el paciente, una vez llegado al laboratorio, se presenta con el encargado de muestra para que se le tome la muestra. El encargado de muestra anota los datos de dicha muestra y procede a tomársela.	
<b>Referencias</b>	RF3
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<p>1- El encargado de muestra selecciona en su bandeja de entrada la solicitud de análisis que vaya a analizar y presiona el botón aceptar.</p> <p>3-El encargado de muestra selecciona el examen que va a realizar y a ponerle los datos correspondiente y selecciona el botón tomar muestra</p> <p>5-El encargado de muestra selecciona la opción Pasar al Laboratorio, selecciona un estado de la muestra y presiona el botón Aceptar.</p>	<p>2- El sistema muestra una ventana con el nombre del paciente, la prioridad que tiene y los análisis a realizar.</p> <p>4- El sistema muestra los datos necesarios que deben tomársele a la muestra.</p> <p>6-El sistema guarda los datos introducidos.</p>
<b>Flujo alternativo 1</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
<p>5.1- El encargado de muestra selecciona la opción Almacenar.</p> <p>7- El encargado de muestra pone las condiciones de almacenaje y presiona el botón Aceptar.</p>	<p>6.1- El sistema activa el botón condiciones de almacenaje.</p> <p>6- El sistema guarda los datos introducidos.</p>
<b>Prioridad:</b> Crítico.	

<b>Caso de uso</b>	
CU-4	Registrar datos de almacenaje
<b>Propósito</b>	Poner los datos de almacenaje a una muestra que lo necesite.
<b>Actores:</b> Encargado de Muestra.	
<b>Resumen:</b> El caso de uso inicia cuando se le toma la muestra al paciente, si el análisis necesita incubación o almacenarlo de alguna otra manera, o sea que no pasara a ser analizado inmediatamente, el encargado de muestra le pone estos datos de almacenaje.	
<b>Referencias</b>	RF4
<b>Acción del actor</b>	<b>Respuesta del sistema</b>



1- El encargado de muestra necesita poner la muestra a almacenar. Selecciona el botón Condiciones de Almacenaje.	2- El sistema muestra una ventana con las condiciones de almacenaje.
3- El encargado de muestra selecciona las condiciones correspondientes.	
4- Presiona el botón aceptar.	
<b>Prioridad:</b> Crítico.	

Caso de uso	
CU-1	Emitir Resultado
<b>Propósito</b>	Emitir el resultado de un análisis después de analizarlo.
<b>Actores:</b> Técnico o Jefe de Área.	
<b>Resumen:</b> El caso de uso inicia cuando se analiza la muestra del paciente. El técnico o el jefe de área (ambos pueden analizar la muestra), después de analizar la muestra emiten un resultado. Dicho resultado se queda en el laboratorio hasta que se verifique que no es erróneo.	
<b>Referencias</b>	RF5
Acción del actor	Respuesta del sistema
1- El actor usuario decide emitir el resultado de un análisis, selecciona en la bandeja de entrada el análisis al cual le emitirá el resultado.	2- El sistema muestra una página con los datos del paciente, de la solicitud de análisis, de la muestra y los campos para ser llenados por el actor usuario con el resultado del análisis.
3-El actor introduce el resultado el botón Guardar.	4-El sistema guarda los datos.
<b>Prioridad:</b> Crítico.	

Caso de uso	
CU-6	Buscar Solicitud
<b>Propósito</b>	Buscar una solicitud.
<b>Actores:</b> Doctor, Recepcionista, Jefe de área.	
<b>Resumen:</b> Este caso de uso inicia cuando el doctor, recepcionista c Jefe de área necesitan buscar una solicitud de análisis de un paciente por cualquier motivo. El caso de uso finaliza cuando se encuentra dicha solicitud.	
<b>Referencias</b>	RF8
Acción del actor	Respuesta del sistema
4- El actor decide hacer la búsqueda de una solicitud.	2- El sistema le muestra la ventana con los datos que debe introducir.
4- El actor introduce los datos y presiona el	4- El sistema busca la solicitud y la muestra.



botón Buscar.	
<b>Flujo alternativo 1</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	4.1- El sistema no encuentra ninguna solicitud de análisis correspondiente a los datos introducidos.
<b>Prioridad:</b> Crítico	

<b>Caso de uso</b>	
CU-7	Buscar Análisis
<b>Propósito</b>	Buscar un análisis.
<b>Actores:</b> Técnico y Jefe de Área.	
<b>Resumen:</b> El caso de uso inicia cuando el técnico o el jefe de área necesitan buscar una análisis para emitir un resultado una vez analizada la muestra. El caso de uso finaliza cuando se encuentra dicho análisis.	
<b>Referencias</b>	RF9
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
4- El actor decide buscar un análisis.	2- El sistema muestra la ventana con los datos que debe introducir.
3- El actor introduce los datos y presiona el botón buscar.	4- El sistema busca el análisis y lo muestra.
<b>Flujo alternativo 1</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	4.1- El sistema no encuentra el análisis.
<b>Prioridad:</b> Crítico	

<b>Caso de uso</b>	
CU-8	Liberar Resultado
<b>Propósito</b>	Liberar un resultado después de verificar que no tenga errores.
<b>Actores:</b> Jefe de Área.	
<b>Resumen:</b> El caso de uso inicia cuando el jefe de área verifica que el resultado del análisis no sea erróneo. De no ser erróneo los libera. El caso de uso finaliza cuando este resultado le llega al doctor que emitió la solicitud y a la recepcionista del laboratorio	
<b>Referencias</b>	RF6
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1- El jefe de área entra al sistema para liberar un	2- El sistema muestra una bandeja de entrada con



análisis. 3- El jefe de área selecciona el análisis a liberar. 5- El jefe de área revisa los datos, de no tener error presiona el botón Liberar.	los análisis pendientes de liberación. 4- El sistema muestra una ventana con datos de la solicitud de análisis, datos de la muestra, del paciente y el resultado de dicho análisis. 6- El sistema libera el este resultado.
<b>Flujo alternativo 1</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
5.1- El jefe de área presiona el botón No Válido. 7- El jefe de área selecciona la opción Volver a realizar análisis y presiona el botón enviar.	6.1- El sistema muestra una ventana para especificar que debe hacerse. La ventana muestra las siguientes opciones: - Volver a realizar análisis. - Volver a analizar muestra - Realizar otro análisis. 8- El sistema envía una notificación a la recepción.
<b>Flujo alternativo 2</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
7.1- El jefe de área selecciona la opción Volver a analizar muestra.	8.1- El sistema envía una notificación al técnico.
<b>Flujo alternativo 3</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
7.1- El jefe de área selecciona la opción Realizar otro análisis.	8.1- El sistema envía una notificación a al Recepción.
<b>Prioridad:</b> Crítico	



### **Conclusiones**

En este capítulo, después de efectuarse un estudio sobre la situación actual de los laboratorios, se realizó la modelación del negocio, definiéndose actores, trabajadores y procesos. Se hizo una investigación para determinar los requerimientos con los que debe contar el sistema a desarrollar, obteniéndose así las funcionalidades que debe tener el mismo. También se hace referencia en este capítulo a propiedades o cualidades que el producto debe tener, o sea los requisitos no funcionales.



## Capítulo 3. Análisis y diseño del sistema.

En el siguiente capítulo se muestra el análisis y diseño del sistema, o sea se traducen los requisitos a una especificación que describe como implementar el sistema o sea los diagramas de clases de análisis correspondiente a cada caso de uso, así como los diagramas de interacción y los diagramas de clases. También se muestra en este capítulo el diseño de la Base de Datos.

### 3.1 Análisis del Sistema.

El objetivo de este flujo de trabajo es traducir los requisitos a una especificación que describe cómo implementar el sistema.

El análisis consiste en obtener una visión del sistema que se preocupa de ver que hace, de modo que solo se interesa por los requisitos funcionales.

#### 3.1.1 Modelo de análisis.

En el modelo de análisis se identifican las clases que describen la realización de los casos de uso, los atributos y las relaciones entre ellas. Con esta información se construye el Diagrama de clases del Análisis. El modelo de análisis es el resultado de la actividad de analizar los Casos de Uso, contiene clases del análisis y sus objetos organizados en paquetes.

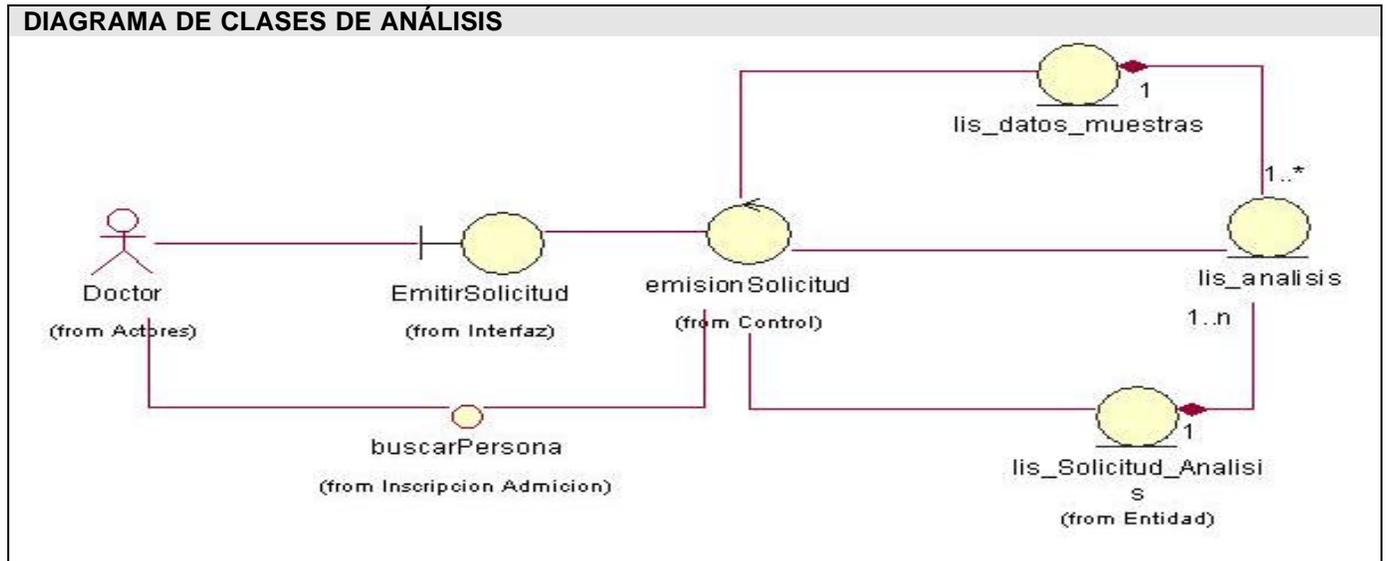
Las clases del análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones de dominio. Tienen atributos y entre ellas se establecen relaciones de asociación, agregación/composición, generalización/especialización y tipos asociativos. Siempre encajan en tres atributos básicos.

- Interfaz: Modelan la interacción entre el sistema y los actores.
- Entidad: Modelan la información del sistema.
- Control: Coordinan el trabajo de las clases.

