Universidad de las Ciencias Informáticas

Facultad 6

Título: Implementación de una aplicación web para la gestión de la información en estudios de docking.

Trabajo de Diploma para optar por el título de
Ingeniero en ciencias Informáticas

Autor(es): Dayamis Ceballos Ortiz
Persy Morell Guerra

Tutor(es): Dr. Ernesto Moreno
Lic. Taymara Hernández

Junio, 2008

"El secreto de la felicidad no está en hacer siempre lo que se quiere, sino en querer siempre lo que se hace"

León Tolstoi

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 ____ días del mes de _____ del año _____.

Dayamis Ceballos Ortiz

_____

Firma del autor

Persy Morell Guerra

_____

Firma del autor

Lic. Taymara Hernández

_____

Firma del tutor

Dr. Ernesto Moreno

_____

Firma del tutor

DATOS DE CONTACTO

Tutores:
Dr. Ernesto Moreno Frías
Centro de Inmunología Molecular
Email:emoreno@cim.sld.cu

Lic. Taymara Hernández Ortega
Universidad de las Ciencias Informáticas
Email:tay@uci.cu

AGRADECIMIENTOS

A nuestros padres por su apoyo incondicional durante todos estos años.

A nuestros tutores Taymara y Moreno que nos apoyaron aun estando lejos.

A todos los amigos que nos brindaron su apoyo y nos acompañaron en estos maravillosos años de alegrías y tristezas.

A Liesner que siempre estuvo preocupado.

Un agradecimiento mayor para la Revolución y para nuestro Comandante Fidel Castro por darnos la oportunidad de formarnos como ingenieros en este proyecto.

DEDICATORIA

A mi mamita y a mi papito por su apoyo interminable. *A mi hermanito que es el motivo fundamental de mi superación.* *A mis abuelitos que siempre estuvieron conmigo.* *A mis tías que siempre estuvieron preocupadas.* *A todos los primos que me cuidaron y me dieron su apoyo.* *A Tony y Susana que confiaron en mi y no me dejaron abandonar la universidad.* *A mis amigas Yudi, Liyanis, Lianet, Yoslaine, Yurima, Lisbet que a pesar de todo siempre estuvieron conmigo en los buenos y malos momentos y se que me quieren mucho.* *A los locos Jeanlup (viriato), Luisito (to Cagao), Frank (mi hermanito), Oscar (primito), Migue (boquita chiquita), que me quisieron un poquito y me soportaron.* *A Yarmay, Yasenia, Yamira, Miguel Angel (Pichi Pichi), Yuniel (Mangon) que tuve la oportunidad de canecerlos al final pero estuvieron en la buenas y malas.* *A Lázaro (el jefe), que siempre estuvo para darme un buen consejo.* *A Yeniel (chaolin) mi amiguito que siempre estuvo ahí cuando necesitaba conversar.* *A conchita, esa vieja loca, pero linda que me quiso y me ayudo mucho.* *A todos mi amigos..... gracias* *Dayamis*	*A mis padres en especial a mi madre Margarita Guerra,* *A mis hermanos y demás familiares;* *A mi novia Yailen Delgado Reyes,* *A mis amigos y compañeros de curso.* *A todos los que contribuyeron a mi formación como ingeniero y oficial de las FAR* *En especial a Fidel Castro Ruz.* *Persy*

RESUMEN

El presente trabajo tiene como objetivo fundamental facilitar el diseño de fármacos mediante el desarrollo de una aplicación informática. La aplicación web está concebida para que sea capaz de almacenar y gestionar toda la información necesaria en la realización del docking, y así aumentar la tasa de éxito del DOCK y minimizar el tiempo de búsqueda de sitios de enlace proteína-ligando, y permita además realizar un estudio posterior a la obtención de las simulaciones. Cuenta con una base de datos para almacenar la información referente a la estructura de proteínas.

PALABRAS CLAVE

- Gestionar
- Docking
- Ligando
- Bases de Datos
- Simulaciones

TABLA DE CONTENIDO

Índice de figura

INTRODUCCIÓN

El diseño de fármacos es un proceso importante para la prevención de varias enfermedades. En el mismo se utiliza la tecnología más apropiada para realizar búsquedas sistemáticas a gran escala con el propósito de encontrar sustancias cada vez con mayor potencia medicinal y menor toxicidad.

Existe un método de simulación por computadora para el desarrollo de fármacos que ha rendido resultados exitosos: Método de reconocimiento molecular *in sílico* o Docking, el cual es ampliamente utilizado en el mundo por compañías privadas e instituciones de investigación pública para el hallazgo de nuevos compuestos con efectos terapéuticos. Si bien la técnica representa una alternativa a los procedimientos de ensayo masivo de laboratorio, también, y de manera más importante, ha sido utilizada en el diseño de fármacos, tanto para llevar a cabo las etapas de identificación del sitio de unión sobre la molécula blanco, como las de construcción y evaluación de los complejos moleculares resultantes.

Las investigaciones y las técnicas en este campo van en ascenso, por lo que cada día aparecen complejos nuevos y las bases de datos donde se guarda esta información se enriquecen mucho más. Existen varias de estas bases de datos a las que se tiene acceso gratuito a través de la red, pero otras son privadas y la información es de más difícil acceso y una vez obtenida se puede analizar realizando algunos cambios.

En la década del 80, comienzan a crearse en Cuba centros biotecnológicos donde se destaca el Centro de Investigaciones Biológicas (CIB), el Centro de Ingeniería Genética y Biotecnología (CIGB), Centro de Inmunoensayo y el Centro de Inmunología Molecular (CIM). Este último es un centro donde opera una instalación de propósitos múltiples para la fabricación de productos terapéuticos y diagnósticos provenientes de la línea de investigación del propio centro o sobre la base de contratos con otras instituciones.

El Centro de Inmunología Molecular ha hecho una compilación de un gran conjunto de los complejos de proteína – ligando extraído del Protein Data Bank (PDB), que se utiliza tanto para la extracción de conocimientos en búsquedas a gran escala [1]como para poner a prueba distintos algoritmos de acoplamiento. La base de datos Protein Data Bank es la fuente principal de estructuras tridimensionales de proteínas determinadas ya sea por los métodos de difracción de rayos X o por los de resonancia magnética nuclear .Sin embargo el número de algoritmos para predecir las interacciones proteína - ligando de la proteína es cada vez mayor y el número de casos de prueba utilizados para

[1] El término "a gran escala", es usado para nombrar el uso de miles de proteínas que son complejos ligando identificados en la PDB y procesado de modo automático.

validar estos métodos es por lo general pequeño. Recientemente, varias bases de datos se han hecho públicas para seguir la comprensión de las interacciones proteína - ligando, utilizando la base de datos PDB como apoyo. No obstante, parece ser difícil de probar los métodos de acoplamiento sobre una gran variedad de los complejos.

La base de datos PDB, es de acceso gratuito a través de la red. Sin embargo, en la mayor parte de los casos los archivos que contiene esta base de datos no pueden ser utilizados directamente en estudios de diseño de fármacos, sino hasta realizar algunas modificaciones. Aunque cada caso tiene sus peculiaridades, puede generalizarse que las modificaciones más comunes que deben hacerse son: la adición de los átomos de hidrógeno, la remoción de las moléculas de agua que están unidas a la superficie de la proteína, y finalmente; la eliminación de ligandos presentes durante la determinación de la estructura tridimensional de la macromolécula.

Debido a todos estos cambios existentes, el CIM necesita desarrollar una nueva base de datos de complejos proteínas -ligando adaptados para las pruebas de los algoritmos de acoplamiento. Esta base de datos tendría la información modificada que se obtiene de la base de datos PDB y la nueva información generada por sus propios investigadores. Junto con la base de datos necesitan una herramienta de software que le permita gestionar la información almacenada en la misma.

Se puede plantear como **problema científico**: ¿Cómo contribuir a la gestión de la información necesaria para el análisis de la formación de complejos proteína-ligando?

Por lo que el **objeto de estudio** es: La información de las interacciones Proteína-Ligando.

El **campo de acción**: La gestión de la información asociada a los estudios de docking.

La mayoría de las bases de datos existentes como Cambridge Structural Database, NCI, ZINC, ChemStar y otros como TOSlab, Aurora Fine Chemicals y AKos GmbH que son privadas, no proporcionan pistas sobre los modos en que estos complejos pueden interactuar con moléculas de proteínas u otras macromoléculas.

De ahí la **necesidad** de crear una aplicación que permita gestionar la información de la base de datos con la información modificada que se obtuvo de la base de datos PDB, que es la principal fuente de información sobre las estructuras de los complejos proteína -ligando y la generada por los investigadores del CIM. Se conoce que en la mayoría de las interacciones proteína-ligando, por sus características bioquímicas, los enlaces son por puentes de hidrógeno y la incorporación de éste átomo a la base de datos constituye un factor fundamental en la realización y optimización de simulaciones de docking.

El **objetivo general** de esta investigación es implementar una herramienta que permita gestionar la información necesaria para realizar docking.

Por tanto **los objetivos específicos** que se derivan son los siguientes:

- ✓ Realizar el análisis de la base de datos y de la herramienta que permita gestionar la información de la base de datos.
- ✓ Diseñar una base de datos y una herramienta que permita gestionar la información de dicha base de datos.
- ✓ Implementar una herramienta que permita gestionar la información contenida en la base de datos.

Para resolver el problema planteado y alcanzar el cumplimiento de los objetivos se trazaron las siguientes **tareas**:

- Estudio del estado actual de las bases de datos que contienen información relacionadas con los procesos de docking.
- Estudio de las metodologías y herramientas disponibles para el desarrollo de la aplicación.
- Estudio del formato de ficheros de la base de datos Protein Data Bank (PDB).
- Estudio de la estructura de los archivos de entrada al software DOCK.
- Diseño de una base de datos que almacene toda la información necesaria para realizar docking.
- Implementación de una herramienta para gestionar la información que está en la base de datos.

Aportes prácticos esperados del trabajo

El desarrollo de esta base de datos y de la herramienta para gestionar la información, le facilitará a los investigadores el análisis para la obtención del mejor sitio de enlace proteína -ligando. La nueva base de datos constituye un valioso recurso para el desarrollo del conocimiento basado en algoritmos de acoplamiento, y de los programas de pruebas de acoplamiento de sistemas grandes de complejos proteína-ligando. Además puede aumentar la tasa de éxitos de simulaciones en el DOCK.

Estructuración del contenido con una breve explicación de sus partes

Capítulo 1: Fundamentación teórica.

En este capítulo se explicarán las técnicas que se aplican para la obtención de fármacos, así como las distintas bases de datos que existen en el mundo para facilitar la obtención del mejor sitio de enlace proteína -ligando. Se tratarán las variadas tendencias y tecnologías que se utilizarán para el desarrollo

de la base de datos y de la herramienta que les permitirá a los investigadores gestionar la información que está contenida en dicha base de datos.

Capítulo 2: Características del sistema.

En este capítulo se describen los casos de uso encontrados. Se mostrará un diagrama de casos de uso del sistema para un mejor entendimiento del mismo. Se definirán los actores del sistema y los requerimientos funcionales que se implementó. Se plantearán los requerimientos mínimos que se debe tener para poder trabajar con la aplicación.

Capítulo 3: Análisis y diseño del sistema.

En este capítulo se muestran los diagramas de clases del diseño y los diagramas de clases del análisis. Además se muestra los patrones de diseño utilizados, y un diagrama de despliegue para mostrar la distribución física del sistema.

Capítulo 4: Implementación y Prueba.

En este capítulo se describe cómo se implementan los elementos del Modelo de diseño, para esto se muestra el Diagrama de componentes, el código fuente de las principales clases y algunas pantallas de la aplicación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

En este capítulo se explicarán las técnicas que se aplican para la obtención de fármacos, así como las distintas bases de datos que existen en el mundo para facilitar la obtención del mejor sitio de enlace proteína -ligando. Se tratarán las variadas tendencias y tecnologías que se utilizarán para el desarrollo de la base de datos y de la herramienta que les permitirá a los investigadores gestionar la información que está contenida en dicha base de datos.

1.1 Diseño de fármacos.

La primera etapa en el desarrollo de fármacos es la elección del sujeto de estudio, también conocido como molécula blanco, sobre la que actuará el medicamento para lograr el efecto deseado. En las diversas investigaciones sobre los fármacos han revelado que la acción terapéutica de estos incide principalmente en las proteínas .Por esta razón, en el desarrollo de medicamentos las proteínas son la primera opción para fungir como molécula blanco. La siguiente etapa del proceso es la identificación de la molécula líder que son aquellos compuestos que muestran tener una actividad significativamente alta sobre el blanco seleccionado.

Los ensayos masivos que se llevan a cabo en las primeras etapas del desarrollo de fármacos requieren de un gran despliegue de recursos puesto que, además de utilizar grandes bancos de datos, es necesario obtener, procesar y almacenar cantidades colosales de resultados experimentales. Por esta razón, en años recientes se ha optado por los estudios *in sílico,* donde una simulación molecular asistida por computadora realiza los ensayos masivos con el fin de hallar moléculas líder de una forma eficiente y relativamente económica.

1.2 Reconocimiento molecular o Docking.

Existe un método de simulación asistida por computadora para el desarrollo de fármacos que ha dado resultados exitosos. El docking, se utiliza ampliamente alrededor del mundo por compañías privadas e instituciones de investigación pública para el hallazgo de nuevos compuestos con efectos terapéuticos. La técnica del docking ha sido utilizada para el diseño de fármacos, tanto para llevar a cabo las etapas de identificación del sitio de unión sobre la molécula blanco, como las de construcción y evaluación de los complejos moleculares resultantes.

Para describir la metodología del docking es útil dividirla en una serie de etapas (Figura 1). *"La primera de ellas consiste en disponer de la estructura tridimensional de la molécula blanco, la cual debe ser acondicionada para los cálculos subsecuentes y sobre la que debe identificarse el sitio de unión de una molécula de prueba. La siguiente etapa requiere poseer un archivo numeroso de ligandos potenciales, esto es, moléculas orgánicas con estructuras tridimensionales conocidas y que se encuentran en condiciones adecuadas para simular su asociación al blanco. La tercera etapa, que es la parte medular del método, consiste en un algoritmo de computadora que toma cada uno de los ligandos de la base datos y lo coloca dentro del sitio de unión en una gran cantidad de orientaciones."*[1]

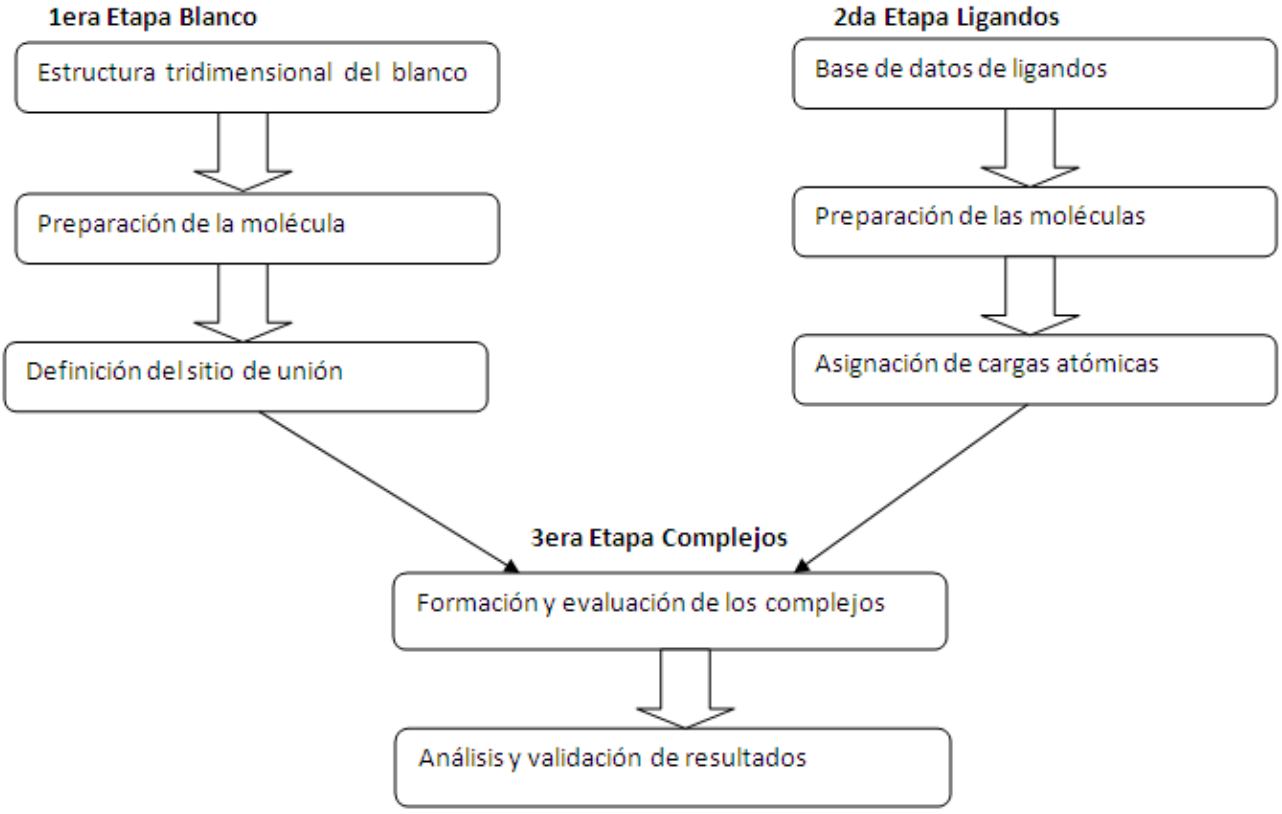


Figura 1 Las etapas de simulación del reconocimiento molecular o docking.

1.3 Selección de ligandos.

Una vez preparada la molécula blanco cuya función se desea alterar y ya seleccionada la cavidad o sitio de unión sobre ella, es necesario realizar una búsqueda de las moléculas orgánicas más afines a este sitio que logren actuar como ligandos potentes. "*Para esta búsqueda se emplean las grandes bases de datos de compuestos orgánicos que están disponibles a través de la red* "[1] y que provienen tanto de fuentes públicas como privadas. Aunque el acceso a las primeras es gratuito, la consulta a las segundas resulta bastante onerosa.

Estas bases de datos, que contienen miles de moléculas distintas, son de gran utilidad en el proceso de *docking*, donde la intención es realizar un muestreo amplio de los compuestos existentes para así incrementar la posibilidad de éxito en la identificación de sustancias potencialmente útiles como fármacos. Muchas de las moléculas contenidas en las bases de datos, que provienen de empresas privadas, poseen las características más comunes observadas en los medicamentos conocidos, situación que no es compartida por las fuentes de información públicas. Para hacer uso de los compuestos presentes en cualquiera de estos dos tipos de bases de datos, es necesario someterlos una serie de etapas de adecuación previas a la simulación por computadora.

El acondicionamiento a que deben sujetarse las moléculas:

a)" deben añadirse átomos de hidrógeno, ya que en muchos complejos proteína-ligando están involucradas las interacciones de puentes de hidrógeno.

b) debe darse una representación tridimensional a las moléculas, pues en la mayoría de los casos sólo se cuenta con una imagen plana o una fórmula codificada de ellas.

c) debe estimarse la carga parcial de cada uno de los átomos que constituyen al compuesto". [2-3]

"En esta última tarea, debe considerarse que las moléculas pequeñas poseen una mayor variedad de tipos de átomo y grupos funcionales que las proteínas conocidas. Por ello, en vez de emplear tablas de valores promedio para la asignación de cargas, se utiliza un algoritmo basado en cálculos de electronegatividad de los átomos involucrados en cada enlace químico"[4].Debido a que la serie de etapas de preparación debe realizarse sobre cada una de las moléculas que integran las enormes bases de datos de compuestos químicos, se han desarrollado paquetes de cómputo que realizan esta ardua tarea de forma automática y eficiente.

1.4 Bases de datos que guardan los ficheros con información de las proteínas.

Las bases de datos biológicos se han convertido en un instrumento importante para ayudar a los científicos a comprender y explicar una serie de fenómenos biológicos. Este conocimiento ayuda a facilitar la lucha contra las enfermedades y ayuda en el desarrollo de medicamentos. El conocimiento

biológico se distribuye entre múltiples bases de datos generales y especializadas. Esto a veces hace que sea difícil garantizar la coherencia de la información. Las bases de datos biológicas tienen referencias cruzadas con otras bases de datos con el número de acceso como una forma de vincular sus conocimientos relacionados con el conjunto.
Ejemplo de algunas bases de datos que hemos estudiado para el desarrollo de nuestro trabajo son:

ProSite

La base de datos ProSite contiene información detallada sobre todos los motivos de secuencia de proteína conocidos (es decir, sitios estructurales o funcionales bien caracterizados), indicando no solamente el motivo hallado, sino también información descriptiva, informes detallados de significatividad y referencias bibliográficas. Además consta de una gran colección de firmas biológicamente significativas que se describen como patrones o perfiles. Cada firma está vinculada a una documentación que da información biológica útil a la familia de proteínas. Ha sido desarrollada por Amos Bairoch en el nodo de EMBnet ExPaSy de Suiza.

DSSP

DSSP es una base de datos de alineamientos de estructura secundaria, enriquecida con información adicional añadida, de todas las proteínas catalogadas en la base de datos PDB. Ha sido desarrollada en el EMBL por Chris Sander.

FSSP

Este es un banco de datos de familias de proteínas estructuralmente parecidas, es decir, de alineamientos múltiples estructurales, desarrollada por el Grupo de Diseño de Proteínas del EMBL. Es de utilidad para hallar nueva información estructural y reglas generales de plegamiento, así como para descubrir nuevas unidades de plegamiento.

HSSP

Es una base de datos de estructuras de proteínas deducidas por homología; es de interés para el análisis estructural de nuevas proteínas, la catalogación de familias de proteínas y los alineamientos estructurales.

InterPro

Es una estrategia integrada de recursos de documentación de las familias de proteínas, dominios y sitios funcionales, se creó en 1999 para unir las principales bases de datos de proteínas en un amplio

recurso. Los resultados se presentan en un único formato amplio, con enlaces a las fuentes originales de datos, así como las bases de datos especializadas. La última versión de InterPro contiene más de 10000 entradas, con 78% de cobertura de todas las proteínas en UniProt.La base de datos está disponible a través de un servidor web (http://www.ebi.ac.uk/interpro).

Protein Data Bank

Es una base de datos en 3-D sobre estructura de las proteínas y ácidos nucleicos. Estos datos, generalmente obtenidos por Cristalografía de rayos X o Resonancia Magnética Nuclear, son enviados por biólogos y bioquímicos de todo el mundo. Están bajo el dominio público y pueden usarse libremente.

Todas esta bases de datos contiene mucha información sobre la estructura de las proteínas, pero no todas pueden utilizarse directamente por los investigadores del CIM en el desarrollo de fármacos ya que la información no puede ser utilizada directamente hasta hacerle varias modificaciones además de que alguna de ellas son privadas .Para el desarrollo de la aplicación nos centramos principalmente en la PDB por que es la base de datos que los investigadores del CIM están utilizando en estos momentos y es una de las más completas.

1.5Herramientas y metodologías utilizadas en la aplicación.

Para el desarrollo de una aplicación de software se deben definir los tipos de tecnologías a utilizar así como las metodologías y herramientas que serán de mayor utilidad para su implementación. A continuación se explica en detalle cada una de las tecnologías que fueron seleccionadas para llevar a cabo la implementación y documentación de la base de datos y de la herramienta para gestionar la información.

1.5.1 Metodologías de desarrollo de software

Cuando se va a desarrollar una aplicación de software se tienen algunas dificultades, algunas de ellas pueden ser: definir qué artefactos deben ser creados, saber cómo organizar las actividades para los desarrolladores por separado y por equipo; por lo que se necesita de una metodología capaz de dirigir estas actividades. Existen distintas metodologías para dirigir las actividades vinculadas al proceso de desarrollo de software, entre ellas se estudiaron las siguientes:

RUP

El Proceso Unificado del Rational (RUP) 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.

El RUP es un proceso de desarrollo de software y además un marco de trabajo genérico que esta basado en componentes y además utiliza UML. Tiene como características principales: estar dirigido a casos de usos, centrado en la arquitectura y es iterativo e incremental. Una característica importante es que permite corregir errores en cada iteración y es flexible a cambios en los requerimientos.

OpenUP (Proceso Unificado Abierto)

Es una versión simplificada del IBM Rational Unified Process (RUP) optimizado para los pequeños proyectos. Conserva las características esenciales del RUP, que incluye el desarrollo iterativo, casos de uso y escenarios de conducción de desarrollo, gestión de riesgos, y el enfoque centrado en la arquitectura.

El uso de OpenUP se debe realizar cuando se tiene un equipo pequeño, cuando se quiere evitar ser cargado excesivamente con metodologías formales improductivas.

Una manera buena de entender que OpenUP es pensar en el como: un proceso ágil, ligero, que promueve el desarrollo del software las prácticas más buenas, haciéndolo un proceso pequeño extensible si es necesario.

Este modelo de desarrollo es perfecto para proyectos pequeños, según el gráfico de roles, pueden participar 4 personas. Y como este modelo tiene las bases del RUP, aun mantiene el análisis de riesgos, que hoy por hoy es uno de los campos que no se puede descuidar.

- Apoya la agilidad en el sentido de que haya un número reducido de las funciones, tareas y artefactos.
- Apoya la flexibilidad, ya que su organización en los paquetes y la separación entre el método y el proceso permite la selección de elementos deseados para crear el proceso.

Por estas características se eligió como la metodología a emplearse durante el proceso de desarrollo de este software.

1.5.2 Herramientas CASE.

Rational Rose

Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML. Utiliza un proceso de desarrollo iterativo controlado .Permite que haya varias personas trabajando a la vez en el proceso iterativo controlado, además proporciona mecanismo para realizar la ingeniería inversa.

Es una de las más poderosas herramientas de modelado visual para el análisis y diseño de sistemas basados en objetos. Se utiliza para modelar un sistema antes de proceder a construirlo. Cubre todo el ciclo de vida de un proyecto:

- Concepción y formalización del modelo.
- Construcción de los componentes.
- Transición a los usuarios.
- Certificación de las distintas fases.

Visual Paradigm

Es una potente herramienta CASE para visualizar y diseñar elementos de software, para ello utiliza el lenguaje UML, proporciona a los desarrolladores una plataforma que les permita diseñar un producto con calidad de una forma rápida. Facilita la interoperabilidad con otras herramientas CASE y se integra con las siguientes herramientas Java: Eclipse/IBM WebSphere, Jbuilder, NetBeans IDE, Oracle Jdeveloper, BEA Weblogic. Está disponible en varias ediciones: Enterprise, Professional, Community, Standard, Modeler y Personal.

Debido a que el sistema operativo que se está utilizando en el proyecto es Ubuntu (distribución de Linux) se decidió utilizar el Visual Paradigm para visualizar y diseñar los elementos de software.

1.5.3 Lenguaje de Modelado

UML

El Proceso Unificado de Modelado (UML) es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos. Ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. Cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

1.5.4 Lenguajes de programación.

PHP

PHP *es un lenguaje de scripting que permite la generación dinámica de contenidos en un servidor web. El significado de sus siglas es HyperText Preprocessor. Entre sus principales características cabe destacar su potencia, su alto rendimiento, su facilidad de aprendizaje y su escasez de consumo de recursos.*[5]

Entre las ventajas podemos citar:

- ✓ "*Muy sencillo de aprender.*
- ✓ *Similar en sintaxis a C y a PERL*
- ✓ *Soporta en cierta medida la orientación a objeto. Clases y herencia.*
- ✓ *El análisis léxico para recoger las variables que se pasan en la dirección lo hace PHP de forma automática. Librándose el usuario de tener que separar las variables y sus valores.*
- ✓ *Excelente soporte de acceso a base de datos.*
- ✓ *La comprobación de que los parámetros son validos se hace en el servidor y no en el cliente (como se hace con javascript) de forma que se puede evitar chequear que no se reciban solicitudes adulteradas. Además PHP viene equipado con un conjunto de funciones de seguridad que previenen la inserción de órdenes dentro de una solicitud de datos.* "[6]

Por que utilizar PHP y no otras opciones:

- "*PHP no soporta directamente punteros, como el C, de forma que no existen los problemas de depuración provocados por estos.*
- *El código PHP es mucho más legible que el de PERL, todo el que haya programado PERL podrá corroborar esta afirmación.*
- *Viene acompañado por una excelente biblioteca de funciones que permite realizar cualquier labor (acceso a base de datos, encriptación, envió de correo, gestión de un e-commerce, XML, creación de PDF ...)*
- *Esta siendo utilizado con éxito en varios millones de sitios web.*
- *Hay multitud de aplicaciones php para resolver problemas concretos (weblogs, tiendas virtuales, periódicos,...) listas para usar.*
- *Es multiplataforma, funciona en todas las plataformas que soporten apache.*
- *Es software libre. Se puede obtener en la web y su código esta disponible bajo la licencia GPL.*

"[6]

JavaScript

L*a característica principal de Javascript, de hecho, es la de ser un lenguaje de scripting, pero, sobre todo, la de ser el lenguaje de scripting por excelencia y, sin lugar a dudas, el más usado. Esta particularidad conlleva una notable serie de ventajas y desventajas según el uso que se le deba dar y teniendo en cuenta la relación que se establece entre el mecanismo cliente-servidor*. [7]

Ventajas de javascript:

- *"Los scripts se pueden desarrollar en un periodo de tiempo relativamente corto.*
- *No requieren mucha memoria ni tiempo adicional de transmisión.*
- *Al incluirse dentro de las mismas páginas HTML se reduce el número de accesos independientes a la red."[8]*

1.5.5 Herramienta de desarrollo

Dreamweaver

Dreamweaver es la herramienta de diseño de páginas web más avanzada, además cumple el objetivo de diseñar páginas web con aspecto profesional. Tiene soporte tanto para edición de imágenes como para animación a través de su integración con otras herramientas. Dreamweaver permite al usuario utilizar la mayoría de los navegadores Web instalados en su ordenador para previsualizar las páginas web. También dispone de herramientas de administración de sitios dirigidas a principiantes como, por ejemplo, la habilidad de encontrar y reemplazar líneas de texto y código por cualquier tipo de parámetro especificado. El panel de comportamientos también permite crear JavaScript básico sin conocimientos de código.

Zent Studio

Zend Studio es un editor de texto para páginas PHP que proporciona un buen número de ayudas desde la creación y gestión de proyectos hasta la depuración del código.

Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Las dos partes se instalan por separado, la del cliente contiene la interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para aprovechar toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor, que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración.

Aptana

Aptana es un entorno gratuito y open source, orientado al desarrollo de aplicaciones web basadas en AJAX. Entre las funcionalidades del mismo están las pistas de código para JavaScript, XML, HTML o CSS, o el potente debugger JavaScript. Aptana es multiplataforma, y se puede descargar en versiones para Windows, Mac OS X, Linux, y como plugin de Eclipse.

Quanta Plus

Quanta Plus es una herramienta de desarrollo web para el entorno de escritorio**.** Quanta Plus proporciona un interfaz de múltiples documentos (MDI) poderosa e intuitiva para los desarrolladores web. Puede incrementar exponencialmente su productividad. A través del uso de acciones personalizadas, guiones y barras de herramientas, puede automatizar casi cualquier tarea.

1.5.6 Gestor de Base de datos

MySQL

Es un gestor de base de datos sencillo de usar y rápido. También es uno de los motores de base de datos más usados en Internet.

Las características principales de MySQL son:

- "*Es un gestor de base de datos. Una base de datos es un conjunto de datos y un gestor de base de datos es una aplicación capaz de manejar este conjunto de datos de manera eficiente.*
- *Es una base de datos relacional. Una base de datos relacional es un conjunto de datos que están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y segura. Para usar y gestionar una base de datos relacional se usa el lenguaje estándar de programación SQL.*
- *Es Open Source. El código fuente de MySQL se puede descargar y está accesible a cualquiera.*
- *Es una base de datos muy rápida, segura y fácil de usar.*
- *Probado con un amplio rango de compiladores diferentes.*
- *Funciona en diferentes plataformas.*
- *Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible. Normalmente no hay reserva de memoria tras*

toda la inicialización para consultas.

- *Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verficación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor.*
- *Los clientes pueden conectar con el servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows de la familia NT (NT,2000,XP, o 2003), los clientes pueden usar named pipes para la conexión. En sistemas Unix, los clientes pueden conectar usando ficheros socket Unix.* "[9]

1.5.7 Otras herramientas utilizadas

AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano. 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 aplicación. Es un término que describe un nuevo acercamiento a usar un conjunto de tecnologías existentes juntas, incluyendo las siguientes: HTML o XHTML, hojas de estilo (Cascading Style Sheets), Javascript, el DOM (Document Object Model), XML, XSLT, y el objeto XMLHttpRequest. Cuando se combinan estas tecnologías en el modelo Ajax, las aplicaciones funcionan mucho más rápido, ya que las interfaces de usuario se pueden actualizar por partes sin tener que actualizar toda la página completa.

YUI-EXT

Yui-ext(YAHOO USER INTERFACE) es el nombre que tuvo originalmente la biblioteca JavaScript desarrollada por Jack Slocum, y que en su versión actual se llama Ext (o también Ext JS). En un comienzo era una extensión a la biblioteca YUI de Yahoo! (de ahí el nombre); ahora puede utilizarse tanto con YUI como con Prototype o jQuery como bibliotecas de base.
Ext facilita el desarrollo de aplicaciones Web con interfaces gráficas de gran calidad, escribiendo una cantidad relativamente pequeña de código.

1.6 Estilo arquitectónico.

Se utilizó la arquitectura en capas ya que esta permite que el desarrollo se pueda llevar a cabo en varios niveles y en caso de algún cambio solo se ataca al nivel requerido sin tener que revisar entre código mezclado.

1. La capa de presentación o interfaz de usuario.

En este caso, esta formada por los formularios y los controles que se encuentran en los formularios. Capa con la que interactúa el usuario.

2. La capa de negocio.

Es la capa que sirve de comunicación entre la capa de presentación y la capa de acceso a datos.

3. La capa de acceso a datos.

Es la capa que controla el acceso a los datos. Puede estar formada por un gestor de base de datos que realiza todo el almacenamiento de datos.

1.7 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. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobada su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reusable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

1.7.1 Patrones utilizados.

1.7.1.1Patrón Adapter (adaptador) (GOF)

“El patrón Adapter (Adaptador) se utiliza para transformar una interfaz en otra, de tal modo que una clase que no pudiera utilizar la primera, haga uso de ella a través de la segunda.”

Propósito

“Convierte la interfaz de una clase en otra interfaz que el cliente espera. Adapter permite a las clases trabajar juntas, lo que de otra manera no podrían hacerlo debido a sus interfaces incompatibles.”

Participantes

“Target define la interfaz específica del dominio que Client usa.

Client colabora con la conformación de objetos para la interfaz Target.

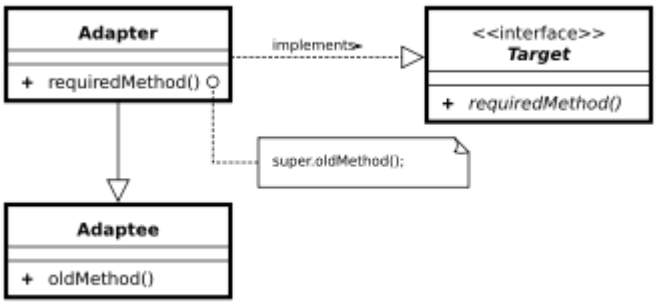
Adaptee define una interfaz existente que necesita adaptarse.

Adapter adapta la interfaz de Adaptee a la interfaz Target.”

Colaboraciones

“Client llama a las operaciones sobre una instancia Adapter. De hecho, el adaptador llama a las operaciones de Adaptee que llevan a cabo el pedido. [10]

Estructura



1.7.1.2 Patrón Comand(comando)(GOF)

“Intención: Encapsula una petición en un objeto. Permite con ello parametrizar a los clientes con diferentes peticiones y almacenar peticiones para deshacerlas en caso necesario
Motivación: Parametrización de las acciones a realizar por los objetos GUI de un framework.
Participantes

COMANDO: Declara la intefaz para ejecutar una operación.

COMANDO_CONCRETO: Define una relación entre un receptor y una acción, redefiniendo la operación ejecutar para que se envíe una petición adecuada al receptor.

CLIENTE: Crea un comando concreto indicándole cuál es su receptor y lo almacena en su emisor

EMISOR: Pide al comando que lleve a cabo una petición.

RECEPTOR: Sabe cómo realizar la operación relacionada con una petición.

Colaboraciones

- *El cliente crea un comando concreto indicándole cuál es su receptor.*
- *El emisor almacena uno o varios comandos concretos.*
- *El emisor solicita al comando que se ejecute.*
- *El comando pide al receptor que realice las operaciones necesarias."[10]*

1.7.1.3 Patrón GRASP Bajo Acoplamiento.

"Solución

Asignar una responsabilidad para mantener bajo acoplamiento.

Problema

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

El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con alto (o fuerte) acoplamiento recurre a muchas otras. Este tipo de clases no es conveniente: presentan los siguientes problemas:

- *Los cambios de las clases afines ocasionan cambios locales.*
- *Son más difíciles de entender cuando están aisladas.*
- *Son más difíciles de reutilizar porque se requiere la presencia de otras clases de las que dependen.*

El Bajo Acoplamiento soporta el diseño de clases mas independientes, que reducen el impacto de los cambios, y también mas reutilizables, que acrecientan la oportunidad de una mayor productividad.

Beneficios

- *No se afectan por cambios de otros componentes.*
- *Fáciles de entender por separado.*
- *Fáciles de reutilizar."[11]*

1.7.1.4 Patrón GRASP Alta Cohesión.

"Solución

Asignar una responsabilidad de modo que la cohesión siga siendo alta.

Problema

¿Cómo mantener la complejidad dentro de límites manejables?

En la perspectiva del diseño orientado a objetos, la cohesión (o, mas exactamente, la cohesión funcional) es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme.

Una clase con baja cohesión hace muchas cosas no afines o un trabajo excesivo. No conviene este tipo de clases pues presentan los siguientes problemas:

- *Son difíciles de comprender*
- *Son difíciles de reutilizar.*
- *Son difíciles de conservar*
- *Son delicadas: las afectan constantemente los cambios*

Beneficios

Mejoran la claridad y la facilidad con que se entiende el diseño.

Se simplifican el mantenimiento y las mejoras en funcionalidad.

A menudo se genera un bajo acoplamiento.

La ventaja de una gran funcionalidad soporta una mayor capacidad de reutilización, porque una clase muy cohesiva puede destinarse a un propósito muy específico."[11]

1.7.1.5 Patrón GRASP Creador

"Solución: Asignarle a la clase B la responsabilidad de crear una instancia de clase A en uno de los siguientes casos:

B agrega 10s objetos A.

B contiene 10s objetos A.

B registra las instancias de los objetos A.

B utiliza específicamente los objetos A.

B tiene los datos de inicialización que serán transmitidos a A cuando este objeto sea creado (así que B es un Experto respecto a la creación de A).

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

La creación de objetos es una de las actividades mas frecuentes en un sistema orientado a objetos. En consecuencia, conviene contar con un principio general para asignar las responsabilidades concernientes a ella. El diseño, bien asignado, puede soportar un bajo acoplamiento, una mayor claridad, el encapsulamiento y la reutilizabilidad.

Explicación: 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 debemos conectar con el objeto producido en cualquier evento.

Beneficios: Se brinda soporte a un bajo acoplamiento, lo cual supone menos dependencias respecto al mantenirniento y mejores oportunidades de reutilización. Es probable que el acoplamiento no aumente, pues la clase creada tiende a ser visible a la clase creador, debido a las asociaciones actuales que nos llevaron a elegirla como el parámetro adecuado."[11]

1.8 Conclusiones

1. Para el desarrollo del sistema se utilizó la metodología OpenUP debido a que es la más adaptable para proyectos cortos y permite entregar un software con calidad.
2. Para el almacenamiento y gestión de los datos que se almacenan se consideró utilizar MySql porque trabaja en múltiples plataformas, se puede acceder a él de forma gratuita y es de mucha calidad, rápido y confiable.
3. El lenguaje utilizado es Javascript por ser multiplataforma y PHP por su fácil trabajo para el desarrollo web.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.

En este capítulo se describen los casos de uso encontrados. Se mostrará un diagrama de casos de uso del sistema para un mejor entendimiento del mismo. Se definirán los actores del sistema y los requerimientos funcionales que se implementó. Se plantearán los requerimientos mínimos que se debe tener para poder trabajar con la aplicación.

2.1 Actores del sistema a automatizar

El actor del sistema puede ser un sistema externo que interactúe con dicho sistema, puede ser o no, una persona. Aquí veremos los actores del sistema en cuestión.

Tabla 1. Definición de actores del sistema a automatizar

Nombre del actor	Descripción
Administrador	Es aquella persona que va a interactuar con el sistema, pero además, puede modificar, insertar y eliminar información en la base de datos.
Laboratorista	Es aquella persona que va a interactuar con el sistema.

2.2 Requisitos Funcionales

Servicios que el sistema le proporciona al usuario.

Los requisitos funcionales de este sistema son:

R-1Autenticar usuario.

R-2Registrar usuario.

R-3Gestionar información

R3.1Insertar información sobre las proteínas.

R3.2 Actualizar información sobre las proteínas.

R3.3 Eliminar información sobre las proteínas.

R-4Guardar reporte

R4.1Salvar reporte.

R4.2 Imprimir reporte.