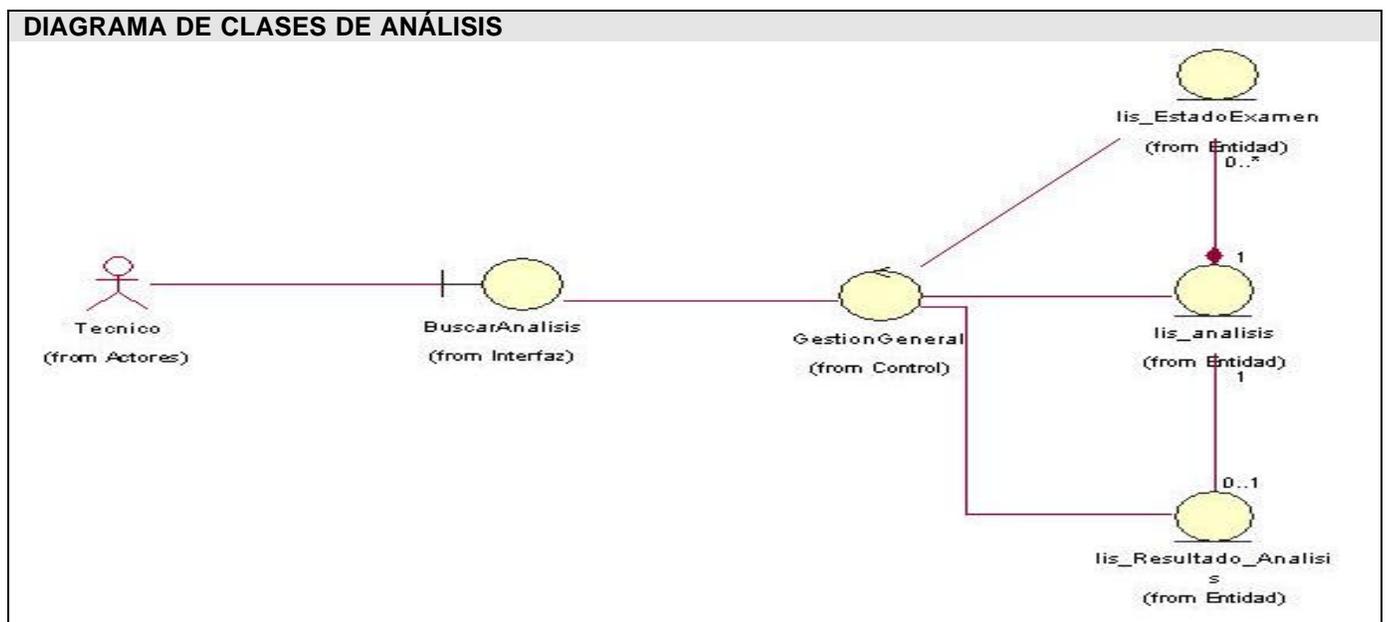


### Diagramas de Clases de Análisis.

#### CU Emitir Solicitud

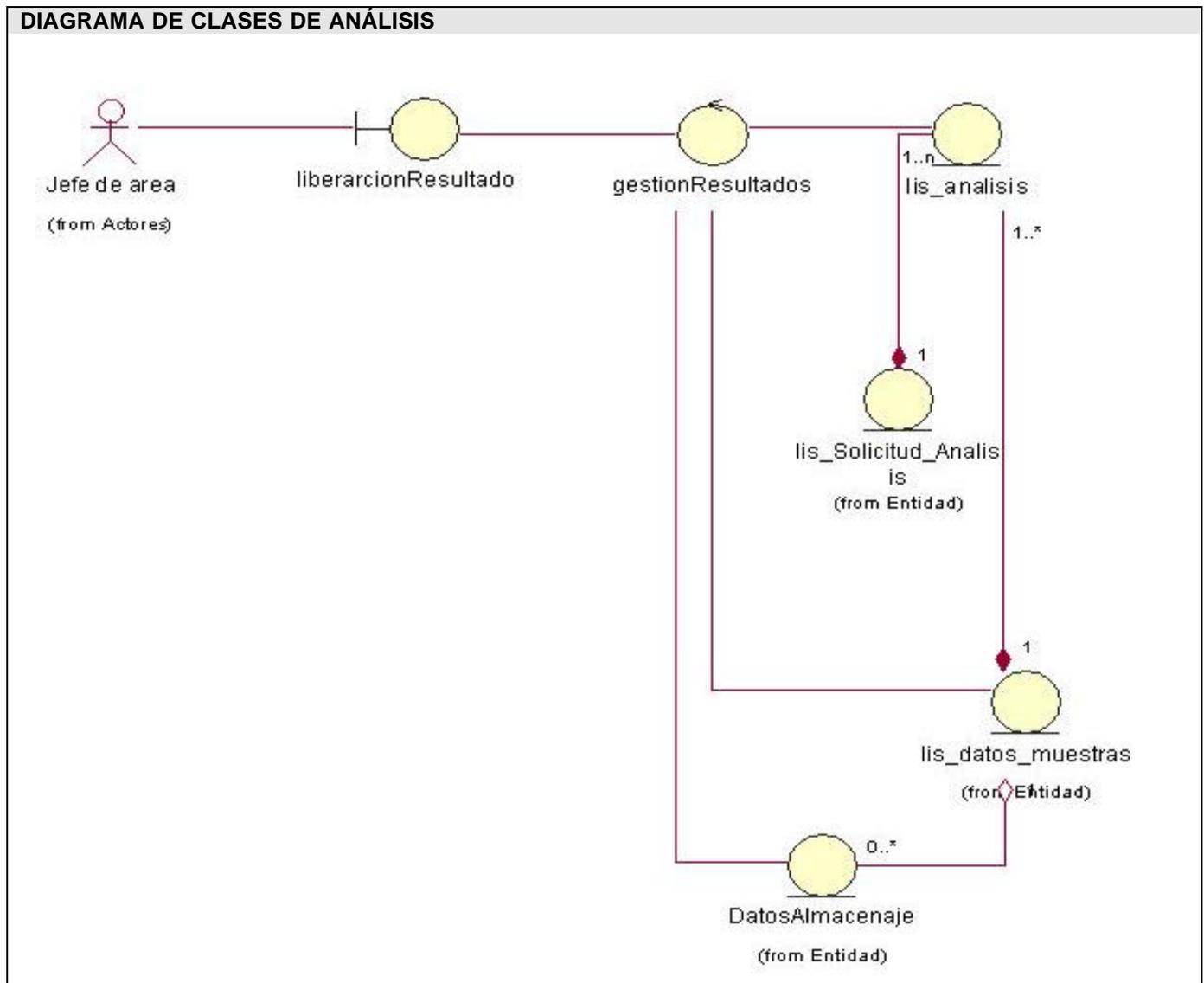


#### CU Buscar Análisis



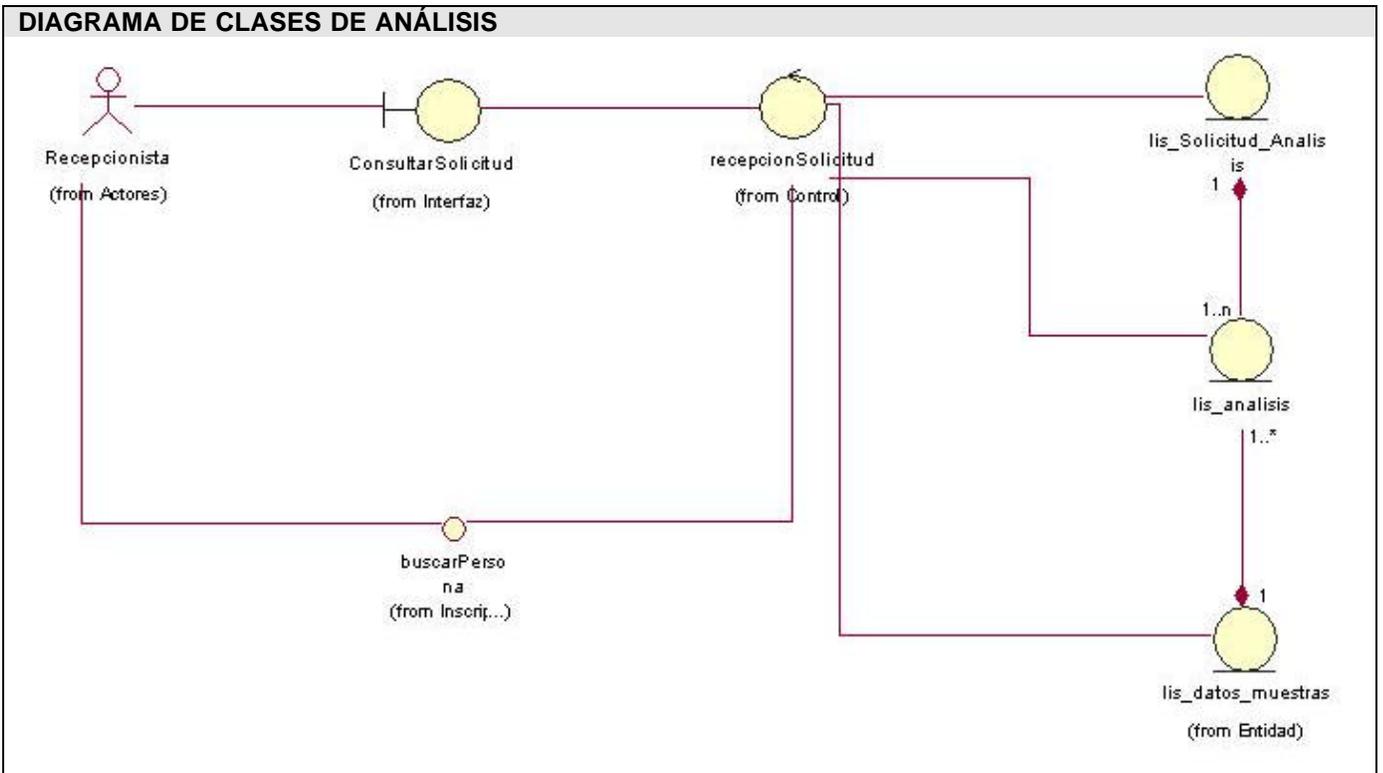


### CU Liberar Resultado

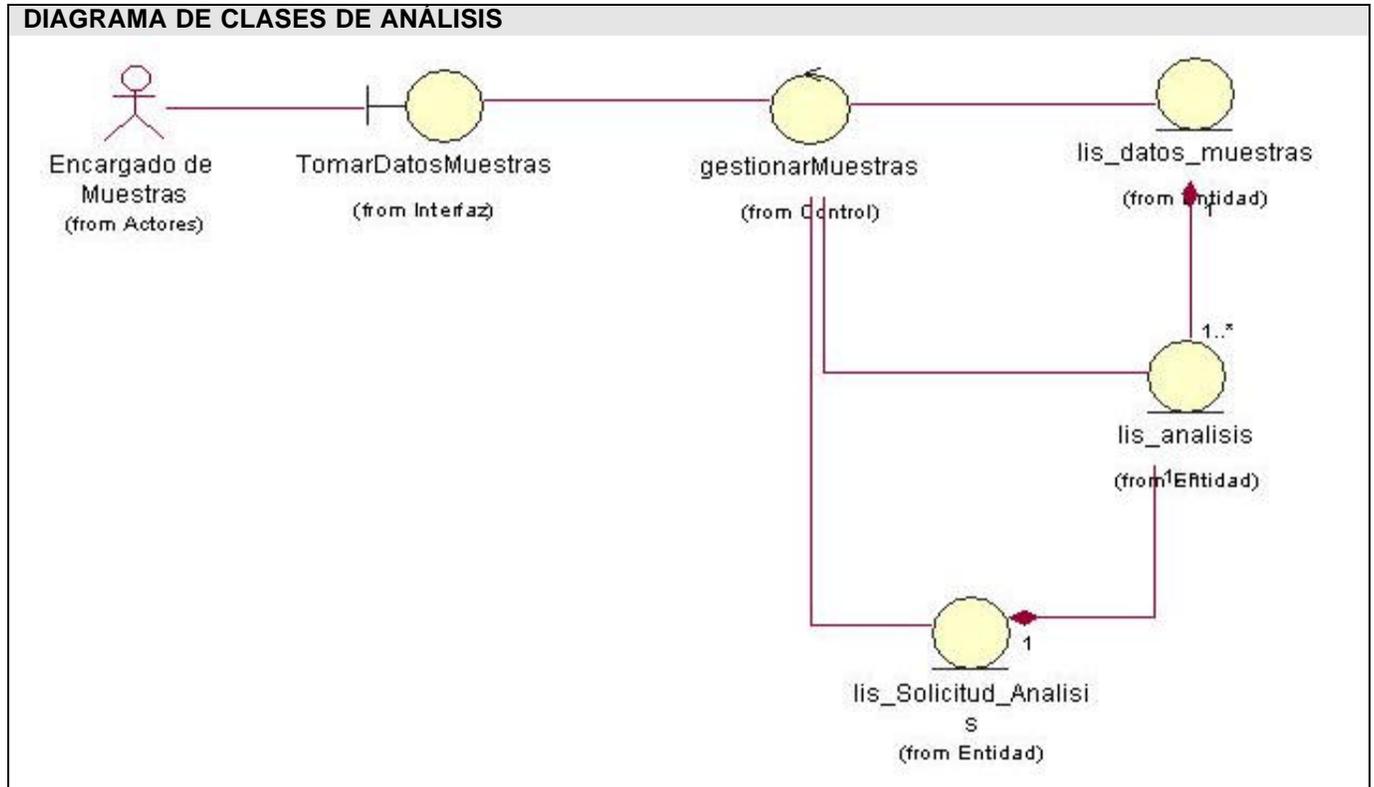
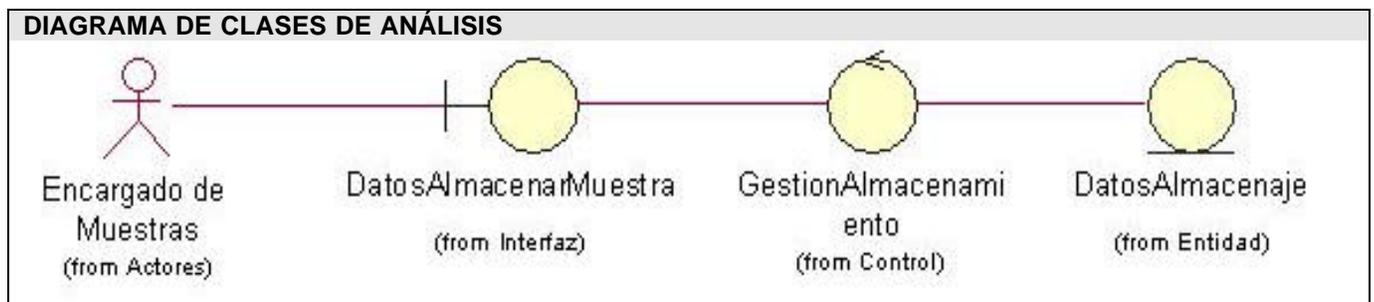




### CU Recepcionar Solicitud





**CU Tomar Datos de Muestra**

**Almacenar Muestra**




## **3.2 Diseño**

El diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva como cumple el sistema sus objetivos.