R-5Gestionar consulta

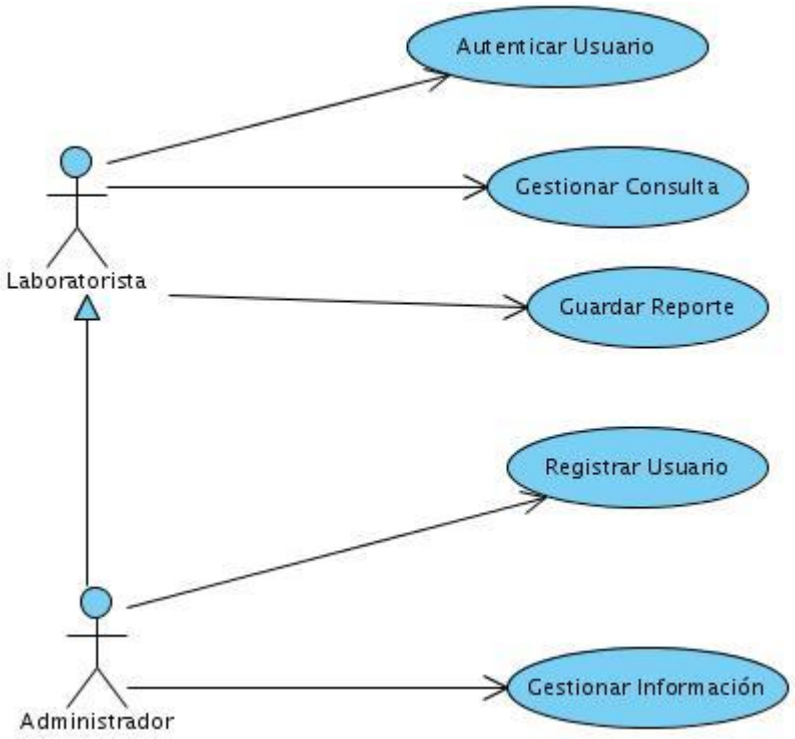
R5.1Realizar consulta en la base de datos.

R5.2Graficar consulta en la base de datos.

R5.3Realizar resumen de consulta en la base de datos.

R5.4Analizar consulta en la base de datos.

2.3Diagrama de casos de uso del sistema a automatizar



2.4 Requisitos no funcionales

Los requisitos no funcionales son restricciones al diseño o funcionamiento del sistema tal como requisitos de funcionamiento, estándares de calidad, requisitos del diseño, etc.

- Apariencia o interfaz externa:
  La interfaz del sistema será a través de la web. Se deberán usar colores y diseño acorde con el usuario final. En este caso, al ser un centro de investigación científica y de servicios médicos se deberán usar colores y diseño sobrios y a la vez refrescantes. Además no debe cargarse de imágenes para que el acceso se facilite.

- Usabilidad:

Se debe garantizar un acceso fácil y rápido a los usuarios. El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de una computadora y de un ambiente web en sentido general.

- Rendimiento:

Los tiempos de respuestas deben ser generalmente rápidos al igual que la velocidad de procesamiento de la información.

- Soporte:
  - Servidor:
    Se requiere de cualquier servidor con las siguientes características:
    PHP 4.1.2 o superior.
    MySQL 3.23.55 o superior.
    Apache 1.3 o superior.

  - Cliente:
    Por parte del cliente se requiere un navegador capaz de interpretar JavaScript .

- Portabilidad:
  El sistema debe ser multiplataforma. Con preferencia a los ambientes Unix .

- Seguridad:
  Garantizar que la información sensible solo pueda ser vista por los usuarios con el nivel de

acceso adecuado. Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que este activo. Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos. Verificación sobre acciones irreversibles (eliminaciones).

- Políticos-culturales:
  Se debe hacer uso correcto del idioma inglés en la interfaz pública de la aplicación, con logotipos e imágenes que se encuentren en correspondencia con el carácter científico y profesional del centro al que va destinada.

- Legales:
  Las herramientas y las tecnologías en que este basado el proyecto deben cumplir con las licencias de software GPL.

- Confiabilidad:
  Validar la captación de datos para evitar entradas inadecuadas.

- Ayuda:
  Se les brindará ayuda a los usuarios donde se requiera y los contenidos estarán vinculados al escenario accedido.

- Software:
  Navegador compatible o superior con Internet Explorer 5.5+, Netscape, Mozilla 1.7 o superior o FireFox 0.9.3 o superior.

- Hardware:
  No existen restricciones específicas en cuanto al servidor o a las máquinas de los usuarios, solo deben tener conexión a red y recursos básicos de una PC.

- Restricciones en el diseño y la implementación:
  La presentación constituirá una capa independiente de la lógica de negocio, centrando su función en la interfaz de usuario y validaciones simples de los datos de entrada. Debe ser una aplicación web desarrollada con la tecnología para creación de páginas dinámicas PHP y base de datos en MySQL. Se utilizarán herramientas de desarrollo que garanticen la calidad de todo el ciclo de desarrollo del producto.

- Requisitos de interfaz interna:

  Todos los componentes del sistema deben desarrollarse siguiendo el principio de máxima cohesión y mínimo acoplamiento.

- Disponibilidad

  El mantenimiento de la aplicación debe ser el menor posible antes los cambios continuos en la base de datos Screening.

2.5 Casos de uso del sistema.

1. Autenticar usuario.
2. Registrar usuario.
3. Gestionar información.
4. Guardar reporte.
5. Gestionar consulta.

2.6 Descripción de los casos de uso del sistema.

Descripción Caso de Uso Gestionar consulta.

Nombre del CU	Gestionar Consulta.
Actores	Laboratorista (Inicia).
Propósito	Permitir el trabajo con la consulta.
Resumen	El Caso de Uso se inicia cuando el usuario selecciona Gestionar Consulta, seleccionando una de las siguientes opciones: • Realizar consulta. • Graficar consulta. • Realizar resumen de consulta. • Analizar consulta. Termina el caso de uso cuando se realiza cualquiera de las opciones anteriores.
Referencias	R 5.1, R 5.2, R 5.3, R 5.4,
Precondiciones	El usuario debe estar autenticado.
Poscondiciones	
Curso Normal de los Eventos	

Acciones del Actor	Respuesta del Sistema
1. El usuario selecciona realizar alguna de las opciones: • Realizar consulta. • Graficar consulta • Realizar resumen de consulta. • Analizar consulta.	2. El sistema, en dependencia de la operación que solicita realizar, hace lo siguiente: - Si el usuario decide realizar una consulta, ir a: **Realizar consulta** ¨. - Si el usuario decide graficar consulta, ir a: ¨ **Graficar consulta¨.** - Si usuario decide Realizar resumen de consulta, ir a Sección ¨ **Realizar resumen de consulta** ¨**.** - Si el usuario decide Analizar consulta, ir a Sección ¨ **Analizar consulta** ¨**.**
	3. Termina el caso de uso Gestionar consulta.
Sección: Realizar consulta	
1. Selecciona la opción select.	2. Muestra la interfaz que le permite realizar consulta.
3. El laboratorista selecciona las tablas.	4. El sistema crea la consulta básica
	5. Realiza la consulta.
	6.Muestra el resultado
Sección: Graficar consulta.	
1. Selecciona la opción graficar.	2. Muestra la interfaz que permite graficar consulta.
3. El laboratorista seleccciona la opción crear tabla.	4. Crea las tablas y las muestra.
5.Selecciona el nombre de la tabla	6.Muestra los campos de la tabla seleccionada
7.Selecciona o no los campos de la tabla	8.Realiza la consulta de modo grafico
	9.Muestra el resultado
Sección: Realizar Resumen de consulta	
1. Selecciona la opción resumir	1. Muestra la interfaz que le permite resumir la consulta.

2. El laboratorista selecciona el campo que desea resumir.	3. Hace un resumen de la consulta
	4. Muestra el resumen de la consulta.
Sección: Analizar consulta	
1. Selecciona la opción analizar.	2. Muestra la interfaz que le permite analizar la consulta.
3. El laboratorista selecciona los campos que desea analizar.	4. El sistema permite analizar la consulta.
Prioridad:	Crítico

Descripción Caso de Uso Autenticar usuario. [Anexo 1].

Descripción Caso de Uso Registrar usuario. [Anexo 2].

Descripción Caso de Uso Guardar reporte. [Anexo 3].

Descripción Caso de Uso Gestionar información. [Anexo 4].

2.7 Conclusiones

En este capítulo se hizo un análisis de todos los procesos que se llevan a cabo en el CIM para realizar screening y docking. Se definieron los actores que interactúan con el sistema. Se plantearon además los requisitos funcionales y no funcionales, que no eran más que las funcionalidades que debe cumplir el software junto con sus cualidades. Se hizo una descripción de cada uno de los casos de uso del sistema.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

En este capítulo se muestran los diagramas de clases del diseño y los diagramas de clases del análisis. Además se muestra los patrones de diseño utilizados, y un diagrama de despliegue para mostrar la distribución física del sistema.

3.1 Diagrama de clases del Análisis.

Diagrama de clases del análisis (Gestionar consulta).

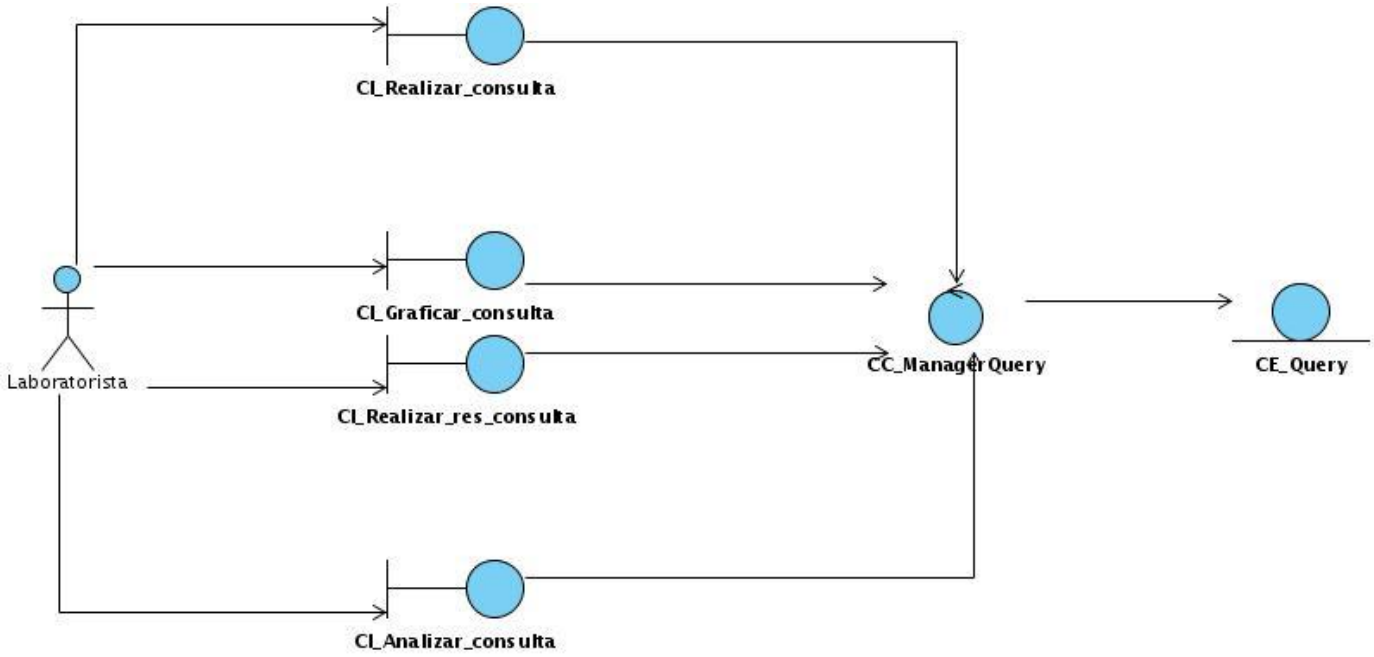


Diagrama de clases del análisis (Autenticar usuario) [Anexo 5].

Diagrama de clases del análisis (Registrar usuario) [Anexo 6].

Diagrama de clases del análisis (Guardar reporte) [Anexo 7].

Diagrama de clases del análisis (Gestionar información) [Anexo 8].

3.2 Estilo arquitectónico utilizado.

Se utilizó la arquitectura en capas. Una capa de interfaz de usuario que seria la capa de presentación, se tiene una capa de negocio (las clases del negocio), y una capa de acceso a datos (las clases de acceso a datos).

3.3 Patrones de diseño utilizados.

Patrón GOF adapter.

Se modificó una función loadData, para cuando se genere una consulta dinámica la muestre se en la interfaz con el formato deseado.

Patrón GOF comand.

Se utiliza cuando el cliente realiza una consulta al servidor haciendo uso de la técnica de AJAX , especificando por el método GET el comando a ejecutar, así como la pagina servidora (receptor).En el cliente se crea la consulta dinámica , que posteriormente es enviada a la clase ManagerData(emisor)que almacena el comando y lo ejecuta en el receptor.

Patrón GRASP Alta cohesión.

Se utilizo en la capa de acceso a datos, en las clases MySql y Conector, con el objetivo de que sus funciones cumplieran estrictamente con las responsabilidades asignadas a dichas clases.

Patrón GRASP Bajo acoplamiento.

La clase ManagerData utiliza la clase ManagerUser, pero la recurrencia a dicha clase es poca por lo que la clase ManagerData es casi independiente

Patrón GRASP Creador.

Se utiliza en la clase ManagerData cuando dicha clase contiene los datos de inicialización de ManagerUser convirtiéndose en un experto respecto a la creación de ManagerUser.

3.4 Diagrama de clases Web del Diseño.

Un requisito fundamental del cliente consiste en hacer que el mantenimiento y disponibilidad de la aplicación fuese el menor posible antes los cambios continuos en la base de datos Screening ,de esta forma la aplicación ha sido diseñada e implementada para que funcione con total independencia de los cambios que ocurran en la base de datos , ya que las funciones de puntuación que utiliza el software DOCK son programadas por el cliente y requieren de información existente en la base de datos Screening , estas funciones pueden devolver información útil al cliente y este a su ves puede alterar la estructura de la base de datos añadiéndole o eliminándole campos y tablas, dependiendo del rol que desempeñe en ese momento.

Diagrama de clases Web del Diseño (Gestionar consulta).

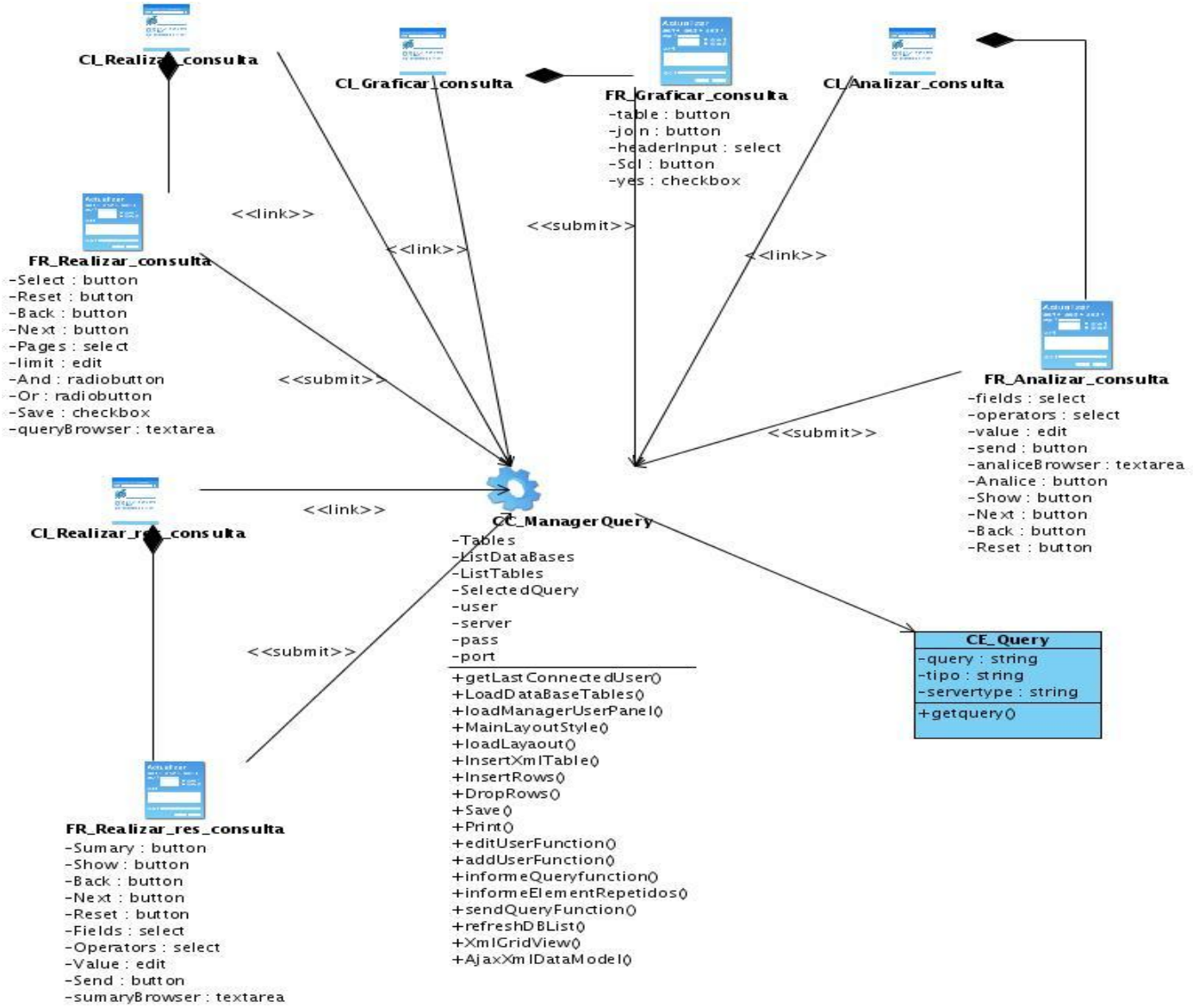


Diagrama de clases Web del Diseño (Autenticar usuario) [Anexo 9].

Diagrama de clases Web del Diseño (Registrar usuario) [Anexo 10].

Diagrama de clases Web del Diseño (Guardar reporte) [Anexo 11].

Diagrama de clases Web del Diseño (Gestionar información) [Anexo 12]

3.5 Diseño de la base de datos.

La aplicación cuenta con una base de datos para facilitar a los usuarios la gestión de la información .El diseño de la base de datos propuesto , teniendo siempre en cuenta que la aplicación es un generador de consulta dinámico .Las consultas son generadas a partir de un esquema relacional de las bases de

datos que no permite implementar las clases persistentes para hacer uso de la Programación Orientada a Objeto (POO).Las clases persistentes representan información de larga duración o que persiste en el tiempo, es decir, es la información que se requiere almacenar para gestionarla en cualquier momento. En nuestro caso una ves que el usuario se autentifica se carga el esquema de la base de datos a partir del cual se generan las consultas dinámicas y se hacen las peticiones al servidor.

Tanto el modelo lógico de datos como el modelo físico de datos fueron generados asumiendo en un momento dado la estructura de la base de datos.

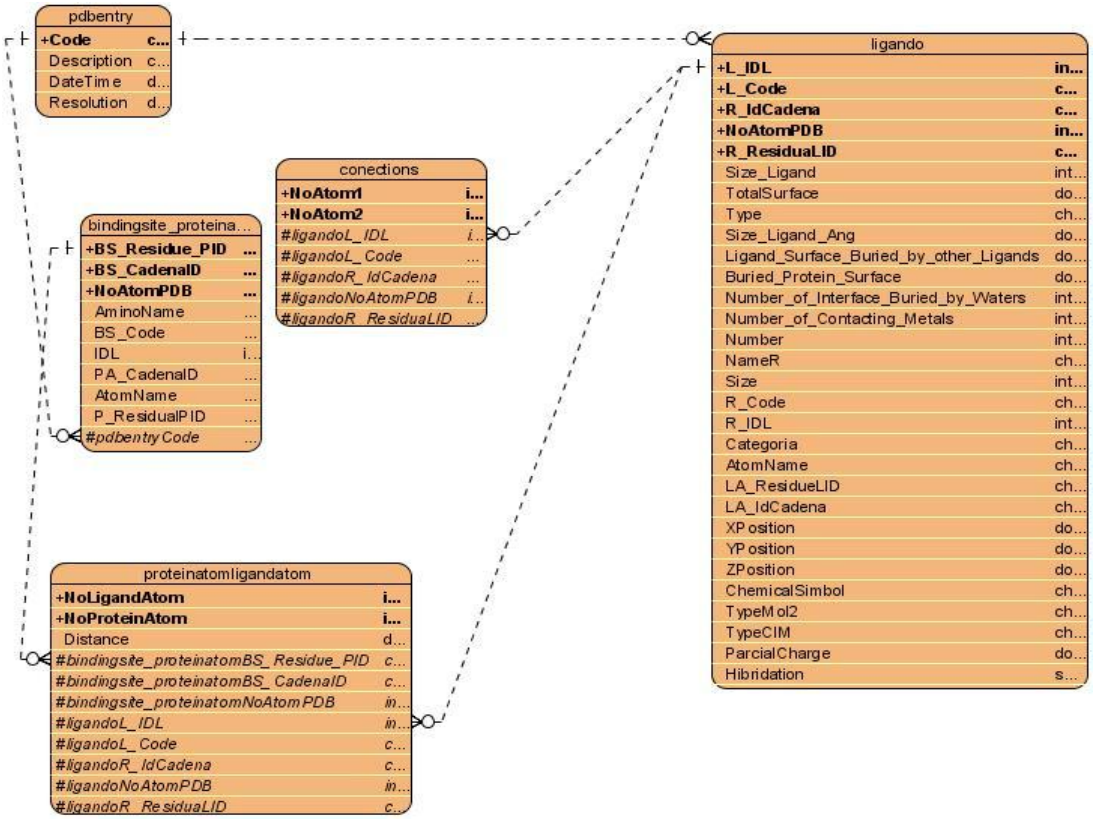


Figura 2 Modelo lógico de datos

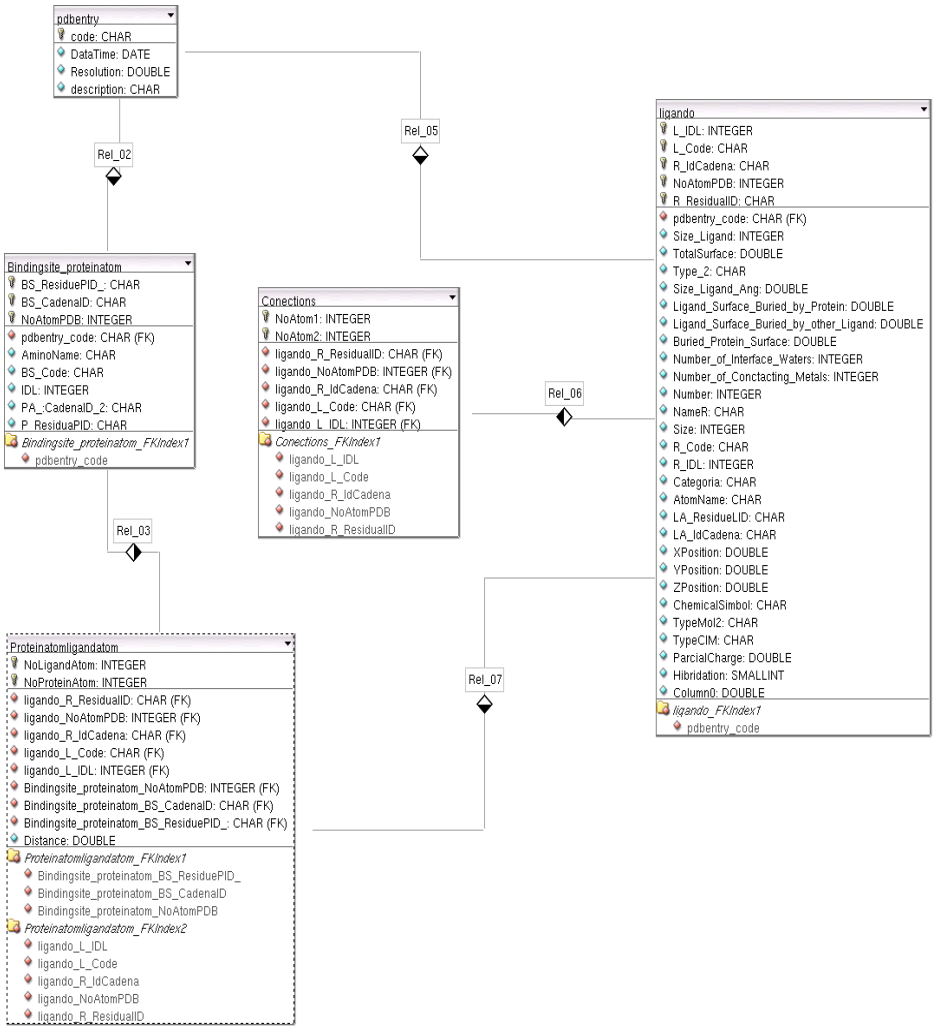


Figura 3 Modelo físico de datos

3.6Diagramas de Interacción.

Diagrama de Secuencia (Gestionar consulta – Graficar consulta).

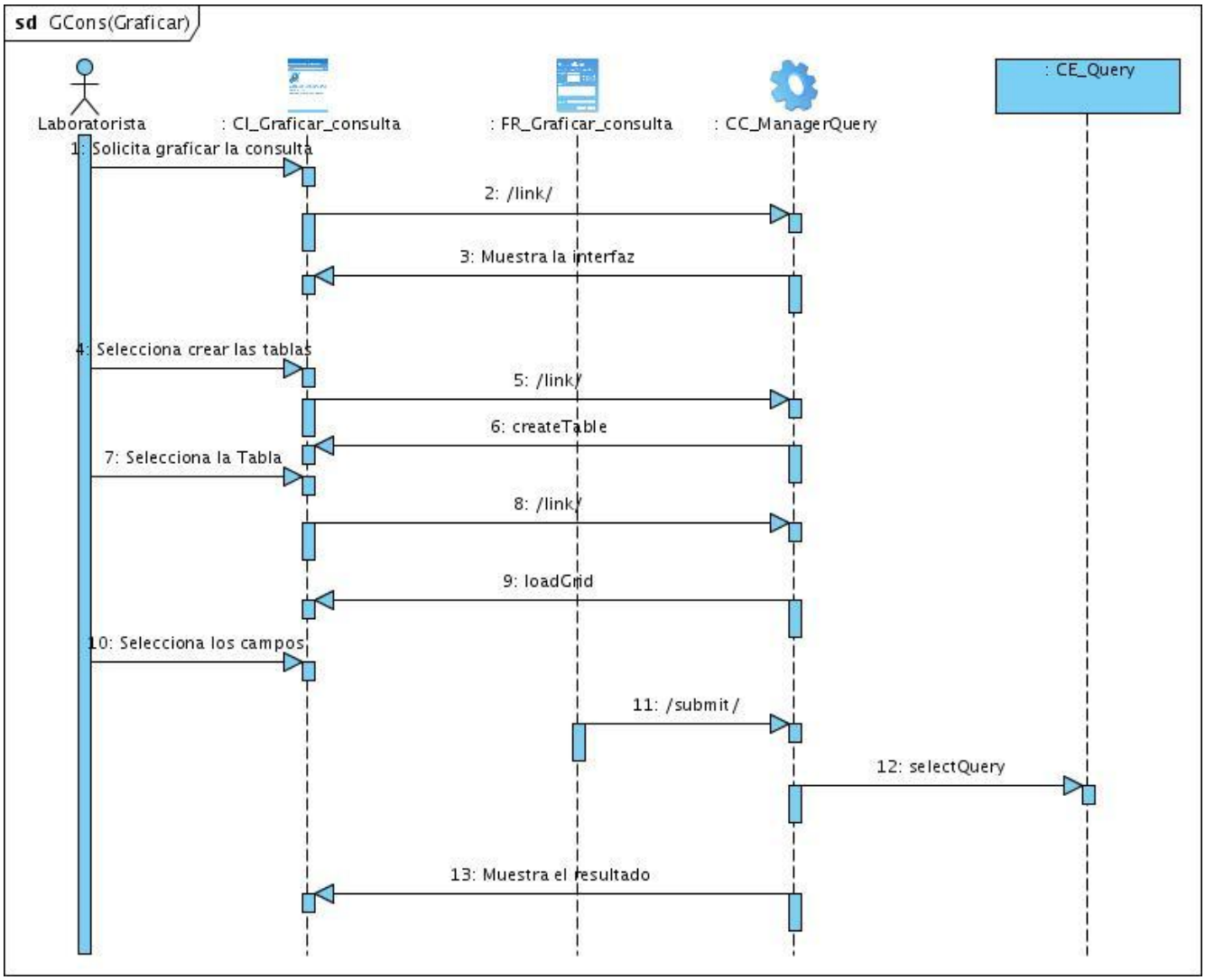


Diagrama de Secuencia (Gestionar consulta – Analizar consulta) [Anexo 13].

Diagrama de Secuencia (Gestionar consulta –Resumir consulta) [Anexo 14].

Diagrama de Secuencia (Gestionar consulta – Realizar consulta) [Anexo 15].

Diagrama de Secuencia (Autenticar usuario) [Anexo 16].

Diagrama de Secuencia (Registrar usuario) [Anexo 17].

Diagrama de Secuencia (Guardar reporte-Salvar reporte) [Anexo 18].

Diagrama de Secuencia (Guardar reporte – Imprimir reporte) [Anexo 19].

Diagrama de Secuencia (Gestionar información-Insertar información) [Anexo 20].

Diagrama de Secuencia (Gestionar información-Actualizar información) [Anexo 21].

Diagrama de Secuencia (Gestionar información-Eliminar información) [Anexo 22].

3.7Modelo de Despliegue.

El diagrama de Despliegue se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes, además muestra las relaciones físicas entre los componentes hardware y software en el sistema final. Los elementos usados por este tipo de diagrama son nodos, componentes y asociaciones.

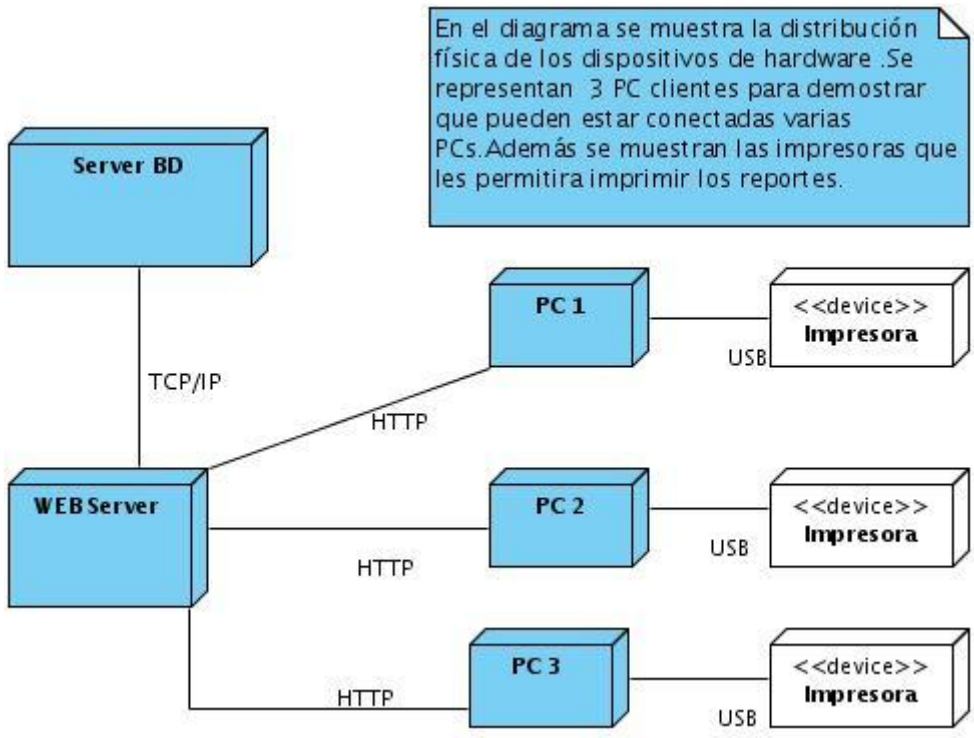


Figura 4 Modelo de despliegue

3.8 Conclusiones

Dando conclusión a este capítulo se conoció como funciona el sistema, además de como sería su distribución física para un mejor funcionamiento. Además se conoce como fue implementado el sistema y como se diseño su base de datos.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

En este capítulo se describe cómo se implementan los elementos del Modelo de diseño, para esto se muestra el Diagrama de componentes, el código fuente de las principales clases y algunas pantallas de la aplicación.

4.1 Diagrama de Componentes

El diagrama de componentes muestra las dependencias lógicas entre componentes software, sean estos componentes fuentes, binarios o ejecutables. Un componente representa una unidad de código que permite mostrar las dependencias en tiempo de compilación y ejecución.

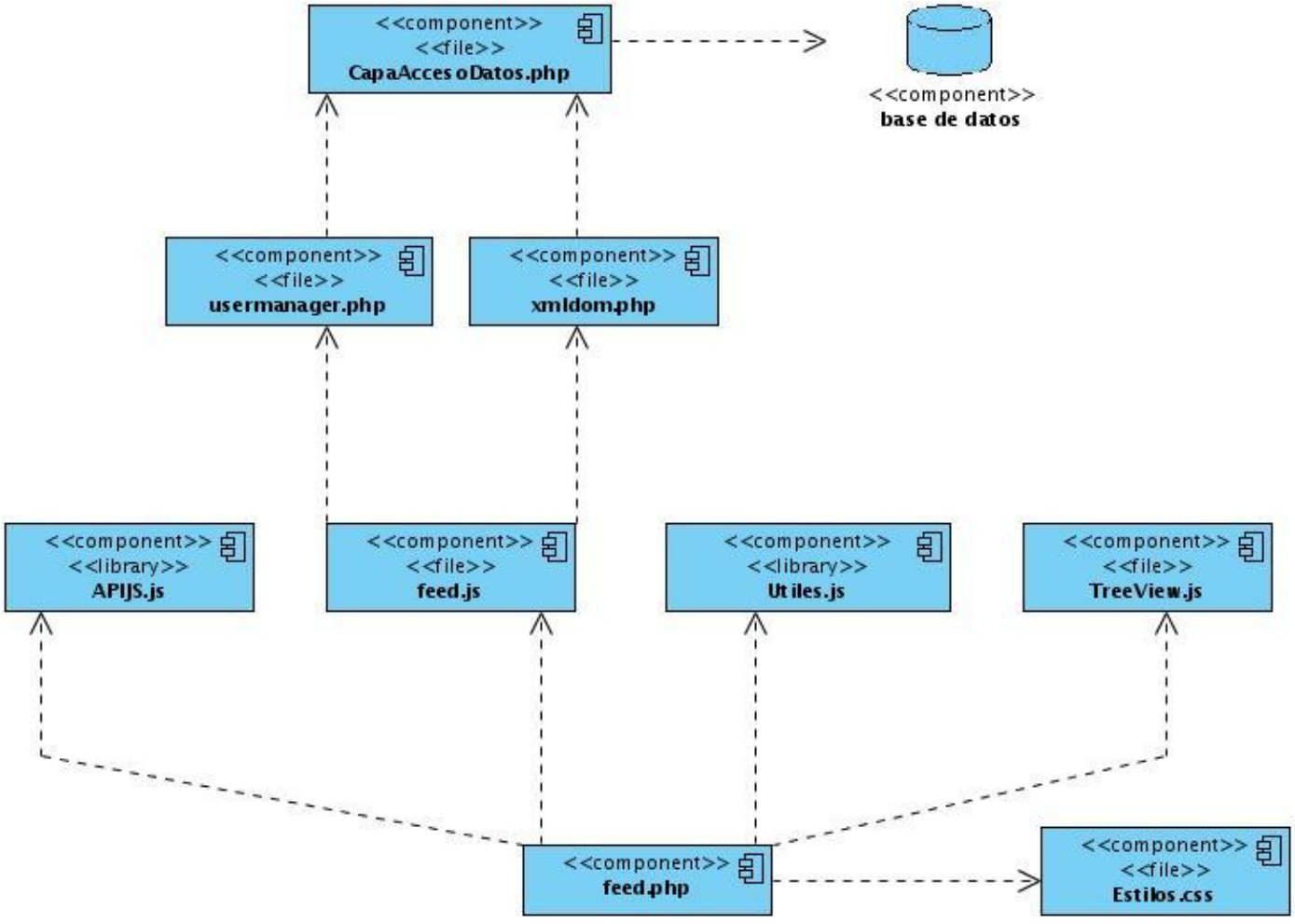


Figura 5 Diagrama de componentes

4.2 Código fuente de principales clases. Descripciones de las mismas.

Aquí se muestran las descripciones de las principales clases implementadas para darle solución al problema.

Nombre: CC_Manager_User	
Tipo de clase: controladora	
Atributo	
$servertype	
$username	
$password	
$host	
$root	
$rootpassword	
$isEnable	
$isRoot	
$isInvitado	
$isExterno	
$isUser	
$isPort	
$dbname	
$isUserSytem	
Para cada responsabilidad:	
Nombre:	ManagerUser($version,$encode,$user,$pass,$host,$db,$port,$server)
Descripción:	Constructor de la clase
Nombre:	set_pass($pass)
Descripción:	Introduce el valor de la contraseña
Nombre:	get_pass()
Descripción:	Devuelve el valor de la contraseña
Nombre:	get_user()
Descripción:	devuelve el nombre de usuario

Nombre:	CheckUser()
Descripción:	Autentifica el usuario y carga el esquema de las bases de datos en caso de que el usuario sea del sistema.
Nombre:	LastConectedUser()
Descripción:	Devuelve el último usuario conectado.
Nombre:	IsInvitado()
Descripción:	Retorna true si el usuario es invitado y falso en caso contrario
Nombre:	IsRoot()
Descripción:	Retorna true si el usuario es administrador y falso en caso contrario
Nombre:	GetHosting()
Descripción:	devuelve el host en formato host: puerto
Nombre:	IsUserSystem()
Descripción:	Retorna true si el usuario es del sistema y falso en caso contrario
Nombre:	IsServerExtern
Descripción:	Retorna true si el usuario está conectándose a un servidor externo y falso en caso contrario
Nombre:	IsEnable()
Descripción:	Retorna true si la conexión está activa para el usuario en cuestión y falso en caso contrario
Nombre:	IsAvaliable()
Descripción:	Retorna true si la conexión está disponible para el usuario en cuestión y falso en caso contrario
Nombre:	RegisterUserOnScreening($user,$pass,$email,$nick,$ip)
Descripción:	registra un usuario en la base de datos screening
Nombre:	EditUserOnScreening($user,$pass,$email,$nick,$ip)
Descripción:	edita los datos de los usuarios de la base de datos screening
Nombre:	RegisterUserOnMysqlSystem($user,$pass,$withGrandOption)
Descripción:	registra un usuario en el gestor MySql
Nombre:	DeleteUserFromMysqlSystem($user,$pass)
Descripción:	Elimina un usuario en el gestor MySql
Nombre:	EditUserOnMysqlSystem($user,$pass,$withGrandOption)
Descripción:	Edita usuarios en el gestor MySql

Nombre: CC_ManagerData	
Tipo de clase: controladora	
Atributo	
$Gestor	
$user	
$db	
$query	
Para cada responsabilidad:	
Nombre:	ManagerData($version,$encode,$usuario,$serverType)
Descripción:	Constructor de la clase
Nombre:	IsAvaliableGestorType($type)
Descripción:	Retorna si el tipo de gestor es correcto
Nombre:	getXMLRoot()
Descripción:	retorna el nodo principal del xml
Nombre:	getXMLItems()
Descripción:	retorna los items del Xml
Nombre:	getXMLFallas()
Descripción:	retorna los nodos de los errores ocurridos durante determinada operación
Nombre:	xmlQuery($conector,$db,$table,$query,$page=0,$limit=30,$type,$AsProcedure)
Descripción:	retorna un xml con los datos asociados a una petición al servidor
Nombre:	Save()
Descripción:	salva la clase como xml
Nombre:	ExecuteMysqlQuery($host,$port,$db,$table,$SQL,$page,$limit,$AsProcedure)
Descripción:	ejecuta una sentencia SQL en el servidor
Nombre:	getPostData()
Descripción:	retorna los datos enviados por el método POST

Nombre: CC_ManagerReport	
Tipo de clase: controladora	
Atributo	
$user	

$mode	
$UrlFile	
$managerUser	
Para cada responsabilidad:	
Nombre:	ManagerReport($UrlFile,$user)
Descripción:	Constructor de la clase
Nombre:	MysqlSaveResult($server,$port,$db,$table,$query,$page,$limit)
Descripción:	Salva la consulta en el formato determinado.
Nombre:	Print()
Descripción:	Imprime el reporte

Nombre: CC_ManagerQuery	
Tipo de clase: controladora	
Atributo	
Tables	
ListDataBases	
ListTables	
SelectedQuery	
user	
server	
pass	
port	
Para cada responsabilidad:	
Nombre:	getLastConnectedUser()
Descripción:	IU para obtener y muestra el último usuario conectado a la aplicación
Nombre:	LoadDataBaseTables()
Descripción:	Carga el esquema de la base de datos a la que el usuario se conecta
Nombre:	loadManagerUserPanel()
Descripción:	carga los paneles de trabajos en dependencia de los roles y/o privilegios del usuario
Nombre:	MainLayoutStyle()
Descripción:	Carga los estilos de las capas de Interfaz de Usuario principales de la aplicación

Nombre:	loadLayaout()
Descripción:	carga la capa principal de Interfaz de Usuario de la aplicación
Nombre:	InsertXmlTable()
Descripción:	IU para insertar una tabla en modo grafico en la aplicación y generar consultas dinámicas de modo gráfico (no en la Base de datos)
Nombre:	InsertRows()
Descripción:	IU para insertar nuevos registros en la aplicación y modelar las consultas de inserción a las tablas de la base de datos
Nombre:	DropRows()
Descripción:	IU para eliminar los registros seleccionados por el usuario directamente en la base de datos
Nombre:	Save()
Descripción:	IU para salvar los reportes en formato Word y Excel
Nombre:	Print()
Descripción:	IU para imprimir los reportes
Nombre:	editUserFunction()
Descripción:	Muestra la IU para editar la información de los usuarios registrados en la aplicación
Nombre:	addUserFunction()
Descripción:	Muestra la IU para adicionar la información de los usuarios que se registran en la aplicación
Nombre:	informeQueryfunction()
Descripción:	Muestra el resultado de un informe de la última consulta de selección realizada.
Nombre:	informeElementRepetidos()
Descripción:	Muestra el resultado de los elementos repetidos en la última consulta de selección realizada.
Nombre:	sendQueryFunction()
Descripción:	Envía las consultas al servidor
Nombre:	refreshDBList()
Descripción:	Actualiza el listado de bases de datos.
Nombre:	XmlGridView()
Descripción:	Muestra el DataGrid con la información resultante de una petición al servidor

Nombre:	AjaxXmlDataModel()
Descripción:	Construye un objeto para controlar las peticiones al servidor y construir un modelos de datos a partir de la respuesta del servidor

4.3 Imagenes de la aplicación

Aquí se muestran algunas imágenes de la aplicación que dan solución a los requisitos funcionales planteados.