### **3.2.1 Diagramas de interacción**

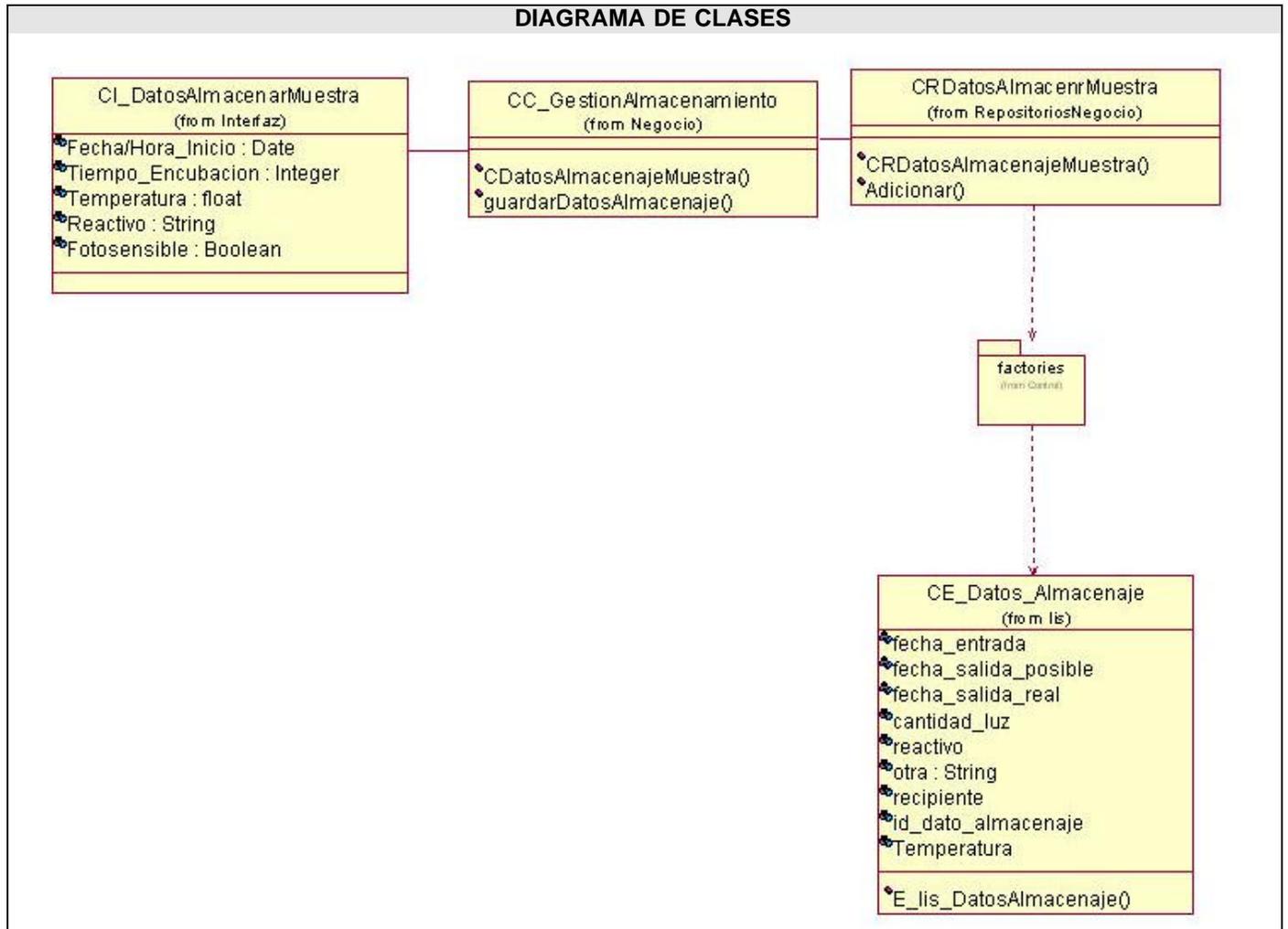
Los diagramas de interacción muestran las interacciones entre objetos mediante transferencia de mensajes entre objetos o subsistemas. Modelan aspectos dinámicos del sistema. Se encuentran referidos en el Anexo #1.

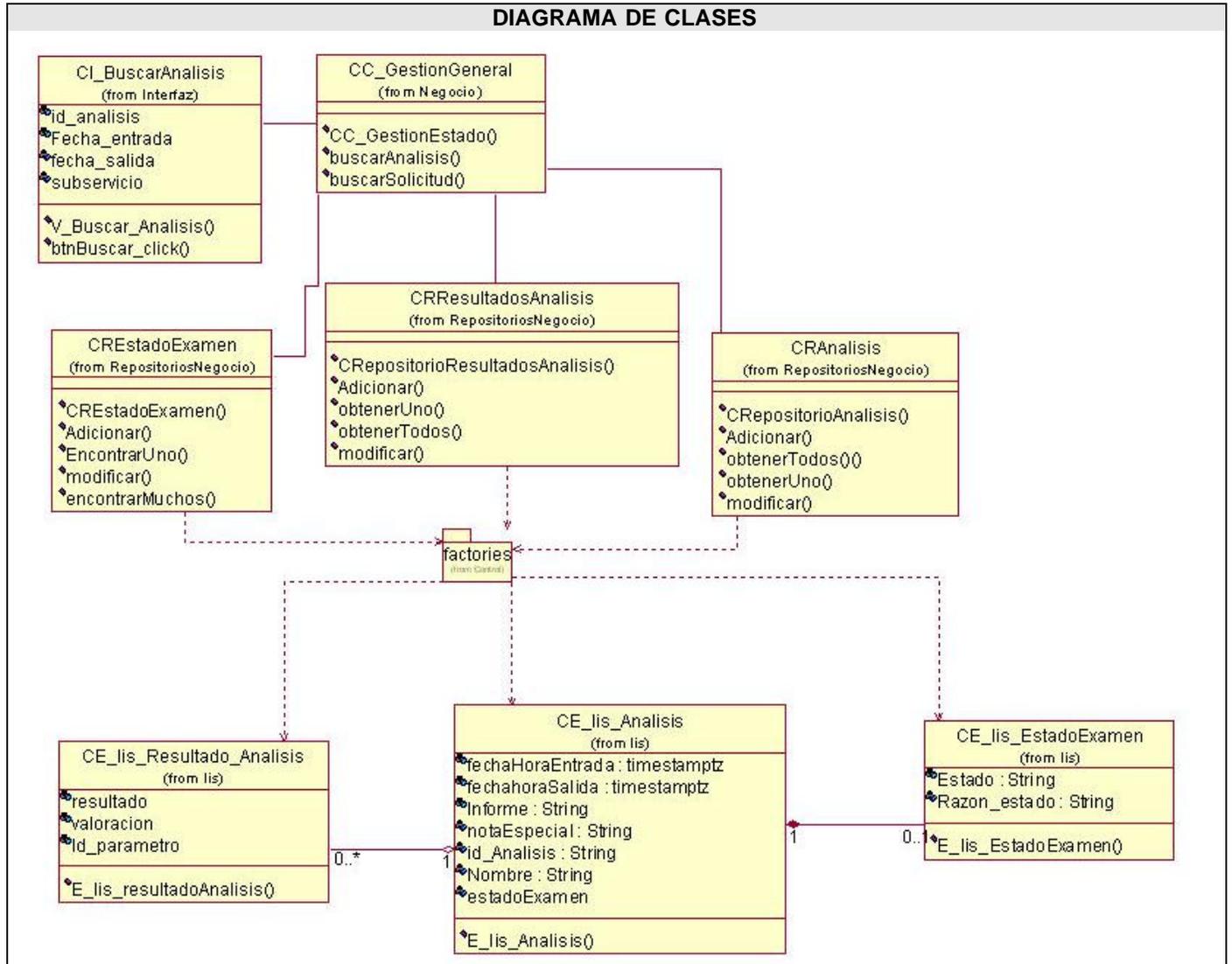
### **3.2.2 Diagrama de clases**

El diagrama de clases es una representación más concreta que el diagrama de clases del análisis. Representa la parte estática del sistema, las clases y sus relaciones.



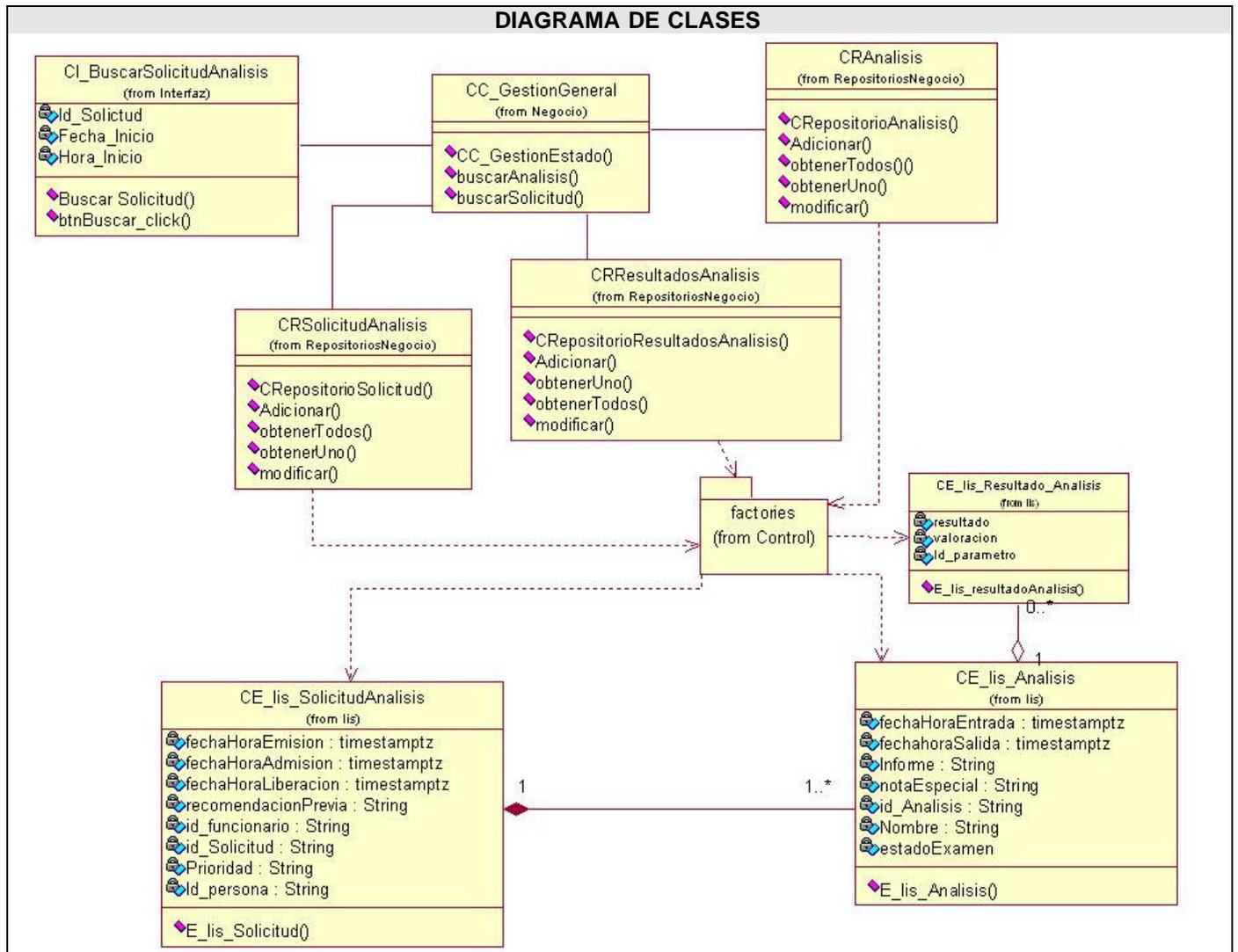
### CU Almacenar Muestra

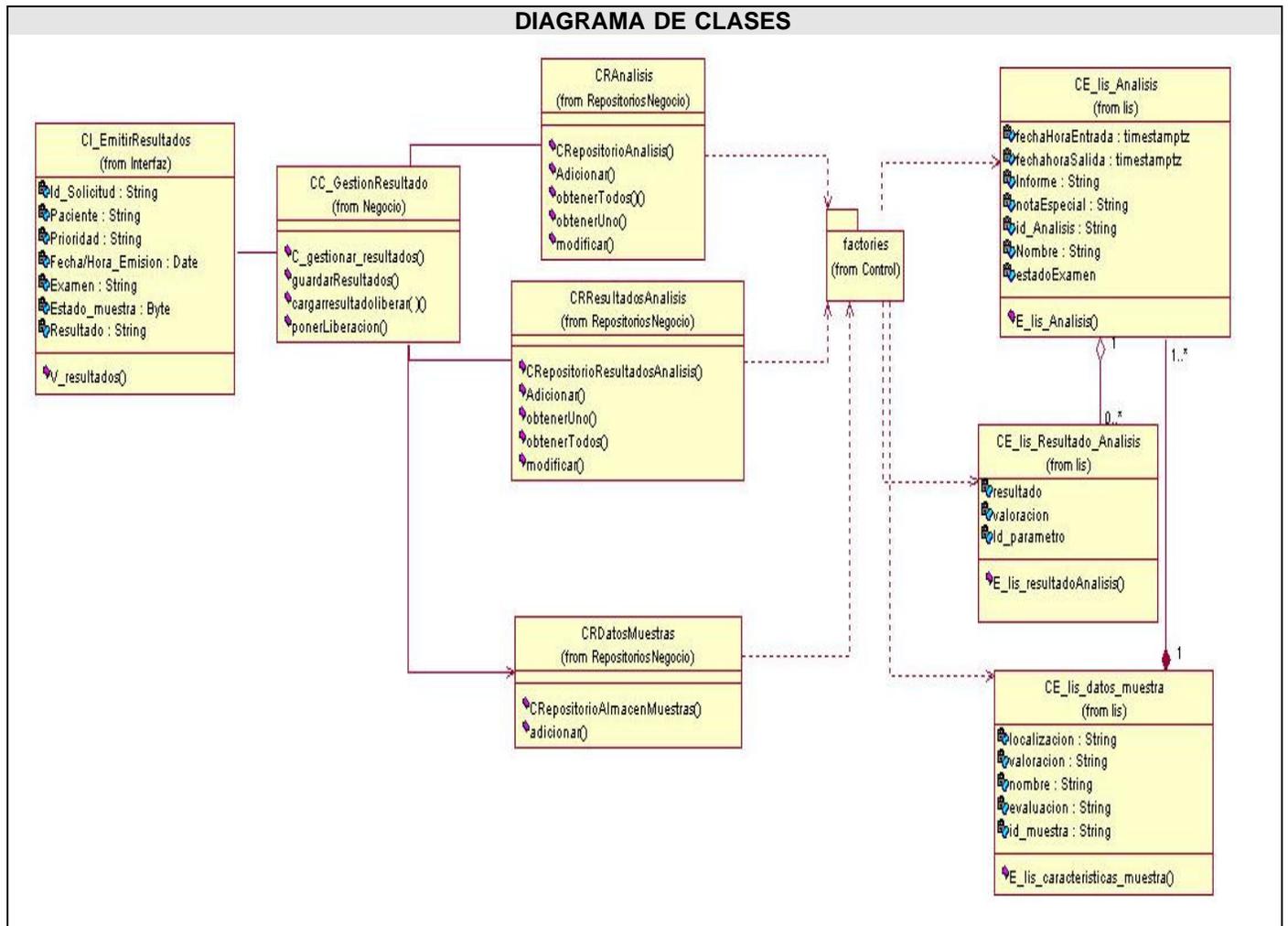


**CU Buscar Análisis**




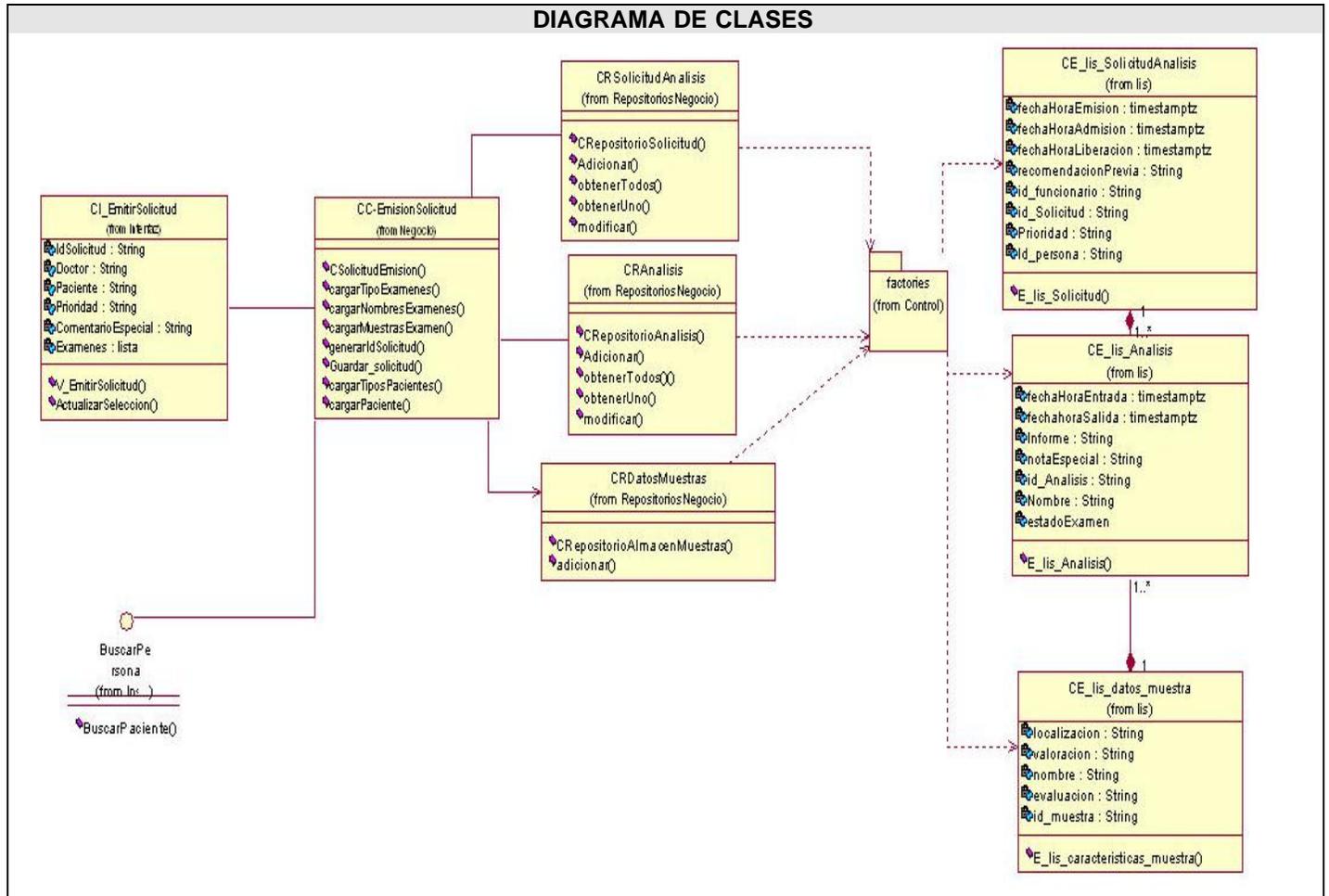
### CU Buscar Solicitud



**CU Emitir Resultado**


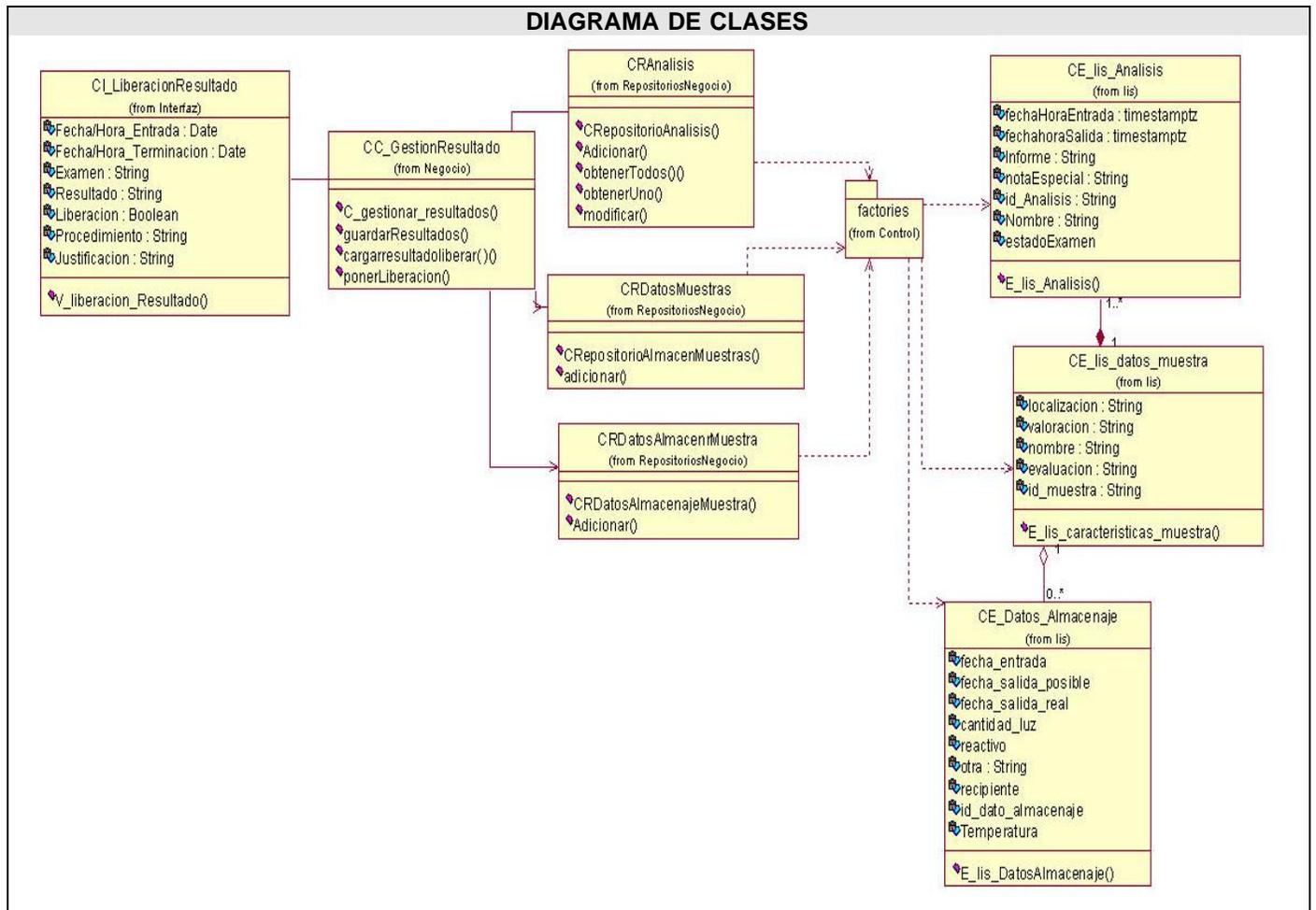


### CU Emitir Solicitud



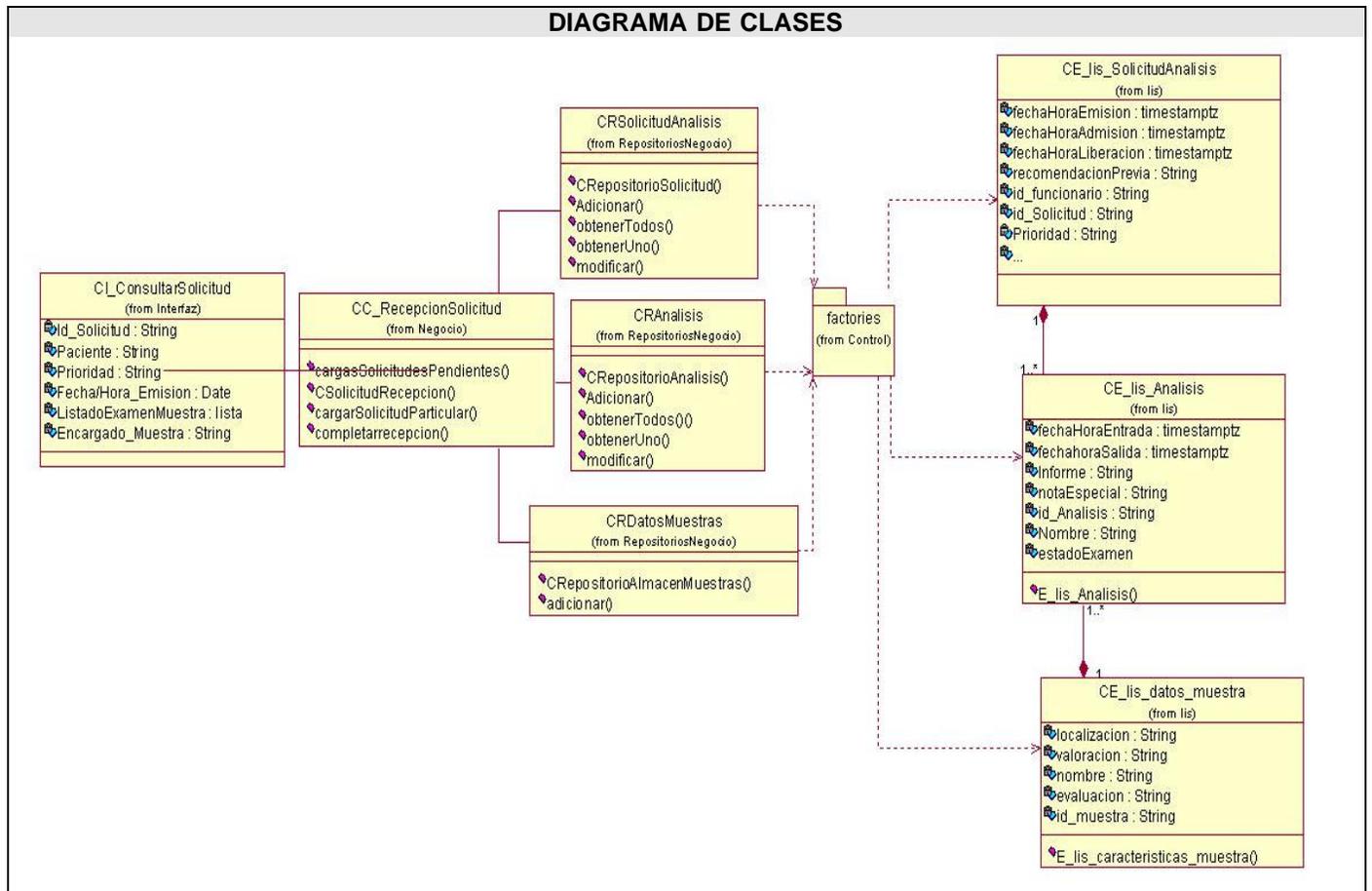


### CU Liberar Resultado



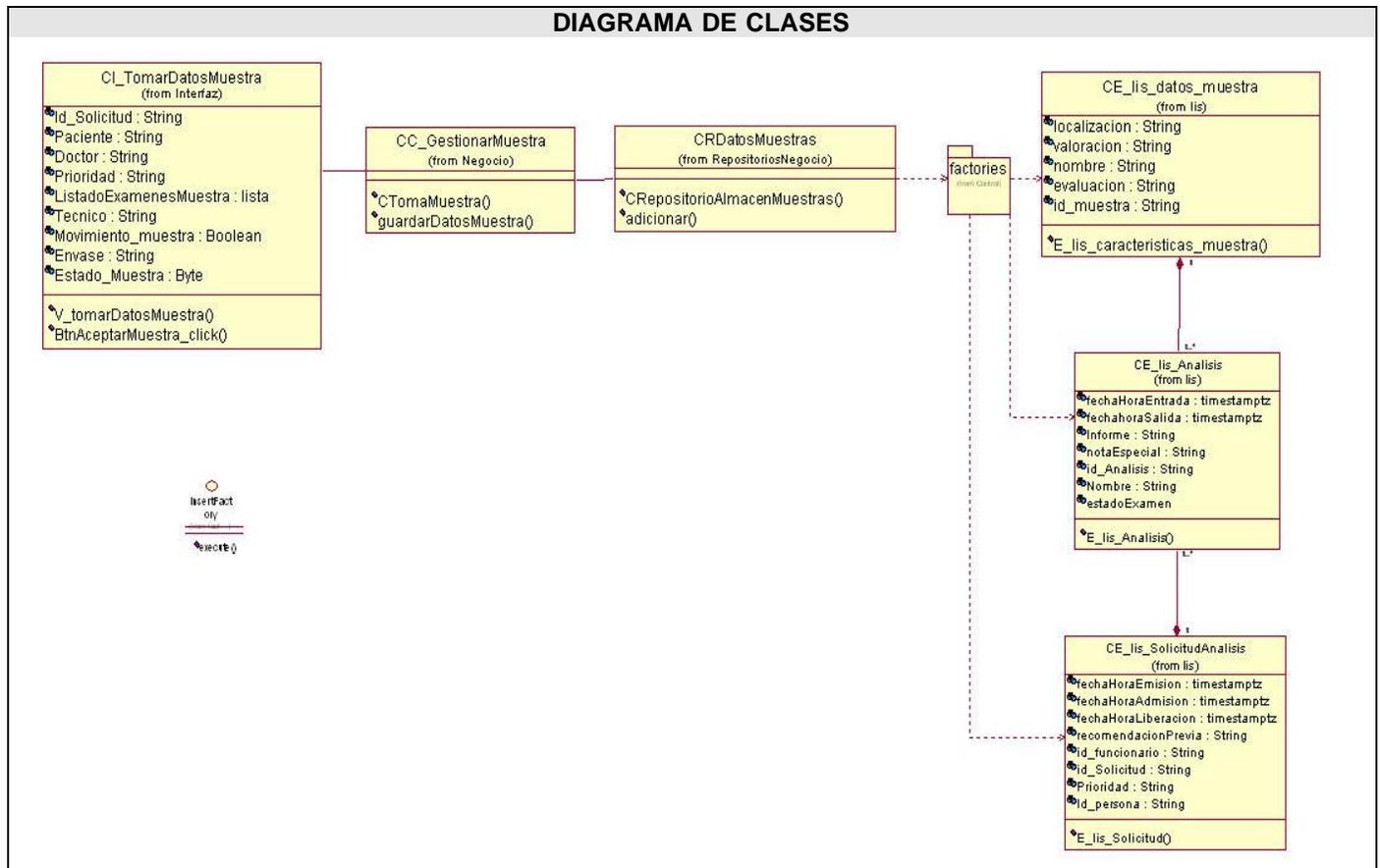


### CU Recepcionar Solicitud





### CU Tomar Datos de Muestra



### 3.2.3 Descripción de las clases.

Las descripciones de las clases se encuentran referidas en el Anexo #2.