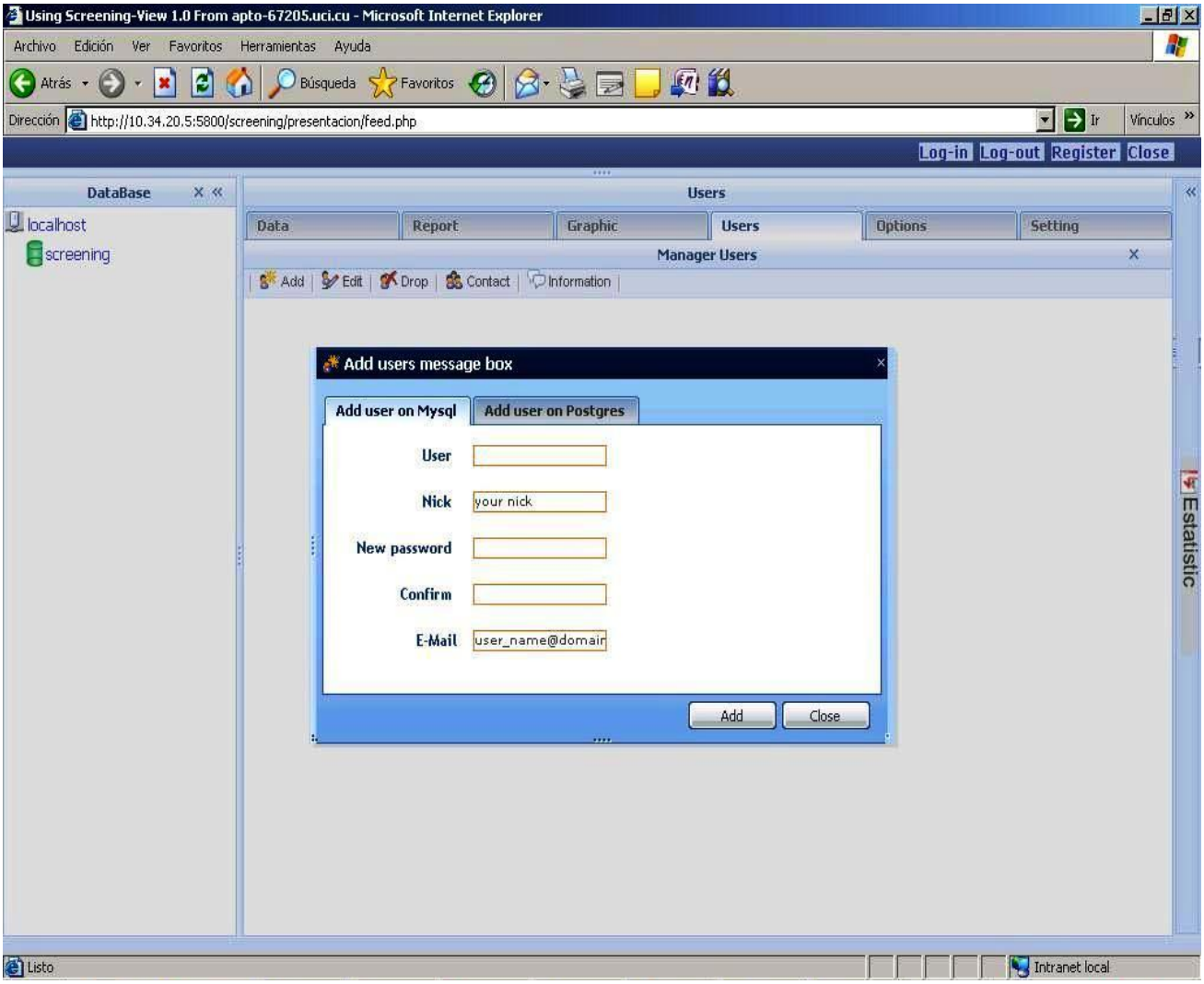


Figura 6 Registrar usuario

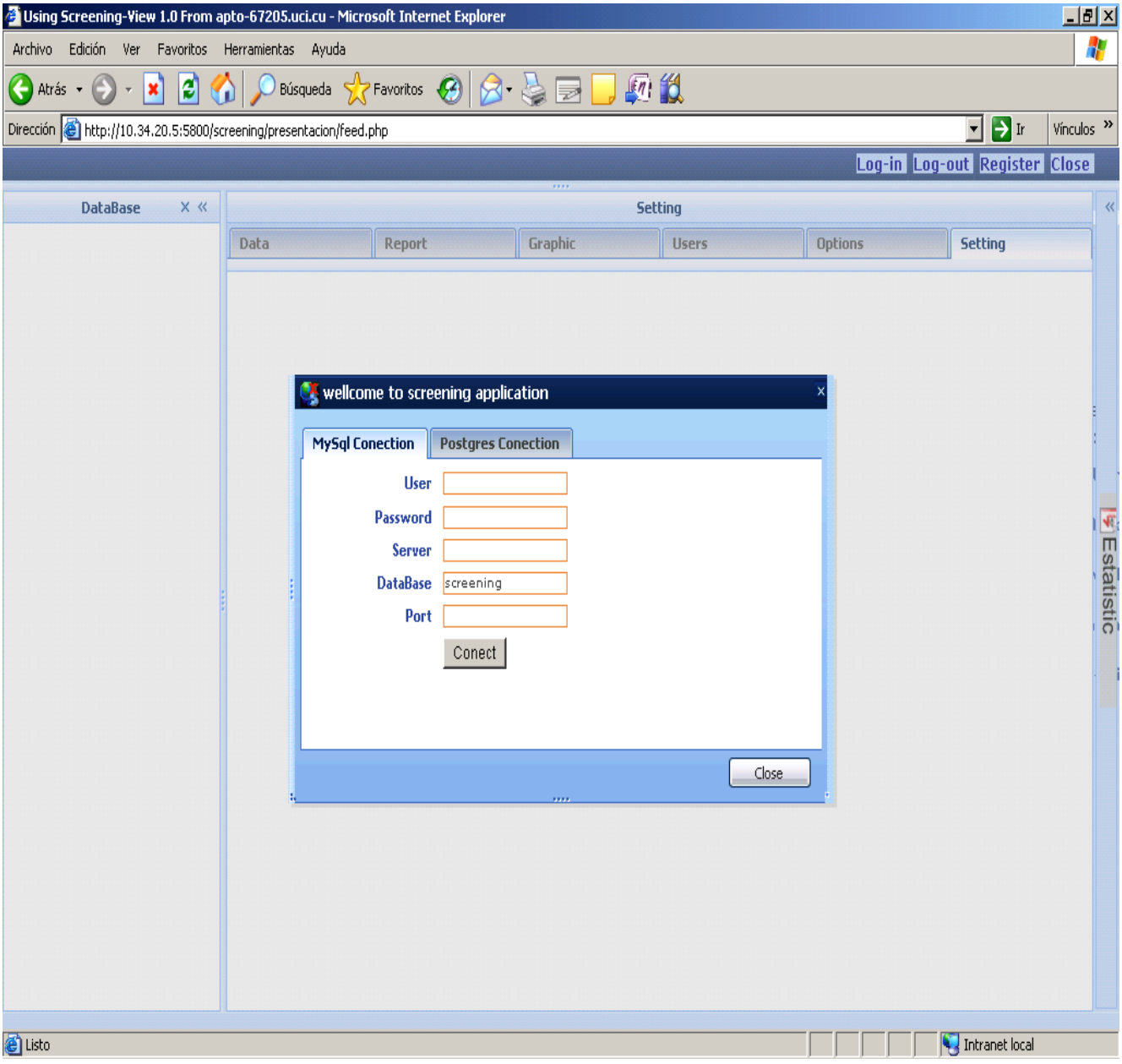


Figura 7 Autenticar usuario

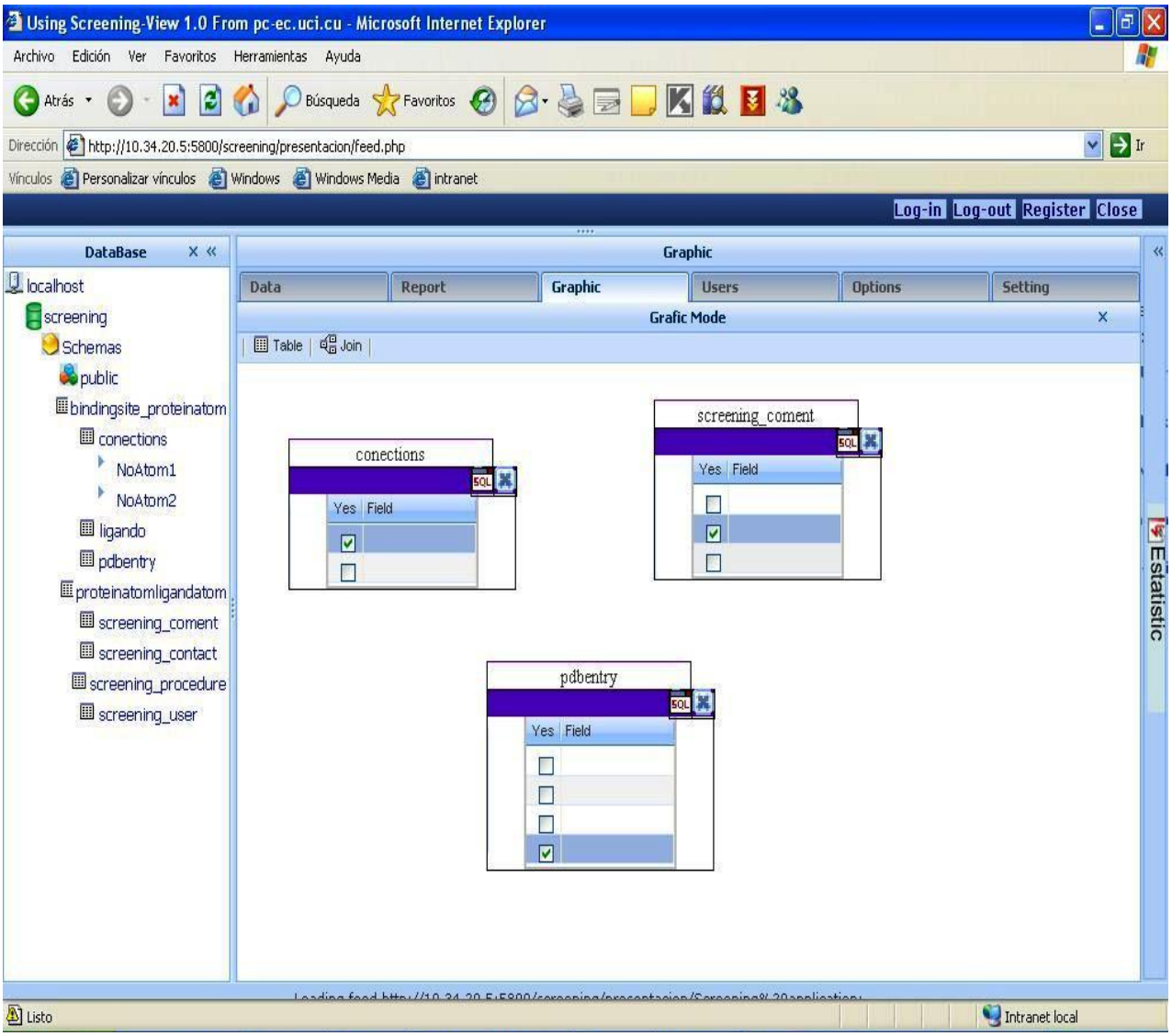


Figura 8 Graficar consulta

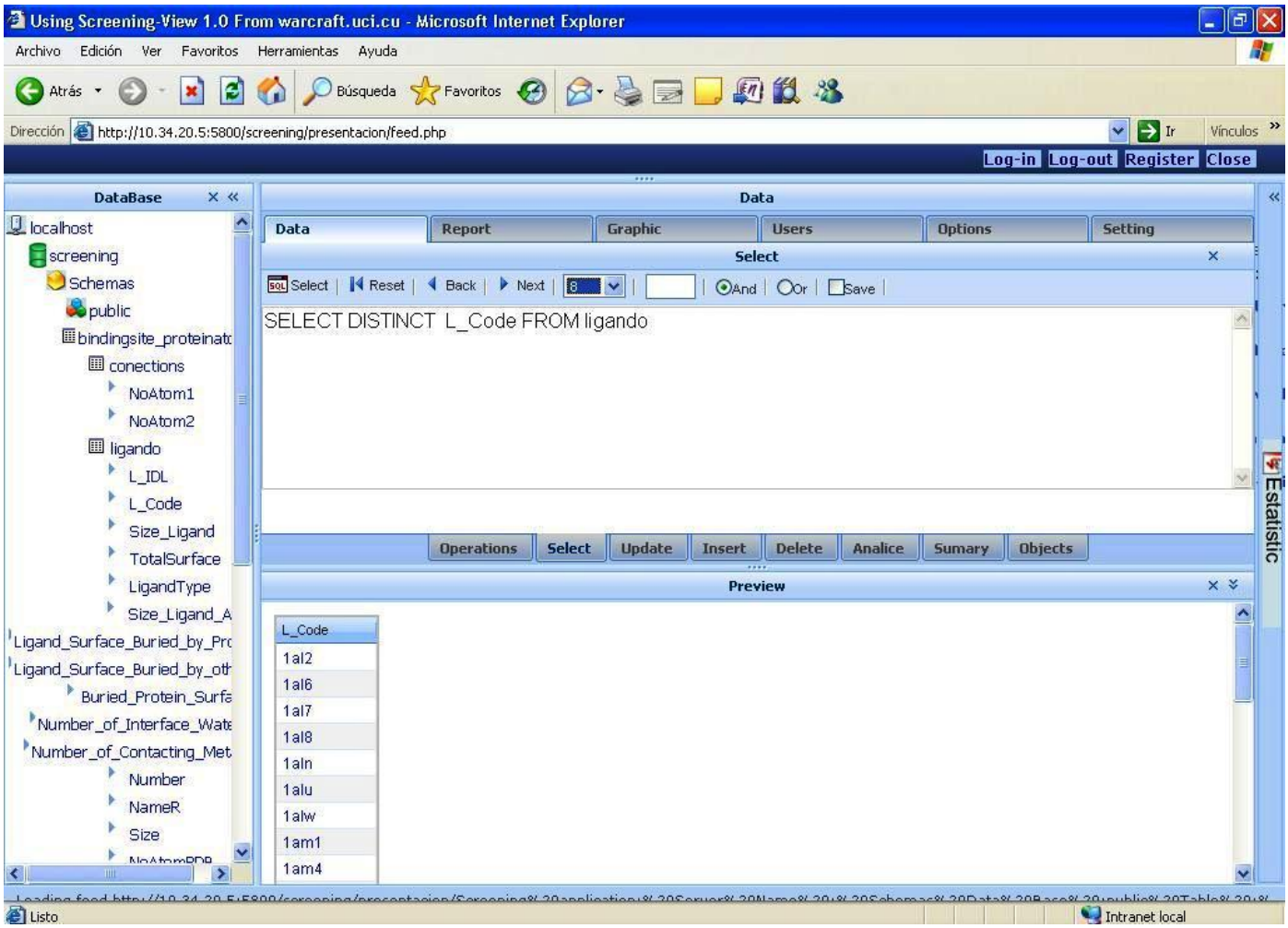


Figura 9 Realizar consulta

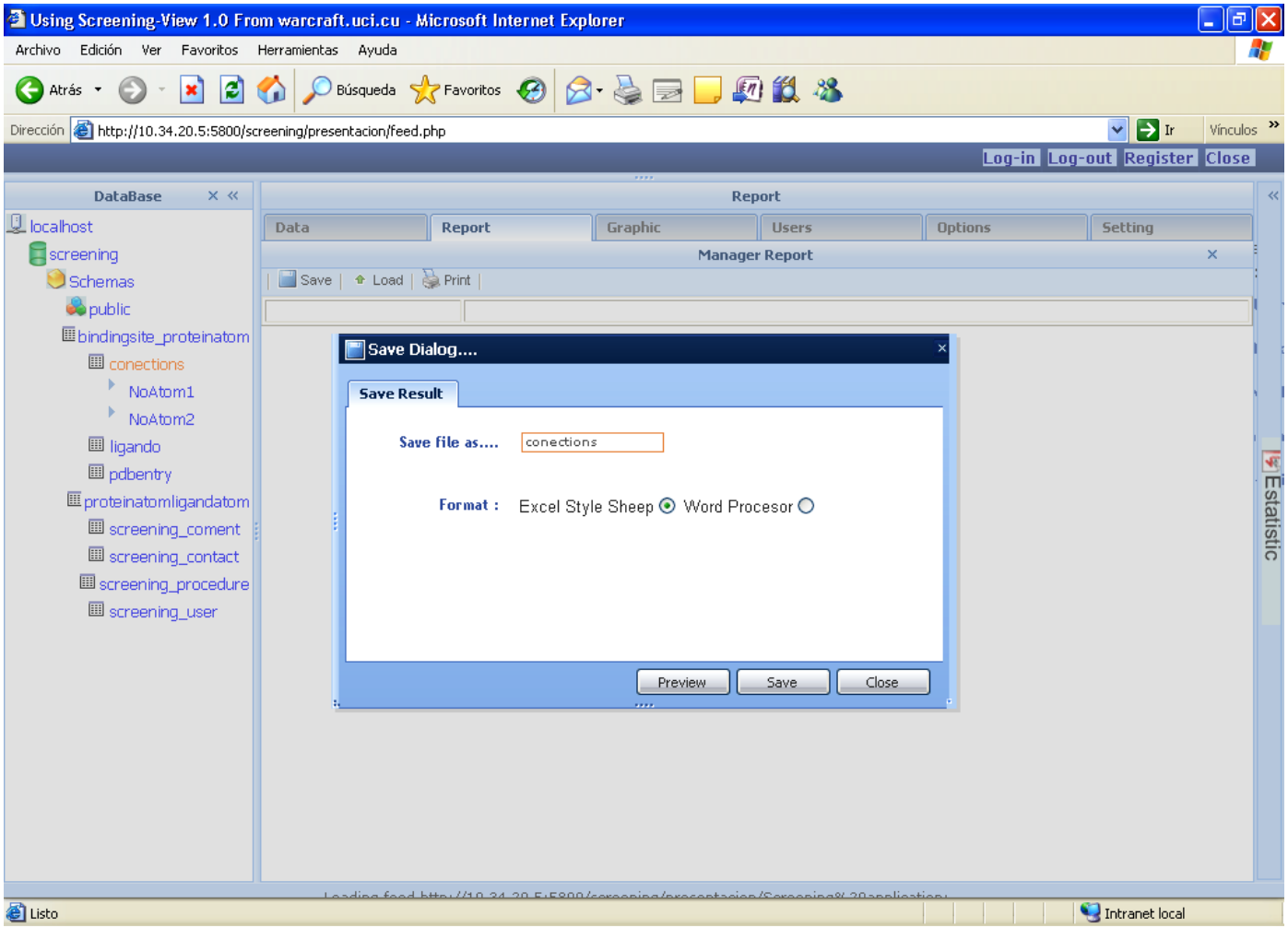


Figura 10 Salvar consulta

4.4 Conclusiones

En este capítulo se mostro un diagrama de componentes y además de explicó el funcionamiento de las principales clases y se mostro algunas pantallas de la aplicación.

CONCLUSIONES

Se obtuvo como producto final una aplicación Web que permite la gestión de información necesaria en estudios de docking para la formación complejos proteína- ligando. Para lograr este objetivo:

- ✓ Se realizó un estudio de las diferentes bases de datos que contienen información sobre las proteínas.
- ✓ Se realizó el análisis y diseño de una aplicación para la gestión de la información en estudios de docking.
- ✓ Se implementó una aplicación en php para la gestión de la información en estudios de docking.

RECOMENDACIONES

- ✓ Incorporar otras formas de análisis con el objetivo de hacer investigaciones más profundas y con mayor resultado.
- ✓ Extender el uso de la aplicación a todas aquellas empresas que se dediquen al estudio de docking.

REFERENCIAS BIBLIOGRÁFICAS

1. Jaqueline Padilla Zúñiga y Arturo Rojo Domínguez .SIMULACIÓN DEL RECONOCIMIENTO ENTRE PROTEÍNAS Y MOLÉCULAS ORGÁNICAS O DOCKING. APLICACIÓN AL DISEÑO DE FÁRMACOS. [Consultado en octubre del 2007].Disponible en: http://laguna.fmedic.unam.mx/mensajebioquimico

2. National Cancer Institute. Consultado en noviembre del 2007 .Disponible en: http://cactus.cit.nih.gov/

3. Weininger D (1998). SMILES, a Chemical Language and Information System. introduction to methodology and encoding rules. J Chem Inf Comput Sci 28:31-36.[Consultado en febrero 2008].Disponible en:http://almost.cubic.uni-koeln.de/cdk/jcp/bib/WEI88

4. Gasteiger J y Marsili M (1980). Iterative partial equalization of orbital electronegativity. A rapid access to atomic charges. Tetrahedron 36:3219-3228. [Consultado en octubre del 2007]. Disponible en: http:www2.ccc.uni-erlangen.de/people/Johann_Gasteiger/

5. Manual de PHP. [Consultado en noviembre del 2007]. Disponible en: http://www.manualdephp.com/manualphp/introduccion-php.html

6. Alvares, R. [consultado en: noviembre, 2007] ? Qué es PHP? .Disponible en: http://www.desarrolloweb.com/articulos/392.php

7. Alvares ,R. [consultado en: noviembre,2007].?Que es Java Script. Disponible en: http://www.desarrolloweb.com/articulos/25.php

8. Wikilearning Comunidades de Wikis libres para aprender). TECNOLOGÍAS CAPACES DE INTEGRARSE EN UNA PÁGINA WEB.[consultado en: febrero,2007].Disponible en: http www.wikilearning.com/tecnologias capaces de integrarse en una pagina web-wkccp-3443-17.htm

9. GRACIA, J. Introdución a MySQL Última actualización: 9 de junio del 2005. [Consultado el: 19 de enero de 2008]. Disponible en: http://www.webestilo.com/mysql/intro.phtml

10. Programación III.I.T.I de sistemas .Patrones de diseño .Félix Prieto curso2007/08. [consultado en: mayo,2008].

11. Larman Craig: UML y patrones, introducción al análisis y diseño orientado a objetos, Félix Varela,2004. [consultado en: mayo,2008].

BIBLIOGRAFÍA

1. PARADIGM, V. Documentation [Consultado el: 18 de enero de 2008]. Disponible en: http://www.visual-paradigm.com/.

2. Setting up a large set of protein-ligand PDB complexes for the development and validation of knowledge-based docking algorithms [Consultado el: 9 de enero de 2008]. Disponible en:http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2008766

3. SCHNEIDER Gisbert; BÖHM Hans-Joachim. Virtual screening and fast automated docking methods.[Consultado en:octubre de 2007].Disponible en:http://cat.inist.fr/aModele=afficheN&cpsidt=13491870

4. DesJarlais. Docking: Successes and Challenges.[Consultado en: octubre de 2007]Disponible en :

http://novacripta.cbm.uam.es/bioweb/courses/MasterBiofis0708/tema07/index.html

5. Brian K. Shoichet.Virtual screening of chemical libraries [Consultado en: noviembre de 2007].Disponible en: http://novacripta.cbm.uam.es/bioweb/courses/MasterBiofis0708/tema07/index.htmlEl

6. BioMed Central .Luis A. Diago, Persy Morell, Longendri Aguilera and Ernesto Moreno. Setting up a large set of protein-ligand PDB complexes for the development and validation of knowledge-based docking algorithms. Disponible en:http://www.biomedcentral.com/1471-2105/8/310/abstract .

7. Venkatraman Mohan, Alan C. Gibbs, Maxwell D. Cummings, Edward P. Jaeger‡and Renee L.DesJarlais. Docking: Successes and Challenges Current Pharmaceutical Design, 2005, 11

8. Jaqueline Padilla Zúñiga y Arturo Rojo Domínguez. Universidad Autónoma Metropolitana-Iztapalapa Departamento de Química. Área de Biofisicoquímica Apartado Postal 55-534, 09340 México. D. F. SIMULACIÓN DEL RECONOCIMIENTO ENTRE PROTEÍNAS YMOLÉCULAS ORGÁNICAS O DOCKING. APLICACIÓN AL DISEÑO DE FÁRMACOS.

9. Ernesto Moreno* and Kalet Leon Center of Molecular Immunology, Havana, Cuba .Geometric and Chemical Patterns of Interaction in Protein–Ligand Complexes and Their Application in Docking.

10. Molecular Docking Web. [En línea] 2002 [Consultado en: diciembre de 2007.] http://mgl.scripps.edu/people/gmm/

11. Larman, Craig: UML y patrones, introducción al análisis y diseño orientado a objetos, Félix Varela,2004.

12. Pressman, Roger S. / Madrid, McGraw-Hill. Ingeniería del Software: un enfoque práctico. Parte I y II [En línea] 2002 [Consultado en: diciembre de 2007.] Disponible en: http://bibliodoc.uci.cu/pdf/reg02689.pdf

13. National Cancer Institute. Consultado en noviembre del 2007 .Disponible en: http://cactus.cit.nih.gov/

14.Weininger D (1998). SMILES, a Chemical Language and Information System. Introduction to methodology and encoding rules. J Chem Inf Comput Sci 28:31-36.[Consultado en febrero 2008].Disponible en:http://almost.cubic.uni-koeln.de/cdk/jcp/bib/WEI88

15.Gasteiger J y Marsili M (1980). Iterative partial equalization of orbital electronegativity. A rapid access to atomic charges. Tetrahedron 36:3219-3228. [Consultado en octubre del 2007]. Disponible en: http:www2.ccc.uni-erlangen.de/people/Johann_Gasteiger/

ANEXOS

Anexo 1Descripción Caso de Uso Autenticar usuario.

Nombre del CUS	Autenticar
Actores	Laboratorista(inicia)
Propósito	Permitir autenticarse.
Resumen	El Caso de Uso se inicia cuando el usuario introduce los datos que se le piden para acceder a la aplicación, estos se verifican y finaliza dándole los permisos y habilitándole la entrada.
Referencias	R1
Precondiciones	El usuario debe estar registrado.
Poscondiciones	Se habilitan las p según los privilegios.

Curso Normal de los Eventos	
Acciones del Actor	**Respuesta del Sistema**
1. El administrador introduce el nombre del usuario y su contraseña.	2. El sistema encripta la contraseña.
	3. Busca el usuario y compara la contraseña.
	4. En caso de ser correcta la contraseña, se le asignan los permisos al usuario para acceder al sistema y este entra al sistema.
	5. Termina el caso de uso Autenticarse.
Curso Alterno	
Acciones del Actor	**Respuesta del Sistema**
	6. En caso de no coincidir la contraseña se le envía un mensaje de: ¨ Contraseña incorrecta ¨.
Prioridad:	Crítico

Anexo 2Descripción Caso de Uso Registrar usuario.

Nombre del CUS	Registrar usuario
Actores	Usuario del Sistema (inicia)

Propósito	Permitir registrarse en el sistema
Resumen	El Caso de Uso se inicia cuando el usuario desea acceder a la aplicación y a todos sus vínculos. El usuario entra los datos solicitados y queda finalmente registrado.
Referencias	R 2
Precondiciones	El usuario no está registrado en la aplicación.
Poscondiciones	Usuario registrado.

Curso Normal de los Eventos	
Acciones del Actor	**Respuesta del Sistema**
1. El usuario selecciona la opción Registrarse.	2. Muestra la interfaz donde se solicitan los datos.
3. Si el usuario decide registrarse introduce todos los datos.	4. Valida los datos ofrecidos por el usuario.
	5. Si los datos están correctos el sistema registra al usuario.
	6. Termina el caso de uso Registrarse
Curso Alterno	
Acciones del Actor	**Respuesta del Sistema**
3. Si el usuario no decide registrarse hace clic en el botón cancelar.	
	5. Si los datos no están correctos muestra un mensaje de ¨Datos incorrectos¨.

Prioridad:	Crítico

Anexo 3Descripción Caso de Uso Guardar reporte.

Nombre del CUS	Guardar reporte
Actores	Laboratorista (inicia)
Propósito	Tener información mas detallada
Resumen	El Caso de Uso se inicia cuando el laboratorista solicita cualquiera de estas opciones

	• Salvar reporte • Imprimir reporte Termina el caso de uso cuando se realiza cualquiera de las opciones anteriores.
Referencias	R 4.1,R4.2
Precondiciones	El usuario debe estar registrado en la base de datos.
Poscondiciones	

Curso Normal de los Eventos	
Acciones del Actor	**Respuesta del Sistema**
● El laboratorista selecciona alguna de las opciones • Salvar reporte • Imprimir reporte	2. El sistema, en dependencia de la operación que solicita realizar, hace lo siguiente: - Si usuario selecciona la opción de Salvar reporte, ir a **Salvar reporte** - Si usuario selecciona la opción de Imprimir reporte, ir a **Imprimir reporte.**
	3. Termina el caso de uso Guardar reporte.
Sección: Salvar reporte	
Acciones del Actor	**Respuesta del Sistema**
1.Selecciona la opción salvar reporte	2. Muestra la interfaz correspondiente para salvar el reporte.
3.El Laboratorista selecciona el formato en que desea salvarlo	4.Salva el reporte
Sección: Imprimir reporte	
Acciones del Actor	**Respuesta del Sistema**
1.Selecciona la opción imprimir	2. Muestra la interfaz correspondiente para imprimir el reporte.
	3,Imprime el reporte

Prioridad:	Crítico

Anexo 4Descripción Caso de Uso Gestionar información.

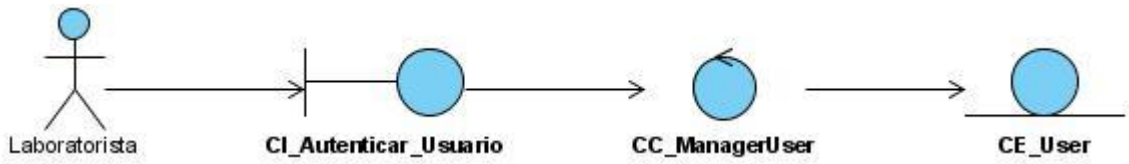
Nombre del CU	Gestionar información

Actores	Administrador
Propósito	Permitir insertar nueva información sobre las proteínas, actualizarla y eliminarla.
Resumen	El Caso de Uso se inicia cuando se solicitan cualesquiera de las siguientes opciones: Insertar información. Actualizar información. Eliminar información. Termina el caso de uso cuando se realiza cualquiera de las opciones anteriores.
Referencias	R 3.1, R 3.2, R 3.3
Precondiciones	El usuario debe estar autenticado.
Poscondiciones	

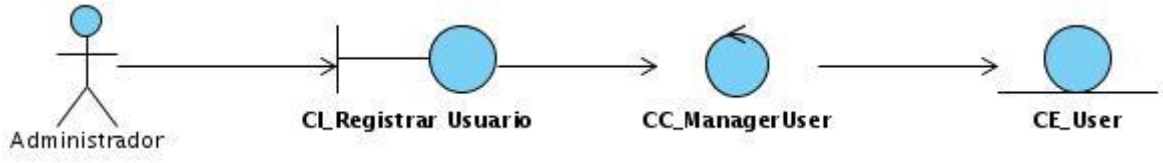
Curso Normal de los Eventos	
Acciones del Actor	**Respuesta del Sistema**
1. El administrador selecciona realizar alguna de las opciones: Insertar información. Actualizar información. Eliminar información.	2. El sistema, en dependencia de la operación que solicita realizar, hace lo siguiente: - Si usuario selecciona la opción de Insertar información, ir a **Insertar información**. - Si usuario selecciona la opción de Actualizar información, ir a **Actualizar información.** - Si usuario selecciona la opción Eliminar información, ir a **Eliminar información.**
	3 Termina el caso de uso Gestionar información
Sección: Insertar información	
	1. Muestra la interfaz correspondiente para insertar nueva información.
2. El administrador selecciona la nueva información que desea agregar.	3. Almacena toda la información en la base de datos.

	4. Termina el caso de uso
Sección: Actualizar información	
	1. Muestra la interfaz que permite actualizar la información.
2. El administrador selecciona los campos a los cuales le va hacer las modificaciones..	3. Muestra la actualización de la información.
Sección: Eliminar información	
	1. Muestra una interfaz para eliminar simulación.
2. El administrador selecciona la información que desea eliminar.	3. El sistema busca en la base de datos la información indicada y la elimina.
	4. Termina el caso de uso.
Prioridad:	Crítico

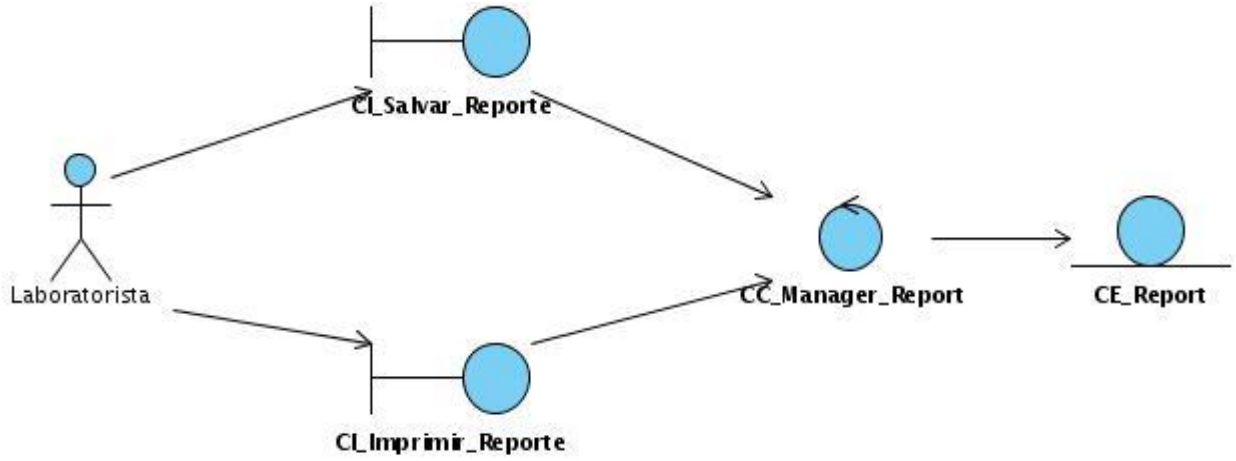
Anexo 5 Diagrama de clases del análisis (Autenticar usuario).



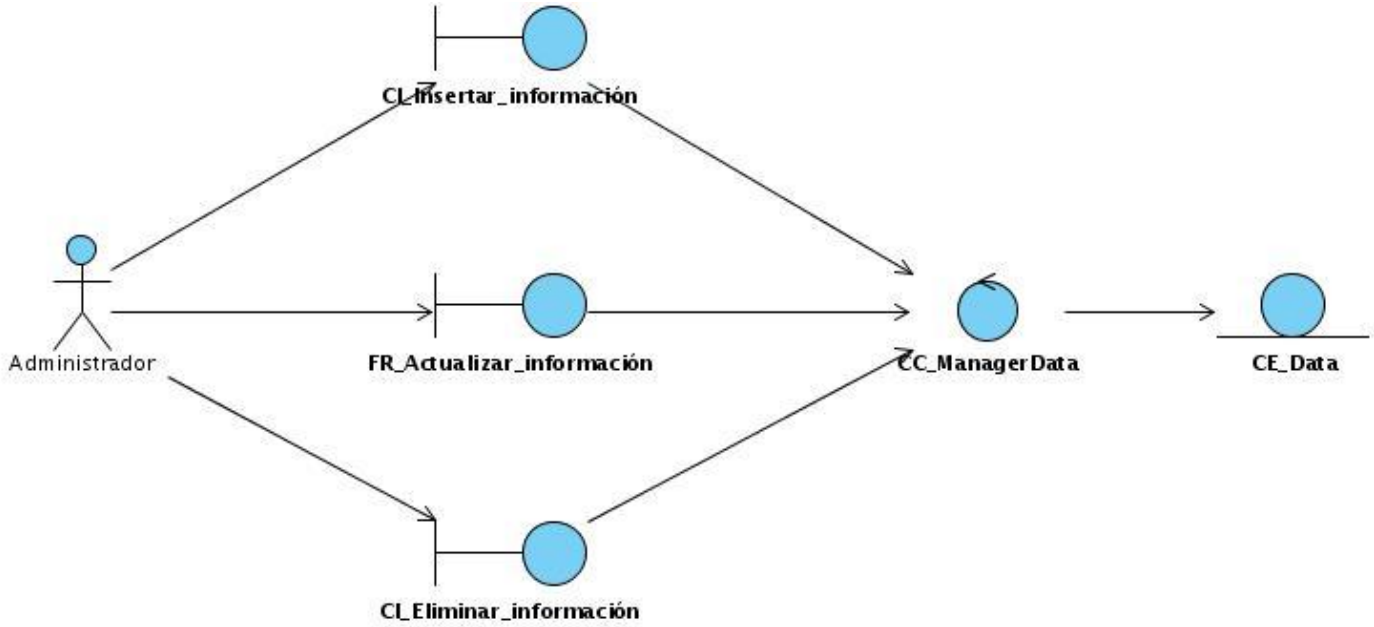
Anexo 6 Diagrama de clases del análisis (Registrar usuario).



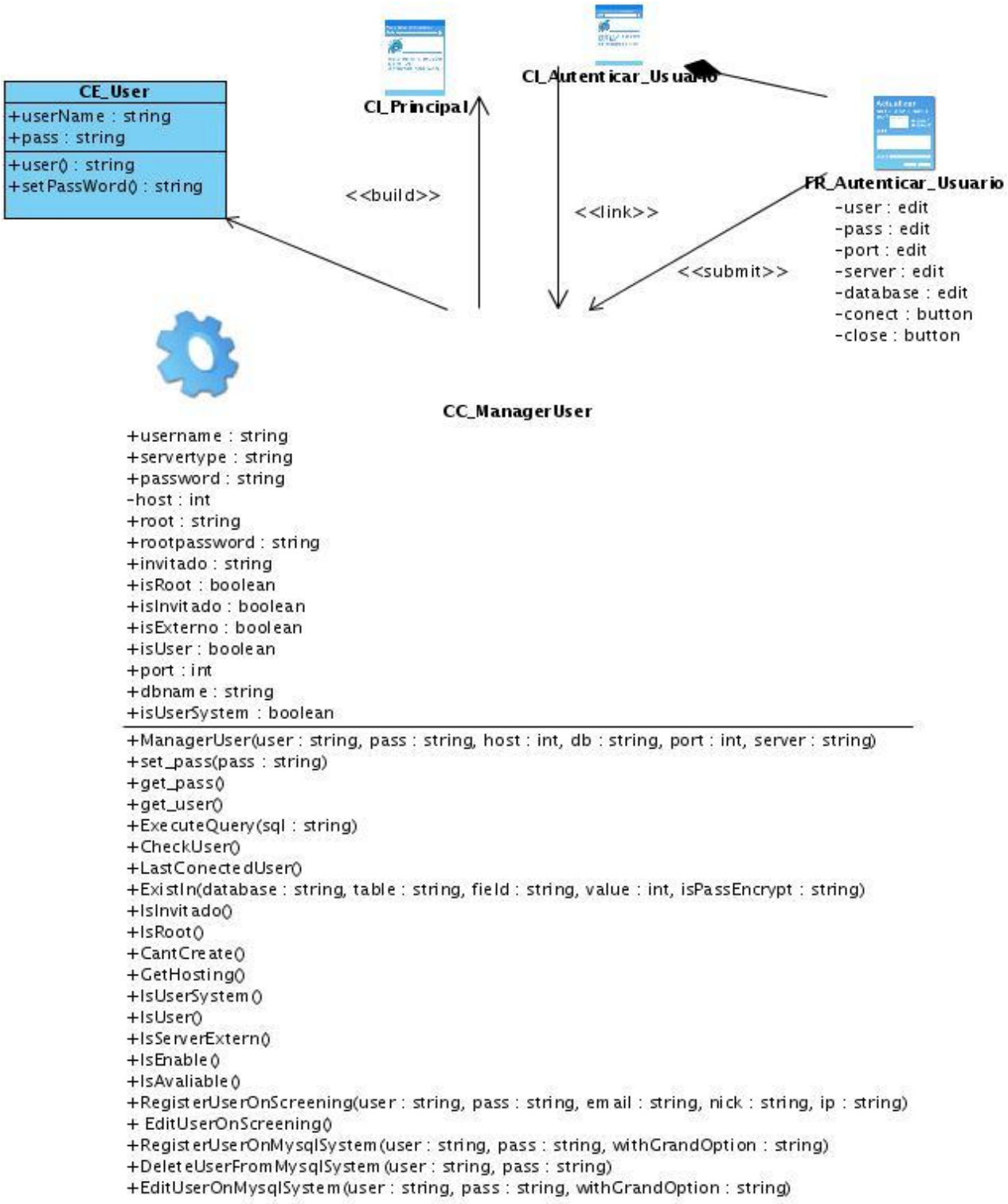
Anexo 7 Diagrama de clases del análisis (Guardar reporte).



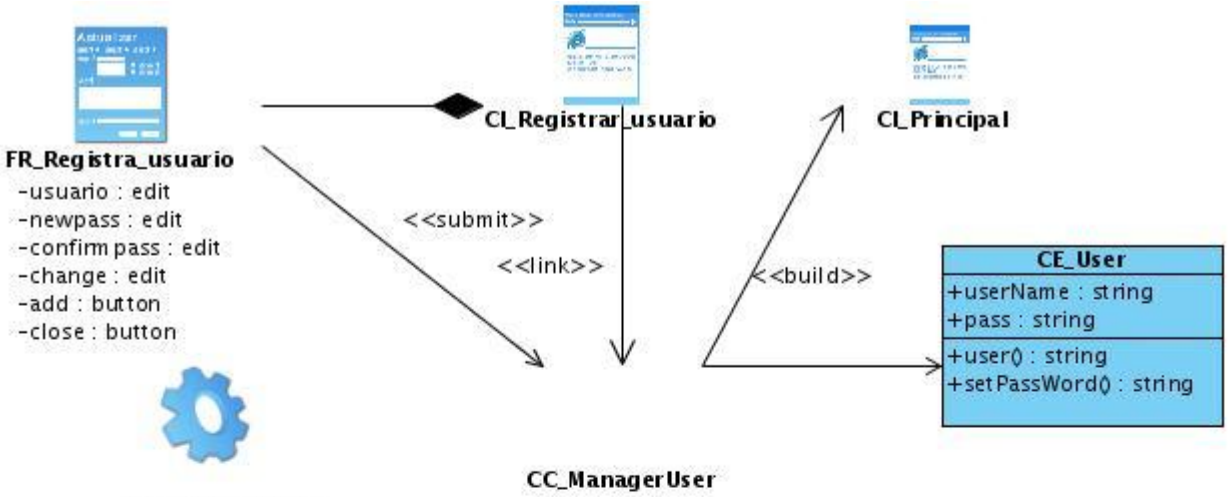
Anexo 8 Diagrama de clases del análisis (Gestionar información).



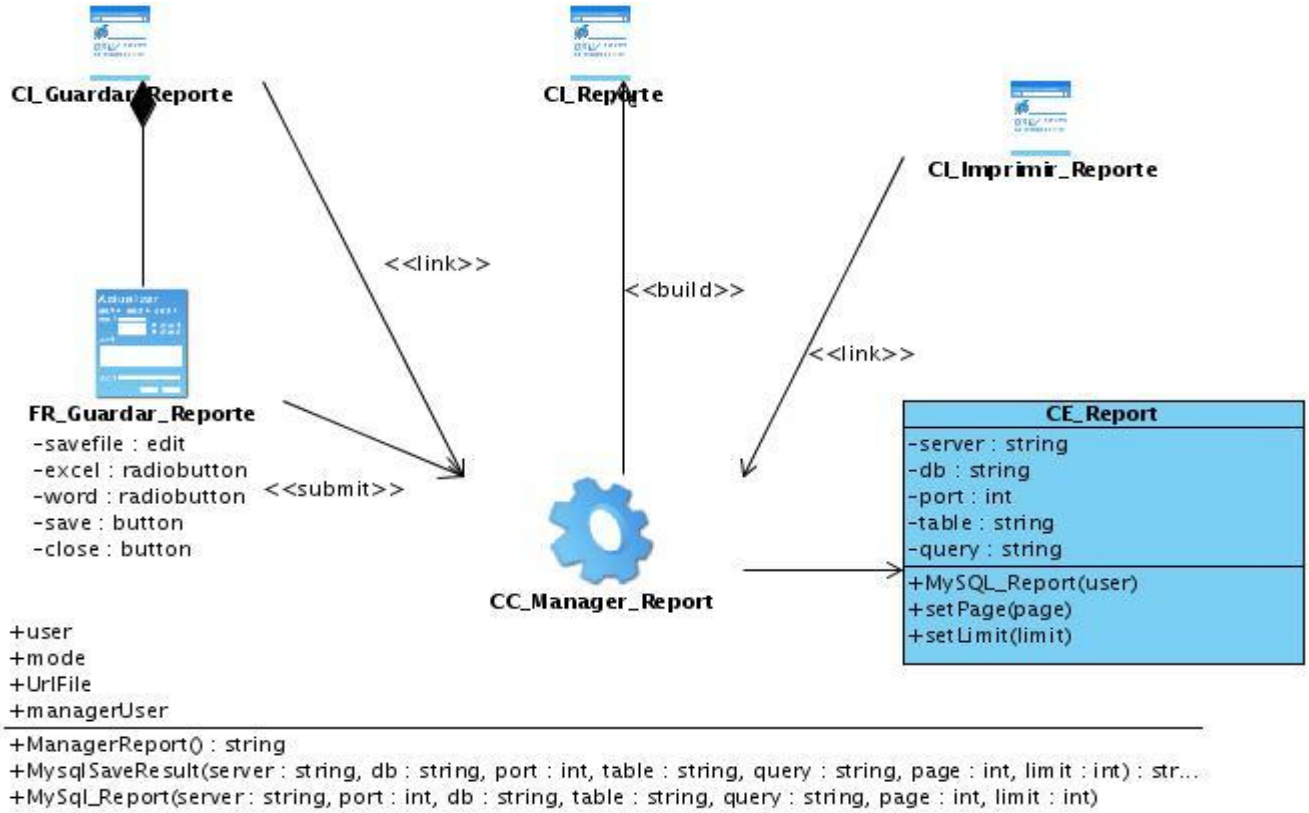
Anexo 9 Diagrama de clases Web del Diseño (Autenticar usuario).



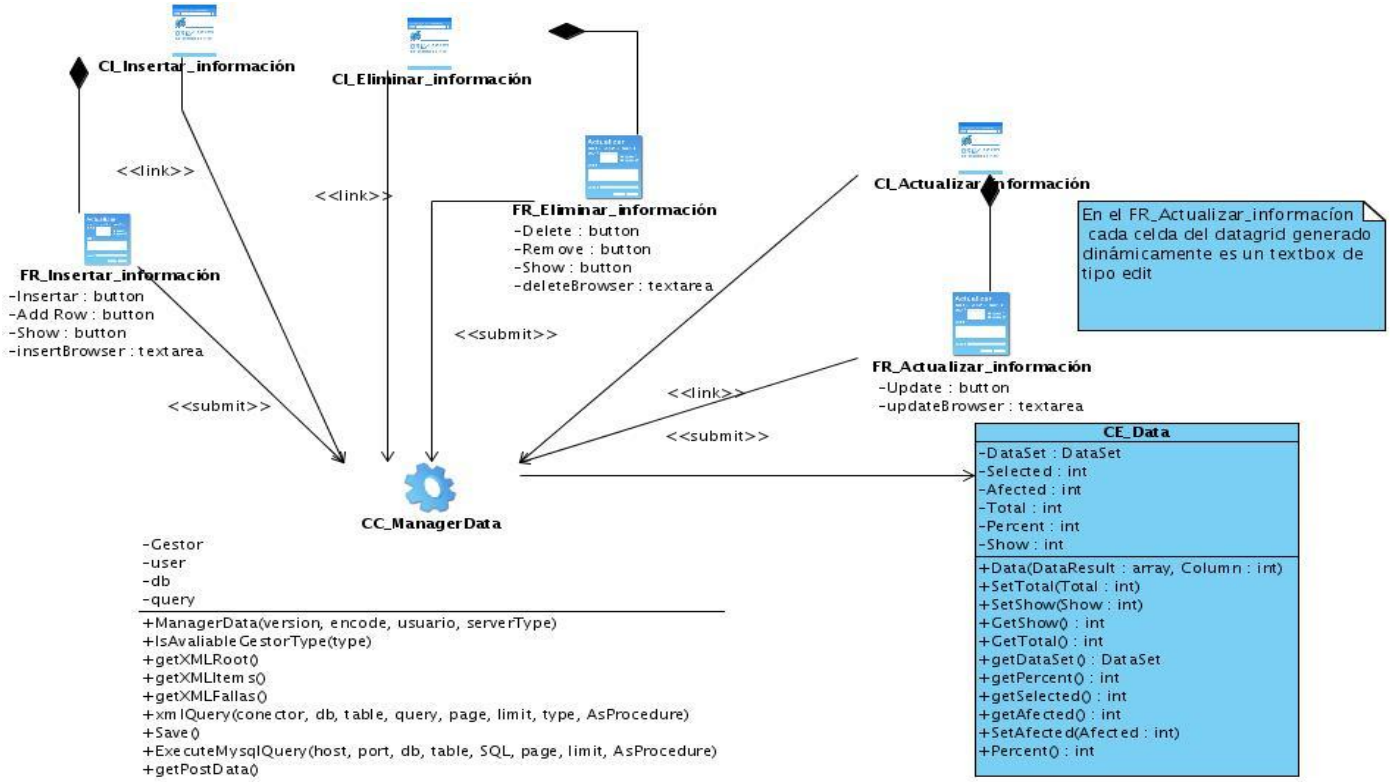
Anexo 10 Diagrama de clases Web del Diseño (Registrar usuario).


FR_Registra_usuario
-usuario : edit
-newpass : edit
-confirm pass : edit
-change : edit
-add : button
-close : button
CL_Registrar_usuario
CL_Principal
<<submit>>
<<link>>
<<build>>
CE_User
+userName : string
+pass : string
+user() : string
+setPassWord() : string
CC_ManagerUser
+username : string
+servertype : string
+password : string
-host : int
+root : string
+rootpassword : string
+invitado : string
+isRoot : boolean
+isInvitado : boolean
+isExterno : boolean
+isUser : boolean
+port : int
+dbname : string
+isUserSystem : boolean
+ManagerUser(user : string, pass : string, host : int, db : string, port : int, server : string)
+set_pass(pass : string)
+get_pass()
+get_user()
+ExecuteQuery(sql : string)
+CheckUser()
+LastConectedUser()
+ExistIn(database : string, table : string, field : string, value : int, isPassEncrypt : string)
+IsInvitado()
+IsRoot()
+CantCreate()
+GetHosting()
+IsUserSystem()
+IsUser()
+IsServerExtern()
+IsEnable()
+IsAvaliable()
+RegisterUserOnScreening(user : string, pass : string, email : string, nick : string, ip : string)
+ EditUserOnScreening()
+RegisterUserOnMysqlSystem(user : string, pass : string, withGrandOption : string)
+DeleteUserFromMysqlSystem(user : string, pass : string)
+EditUserOnMysqlSystem(user : string, pass : string, withGrandOption : string)

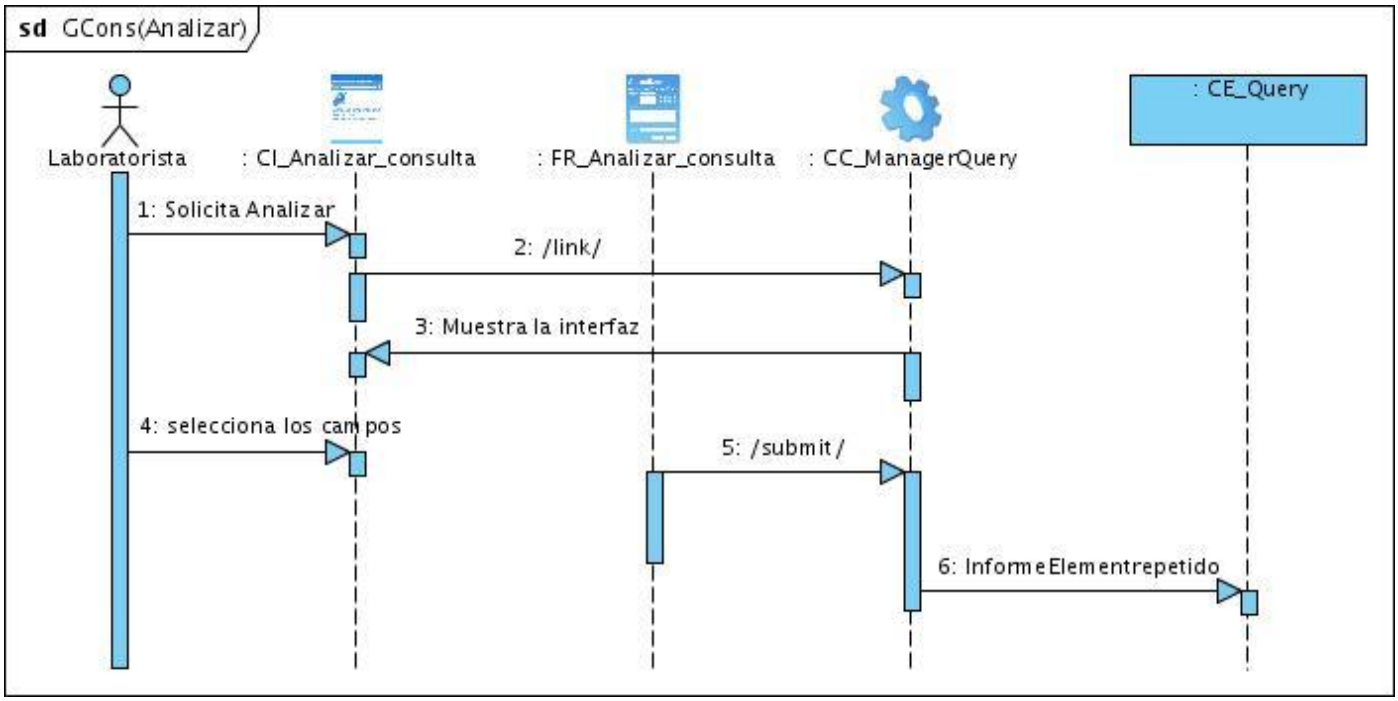
Anexo 11 Diagrama de clases Web del Diseño (Guardar reporte).



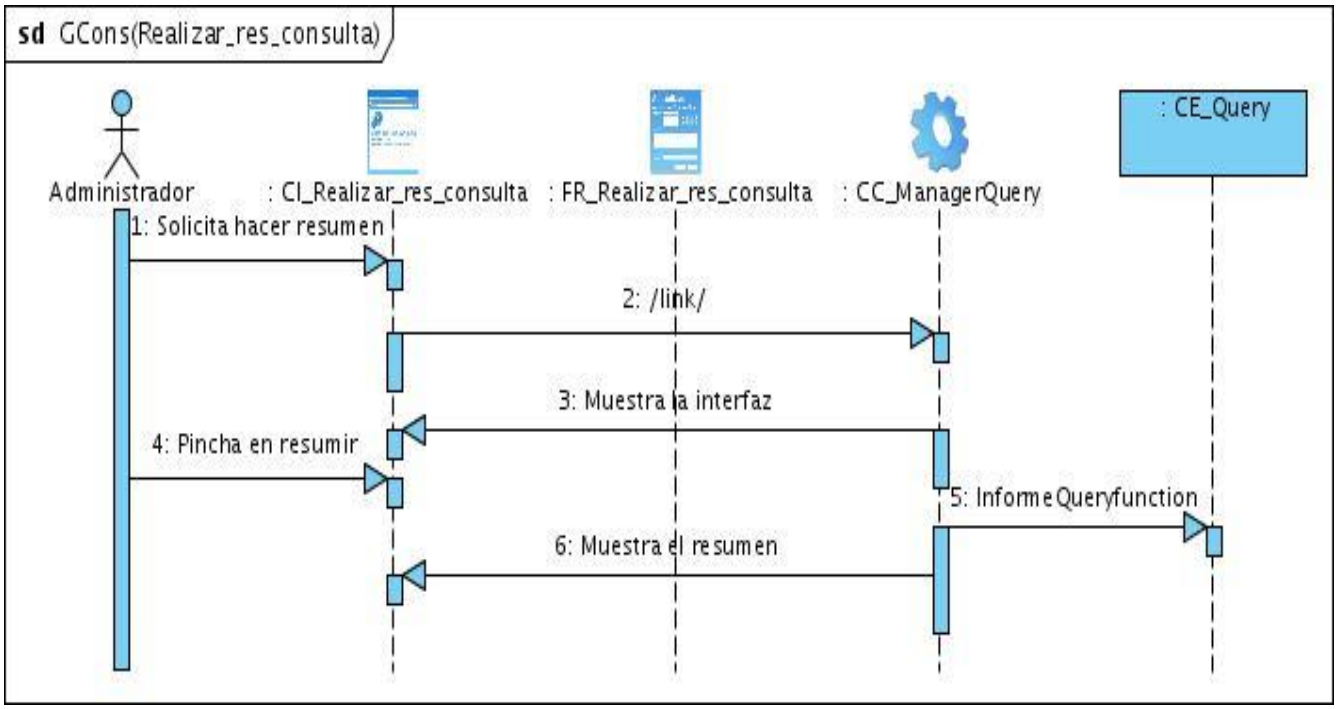
Anexo 12 Diagrama de clases Web del Diseño (Gestionar información).



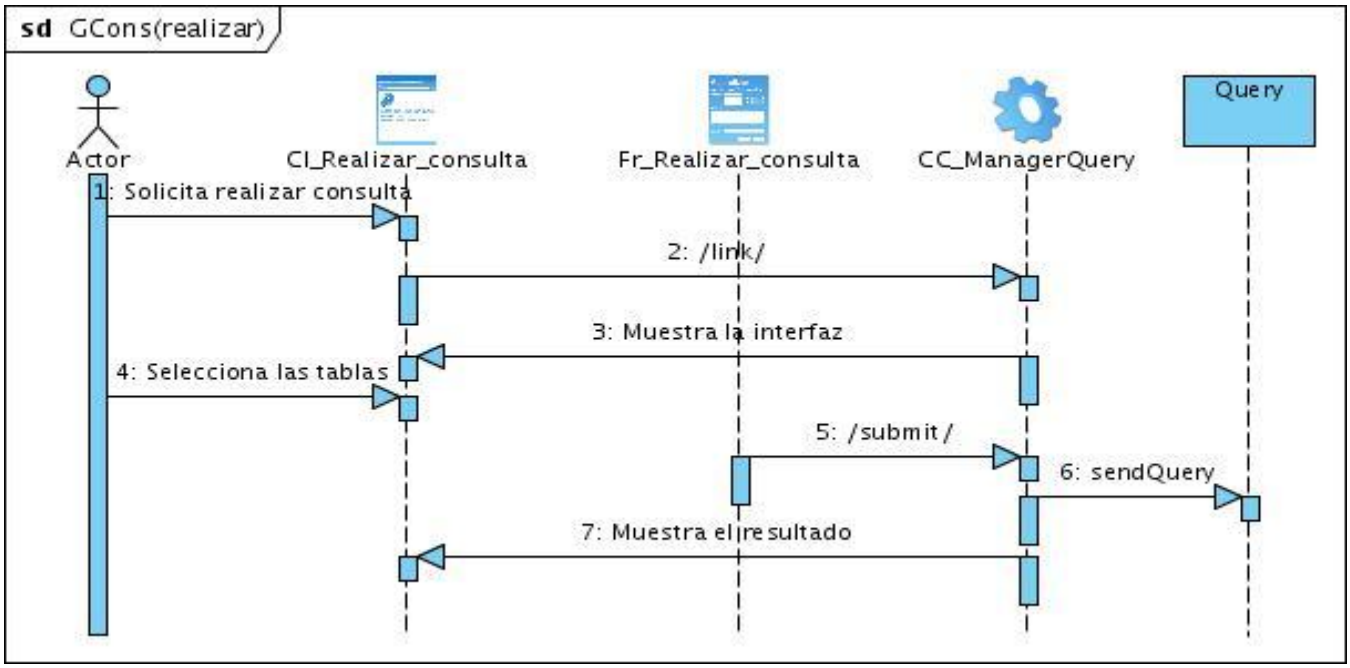
Anexo 13 Diagrama de Secuencia (Gestionar consulta – Analizar consulta).



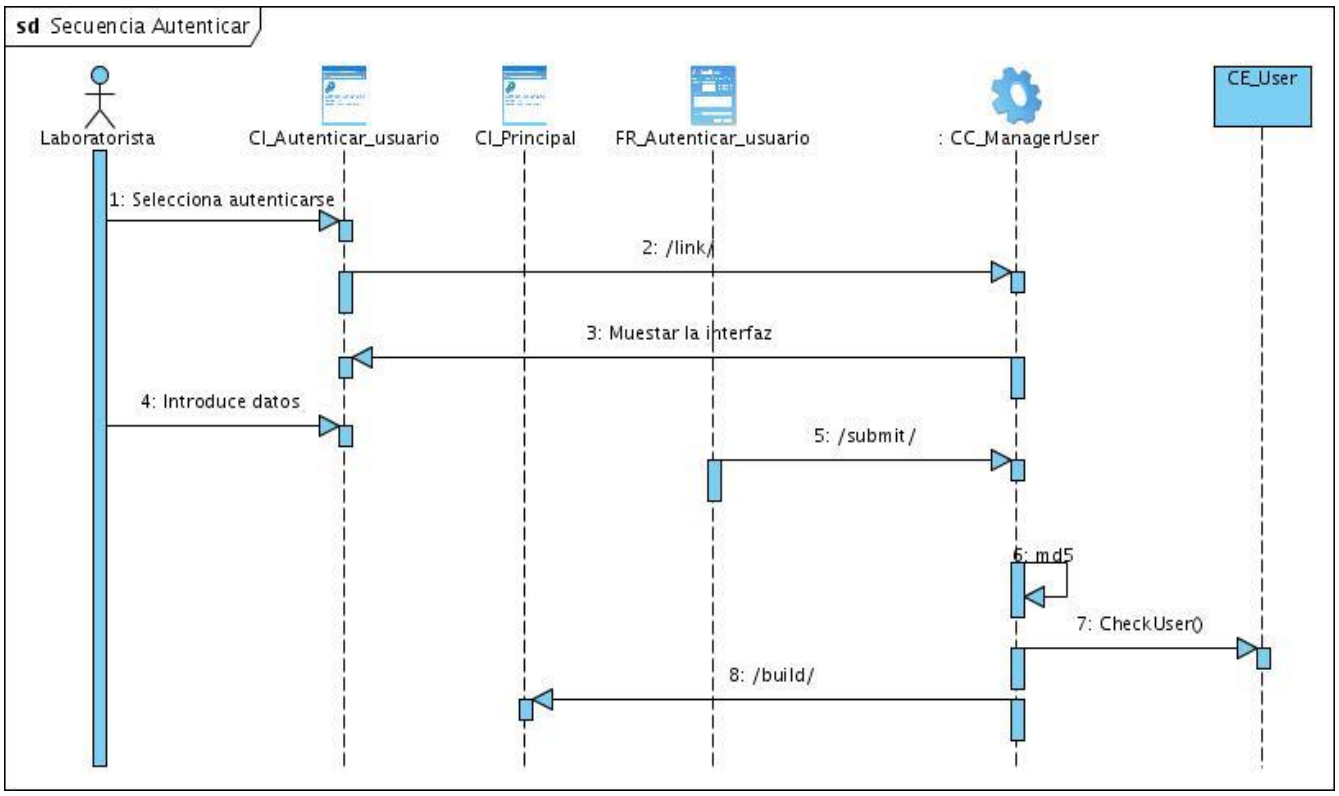
Anexo 14 Diagrama de Secuencia (Gestionar consulta –Resumir consulta).



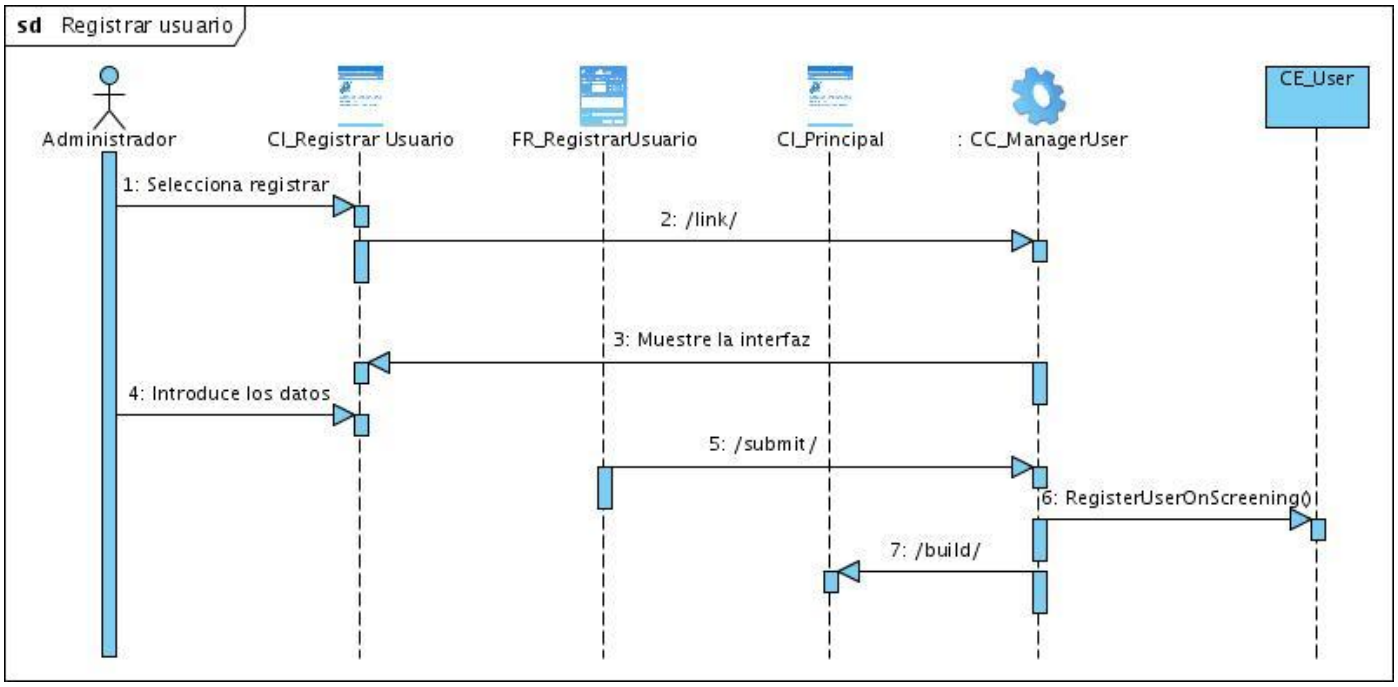
Anexo 15 Diagrama de Secuencia (Gestionar consulta – Realizar consulta).



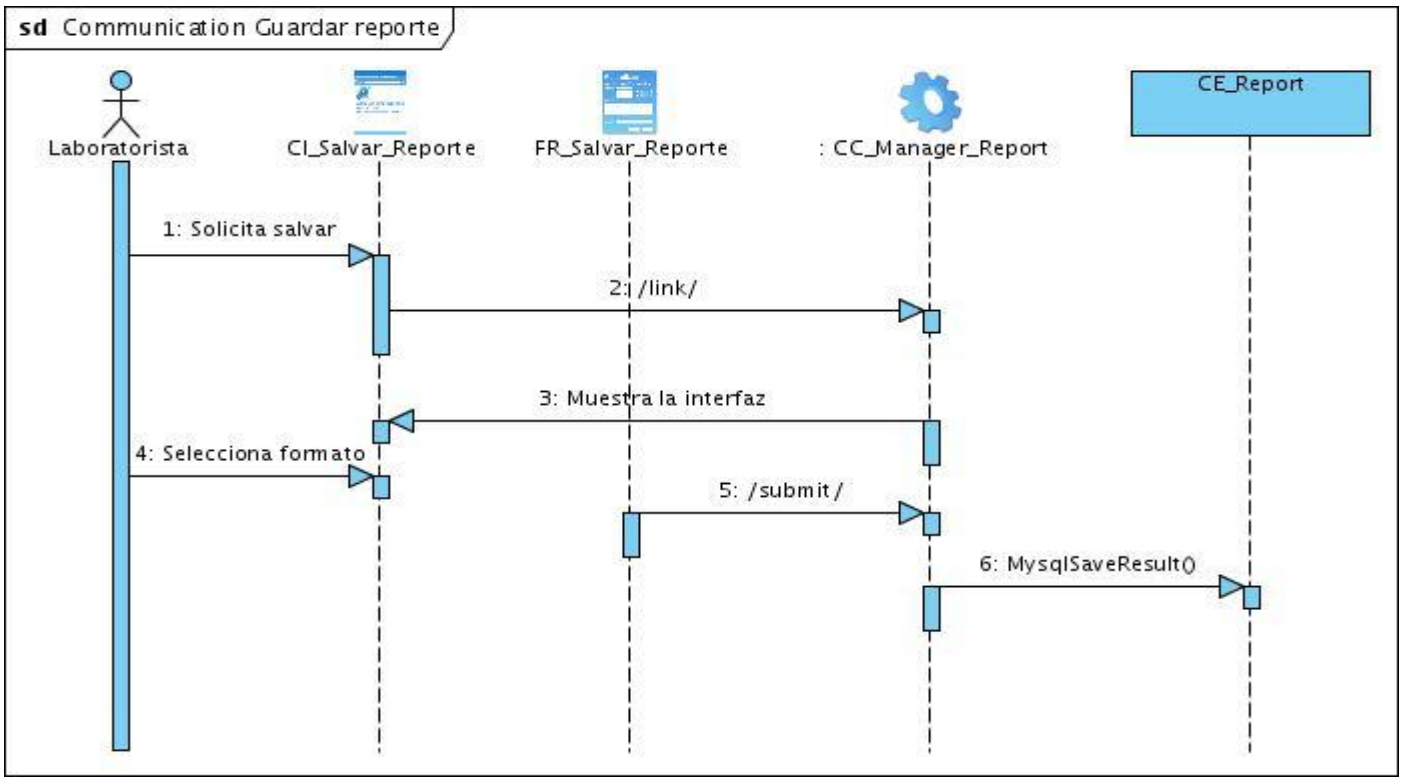
Anexo 16 Diagrama de Secuencia (Autenticar usuario).



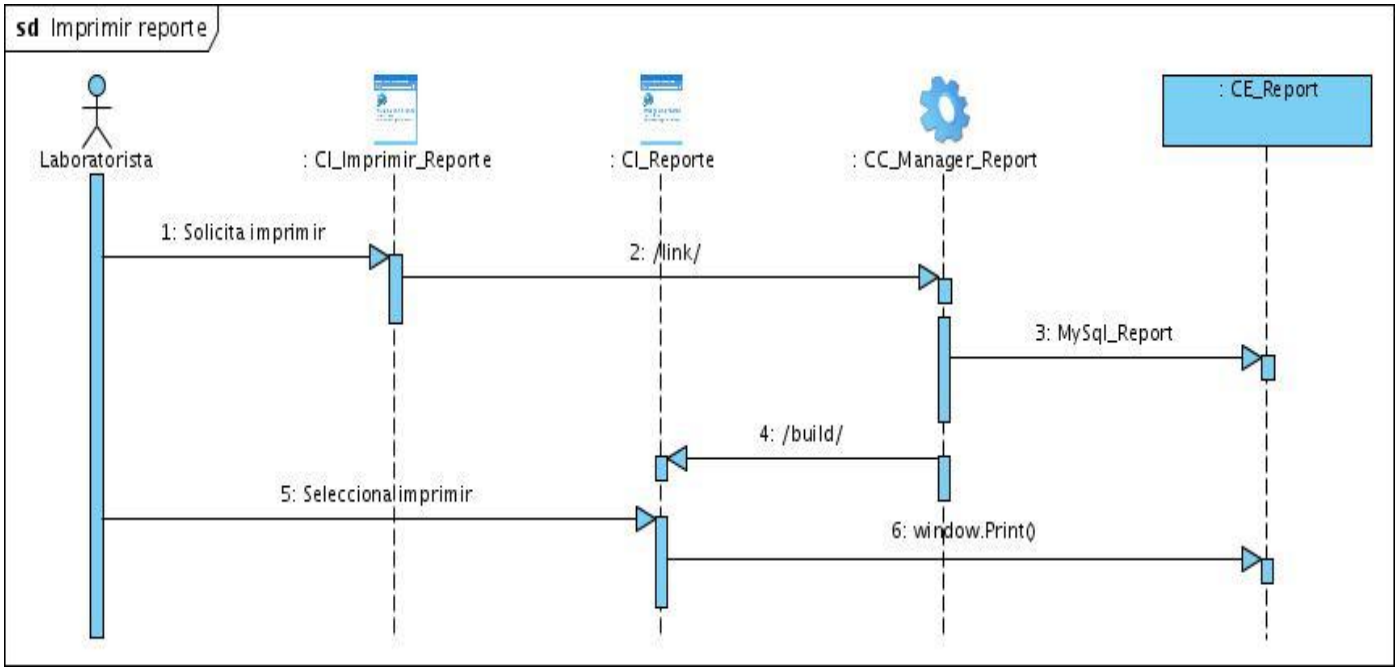
Anexo 17 Diagrama de Secuencia (Registrar usuario).



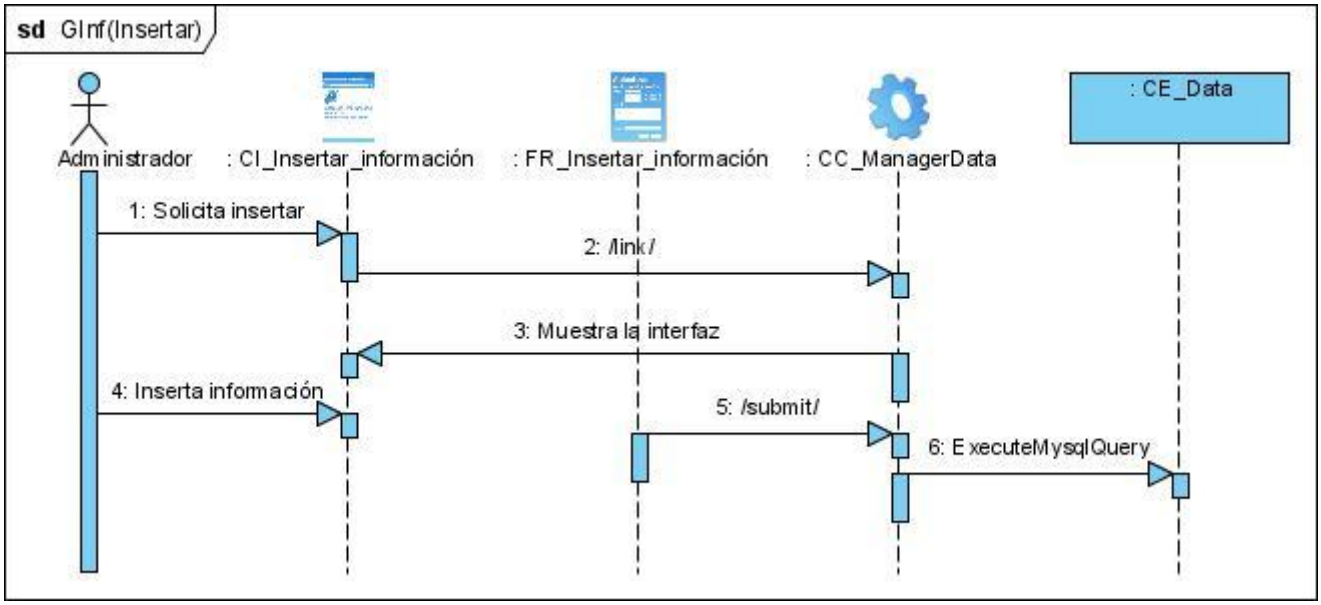
Anexo 18 Diagrama de Secuencia (Guardar reporte-Salvar reporte).



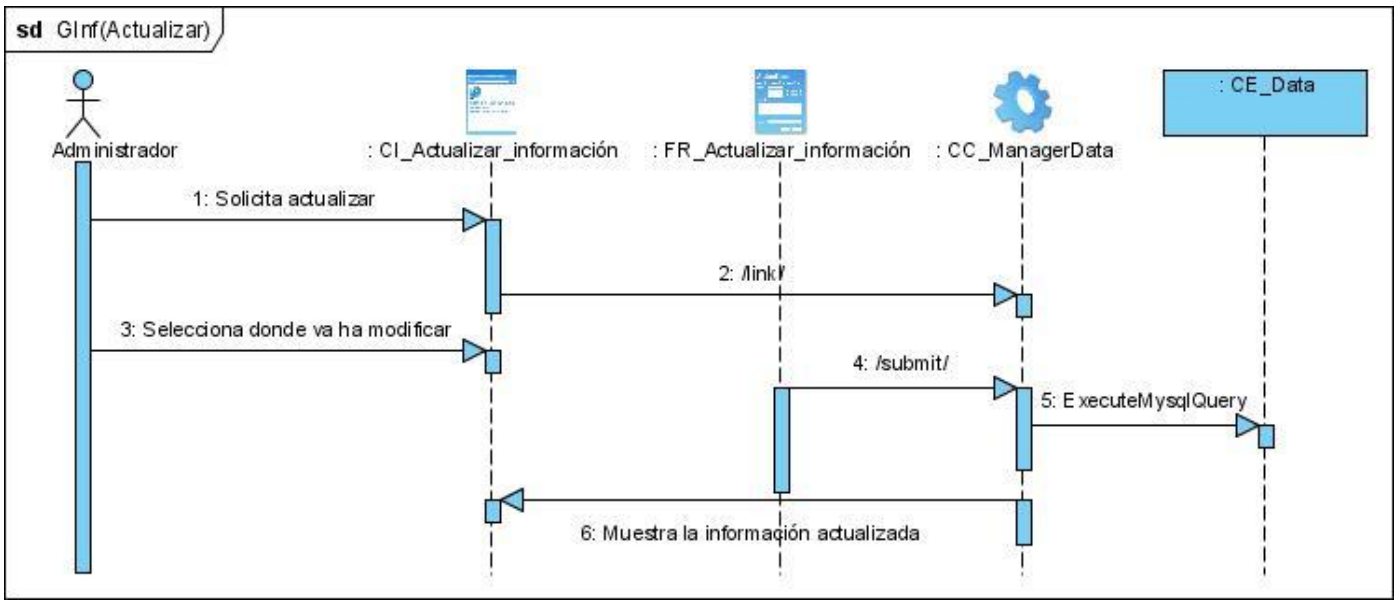
Anexo 19 Diagrama de Secuencia (Guardar reporte – Imprimir reporte).



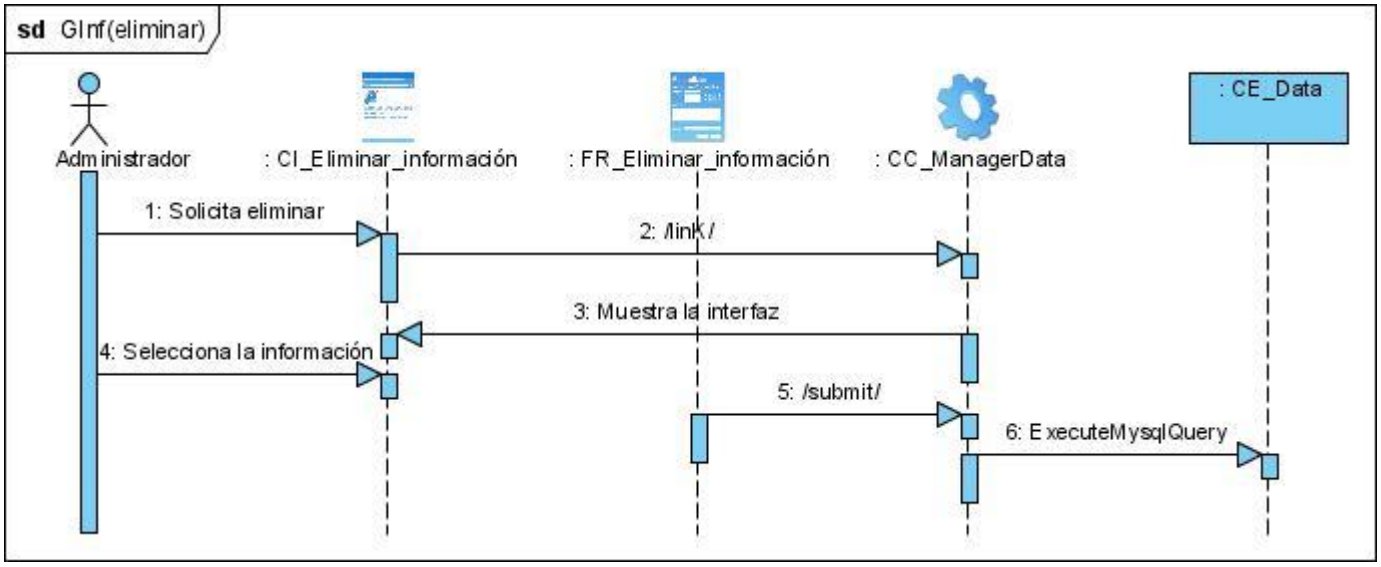
Anexo 20 Diagrama de Secuencia (Gestionar información-Insertar información).



Anexo 21 Diagrama de Secuencia (Gestionar información-Actualizar información)



Anexo 22 Diagrama de Secuencia (Gestionar información-Eliminar información).



GLOSARIO

- Algoritmo: Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.
- Caso de uso: Fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor
  para sus actores.
- Docking: Método de simulación por computadora para el desarrollo de fármacos.
- Fármaco: Medicamento.
- In sílico: Es una expresión que significa “hecho por computador o vía simulación computacional.
- Ligando: Los iones o moléculas que rodean a un metal en un complejo.
- Macromolécula: Molécula de gran tamaño, generalmente de muy elevado peso molecular y de origen natural o sintético.
- Molécula: Unidad mínima de una sustancia que conserva sus propiedades químicas. Puede estar formada por átomos iguales o diferentes.
- Molécula blanco: Molécula sobre la que actuará el medicamento para lograr el efecto terapéutico deseado.
- PERL (Lenguaje Práctico para la Extracción e Informe): Es un lenguaje de programación es propósito general originalmente desarrollado para la manipulación de texto y que ahora es utilizado para un amplio rango de tareas incluyendo administración de sistemas y desarrollo web.
- Simulaciones: Fingimiento, presentación de algo como real.