### 3.2.4 Definiciones de diseño que se aplican

La aplicación esta dirigida a profesionales de la salud, específicamente a los que trabajan en los laboratorios clínicos, en muchos casos con conocimiento mínimo del trabajo con computadoras. Teniendo esto en cuenta los esfuerzos del diseño están enfocados en lograr una interfaz sencilla y fácil de usar por los usuarios. Es por ello que el sistema debe adaptarse al ambiente de trabajo del usuario final para que no haya resistencia al cambio. Para lograr lo que se ha planteado anteriormente, el diseño de la aplicación se basa en los 7 Principios del Diseño Universal o Diseño para Todos, estos principios se centran en el



diseño de aplicaciones teniendo en cuenta la cultura, el conocimiento, el ambiente, que influyan sobre los usuarios a los que va dirigido el producto. Estos principios son:

1. **Uso equiparable.** El diseño es útil y vendible a personas con diversas capacidades.
  - Que proporcione las mismas maneras de uso para todos los usuarios: idénticas cuando es posible, equivalentes cuando no lo es.
  - Que evite segregar o estigmatizar a cualquier usuario.
  - Las características de privacidad, garantía y seguridad deben estar igualmente disponibles para todos los usuarios.
  - Que el diseño sea atractivo para todos los usuarios.
  
2. **Uso flexible** El diseño se acomoda a un amplio rango de preferencias y habilidades individuales.
  - Que ofrezca posibilidades de elección en los métodos de uso.
  - Que pueda accederse y usarse tanto con la mano derecha como con la izquierda.
  - Que facilite al usuario la exactitud y precisión.
  - Que se adapte al paso o ritmo del usuario.
  
3. **Simple e intuitivo** El uso del diseño es fácil de entender, atendiendo a la experiencia, conocimientos, habilidades lingüísticas o grado de concentración actual del usuario.
  - Que elimine la complejidad innecesaria.
  - Que sea consistente con las expectativas e intuición del usuario.
  - Que se acomode a un amplio rango de alfabetización y habilidades lingüísticas.
  - Que dispense la información de manera consistente con su importancia.
  - Que proporcione avisos eficaces y métodos de respuesta durante y tras la finalización de la tarea.
  
4. **Información perceptible** El diseño comunica de manera eficaz la información necesaria para el usuario, atendiendo a las condiciones ambientales o a las capacidades sensoriales del usuario.
  - Que use diferentes modos para presentar de manera redundante la información esencial (gráfica, verbal o táctilmente)
  - Que proporcione contraste suficiente entre la información esencial y sus alrededores.



- Que amplíe la legibilidad de la información esencial.
  - Que diferencie los elementos en formas que puedan ser descritas (por ejemplo, que haga fácil dar instrucciones o direcciones).
  - Que proporcione compatibilidad con varias técnicas o dispositivos usados por personas con limitaciones sensoriales.
5. **Con tolerancia al error** El diseño minimiza los riesgos y las consecuencias adversas de acciones involuntarias o accidentales.
- Que disponga los elementos para minimizar los riesgos y errores: elementos más usados, más accesibles; y los elementos peligrosos eliminados, aislados o tapados.
  - Que proporcione advertencias sobre peligros y errores.
  - Que proporcione características seguras de interrupción.
  - Que desaliente acciones inconscientes en tareas que requieren vigilancia
6. **Que exija poco esfuerzo físico** El diseño puede ser usado eficaz y confortablemente y con un mínimo de fatiga.
- Que permita que el usuario mantenga una posición corporal neutra.
  - Que utilice de manera razonable las fuerzas necesarias para operar.
  - Que minimice las acciones repetitivas.
  - Que minimice el esfuerzo físico continuado.
7. **Tamaño y espacio para el acceso y uso** Proporciona un tamaño y espacio apropiados para el acceso, alcance, manipulación y uso, atendiendo al tamaño del cuerpo, la postura o la movilidad del usuario.
- Que proporcione una línea de visión clara hacia los elementos importantes tanto para un usuario sentado como de pie.
  - Que el alcance de cualquier componente sea confortable para cualquier usuario sentado o de pie.
  - Que se acomode a variaciones de tamaño de la mano o del agarre.
  - Que proporcione el espacio necesario para el uso de ayudas técnicas o de asistencia personal



### **3.2.5 Concepción de la ayuda.**

La ayuda debe estar accesible como parte del menú en todas las pantallas del sistema, con el objetivo de que el usuario vea la información que necesita en ese momento. Esta cuenta con informaciones generales de la aplicación, explicaciones de la funcionalidad del sistema. En cada pantalla debe mostrarse cómo realizar aquellas operaciones que estén relacionadas con la posición donde se encuentre el usuario en dicho momento.

Además se tendrá en cuenta la confección de manuales de usuario y será entregado a los usuarios que interactuarán con el sistema en formato digital o en papel, que explicará de forma detallada las principales funcionalidades y opciones que brinda el software.

También deben coordinarse cursos de capacitación o entrenamiento en el uso de la aplicación, dirigida a todos los usuarios potenciales antes y durante la implantación oficial de este producto en el los hospitales.

### **Conclusiones**

Esta fase de análisis y diseño, es una de las más importantes, pues proporciona una idea de lo que realmente es el software y es donde se materializan con precisión los requerimientos del cliente. En este capítulo se modelaron los diagramas de clases del análisis, representando las clases y las relaciones entre ellas, con el fin de comprender de una mejor manera las funcionalidades del sistema. También se obtuvo los diagramas de clases del diseño, donde se modela la forma en que el sistema debe darle solución a los requerimientos funcionales y no funcionales.

## **Conclusiones**

Una vez concluida la investigación, se realizó el diseño del sistema para la gestión y procesamiento de la información de los laboratorios clínicos y se dieron cumplimiento a los objetivos planteados, obteniéndose los siguientes resultados:

- ✓ Se analizó el problema de los laboratorios y la situación actual y se determinó que el estado de informatización en los laboratorios es muy bajo.
- ✓ Se realizó una investigación sobre los sistemas existentes tanto en Cuba como el extranjero.
- ✓ Se realizó el análisis y diseño del sistema de información de laboratorios que permite controlar la gestión de información y los servicios que se prestan en los laboratorios de los diferentes centros de atención médica.



## **Recomendaciones.**

Se recomienda que se continúe este proyecto, para mejorar la gestión de los laboratorios clínicos y llegar a tener un servicio de excelencia y además:

- ✓ Contar con un asesor técnico especializado en la materia.
- ✓ Comenzar a usar técnicas de estimación de tamaño de los proyectos.
- ✓ Que se desarrollen los submódulos de Logística, Estadísticas y Calidad.



### Referencias Bibliográficas

- [1] Yoenny Pérez Romero, Yovannys Sánchez Corales. Registro de Partos y Nacimientos para el Sistema Integral de Salud. Instituto Superior Politécnico “José Antonio Echeverría”. Ciudad de La Habana. 2005. Total de páginas: 144. [Revisado el 26 de noviembre de 2007]
- [2] Red de Salud en Cuba. (1999- 2000) [Revisado el 20 de mayo de 2007] Disponible en: [http://www.sld.cu/sistema\\_de\\_salud/aspectos.html#principios](http://www.sld.cu/sistema_de_salud/aspectos.html#principios)
- [3] Ariel Delgado Ramos, Maria Vidal Ledo. Informática en la salud pública cubana. (2006)[Revisado el 20 de mayo de 2007] Disponible en: [http://bvs.sld.cu/revistas/spu/vol32\\_3\\_06/spu15306.htm](http://bvs.sld.cu/revistas/spu/vol32_3_06/spu15306.htm)
- [4] Caridad Guzmán Ravelo. Revolución Científico-Técnica. Su Impacto Social y Tecnológico en el Laboratorio Clínico. (2004)[Revisado el 20 de mayo de 2007]. Disponible en: [http://www.cpicmha.sld.cu/hab/vol10\\_1\\_04/hab07104.htm](http://www.cpicmha.sld.cu/hab/vol10_1_04/hab07104.htm)
- [5] Tiempo21. Consolida Cuba informatización del Sistema de Salud. (2007)[Revisado el 20 de mayo de 2007]. Disponible en: [http://www.tiempo21.islagrande.cu/Cuba/Febrero07/cuba\\_informatizacion\\_salud\\_publica\\_070213.htm](http://www.tiempo21.islagrande.cu/Cuba/Febrero07/cuba_informatizacion_salud_publica_070213.htm)
- [6] Pascual Mostaza, C/ Raventos, J/ Pelegri, D/ Bedini. Base de Datos de Documentación Médica Española. [Revisado el 20 de mayo de 2007] Disponible en: <http://www.uv.es/~docmed/documed/documed/1483.html>
- [7] Colaboradores de Wikipedia. Análisis clínico (2007) [Revisado el 20 de mayo de 2007] Disponible en: [http://es.wikipedia.org/wiki/An%C3%A1lisis\\_cl%C3%ADnicos](http://es.wikipedia.org/wiki/An%C3%A1lisis_cl%C3%ADnicos)
- [8] SOFTEL, Manual de Usuario GALEN LAB, Edición de la versión 5.0. [Revisado el 15 de noviembre de 2006]
- [9] Coya Laboratorios. Manual de Usuario. (2006)[Revisado el 15 de noviembre de 2006]. Disponible en: [http://www.coyasistemas.com.ar/laboratorios.htm/Coya\\_Laboratorios\\_2.1\\_Manual\\_de\\_Usuario\\_2006](http://www.coyasistemas.com.ar/laboratorios.htm/Coya_Laboratorios_2.1_Manual_de_Usuario_2006).



## REFERENCIAS BIBLIOGRÁFICAS

- [10] Health Care Internacional Services. (2005). [Revisado el 15 de noviembre de 2007] Disponible en: <http://www.hcisonline.com/online/sp/software.asp?id=6>
- [11] Soluciones informáticas para la Gestión Hospitalaria. (2003)[Revisado el 19 de febrero de 2007]. Disponible en: [http://www.siemens.com.ar/sites/internet/legacy/siemed/soluciones\\_informaticas\\_para\\_la\\_gestion\\_hospitalaria.htm](http://www.siemens.com.ar/sites/internet/legacy/siemed/soluciones_informaticas_para_la_gestion_hospitalaria.htm)
- [12] Características del Sistema de Información para Laboratorios Clínicos DLabo (2004) [Revisado el 19 de febrero de 2007] Disponible en: <http://www.esicomsa.com/dlabo.html>
- [13] Colaboradores de Wikipedia. Tecnologías de la información y la comunicación (2007) [Revisado el 5 de marzo de 2007]. Disponible en: [http://es.wikipedia.org/wiki/Tecnolog%C3%ADas\\_de\\_la\\_informaci%C3%B3n](http://es.wikipedia.org/wiki/Tecnolog%C3%ADas_de_la_informaci%C3%B3n)
- [14] Colaboradores de Wikipedia. Internet (2007) [Revisado el 5 de marzo de 2007] Disponible en: <http://es.wikipedia.org/wiki/Internet>
- [15] Colaboradores de Wikipedia. Aplicación Web (2007) [Revisado el 5 de marzo de 2007]. Disponible en: [http://es.wikipedia.org/wiki/Aplicaci%C3%B3n\\_web](http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web)
- [16] Tutorial de AJAX (2006)[Revisado el 5 de marzo de 2007]. Disponible en: <http://www.cristalab.com/tutoriales/162/tutorial-de-ajax>
- [17] Francisco Recio, David Provencio. Ventajas de .Net. [Revisado el 25 de marzo de 2007]. Disponible en: <http://www.desarrolloweb.com/articulos/1329.php>.
- [18] Colaboradores de Wikipedia. JavaScript (2007) [Revisado el 8 de marzo de 2007]. Disponible en: <http://es.wikipedia.org/wiki/JavaScript>
- [19] ¿Por qué ASP.NET? (2006) [Revisado el 25 de marzo de 2007]. Disponible en: <http://www.subgurim.net/Articulos/asp-net-general-articulo3.aspx>
- [20] Colaboradores de Wikipedia. C Sostenido (2007) [Revisado el 25 de marzo de 2007]. Disponible en: [http://es.wikipedia.org/wiki/C\\_Sostenido](http://es.wikipedia.org/wiki/C_Sostenido)



## REFERENCIAS BIBLIOGRÁFICAS

- [21] José H. Canós, Patricio Letelier, Maria Carmen Penadés. Metodologías ágiles en el Desarrollo de Software. [Revisado el 22 de enero de 2007]. Disponible en: <http://www.willydev.net/descargas/prev/TodoAgil.Pdf>
- [22] Colaboradores de Wikipedia. Programación Extrema (2007) [Revisado el 22 de enero de 2007]. Disponible en: [http://es.wikipedia.org/wiki/Programaci%C3%B3n\\_Extrema](http://es.wikipedia.org/wiki/Programaci%C3%B3n_Extrema)
- [23] IDEM 26
- [24] Carlos Reynoso. Métodos Heterodoxos en Desarrollo de Software. (2004)[Revisado el 22 de enero de 2007]. Disponible en: [http://www.microsoft.com/spanish/msdn/arquitectura/roadmap\\_arq/heterodox.asp#8](http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/heterodox.asp#8)
- [25] Dónde aprender más sobre SCRUM. (2005)[Revisado el 25 de mayo de 2007]. Disponible en: [http://www.agile-spain.com/agilev2/donde\\_aprender\\_mas\\_sobre\\_scrum](http://www.agile-spain.com/agilev2/donde_aprender_mas_sobre_scrum)
- [26] Joaquin Gracia. Gestión de proyectos con SCRUM. (2006)[Revisado el 25 de mayo de 2007]. Disponible en: <http://www.ingenierosoftware.com/equipos/scrum.php>
- [27] María A. Mendoza Sánchez. Metodologías De Desarrollo De Software. (2004)[Revisado el 15 de diciembre de 2006]. Disponible en: [http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html)
- [28] Colaboradores de Wikipedia. Lenguaje Unificado de Modelado (2007) [Revisado el 15 de diciembre de 2006]. Disponible en: <http://es.wikipedia.org/wiki/UML>
- [29] Rational Rose. (2001) [Revisado el 15 de diciembre de 2006]. Disponible en: [http://www.indudata.com/1rational\\_rose.htm](http://www.indudata.com/1rational_rose.htm)
- [30] Case Studio 2. (2003)[Revisado el 26 de marzo de 2007]. Disponible en: <http://programas.navegalis.com/software-producto/31.php>
- [31] Jacobson, Booch, Rumbaugh, Addison – Wesley. El Proceso Unificado de Software. Object Technology Series (2000)[Revisado el 26 de marzo de 2007]
- [32] Colaboradores de Wikipedia. Patrones de Diseño (2007) [Revisado el 16 de febrero]. Disponible en: [http://es.wikipedia.org/wiki/Abstract\\_Factory\\_%28patr%C3%B3n\\_de\\_dise%C3%B1o%29](http://es.wikipedia.org/wiki/Abstract_Factory_%28patr%C3%B3n_de_dise%C3%B1o%29)



### Bibliografía

1. Aspectos generales del Sistema Nacional de Salud (2000). Disponible en: [http://www.sld.cu/sistema\\_de\\_salud/aspectos.html](http://www.sld.cu/sistema_de_salud/aspectos.html).
2. Booch, Grady, Jacobson, Ivar y Rumbaugh, James. El Proceso Unificado de Desarrollo de Software. La Habana. Felix Varela, 2004. 299.
3. Canós José, Letelier Patricio, Penadés Maria C. Metodologías ágiles en el Desarrollo de Software. Disponible en: <http://www.willydev.net/descargas/prev/TodoAgil.Pdf>
4. Colaboradores de Wikipedia. PostgreSQL (2007) Disponible en: <http://es.wikipedia.org/wiki/PostgreSQL>
5. De la Torre, Aníbal (2006). Lenguajes del lado servidor o cliente. Disponible en: [http://www.adelat.org/media/docum/nuke\\_publico/lenguajes\\_del\\_lado\\_servidor\\_o\\_cliente.html](http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html)
6. Delgado Ariel, Vida Maria. Informática en la salud pública cubana. (2006) Disponible en: [http://bvs.sld.cu/revistas/spu/vol32\\_3\\_06/spu15306.htm](http://bvs.sld.cu/revistas/spu/vol32_3_06/spu15306.htm)
7. Larman, Craig. UML y Patrones. Introducción al análisis y diseño orientado a objetos. La Habana, Felix Varela, 2004.506.
8. Menéndez-B Asensio Rafael. Metodologías de desarrollo de Software. Disponible en: [http://www.wikilearning.com/metodologia\\_de\\_desarrollo\\_de\\_software-wkccp-3617-1.htm](http://www.wikilearning.com/metodologia_de_desarrollo_de_software-wkccp-3617-1.htm)
9. Pérez Romero, Yoenny, Sánchez Corales, Yovannys. Registro de Partos y Nacimientos para el Sistema Integral de Salud. Instituto Superior Politecnico “Jose Antonio Echeverría”, Facultad de Ingeniería Industrial. Ciudad de la Habana, 2005.132
10. Pérez García, Reñiré. Impacto de la Informatización en la Sociedad Cubana. Ciencia, tecnología y sociedad, (abril 2005). Disponible en: <http://www.monografias.com/trabajos24/informatizacion-cuba/informatizacion-cuba.shtml>
11. Pressman, Roger S. Ingeniería del Software. *Un enfoque práctico*. La Habana, Felix Varela, 2005. 600.
12. Recio Francisco, Provencio David. Ventajas de .Net. Disponible en: <http://www.desarrolloweb.com/articulos/1329.php>



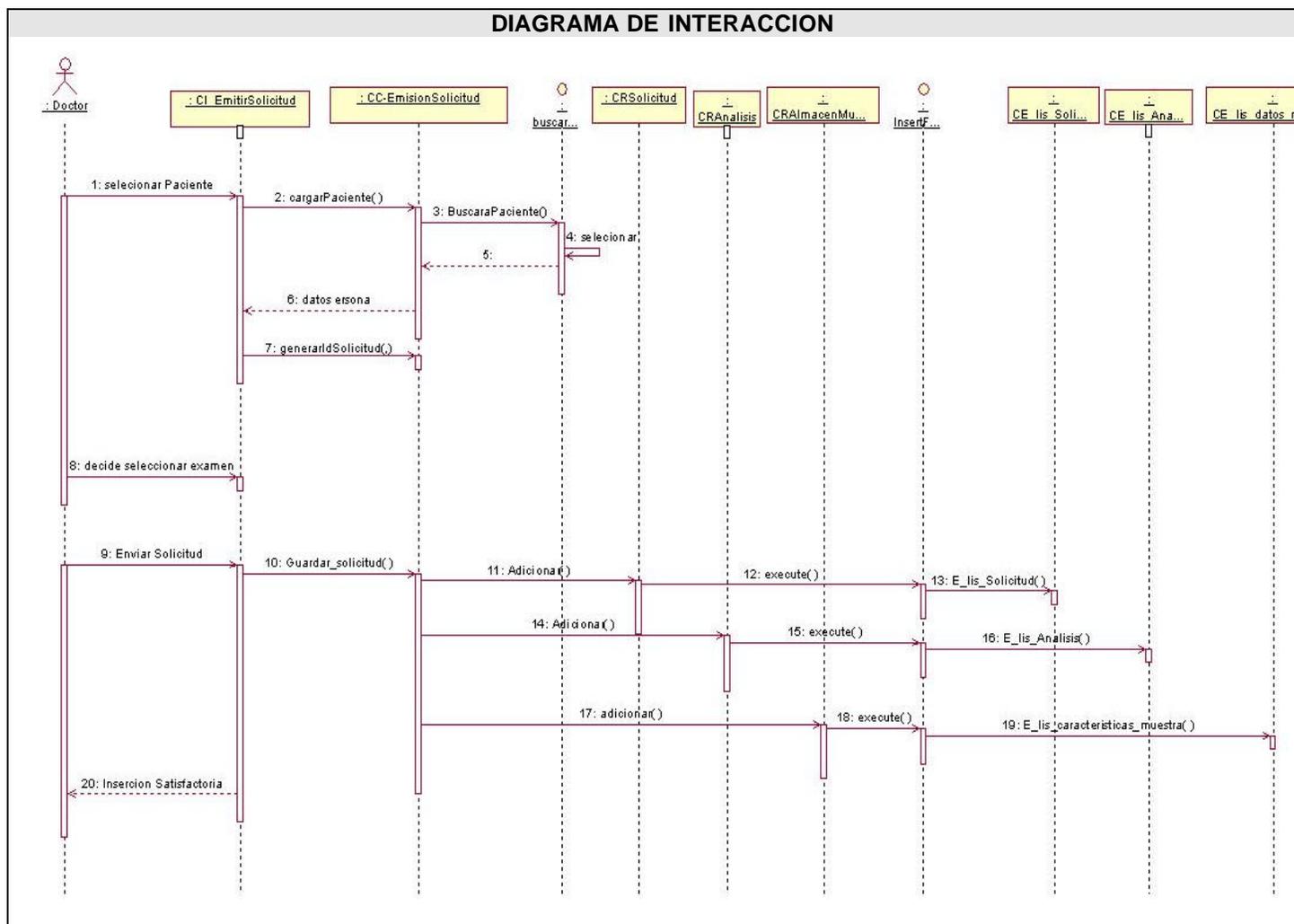
## BIBLIOGRAFÍA

13. Rodríguez Montero, Rafael Y. Sistema Integrado para la Gestión Bibliotecaria SIGeBi. Instituto Superior Politécnico “Jose Antonio Echeverría”, Facultad de Ingeniería Industrial. Ciudad de la Habana, 2004.93
14. Sánchez Mendoza, María A. Metodologías De Desarrollo De Software. (2004). Disponible en: [http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_software\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html)
15. Teruel Alejandro. Arquitectura de capas en Sistemas de Información. (2000). Disponible en: <http://www ldc.usb.ve/~teruel/ci3715/clases/arqCapas2.html>

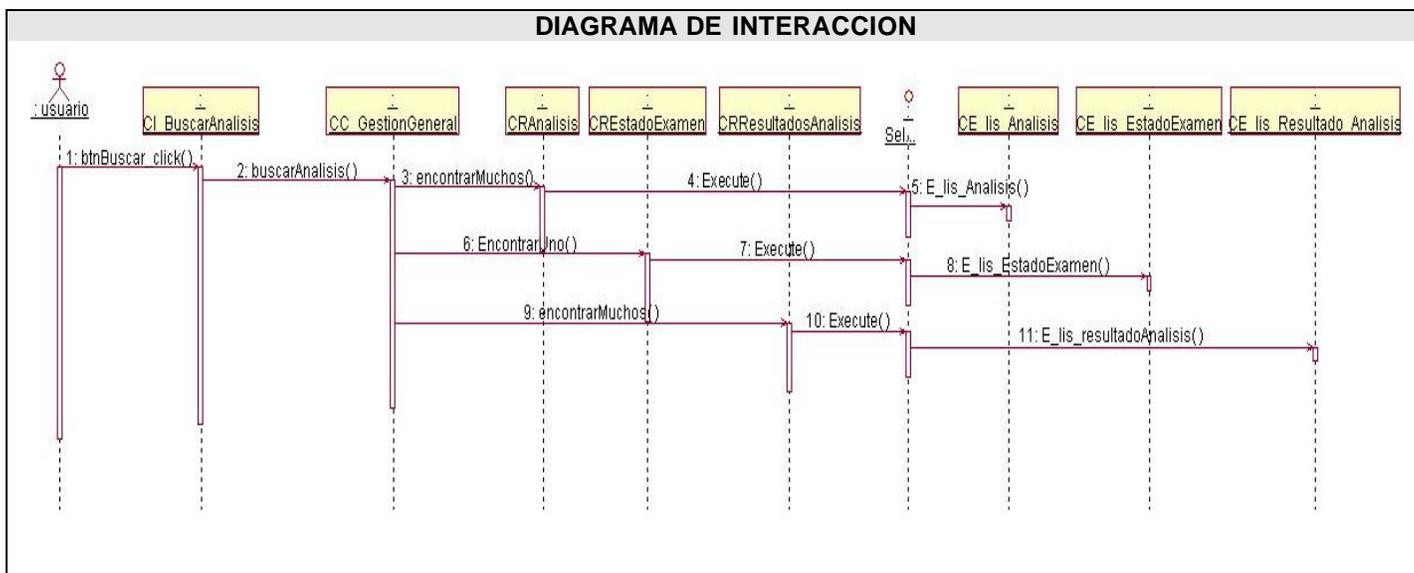
# Anexos

## Anexo #1. Diagramas de Interacción

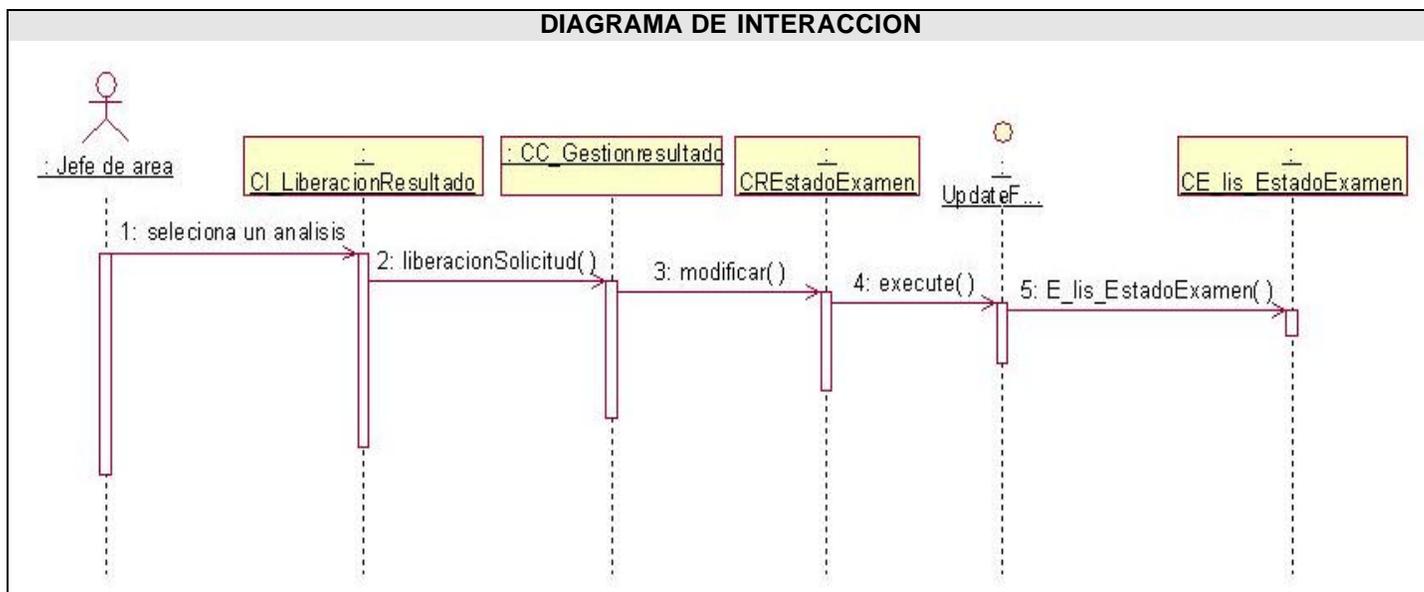
### CU Emitir Solicitud



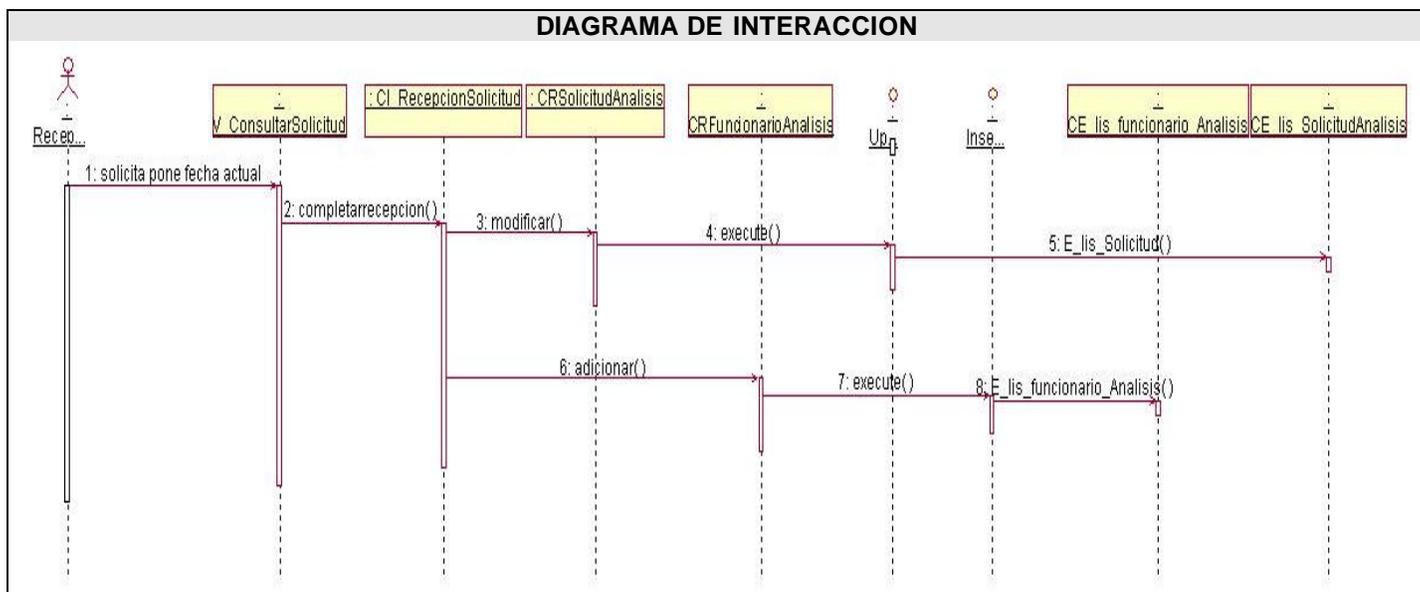
### CU Buscar Análisis



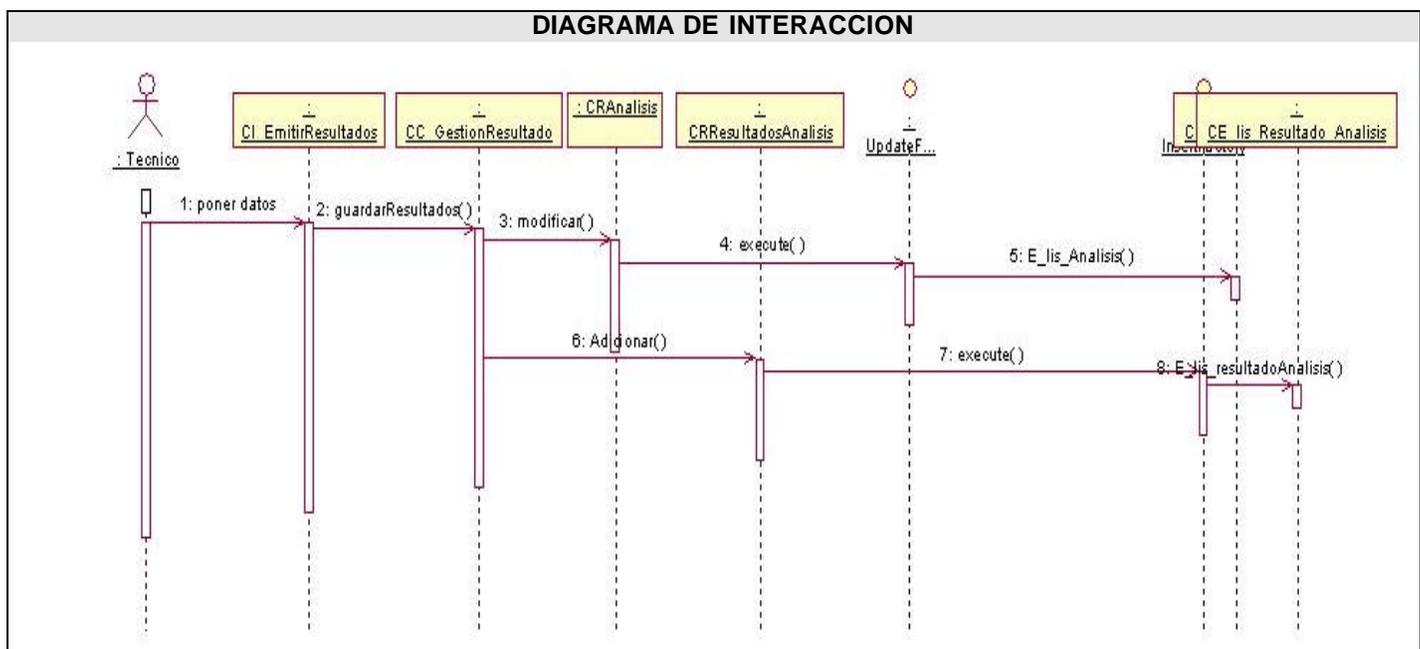
### CU Liberar Resultado



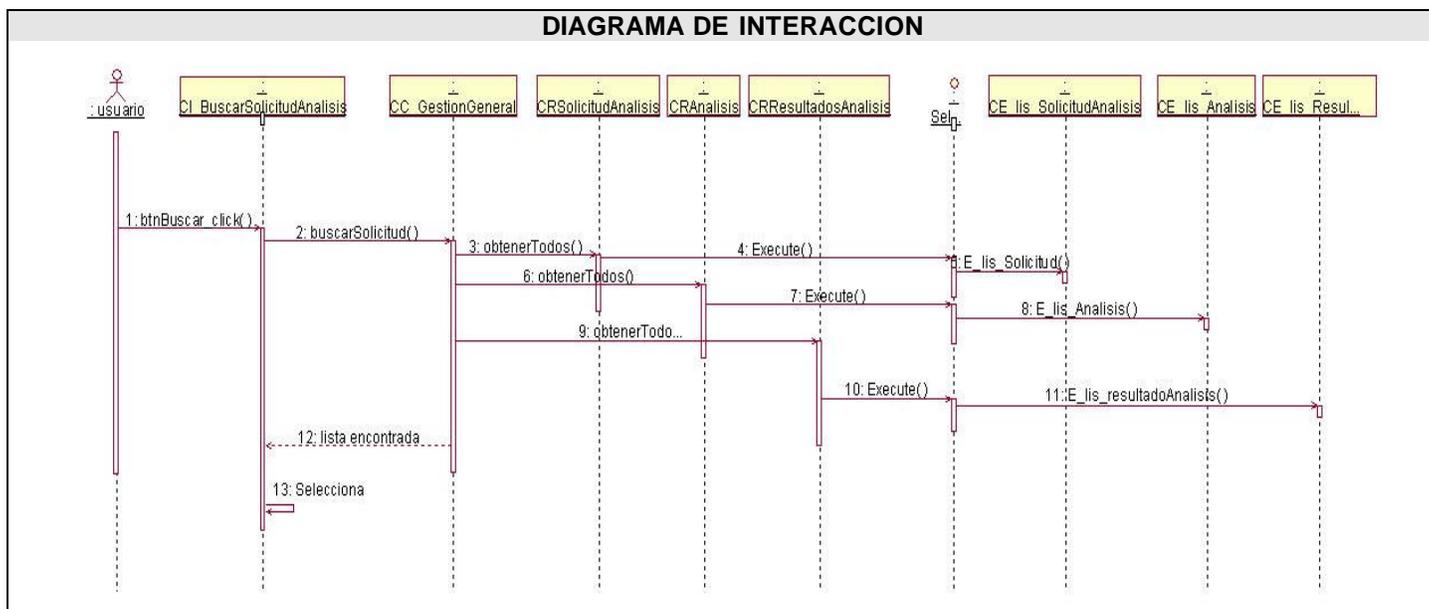
### CU Recepcionar Solicitud



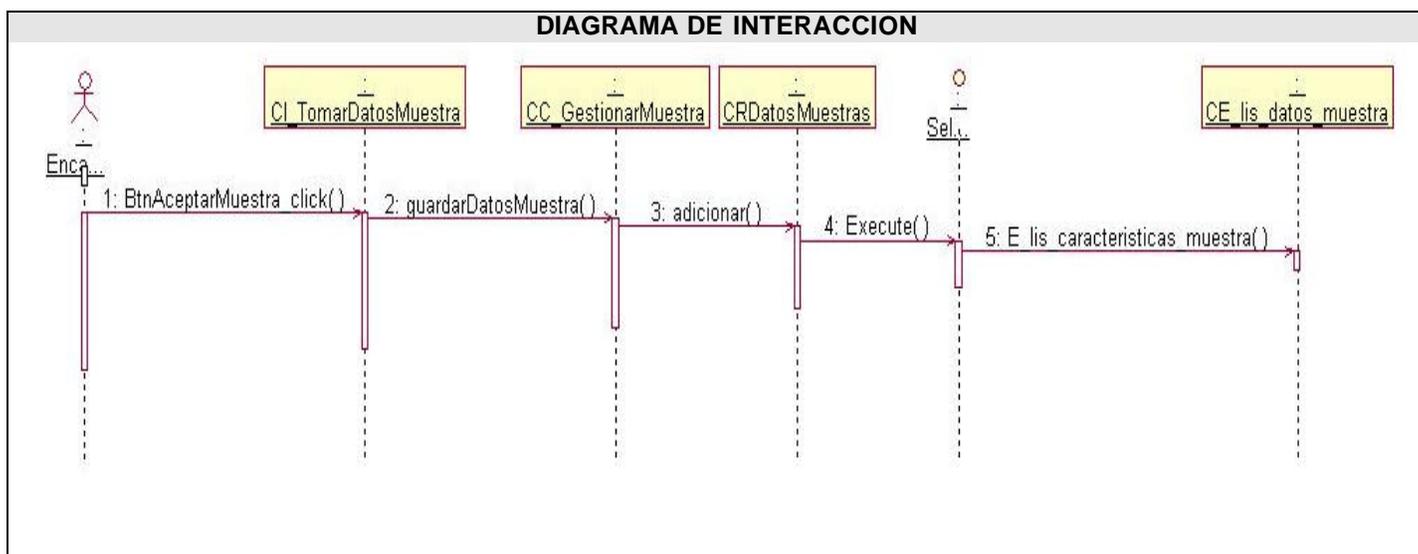
### CU Emitir Resultado



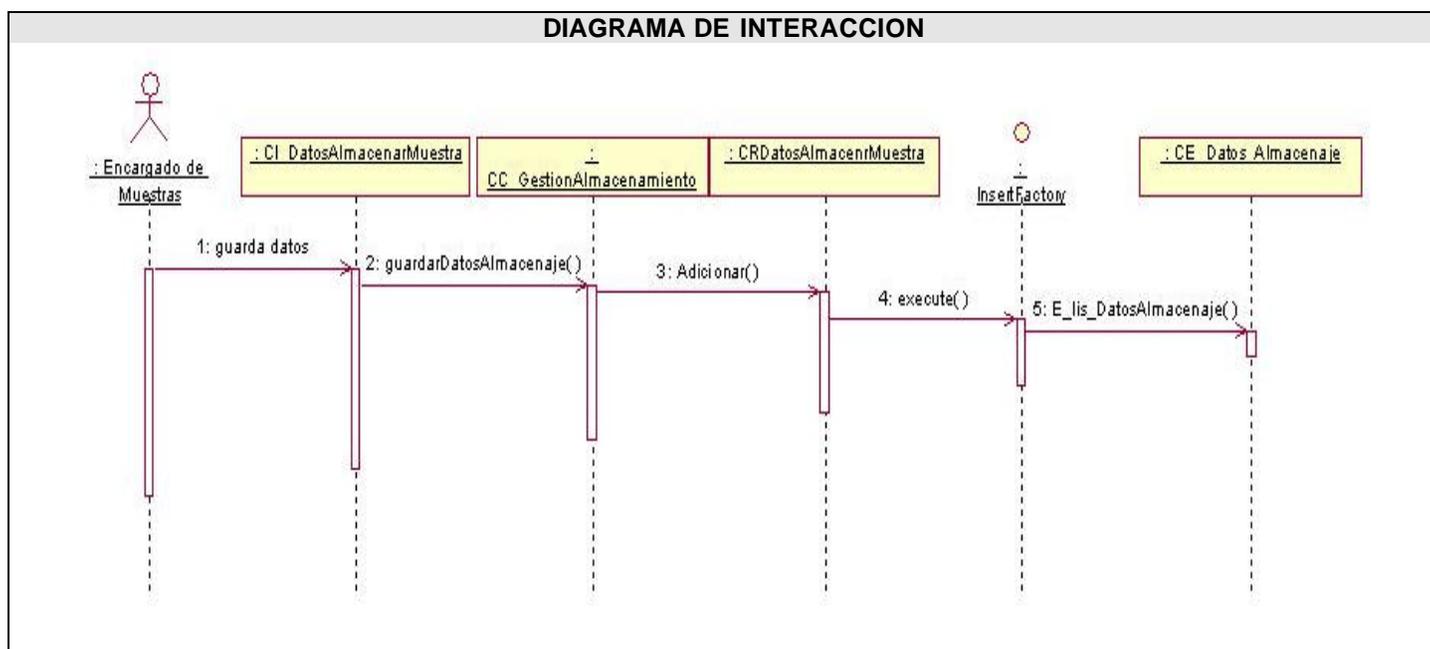
## CU Buscar Solicitud



## CU Tomar Datos de Muestra



## CU Almacenar Muestra



### Anexo #2 Descripción de las clases.

<b>Nombre: CI_Datos_Almacenar_Muestra</b>	
<b>Tipo de clase: interfaz</b>	
<b>Atributo</b>	<b>Tipo</b>
Fecha/Hora_Inicio	Date
Tiempo_Encubacion	Integer
Temperatura	Float
Reactivo	String
Fotosensible	Boolean
<b>Para cada responsabilidad:</b>	
Nombre:	CI_Datos_Almacenar_Muestra
Descripción:	Constructor

<b>Nombre: CI_Buscar_Análisis</b>	
<b>Tipo de clase: interfaz</b>	
<b>Atributo</b>	<b>Tipo</b>
Id_Análisis	String

Fecha_entrada	Date
fecha_salida	Date
subservicio	String
<b>Para cada responsabilidad:</b>	
Nombre:	Buscar_Análisis.
Descripción:	Constructor.

<b>Nombre: CI_Buscar_Solicitud_Análisis</b>	
<b>Tipo de clase: interfaz</b>	
<b>Atributo</b>	<b>Tipo</b>
Id_Solicitud	String
Fecha_Inicio	Date
Hora_Inicio	Date
<b>Para cada responsabilidad:</b>	
Nombre:	CI_Buscar_Solicitud_Análisis.
Descripción:	Constructor.

<b>Nombre: CI_Emitir_Resultado</b>	
<b>Tipo de clase: interfaz</b>	
<b>Atributo</b>	<b>Tipo</b>
Id_Solicitud	String
Paciente	String
Prioridad	String
Fecha/Hora_Emision	Date
Examen	String
Resultado	String
Estado_Muestra	Byte
<b>Para cada responsabilidad:</b>	
Nombre:	CI_Emitir_Resultado
Descripción:	Constructor.

<b>Nombre: CI_Emitir_Solicitud</b>	
<b>Tipo de clase: interfaz</b>	
<b>Atributo</b>	<b>Tipo</b>
Id_Solicitud	String
Doctor	String
Paciente	String
Prioridad	String
ComentarioEspecial	String
Exámenes	lista
<b>Para cada responsabilidad:</b>	

Nombre:	CI_Emitir_Solicitud
Descripción:	Constructor.
Nombre:	Actualizar Selección

<b>Nombre: CI_Liberación_Resultado</b>	
<b>Tipo de clase: interfaz</b>	
<b>Atributo</b>	<b>Tipo</b>
Fecha/Hora_Entrada	Date
Fecha/Hora_Terminación	Date
Examen	String
Resultado	String
Liberación	Boolean
Procedimiento	String
Justificación	String
<b>Para cada responsabilidad:</b>	
Nombre:	CI_Liberación_Resultado
Descripción:	Constructor.

<b>Nombre: CI_Consultar_Solicitud</b>	
<b>Tipo de clase: interfaz</b>	
<b>Atributo</b>	<b>Tipo</b>
Id_Solicitud	String
Paciente	String
Prioridad	String
Fecha/Hora_Emision	Date
Encargado_Muestra	lista
ListadoExamenMuestra	String
<b>Para cada responsabilidad:</b>	
Nombre:	CI_Consultar_Solicitud
Descripción:	Constructor

<b>Nombre: CI_Tomar_Datos_Muestra</b>	
<b>Tipo de clase: interfaz</b>	
<b>Atributo</b>	<b>Tipo</b>
Id_Solicitud	String
Paciente	String
Doctor	String
Prioridad	String
ListadoExamenMuestra	Lista
Técnico	String
Movimiento_muestra	Boolean

Envase	String
Estado_Muestra	Byte
<b>Para cada responsabilidad:</b>	
Nombre:	CI_Tomar_Datos_Muestra
Descripción:	Constructor

<b>Nombre: CC_GestionAlmacenamiento</b>	
<b>Tipo de clase: control</b>	
<b>Para cada responsabilidad:</b>	
Nombre:	CC_GestionAlmacenamiento
Descripción:	Constructor
Nombre:	guardarDatosAlmacenaje
Descripción:	Gestiona y controla la salva de los datos de una muestra.

<b>Nombre: CC_GestionGeneral</b>	
<b>Tipo de clase: control</b>	
<b>Para cada responsabilidad:</b>	
Nombre:	CC_GestionEstado
Descripción:	Constructor
Nombre:	buscarAnalisis
Descripción:	Gestiona la búsqueda de un análisis o un grupo de análisis en la base de datos.
Nombre:	buscarSolicitud
Descripción:	Gestiona la búsqueda de una solicitud o un grupo de solicitudes en la base de datos.

<b>Nombre: CC_GestionResultado</b>	
<b>Tipo de clase: control</b>	
<b>Para cada responsabilidad:</b>	
Nombre:	CC_GestionResultado
Descripción:	Constructor
Nombre:	guardarResultados
Descripción:	Gestiona la salva de los resultados.
Nombre:	cargarresultadoliberar
Descripción:	Gestiona la búsqueda de un resultado a liberar.
Nombre:	ponerLiberacion
Descripción:	Gestionar salva de estado de liberación o rechazo.

<b>Nombre: CC_EmisionSolicitud</b>	
<b>Tipo de clase: control</b>	
<b>Para cada responsabilidad:</b>	

Nombre:	CC_EmisionSolicitud
Descripción:	Constructor
Nombre:	cargarTipoExámenes
Descripción:	Gestiona la búsqueda de los tipos de exámenes disponibles en el laboratorio.
Nombre:	cargarNombresExámenes
Descripción:	Gestiona la búsqueda de los nombre de los exámenes disponibles en el laboratorio.
Nombre:	cargarMuestrasExamen
Descripción:	Gestiona la búsqueda de los nombre de las muestras tomables en el laboratorio para un examen.
Nombre:	generarIdSolicitud
Descripción:	Genera un id para cada solicitud que será único y estará compuesto por: 8 caracteres son la fecha, los siguientes 4 la hora, luego el registro profesional (integro), número de HC (integro).
Nombre:	Guardar_solicitud
Descripción:	Gestiona la salva de los datos de las solicitudes.
Nombre:	cargarPrioridadPaciente
Descripción:	Gestiona la búsqueda de la prioridad de un paciente que esta dada en; Normal, urgencia, emergencia.

<b>Nombre: CC_GestionarMuestra</b>	
<b>Tipo de clase: control</b>	
<b>Para cada responsabilidad:</b>	
Nombre:	CC_GestionarMuestra
Descripción:	Constructor
Nombre:	guardarDatosMuestra
Descripción:	Gestiona salva de los datos correspondientes a una muestra.

<b>Nombre: CE_lis_datos_muestra</b>	
<b>Tipo de clase: entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
Id_muestra	String
nombre	String
valoracion	String
localizacion	String
<b>Para cada responsabilidad:</b>	
Nombre:	CE_lis_datos_muestra
Descripción:	Constructor

<b>Nombre: CE_lis_Análisis</b>	
<b>Tipo de clase: entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
id_Análisis	String
nombre	String
Fecha/hora/Salida	Datetime
Fecha/Hora/Entrada	datetime
Informe	String
Estado_Examen	String
<b>Para cada responsabilidad:</b>	
Nombre:	CE_lis_Análisis
Descripción:	Constructor

<b>Nombre: CE_lis_Solicitud_Análisis</b>	
<b>Tipo de clase: entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
id_Solicitud	String
id_Persona	String
Fecha/Hora/Emisión	Datetime
Fecha/Hora/Admisión	datetime
prioridad	String
Id_Funcionario	String
Fecha/Hora_Emisión	Date
Fecha/Hora_Admisión	Date
<b>Para cada responsabilidad:</b>	
Nombre:	CE_lis_Solicitud_Análisis
Descripción:	Constructor

<b>Nombre: CE_Datos_Almacenaje</b>	
<b>Tipo de clase: entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
id_dato_almacenaje	String
Temperatura	Double
recipiente	String
reactivo	String
cantidad_luz	Bool
fecha_entrada	Datetime
fecha_salida_posible	Datetime
fecha_salida_real	Datetime
<b>Para cada responsabilidad:</b>	

Nombre:	CE_Datos_Almacenaje
Descripción:	Constructor

<b>Nombre: CE_lis_Resultado_Análisis</b>	
<b>Tipo de clase: entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
Id_parametro	String
valoracion	String
resultado	String
<b>Para cada responsabilidad:</b>	
Nombre:	CE_lis_Resultado_Análisis
Descripción:	Constructor

<b>Nombre: CE_lis_Estado_Examen</b>	
<b>Tipo de clase: entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
Estado	String
Razón_estado	String
<b>Para cada responsabilidad:</b>	
Nombre:	CE_lis_Estado_Examen
Descripción:	Constructor

## Glosario de Términos

**Arquitectura:** Estructura organizativa de un sistema que incluye su descomposición en partes, su conectividad, mecanismos de interacción y principios de guía que proporciona información sobre el diseño del mismo.

**Artefactos:** Los productos tangibles del proceso, como por ejemplo, el modelo de casos de uso, el código fuente, etc.

**Clases:** Descriptor de un conjunto de objetos que comparten los mismos atributos., operaciones, métodos, relaciones y comportamientos. Una clase representa un concepto dentro del sistema que se está modelando.

**DOM:** (Document Object Model) El Modelo de Objetos de Documento es una forma de representar los elementos de un documento estructurado (tal como una página web HTML o un documento XML) como objetos que tienen sus propios métodos y propiedades.

**Frames:** (marcos) es un elemento que permite dividir la pantalla en varias áreas independientes unas de otras, y por tanto con contenidos distintos, aunque puedan estar relacionados.

**Hipertexto:** Formato que se le aplica a un texto, en el cual se representan palabras claves (en la mayoría de los casos subrayadas o con otros colores) las cuales dan acceso a un información determinado.

**HTML:** lenguaje diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web.

**HTTP:** protocolo usado para la transferencia de documentos WWW. Estas transferencias requieren un programa cliente http en un extremo de la comunicación y un servidor http en el otro.

**IDE:** es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios.

**IFrame:** Marco Incorporado. Es un elemento HTML que permite insertar o incrustar un documento HTML dentro de un documento HTML principal

**Métodos:** Utilizado principalmente en programación orientada a objetos, el término se refiere a las porciones de código asociadas exclusivamente con una clase (se los denomina entonces métodos de clase o métodos estáticos) o con un objeto (en este caso métodos de instancia).

**Modelo Cliente-Servidor:** Esta arquitectura consiste básicamente en que un programa, el Cliente informático realiza peticiones a otro programa, el servidor, que les da respuesta.

**Multihilo:** Es cuando un proceso tiene múltiples hilos de ejecución los cuales realizan actividades distintas.

**Multiplataforma:** Es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

**Plataforma:** Se refiere al sistema operativo o a sistemas complejos que a su vez sirven para crear programas, como las plataformas de desarrollo.

**Protocolo:** Es para la comunicación de datos a través de una red de paquetes conmutados

**SDK:** (Software Development Kit o kit de desarrollo de software) es generalmente un conjunto de herramientas de desarrollo que le permite a un programador crear aplicaciones para un sistema bastante concreto,

**Servidor:** Una aplicación informática o programa que realiza algunas tareas en beneficio de otras aplicaciones llamadas clientes.

**Sistema Operativo:** Es un conjunto de instrucciones destinados a permitir la comunicación del usuario con una aplicación y gestionar sus recursos de una forma eficaz

**Triggers:** Es un evento que se ejecuta cuando se cumple una condición establecida al realizar una operación de inserción (insert), actualización (update) o borrado (delete).

**Vistas:** Es un resultado de una consulta SQL de cero, una o varias tablas.

**XHTML:** lenguaje extensible de marcado de hipertexto, es la versión XML de HTML, por lo que tiene las mismas funcionalidades, pero cumple las especificaciones de XML.

**XML:** (**X**tensible **M**arkup **L**anguage) Lenguaje de Etiquetado Extensible, muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones.