

Universidad de las Ciencias Informáticas
Facultad 6



***Título:** Servicio de Química Computacional para la
Plataforma de Servicios Bioinformáticos.*

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

Autores: Liuva Rosabal Valdés.

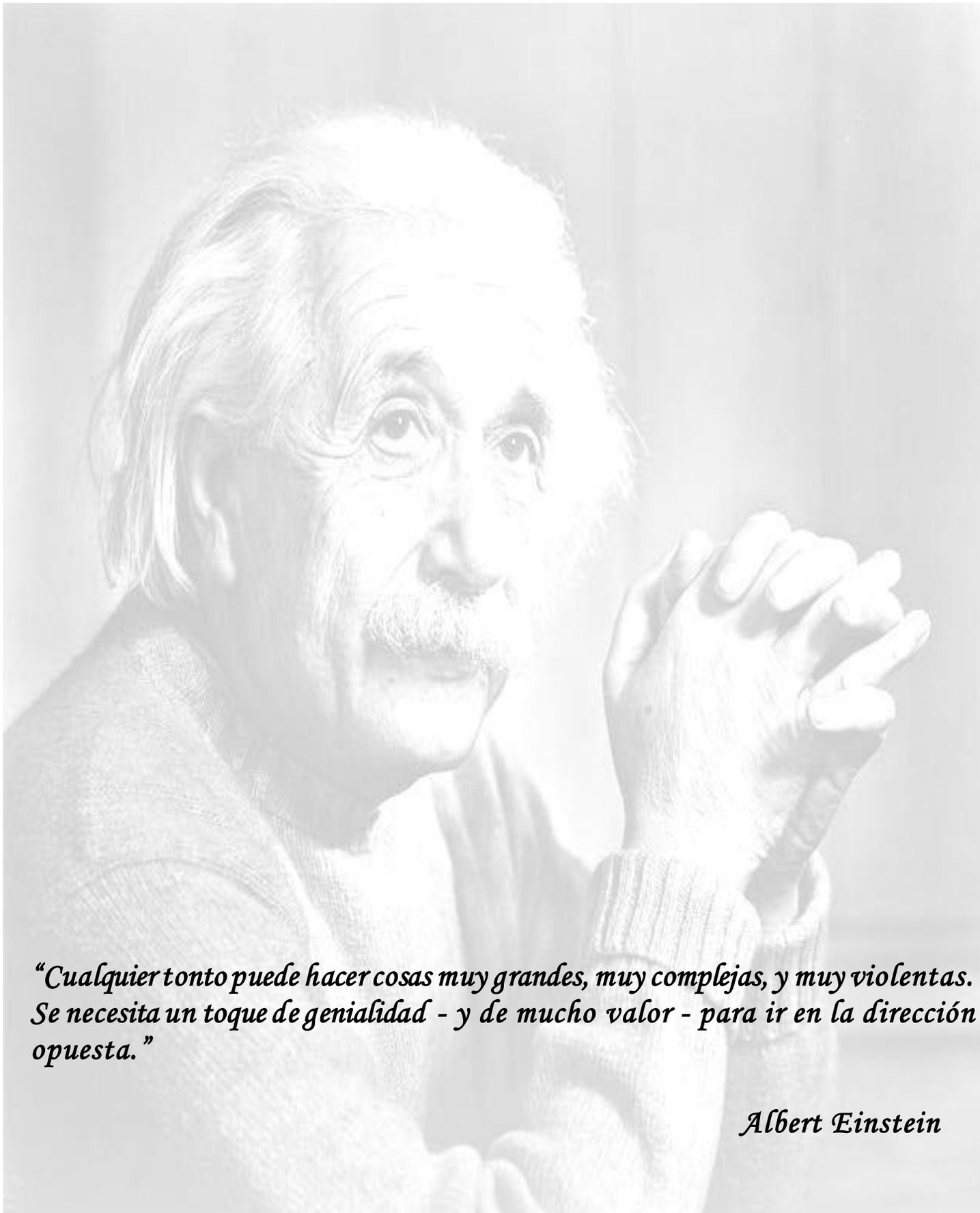
Alexei De Quesada Peña.

Tutores: MsC. Taymara Hernández Ortega.

Ing. Osvel Chávez Hernández.

La Habana, Junio 2013

“Año 55 de la Revolución”



“Cualquier tonto puede hacer cosas muy grandes, muy complejas, y muy violentas. Se necesita un toque de genialidad - y de mucho valor - para ir en la dirección opuesta.”

Albert Einstein

Declaración de autoría

Declaramos ser autores del presente trabajo de diploma 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 _____.

Liuva Rosabal Valdés

Alexei De Quesada Peña

Firma del Autor

Firma del Autor

MSc. Taymara Hernández Ortega

Ing. Osvel Hernández Chávez

Firma del Tutor

Firma del Tutor

Autores:

Liuva Rosabal Valdés.

Universidad de las Ciencias Informáticas, La Habana, Cuba

Email: lrosabal@estudiantes.uci.cu

Alexei De Quesada Peña.

Universidad de las Ciencias Informáticas, La Habana, Cuba

Email: adquesada@estudiantes.uci.cu

Tutores:

MsC. Taymara Hernández Ortega.

Universidad de las Ciencias Informáticas, La Habana, Cuba

Email: tay@uci.cu

Ing. Osvel Chávez Hernández.

Universidad de las Ciencias Informáticas

Email: ochavez@uci.cu

Agradecimientos

Difíciles el camino para llegar a donde soñamos desde nuestros inicios escolares. Por fin ha llegado el momento de demostrar que somos capaces de enfrentarnos al mundo como profesionales. En nuestro cursar por esta universidad y para la realización final de este trabajo pasamos momentos difíciles, y precisamente en estos momentos hubieron personas que nos dieron su apoyo incondicional, es por ello que le dedicamos estas líneas como agradecimientos.

A mi madre que me dio la posibilidad de vivir, que siempre espera lo mejor de mí y a quien le debo el haber llegado hasta aquí y mi espíritu de seguir adelante. A mi hermano por apoyarme en todos los momentos de mi vida. A mi padre por sus enseñanzas y aunque está lejos sé que estará muy orgulloso de mí. A mi novio Daldís por su compañía y apoyo en estos últimos años de la Universidad.

A Karel, Yudy, Mayde y Martha por ser excelentes compañeros con los cuales siempre pude contar.

A Ale, mi compañero de tesis por su entusiasmo en este trabajo de diploma.

A mis tutores Taymara y Osvel por su ayuda incondicional y experiencia brindada.

A la Revolución por haberme dado la oportunidad de estudiar en esta Universidad de excelencia.

En general a todas las personas que me han ayudado, que han compartido estos años conmigo.

A todos ustedes les estoy eternamente agradecida y de corazón les digo: muchas gracias.

Líuva

Agradecimientos

A Dios por ser la mano derecha que guía mi vida.

A mi familia por su apoyo incondicional y su confianza.

A mi novia por estar a mi lado en todo momento.

A mis amigos, que son mi otra familia.

A Lúva, por su esfuerzo y dedicación en este trabajo.

A mis tutores Taymara y Osvel por su guía y paciencia.

A todas las personas que de una forma u otra han contribuido a hacer realidad este trabajo.

A todos muchas gracias.

Ale

Dedicatoria

A mi madre, por sus esfuerzos que hoy son mis logros.

A mi hermano.

A mi padre.

Líuva

A mis padres, a mi hermana y a mi novia, porque siempre están a mi lado.

A todos mis amigos.

Ale

Resumen

Hoy en día, los cálculos de Química Computacional tienen un alto valor en la predicción de estructuras y funciones de las biomoléculas. En un vínculo estrecho con ciencias emergentes como la Bioinformática, estos cálculos benefician a la industria farmacéutica actual, principalmente en la etapa de diseño del fármaco. Para muchos centros de investigación de nuestro país realizar este tipo de cálculo constituye un desafío, no todos los especialistas cuentan con los recursos computacionales o algún servidor en la red nacional que brinde un servicio para la realización de este tipo de cálculos. Por esta razón el presente trabajo se enmarca en la concepción e implementación de un Servicio de Química Computacional que forma parte de la Plataforma de Servicios Bioinformáticos (PSBIO) del centro de Tecnología y Gestión de Datos (DATEC) de la Universidad de las Ciencias Informáticas. El mismo brinda la posibilidad de realizar estudios a los investigadores de universidades y centros del país que promueven y desarrollan la Química Computacional. Con este fin se realiza un análisis de los principales programas de cálculos químicos-teóricos y de las tecnologías para desarrollar el sistema. El proceso de desarrollo de *software* estuvo guiado por la metodología OpenUP y se utilizó como lenguaje de programación Java. Finalmente se obtuvo un portlet que brinda la posibilidad de realizar cálculos químicos-teóricos usando los programas Mopac2012 y Gamess.

Palabras clave: Servicio de Química Computacional, Bioinformática, Plataforma de Servicios Bioinformáticos, portlet.

INTRODUCCIÓN	2
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	6
1.1. LA QUÍMICA COMPUTACIONAL	6
1.2. PROGRAMAS DE QUÍMICA COMPUTACIONAL	7
1.2.1. <i>Gaussian</i>	7
1.2.2. <i>Molpro</i>	7
1.2.3. <i>Q-Chem</i>	8
1.2.4. <i>Amsol</i>	8
1.2.5. <i>Mopac</i>	8
1.2.6. <i>Gamess</i>	8
1.2.7. <i>Gabedit</i>	9
1.3. RECURSOS DE QUÍMICA COMPUTACIONAL EN LA WEB.....	10
1.3.1. <i>Servicio web</i>	10
1.3.2. <i>Arquitectura Orientada a Servicios</i>	11
1.4. TECNOLOGÍAS Y LENGUAJES DE PROGRAMACIÓN PARA EL DESARROLLO DE APLICACIONES WEB	12
1.4.1. LENGUAJES Y PLATAFORMA DE DESARROLLO	12
1.5. APACHE TOMCAT	13
1.6. MARCO DE TRABAJO.....	13
1.6.1. <i>Hibernate</i>	13
1.6.2. <i>Apache Axis2</i>	14
1.6.3. <i>Spring Framework</i>	14
1.7. <i>Componente de la plataforma J2EE</i>	15
1.7.1. <i>Contenedor de Portlets</i>	15
1.8. PLATAFORMA DE SERVICIOS BIOINFORMÁTICOS	16
1.9. SISTEMAS GESTORES DE BASES DE DATOS	16
1.9.1. <i>PostgreSQL 9.1</i>	16
1.10. ENTORNO DE DESARROLLO IDE	17
1.10.1. <i>Eclipse</i>	17
1.11. METODOLOGÍA DE DESARROLLO.....	17

1.12. LENGUAJE DE MODELADO.....	18
1.13. HERRAMIENTA CASE.....	18
CAPÍTULO 2. CARACTERÍSTICAS DEL SISTEMA.....	19
2.1. DESCRIPCIÓN DE SISTEMA.....	19
2.2. MODELO DEL DOMINIO.....	19
2.3. ESPECIFICACIÓN DE REQUISITOS.....	20
2.3.1. <i>Requisitos funcionales</i>	20
2.3.2. REQUISITOS NO FUNCIONALES.....	21
2.4. DIAGRAMA DE CASOS DE USO DEL SISTEMA.....	23
2.5. DEFINICIÓN DE LOS CASOS DE USO DEL SISTEMA.....	23
2.5.1. <i>Actores del sistema</i>	24
2.5.2. <i>Descripción de los Casos de Usos del sistema</i>	24
CAPÍTULO 3. DISEÑO DEL SISTEMA.....	26
3.1. ARQUITECTURA DEL SISTEMA.....	26
3.1.1. <i>Estilos y patrones arquitectónicos</i>	26
3.1.2. <i>Patrones de diseño</i>	27
3.2. DIAGRAMA DE CLASES DEL DISEÑO.....	30
3.3. DIAGRAMAS DE INTERACCIÓN.....	33
3.3.1. <i>Diagramas de secuencia</i>	33
3.4. MODELO DE DATOS.....	34
3.4. VISTA DE DESPLIEGUE.....	35
CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBA.....	36
4.1. MODELO DE IMPLEMENTACIÓN.....	36
4.1.1. <i>Diagrama de componentes</i>	36
4.2. PRUEBAS.....	38
4.2.1. <i>Casos de pruebas de caja negra</i>	38
4.3. RESULTADOS.....	43
4.3.1. <i>Servicio de química computacional con Gamess</i>	43
4.3.2. <i>Servicio de química computacional con Mopac</i>	45
4.4. INTERFACES DE LA APLICACIÓN.....	50

CONCLUSIONES GENERALES	53
RECOMENDACIONES.....	54
REFERENCIAS BIBLIOGRÁFICAS.....	55
BIBLIOGRAFÍA.....	58
ANEXOS	61
DESCRIPCIÓN TEXTUAL DE CASOS DE USOS.....	61
DIAGRAMA DE CLASES DEL DISEÑO	65
DIAGRAMA DE SECUENCIAS.....	66
DIAGRAMA DE COMPONENTE.....	66

Índice de figuras

Figura 1. Componentes básicos de una arquitectura SOA.....	11
Figura 2. Modelo Cliente-Servidor	12
Figura 3. El patrón MVC con Spring Framework.....	14
Figura 4. Modelo de Dominio.....	20
Figura 5. Diagrama de Casos de Uso del sistema.....	23
Figura 6. Patrón Modelo Vista Controlador	27
Figura 7. Diagrama de clases del diseño Realizar cálculos químicos teóricos con Mopac	30
Figura 8. Diagrama de clases del diseño Realizar cálculos químicos teóricos con Gamess.....	31
Figura 9. Diagrama de clases del diseño para el paquete Objeto de Acceso a Datos	31
Figura 10. Diagrama de clases del diseño para el paquete Cliente de Servicios Web con Mopac.....	32
Figura 11. Diagrama de clases del diseño para el paquete Cliente de Servicios Web con Gamess.....	32
Figura 12. Diagrama de secuencia para Realizar cálculos químicos teóricos con Mopac	33
Figura 13. Diagrama de secuencia para Realizar cálculos químicos teóricos con Gamess	34
Figura 14. Modelo de Datos.....	34
Figura 15. Vista de despliegue	35
Figura 16. Diagrama de Componentes del CU Realizar cálculos químicos teóricos con Mopac	37
Figura 17. Diagrama de Componentes del CU Realizar cálculos químicos teóricos con Gamess	38
Figura 18. Estructura del fichero de entrada para Mopac.....	46
Figura 19. Tiempos de respuesta de Mopac en distintos entornos de ejecución	47
Figura 20. Tiempos de respuesta variando la cantidad de PC	48
Figura 21. Speed-up para un problema de 100 moléculas	48
Figura 22. Eficiencia en el entorno distribuido	49

Figura 23. Interfaz Principal	50
Figura 24. Interfaz Administrar Ejecuciones.....	50
Figura 25. Interfaz Crear Ejecución con Mopac	51
Figura 26. Interfaz Administrar Resultados	51
Figura 27. Diagrama del Diseño del CU Gestionar Ejecución	65
Figura 28. Diagrama de secuencia del CU Gestionar Ejecución de la sección Crear Ejecución	66
Figura 29. Diagrama de componentes del CU Gestionar Ejecución de la sección Crear Ejecución.....	66

Índice de tablas

Tabla 1. Actores del Sistema	24
Tabla 2. Descripción del CU "Realizar cálculos químicos teóricos con Mopac"	24
Tabla 3. Descripción del CU "Realizar cálculos químicos teóricos con Gamess"	25
Tabla 4. Variables para el Caso de Prueba Crear Ejecución.....	39
Tabla 5. Caso de prueba 1: Crear Ejecución.....	39
Tabla 6. Variables para el Caso de Prueba Modificar Ejecución.....	40
Tabla 7. Caso de prueba 2: Modificar Ejecución.....	40
Tabla 8. Variables para el Caso de Prueba Buscar Ejecución.....	42
Tabla 9. Caso de prueba 3: Buscar Ejecución	42
Tabla 10. Caso de prueba 4: Eliminar Ejecución.....	42
Tabla 11. Tiempos de respuesta para una molécula utilizando varios métodos <i>Semiempíricos</i>	45
Tabla 12. Descripción del CU Gestionar Ejecuciones	61
Tabla 13. Descripción del CU Administrar Resultados.....	63

Introducción

Los procesos químicos pueden ser explicados mediante la descripción matemática tanto de los sistemas que lo componen como del mecanismo de interacción entre ellos. En este caso se refiere el uso de los métodos teóricos de la química como área de investigación para explicar la naturaleza de estos fenómenos.

De acuerdo a las características del problema a abordar se utilizan diferentes métodos, agrupados en dos fundamentos teóricos: los basados en las leyes de la mecánica clásica y los basados en las leyes de la mecánica cuántica (1). Se tienen indistintamente en este grupo los métodos *Semiempíricos*¹, *ab initio*², *Teoría Funcional de la Densidad (DFT por sus siglas en inglés)*, *Mecánica Molecular (MM)*, así como los de *Dinámica Molecular* y *Monte Carlo*.

Cuando los métodos matemáticos para la descripción de los sistemas químicos están lo suficientemente bien descritos como para poder ser implementados en un determinado *software*, se emplea para referirse a ellos el término Química Computacional. Los avances en esta área han sido posible debido en gran medida a que el *software* ha tenido un desarrollo vertiginoso en los últimos años, posibilitando el surgimiento de programas que permiten el empleo de estos métodos sin necesidad de un entendimiento a profundidad de los complejos procedimientos teóricos que estos encierran (2). A pesar de ello, aún quedan limitaciones en su empleo, debidas principalmente al costo computacional que implica el uso de algunos basamentos teóricos como los de *Coupled Clusters (C.C por sus siglas en inglés)* en *ab initio* para sistemas de más de 100 átomos.

Los estudios computacionales de química pueden ser usados para predecir estructuras moleculares hasta la fecha totalmente desconocidas o explorar mecanismos de reacción difíciles de dilucidar haciendo uso de métodos experimentales. Con el empleo de estos, se logran acortar los ciclos de desarrollo de los productos y/o procesos, además de diseñar nuevos productos y procesos u optimizar los existentes.

En las ciencias de la vida, la Química Computacional tiene un alto valor en la predicción de estructuras y funciones de las biomoléculas. En un vínculo estrecho con ciencias emergentes como la Bioinformática, estos métodos benefician a la industria farmacéutica actual, principalmente en la etapa de diseño del fármaco.

¹ Métodos que hacen muchas aproximaciones y obtienen algunos parámetros de datos empíricos.

² Métodos que no incluyen ningún parámetro empírico o semiempírico en sus ecuaciones.

La alta demanda de cómputo que exige el empleo de varios de estos métodos, ha hecho necesario para numerosas instituciones en el mundo contar con potentes ordenadores dedicados a realizar este tipo de estudios. Además han creado proyectos y redes de colaboración entre centros para el uso de estos recursos a través de la web. Desafortunadamente en el caso de la Química Computacional, no ha ocurrido lo que en otras áreas y estos servicios han quedado limitados a entidades relacionadas por intereses comunes.

En Cuba el empleo de estos métodos cobra importancia en renglones como la Biotecnología, la Industria Farmacéutica, Ciencia de los Materiales, Nanociencias, etc. Por lo que existen instituciones que cuentan con recursos computacionales para realizar cálculos de Química Computacional. Se pueden mencionar entre ellos el Centro de Ingeniería Genética y Biotecnología (CIGB), el Centro de Inmunología Molecular (CIM) y además algunos grupos de investigación, como los pertenecientes a las universidades de la Habana y Villa Clara, por solo citar algunos ejemplos. Sin embargo no todos los especialistas cuentan con los recursos computacionales o algún servidor en la red nacional que brinde un servicio para la realización de este tipo de cálculos.

La Universidad de las Ciencias Informáticas (UCI) cuenta con varios centros de desarrollo de *software*, donde se encuentra el Centro de Tecnología y Gestión de Datos (DATEC). En el departamento de Bioinformática de este centro se desarrolla una Plataforma de Servicios Bioinformáticos (PSBIO), la cual tiene como principal objetivo poner a disposición de la comunidad científica del país un grupo de servicios básicos asociados al área de la Bioinformática. Hasta la fecha dicha plataforma no brinda la posibilidad a los usuarios de realizar cálculos que utilicen los métodos englobados en el área de la Química Computacional. Lo que sería de gran utilidad debido al aporte que brindan estos métodos al estudio de sistemas biológicos, por lo que se plantea como **problema a resolver**: ¿Cómo garantizar a los especialistas del país el manejo de herramientas básicas para realizar cálculos de Química Computacional a través de la Plataforma de Servicios Bioinformáticos de la Universidad de las Ciencias Informáticas?

Se define como **objeto de estudio** el desarrollo de servicios de cálculos científicos, enmarcado en el **campo de acción** el desarrollo de servicios de cálculo en la Plataforma de Servicios Bioinformáticos.

Como **objetivo general** se plantea: Desarrollar un Servicio de Química Computacional para la Plataforma de Servicios Bioinformáticos de la UCI.

Desglosado en los siguientes **objetivos específicos**:

- ❖ Caracterizar las técnicas y herramientas necesarias para el desarrollo del Servicio de Química Computacional.
- ❖ Definir los requisitos funcionales del Servicio de Química Computacional.
- ❖ Diseñar el Servicio de Química Computacional.
- ❖ Implementar el Servicio de Química Computacional.
- ❖ Validar el Servicio de Química Computacional.

Para dar cumplimiento a los objetivos específicos se trazaron las siguientes **tareas de investigación**:

- ❖ Selección de las herramientas que realizan cálculos de Química Computacional.
- ❖ Identificación de los requisitos funcionales para el Servicio de Química Computacional.
- ❖ Implementación de los requisitos funcionales identificados.
- ❖ Diseño de los casos de pruebas para la realización de pruebas al Servicio de Química Computacional.
- ❖ Aplicación de los casos de pruebas para validar Servicio de Química Computacional.

Arrojando como **posible resultado** un Servicio de Química Computacional en la Plataforma de Servicios Bioinformáticos, que utilice los recursos computacionales que existen y están disponibles en nuestra Universidad.

El presente trabajo se encuentra estructurado en cuatro capítulos, resumidos de la siguiente forma:

Capítulo 1: Fundamentación teórica.

Abarca temas relacionados con la Química Computacional y los principales programas englobados en esta área. Se describen las tecnologías, herramientas y metodologías a utilizar en la implementación de la solución siguiendo las directrices propuestas por la Plataforma de Servicios Bioinformáticos.

Capítulo 2: Características del sistema.

Durante el desarrollo de este capítulo se brinda un fundamento de la solución de la propuesta. Se presentan los requisitos funcionales y no funcionales, se identifican los casos de uso del sistema y se describen los mismos.

Capítulo 3: Diseño del sistema.

Se analizan los Casos de Usos del Sistema para diseñar las clases que se implementan, se presentan los Diagramas de Secuencia del Diseño, los Diagramas de las Clases del diseño con sus relaciones, los patrones arquitectónicos y de diseños utilizados, así como el modelo de datos y la vista de despliegue.

Capítulo 4: Implementación y Prueba.

El contenido que se aborda en este capítulo está basado fundamentalmente en la implementación del sistema y responde a los requisitos previamente establecidos, se especifican las pruebas a las que fue sometida la aplicación.

Capítulo 1

Fundamentación teórica

En el presente capítulo se aborda el fundamento teórico relacionado con la Química Computacional. Se fundamenta la selección de los programas de cálculos. Además se analizan las principales herramientas y tecnologías web para el desarrollo de la solución propuesta, teniéndose en cuenta las directrices propuestas por la Plataforma de Servicios Bioinformáticos.

1.1. La Química Computacional

La Química Computacional es el campo de la ciencia dedicado al desarrollo de modelos moleculares (3), existen dos grandes áreas en la Química Computacional, la Química Cuántica (QC), que permite modelar las densidades electrónicas y que ayuda a entender el comportamiento de los sistemas de una forma relativamente rigurosa, y las técnicas de MM, que imitan los resultados cuánticos, sin calcular las densidades electrónicas.

La Química Computacional abarca una gran diversidad de métodos, estos pueden ser agrupados atendiendo a dos bases teóricas fundamentales: (3,4)

- ❖ Los métodos basados en las leyes de la mecánica clásica, los cuales adaptan las ecuaciones de la física clásica a las interacciones atómicas y moleculares. Entre estos métodos se encuentran los de mecánica molecular, los métodos de Monte Carlo y de dinámica molecular.
- ❖ Los métodos que se basan en las leyes de la mecánica cuántica, los cuales describen propiedades físicas y químicas de los sistemas en sus detalles atómicos. Estos métodos se dividen en dos grupos fundamentales:
 - los métodos que describen a los sistemas mediante una función de onda con el empleo de la ecuación de *Schorödinguer*³ y que, según el nivel de teoría utilizado, se subdividen en *ab initio* y *Semiempíricos*.

³ Ecuación base para casi todos los métodos mecanocuánticos de la química computacional.

- los métodos *DFT* mediante los cuales se calculan las propiedades de un sistema electrónico partiendo directamente de la densidad electrónica.

La selección del método que debe ser utilizado en cada problema científico en concreto, es dependiente de las características del sistema en particular, tales como: los medios de los que se dispone, el tamaño del sistema a estudiar y la precisión de los datos que deseamos obtener. (5)

1.2. Programas de Química Computacional

En la actualidad existen varios programas que implementan los métodos antes mencionados y son comúnmente utilizados por distintos grupos de investigación, enmarcados en el área de la Química Computacional.

1.2.1. Gaussian

Gaussian es un programa para la química cuántica publicado inicialmente en 1970 por John Pople y su grupo de investigación en la Universidad de Carnegie-Mellon, desarrollado y licenciado por Gauss, Inc (5). Este programa es conocido por su gran sencillez de uso y su amplia variedad de métodos y técnicas teóricas especializadas en métodos *ab initio*. Gaussian 09 es la última versión de programas de Gaussian, puede ejecutarse en sistemas con una CPU y en paralelo en memoria compartida (6). Esta versión contiene GaussView 5, la cual es una interfaz gráfica avanzada y potente que brinda facilidades de uso a los usuarios de Gaussian. El programa puede ser comprado como código fuente o archivos ejecutables. Posee versiones para los sistemas operativos Windows y Linux.

1.2.2. Molpro

Molpro es un paquete de *software* utilizado para los cálculos *ab initio*, desarrollado por Peter Knowles en la Universidad de Cardiff, en colaboración con otros autores (7,8). El *software* y la documentación de Molpro es material con *copyright* de University College de Cardiff Consultants Limited y se distribuye bajo licencia propietaria a los usuarios finales (8), además cuenta con una licencia académica de 30 días. En su versión Molpro2012 incluye varios métodos con implementación paralela. MolproView es la herramienta disponible para la visualización gráfica de los resultados de Molpro. Contiene versiones disponibles para el sistema operativo Linux.

1.2.3. Q-Chem

Q-Chem es un *software* de química computacional comercial distribuido por Q-Chem, Inc., una compañía de *software* con sede en Pittsburgh, Pennsylvania, EE.UU (9). Es un programa *ab initio* diseñado para cálculos eficientes en grandes moléculas. Q-Chem es un producto de *software* comercial. Su última versión es Q-Chem 4.0.1, la cual contiene paralelización de sus métodos. Soporta plataformas como: Linux, Microsoft Windows y MAC OS X.

1.2.4. Amsol

Amsol es un programa de mecánica cuántica *Semiempíricos* desarrollado y distribuido por la Universidad de Minnesota Departamento de Química. El programa contiene un código fuente, un conjunto de pruebas y documentación. La última versión del programa es Amsol 7.1 (diciembre del 2004) (10). La Universidad de Minnesota concede una licencia no exclusiva, intransferible y perpetua para utilizar el *software*. La Universidad retiene el derecho irrevocable a practicar el *software* por su propia educación, investigación y con fines de desarrollo y se reserva todos los derechos para entrar en acuerdos académicos, comerciales y de usuario con otros partidos para el *software*. Amsol corre sobre la plataforma UNIX.

1.2.5. Mopac

Mopac es un paquete de orbitales molecular que realiza cálculos *Semiempíricos* para el estudio de estructuras de estado sólido y reacciones moleculares, su versión más reciente es Mopac2012. Es utilizado en el procedimiento de hipersuperficies de múltiples mínimos (MMH) desarrollo en Cuba por Montero y colaboradores (11,12). Puede ejecutarse sobre los sistemas operativos Linux y Windows, además posee una licencia académica para su uso. Sus ejecutables están disponibles gratuitamente para fines académicos y su licencia académica debe ser renovada anualmente (12).

1.2.6. Gamess

Gamess es mantenido por miembros del grupo de investigación de Gordon de la Universidad de Iowa State. Posee una alta calidad, lo cual hace que sea uno de los programas favoritos de Química Computacional, especializado en cálculos *ab initio* después de Gaussian, para la comunidad de investigadores académicos (13). Está disponible sin costo para usuarios académicos e industriales, contiene paralelizaciones de sus métodos y su última versión es la de Mayo 2012 (13,14). Puede ejecutarse sobre los sistemas operativos Linux, Microsoft Windows, MAC OS X. Gamess a diferencia de

Gaussian, no posee una interfaz gráfica de usuario para poder crear archivos de entrada, constituyendo un paquete de *software* menos amigable que Gaussian.

1.2.7. Gabedit

Gabedit es una interfaz gráfica de usuario para paquetes de Química Computacional como Gamess, Gaussian, Molcas, Molpro, Orca y Q-Chem. Puede mostrar una variedad de resultados de cálculo, incluyendo soporte para la mayoría de los principales formatos de archivo moleculares. Está disponible gratis para su descarga en su sitio oficial (15).

Gabedit permite:

- ❖ Construir moléculas por átomo, anillo, grupo, de aminoácidos y nucleótidos.
- ❖ Crea archivo de entrada para los paquetes Gamess, Gaussian, Molcas, Molpro, Mopac, Orca y Q-Chem.
- ❖ Lee la salida de los paquetes Gamess, Gaussian, Molcas, Molpro, Mopac, Orca y Q-Chem, y soporta un número de otros formatos.
- ❖ Muestra orbitales moleculares o la densidad de electrones como mapas de contorno o parcelas de cuadrícula 3D y salida a una serie de formatos gráficos.
- ❖ Anima vibraciones moleculares, contornos, isosuperficies y rotación.

Fundamento de la selección

Luego de la caracterización de los principales programas de cálculo de Química Computacional se concluye que Gaussian es el *software* más empleado en esta área de investigación, dado a su robustez y facilidad de uso. Sin embargo posee como limitante ser un *software* bajo licencia privativa y pese a los intentos de obtención de esta por la Comunidad Científica Cubana, no ha podido ser adquirido por el hecho de ser un producto desarrollado por la compañía estadounidense Gaussian, Inc.

Otros programas como Molpro, Q_Chem y Amsol también poseen licencia privativa, por lo que se descartan de la selección.

Se decide entonces seleccionar el Gamess como paquete *ab initio* de Química Computacional por mostrar resultados muy similares a los obtenidos con la aplicación de los métodos implementados en Gaussian y además por ser *software* libre.

Se selecciona también a Mopac2012, pues posee una licencia académica para su uso y es reconocida como una de las aplicaciones vanguardias en la resolución de problemas haciendo uso de métodos *Semiempíricos*. Es utilizado en los procedimientos MMH, implementado y utilizado por científicos cubanos. Por la complejidad que conlleva crear los datos de entrada para los programas antes mencionados, se decide incluir la aplicación Gabedit, la cual permite crear ficheros y realizar análisis de los resultados de los programas Mopac2012 y Gamess.

1.3. Recursos de química computacional en la web

Internet es la herramienta más usada de las Tecnologías de la Informática y las Comunicaciones en la actualidad y cuando se habla de procesamiento y gestión de datos, relacionados a ciencias como la Bioinformática, juega un papel fundamental como plataforma de intercambio de información.

La Red Latinoamericana de Química publica cada año un directorio de las instituciones y profesionales de la química en América Latina, con la finalidad de promover la comunicación y la colaboración entre los investigadores latinoamericanos. Esta presenta los recursos de química en el mundo, *software* de Química Computacional, bases de datos, entre otros.

En el mundo existen numerosos centros e instituciones que poseen potentes ordenadores dedicados a realizar estudios científicos, entre ellos se encuentra el Centro de Supercomputación de Galicia (CESGA), que proporciona servicios de cálculo y comunicaciones a la comunidad científica y a la industria principalmente en Europa (16). Además el Centro Nacional de Supercomputación y la Red Española de Supercomputación (BSC-CNS por sus siglas en inglés) cuyo objetivo es promover la investigación y desarrollo en Ciencias de la Computación, Ciencias de la Vida y Ciencias de la Tierra, así como brindar soporte de supercomputación para investigaciones externas (17). Para tener acceso a los recursos computacionales que se brindan en estas instalaciones, los usuarios deben estar asociados a instituciones o entidades relacionadas a estos centros, o en otros casos pagar para hacer uso de los servicios de cálculo científico que se brindan a través de la web.

1.3.1. Servicio web

Una de las tendencias actuales, dada la dificultad de integrar las funcionalidades de diversas aplicaciones y la gran cantidad de datos que estas intercambian, es la creación de *web services* (servicios web, en español) (18). Estos permiten la interacción computadora-computadora para realizar operaciones de cómputos que procesen gran cantidad de información, y además brindan la posibilidad de crear

aplicaciones más complejas que reutilicen las funcionalidades de otras, logrando de esta manera gran interoperabilidad entre ellas e independencia de las plataformas en la que están construidas.

Utilizando la Web como plataforma, los usuarios, de forma remota, pueden solicitar un servicio que algún proveedor ofrezca en la red. Pero para que esta interacción funcione, deben existir unos mecanismos de comunicación estándares entre diferentes aplicaciones. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios web se ha creado el organismo WS-I⁴ (*Web Services Interoperability* por sus siglas en inglés), encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares. (18)

1.3.2. Arquitectura Orientada a Servicios

Para aprovechar el potencial de los servicios web estos deben ser diseñados cuidadosamente bajo los conceptos y principios de la orientación a servicios propuestos por la Arquitectura Orientada a Servicios (SOA por sus siglas en inglés). SOA provee una serie de conceptos que garantizan un mejor diseño de aplicaciones en cuanto a seguridad, administración de servicios, confiabilidad, separación en unidades lógicas de procesamiento y calidad de los servicios brindados (18). Los componentes básicos de SOA se representan en la Figura 1.



Figura 1. Componentes básicos de una arquitectura SOA

SOA organiza las funcionalidades en servicios web que pueden ser consumidos por aplicaciones clientes, las cuales intercambian información con el proveedor de servicios mediante el Protocolo Simple de Acceso a Objeto (SOAP por sus siglas en inglés). El proveedor publica los servicios en un registro denominado *UDDI* (siglas en inglés de *Universal Description, Discovery and Integration*), el cual especifica

⁴ Conjunto de especificaciones estándares recomendadas para desarrollar servicios web.

mediante documentos descriptores de servicios web (*WSDL*⁵ por sus siglas en inglés) las funciones y los parámetros necesarios para poder utilizar el servicio web.

Para la implementación del Servicio de Química Computacional se optó por el uso de una Arquitectura Orientada a Servicios siguiendo los principios de orientación a servicios que esta propone.

1.4. Tecnologías y lenguajes de programación para el desarrollo de aplicaciones web

El nacimiento de Internet marcó un cambio en casi todos los sectores de la sociedad, muchas de las empresas que actualmente dominan en el mundo basan su negocio en la red de redes. La arquitectura cliente-servidor es el centro de Internet y la WWW (por sus siglas en inglés *World Wide Web*), posee un estilo de llamada y retorno, también conocido como petición-respuesta (*request-response*) (19). En este modelo el cliente que no es más que un usuario utilizando un navegador solicita un recurso al servidor y este responde en un formato que es interpretado por el navegador que finalmente muestra al usuario la respuesta. Ambos son los componentes fundamentales el modelo cliente-servidor (Figura 2)

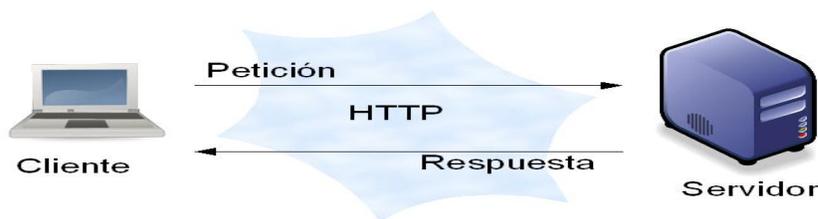


Figura 2. Modelo Cliente-Servidor

1.4.1. Lenguajes y plataforma de desarrollo

Actualmente existen muchos lenguajes de programación para desarrollar aplicaciones web, entre los más utilizados se pueden citar PHP, Python, C# y Java. Dada las características de robustez de Java, de ser multiplataforma, además de ser software libre, es muy usado en el mundo del desarrollo de aplicaciones.

Java es un lenguaje de programación sencillo, orientado a objetos, de propósito general e independiente de la plataforma de desarrollo. Es toda una tecnología orientada al desarrollo de *software* con el cual se puede realizar cualquier tipo de programa (20). Hoy en día, la tecnología Java ha cobrado auge en el ámbito de Internet gracias a su plataforma J2EE (por sus siglas en inglés Java 2 Enterprise Edition).

⁵ Describe la interfaz pública a los servicios Web, está basado en XML.

La Java 2 Enterprise Edition es la extensión de Java para el desarrollo web, dispone de varias herramientas de código abierto como IDEs, servidores, marcos de trabajo y APIs (por sus siglas en inglés *Applications Programming Interface*). La plataforma (J2EE) engloba dentro de sí un conjunto de especificaciones APIs relacionadas tales como JDBC (por sus siglas en inglés Java Database Connectivity), RMI (por sus siglas en inglés Remote Method Invocation), JMS (por sus siglas en inglés Java Message Service), Servicios Web, XML (por sus siglas en inglés eXtensible Markup Language), etc., además configura algunas especificaciones únicas como Enterprise JavaBeans, *servlets*, Java Server Pages, portlets y varias tecnologías de servicios web, que le permiten al desarrollador crear una aplicación empresarial portable entre plataformas y a la vez integrable con otras tecnologías. (21)

Las aplicaciones creadas bajo esta plataforma pueden ser desplegadas en cualquier sistema operativo. Al utilizar Java como lenguaje de programación, el cual está bajo la licencia GNU/GPL y disponer de la máquina virtual de Java, para la ejecución del código, la cual es completamente libre, J2EE es una opción acertada de acuerdo con las políticas de migración al *software* libre en nuestro país.

1.5. Apache Tomcat

Es un servidor web de código abierto que une la tecnología Java Servlet y Java Server Pages. Apache Tomcat fue escrito en Java, por lo que funciona en cualquier sistema operativo que disponga de la máquina virtual de Java, es multiplataforma. Es desarrollado y publicado bajo la licencia Apache 2.0, funciona como un contenedor de *servlets* y servidor HTTP desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Apache Tomcat implementa las especificaciones de los *servlets* y de Java Server Pages (JSP) de Sun Microsystems. (22)

1.6. Marco de trabajo

Un marco de trabajo (*framework*, en inglés) es una pieza de *software* estructural. Se dice estructural porque estructura es el objetivo de un *framework* que especifica todo requerimiento funcional. Un marco de trabajo trata de hacer generalizaciones acerca de las tareas comunes e intenta proveer una plataforma donde las aplicaciones pueden ser rápidamente construidas.

1.6.1. Hibernate

Es una herramienta *software* libre de mapeo objeto-relacional para ambientes de desarrollo en Java. Hibernate facilita el mapeo de atributos entre una Base de Datos (BD) relacional tradicional y el modelo orientado a objetos de una aplicación. Posee un lenguaje de consultas propio llamado Hibernate Query

Lenguaje (HQL) que puede ser usado por los programadores sin tener conocimientos de SQL. Brinda diferentes beneficios como son la generación del código Java a partir del modelo de base datos y viceversa, así como el mapeo de clases. (23)

1.6.2. Apache Axis2

Es un servidor web, de código abierto que une la tecnología Java Servlet y Java Server Pages. Es desarrollado en un entorno abierto y participativo y publicado bajo la licencia Apache 2.0. Está destinado a ser una colaboración de los desarrolladores de los mejores de su clase de todo el mundo. En él se pueden montar aplicaciones de misión crítica a través de la Web. Es un servidor HTTP y un contenedor de servlets (clases de Java), cargándolos y ejecutándolos de forma dinámica. Es compatible con los protocolos de comunicación SOAP 1.1 y 1.2 (24).

1.6.3. Spring Framework

Es un marco de trabajo para el desarrollo de aplicaciones web en Java, de código abierto y está bajo la licencia Apache (25). Presenta un arquitectura en capas, que le permite al desarrollador decidir que componentes utilizar, además de constituir un framework coherente para el desarrollo de aplicaciones J2EE.

Cada módulo o componente presente en el framework Spring puede entenderse como un ente aparte, o puede usarse combinándolo con uno o varios de los restantes. Spring provee soporte para validación de datos, integración con otros marcos de trabajo, manipulación de base de datos, integración con herramientas de mapeo relacional (*ORM* por sus siglas en inglés) como Hibernate, patrones de acceso a datos, facilita la implementación del modelo vista controlador y el desarrollo de portlets. La siguiente imagen muestra el patrón MVC con Spring Framework.

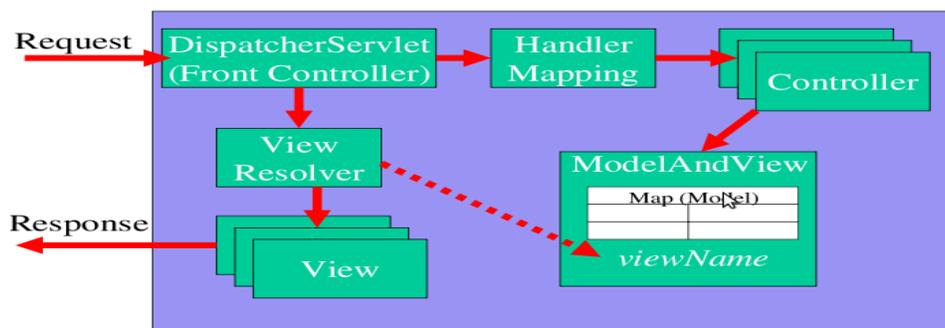


Figura 3. El patrón MVC con Spring Framework

El proceso de atender las solicitudes se realiza cuando: una petición llega al controlador frontal, este mediante un proceso de búsqueda encuentra el controlador destinado a atender dicha petición, invoca al controlador y este devuelve una vista con un modelo de datos, nuevamente el controlador frontal busca el nombre de la vista (la página JSP) que va a ser mostrada y finalmente se envía al usuario la página con la información a mostrar.

1.7. Componente de la plataforma J2EE

Dentro de J2EE existen componentes que permiten combinar varias potencialidades de la plataforma, incluyendo el desarrollo de interfaces de usuario, ejemplo de ello lo constituyen los portlets.

Un portlet es un componente de interfaz web reutilizable y modulares gestionadas, visualizadas en un portal web, producen fragmento de lenguaje de marcado y puede ser mostrado en una página de un portal (26). Una especificación de petición JSR es la encargada de regir el comportamiento de ciertos componentes web dentro de la plataforma J2EE, específicamente las números 168 y 286 están encargadas de proporcionar el estándar para el desarrollo de portlets.

Un portlet posee un ciclo de vida que es gestionado por el portal o contenedor web, tiene 3 estados (Minimizado, Normal, Maximizado) y tres modos de visualizarse (Vista, Edición y Ayuda). El contenido generado por un portlet es llamado fragmento que pueden ser incluidos como componentes de interfaz de usuario en portales web denominados contenedores de portlets, entre ellos tenemos: JBoss Portal, Liferay Portal y Stringbeans Portal.

1.7.1. Contenedor de Portlets

Liferay Portal es un portal de gestión de contenidos de código abierto implementado en Java y su contenido está basado en portlets. Ofrece una arquitectura de temas, que permite llevar a cabo cambios en la apariencia del portal sin cambiar el código fuente de Liferay. (27)

- ❖ Liferay por ser un portal basado en Java se ejecuta sobre múltiples sistemas operativos: BSD (FreeBSD, NetBSD, OpenBSD), Linux (Fedora, Novell, Ubuntu, Debian), Solaris, Mac OS X y Windows.
- ❖ Liferay emplea Hibernate como herramienta ORM para la capa de persistencia, lo que facilita que soporte cualquier base de datos como MySQL, Oracle, PostgreSQL.

- ❖ Liferay se puede desplegar en muchos servidores como: JBoss, SunGlassFish y Apache Tomcat.
- ❖ Liferay soporta tecnologías del tipo JSR-168 y 286, JSR-220 (Hibernate), AJAX, Spring-MVC, Struts.

Liferay es un contenedor de portlet, los cuales son visualizados y gestionados por Liferay, de esta forma, se podrían implementar portlets específicos para cada servicio bioinformático, de forma independiente y luego desplegarse en el Portal Web basado en Liferay.

1.8. Plataforma de Servicios Bioinformáticos

La Plataforma de Servicios Bioinformáticos de la Universidad de las Ciencias Informáticas es un sistema que integra un conjunto de herramientas de uso común en la Bioinformática, así como los productos desarrollados por el departamento de Bioinformática, con el propósito de brindar servicios que puedan ser consumidos por aplicaciones o usuarios a través de un Portal Web y proporcionar acceso a los recursos computacionales existentes en la Institución. Esta plataforma tiene una arquitectura orientada a servicios, implementados sobre la Plataforma de desarrollo Java 2 Enterprise Edition, de cara al usuario posee interfaces web en forma de Portlets, los cuales se encargan de gestionar los distintos servicios que brinda la Plataforma de Servicios Bioinformáticos, y son controlados por el contenedor de portlets: Liferay Portal. La plataforma utiliza los recursos computacionales existentes en la Universidad, para aumentar las potencialidades de cómputo que la misma demanda.

1.9. Sistemas gestores de bases de datos

Un sistema gestor de bases de datos o RDBMS (siglas del inglés *Relational Data Base Management System*), se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos. Ayuda a realizar las siguientes acciones: definición de los datos, mantenimiento de la integridad de los datos dentro de la base de datos, control de la seguridad y privacidad de los datos, y manipulación de los datos. Existen RDBMS muy populares, entre los que se pueden mencionar: Oracle, MySQL y PostgreSQL.

1.9.1. PostgreSQL 9.1

PostgreSQL 9.1 es un sistema gestor de base de datos objeto-relacional, bajo licencia BSD. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio

público. Es el sistema de gestión de bases de datos de código abierto más avanzado del mundo y en sus últimas versiones posee muchas características que solo se podían ver en productos comerciales de alto calibre (28). PostgreSQL 9.1 utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Posee una documentación muy bien organizada, pública y libre, además se ejecuta en casi todos los principales sistemas operativos: Linux, Unix, Mac OS, Beos y Windows. Tiene el respaldo de comunidades muy activas como las de PHP, C, C++, Perl y Python.

1.10. Entorno de desarrollo IDE

Un IDE es una aplicación compuesta por un conjunto de herramientas útiles para un desarrollador. Puede ser exclusivo para un lenguaje de programación o utilizarse para varios. Está compuesta por un editor de código (con facilidades como resaltado de sintaxis, completamiento de código y navegación entre clases), un compilador y herramientas de automatización de la compilación, un depurador y en algunos casos un constructor de interfaz gráfica.

1.10.1. Eclipse

Eclipse Foundation es una comunidad de código abierto con proyectos enfocados en proveer una plataforma de desarrollo de marcos de trabajo y herramientas para desarrollar y gestionar los ciclos de vida en el desarrollo de *software*, la plataforma de desarrollo es denominada Proyecto Eclipse (29). Eclipse no sólo trae editores de código para Java, se puede utilizar para desarrollar en otros lenguajes como PHP, C y C++. Posibilita la agregación de componentes ya que su arquitectura está basada en complementos.

1.11. Metodología de desarrollo

Es una metodología de desarrollo de *software* ágil, de código abierto, que preserva las características fundamentales del Proceso Unificado de Desarrollo (RUP por sus siglas en inglés) (30). Su objetivo es asegurar la producción de *software* de calidad dentro de los plazos propuestos. OpenUP es iterativo e incremental, centrado en la arquitectura y guiado por casos uso, está diseñado para equipos de desarrollo pequeños y se organiza en cuatro áreas fundamentales de contenido: comunicación y colaboración, intención, solución y administración. La mayoría de los elementos de OpenUP están declarados para fomentar el intercambio de información entre los equipos de desarrollo y mantener un entendimiento compartido del proyecto, sus objetivos, alcance y avances.

1.12. Lenguaje de modelado

Lenguaje de Modelado Unificado (*UML* por sus siglas en inglés), es el lenguaje de modelado de sistemas de *software* más conocido y utilizado en la actualidad; está respaldado por el *OMG* (*Object Management Group*). Permite la especificación, visualización, construcción y documentación de elementos de la Ingeniería del *Software*. Se aplica para dar soporte a una determinada metodología de desarrollo de *software* (como *RUP*), no especifica cual usar. Una de las metas principales de *UML* es proporcionar herramientas de interoperabilidad para el modelado visual de objetos en el proceso de construcción del *software*. (31)

1.13. Herramienta CASE

Visual Paradigm para *UML* soporta el ciclo de vida completo del desarrollo de *software*, análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El *software* ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite diseñar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas, además permite generar reportes y documentación. (31)

Conclusiones del Capítulo

En este capítulo se abordaron los métodos relacionados a la resolución de problemas de Química Computacional. Se describieron los principales *software* que implementan estos métodos, haciendo énfasis en los más empleados por los investigadores. Se determinó el uso de *Mopac2012* y *Gamess* como herramientas para realizar los cálculos químicos-teóricos y *Gabedit* como aplicación que permite modelar los datos de entrada para dichas herramientas. Se describieron las tecnologías, herramientas y metodologías a utilizar en la implementación de la solución siguiendo las directrices propuestas por la Plataforma de Servicios Bioinformáticos.

Capítulo 2

Características del sistema

En este capítulo se describen los requisitos funcionales y requisitos no funcionales de la aplicación a desarrollar. Se realiza una descripción de los actores que interactúan con el sistema, así como de los casos de usos identificados. Se presenta el diagrama de clases del modelo de dominio, el diagrama de casos de uso del sistema.

2.1. Descripción de sistema

El sistema que se propone desarrollar es un Servicio de Química Computacional que servirá a los investigadores de universidades y centros del país que realizan cálculos englobados en el área de la Química Computacional. Este servicio será consumido por aplicaciones o usuarios a través de un Portal Web. De cara al usuario se tendrá una interfaz web implementada en forma de portlets. El mismo puede ser reutilizado por otras aplicaciones indistintamente de la tecnología en que estas han sido elaboradas.

2.2. Modelo del dominio

El modelo de dominio (ver figura 4) es una representación visual estática del entorno real del problema, incluye el vocabulario del dominio significativo desde el punto de vista de la arquitectura para de esta forma ayudar al entendimiento de ella. (32)

Para un mejor entendimiento se describen cada una de las clases existentes en el diagrama de clases del dominio.

- ❖ **Especialista:** Usuario que interactúa con el Portal Web.
- ❖ **Portal de Servicios Bioinformáticos:** Portal Web que contiene un conjunto de aplicaciones en forma de portlets.
- ❖ **Portlets de Servicios:** Conjunto de aplicaciones web contenidas en el Portal Web.
- ❖ **Servicios Web:** Funcionalidades que serán utilizadas por las aplicaciones publicadas en el Portal Web.
- ❖ **Games:** Programa que permite realizar cálculos químicos teóricos.

- ❖ **Mopac:** Programa que permite realizar cálculos químicos teóricos.
- ❖ **T_Arenal:** Plataforma de tareas distribuidas integrada a la Plataforma de Servicios Bioinformáticos.

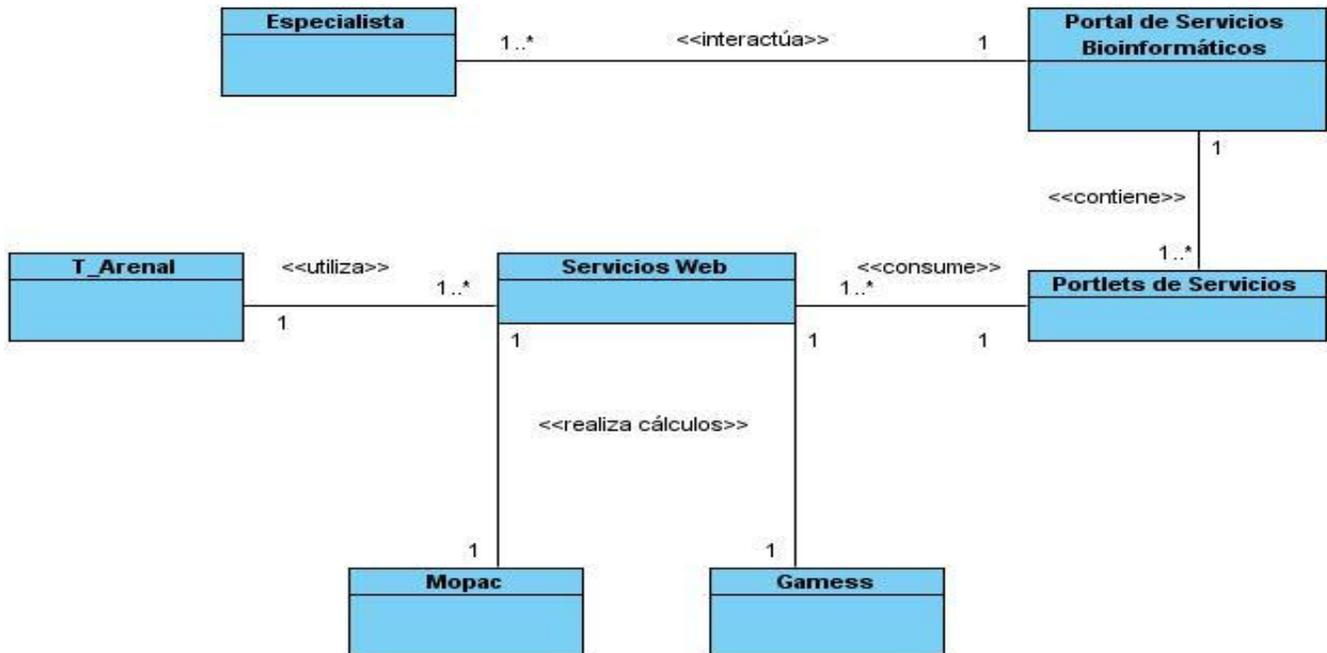


Figura 4. Modelo de Dominio

2.3. Especificación de requisitos

La especificación de requisitos es una descripción completa del comportamiento del sistema que se va a desarrollar. Para la especificación de los requisitos el cliente debe tener en cuenta además de las funcionalidades o requisitos funcionales, los requisitos no funcionales del *software*.

2.3.1. Requisitos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. Se mantienen invariables sin importar con qué propiedades o cualidades se relacionen. Definen las funciones que el sistema será capaz de realizar.

RF1: Realizar cálculos químicos teóricos con Mopac.

RF2: Realizar cálculos químicos teóricos con Gamess.

RF3: Gestionar ejecución

RF3.1: Crear ejecución.

RF3.2: Modificar ejecución.

RF3.3: Buscar ejecución.

RF3.4: Eliminar ejecución.

RF3.5: Listar ejecución.

RF4: Administrar resultados

RF4.1: Listar resultados.

RF4.2: Buscar resultados.

RF4.3: Eliminar resultado.

RF5: Descargar resultados.

RF6: Descargar Gabedit.

2.3.2. Requisitos no funcionales

Son las restricciones que afectan a los servicios o funciones del sistema, tales como restricciones de tiempo, definición de estándares y otras condiciones necesarias para que el sistema pueda funcionar.

Son propiedades o cualidades que el producto debe tener, estas son las que hacen al producto atractivo, usable, rápido y confiable.

❖ **Apariencia o Interfaz externa**

La aplicación deberá tener una interfaz externa sencilla, amigable y fácil de entender para el usuario, de esta forma se evita que este se pierda dentro de la aplicación. Además su funcionamiento debe ser de fácil comprensión para todo tipo de usuario.

❖ **Usabilidad**

El sistema podrá ser usado por aquellos usuarios que posean conocimientos básicos en el campo de la Bioinformática y en el uso de aplicaciones web.

❖ Seguridad

La aplicación contará con protección contra acciones no autorizadas y que puedan afectar la integridad de los datos por lo que cada especialista solo tendrá acceso a su propia información.

❖ Confidencialidad

Los especialistas tienen que estar autenticados en el Portal de Servicios Bioinformáticos de la UCI para acceder a todas las funcionalidades del sistema.

❖ Integridad

La plataforma garantiza la integridad de los datos a los usuarios autorizados.

❖ Disponibilidad

El sistema debe estar disponible para su utilización las 24 horas del día.

❖ Software

Al ser una aplicación cliente servidor, los requisitos de *software* de la herramienta son diferentes para cada una de las partes. En el lado del cliente debe existir un navegador web que soporte Java Script. Por el lado del servidor se debe instalar en el servidor web el Apache Tomcat 6.0.26 con la aplicación Liferay Portal 6.0.5, JDK6, PostgreSQL 9.1. Y en el servidor de servicios web debe estar instalado Mopac2012 y Gamess para realizar cálculos de Química Computacional.

❖ Hardware

En el caso de las PC clientes se debe contar con un microprocesador Intel Pentium II o superior, con un mínimo de 128MB de memoria RAM y tarjeta de red para la conexión de red con el Portal de Servicios Bioinformáticos de la UCI. En el servidor web se recomienda contar con un microprocesador Intel Pentium IV o superior, 2GB de memoria RAM, un disco duro con capacidad disponible de almacenamiento de 160GB y una tarjeta de red.

❖ Soporte

El sistema se agregara al Portal de Servicios Bioinformáticos para realizar las pruebas pilotos y de despliegue, luego quedara instalada en la plataforma para su uso.

❖ Rendimiento

El sistema está concebido para un ambiente cliente - servidor por lo que la rapidez del servicio dependerá del volumen de la información que necesite procesar y de la cantidad de corridas en la cola de ejecución, es por ello que se propone su integración a la Plataforma de Cálculos Distribuidos: T-Arenal.

2.4. Diagrama de Casos de Uso del Sistema

Un Diagrama de Casos de Uso muestra la relación entre los actores y los casos de uso del sistema, de esta forma se representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa. (33). La figura 5 muestra el diagrama de Casos de Uso del sistema desarrollado.

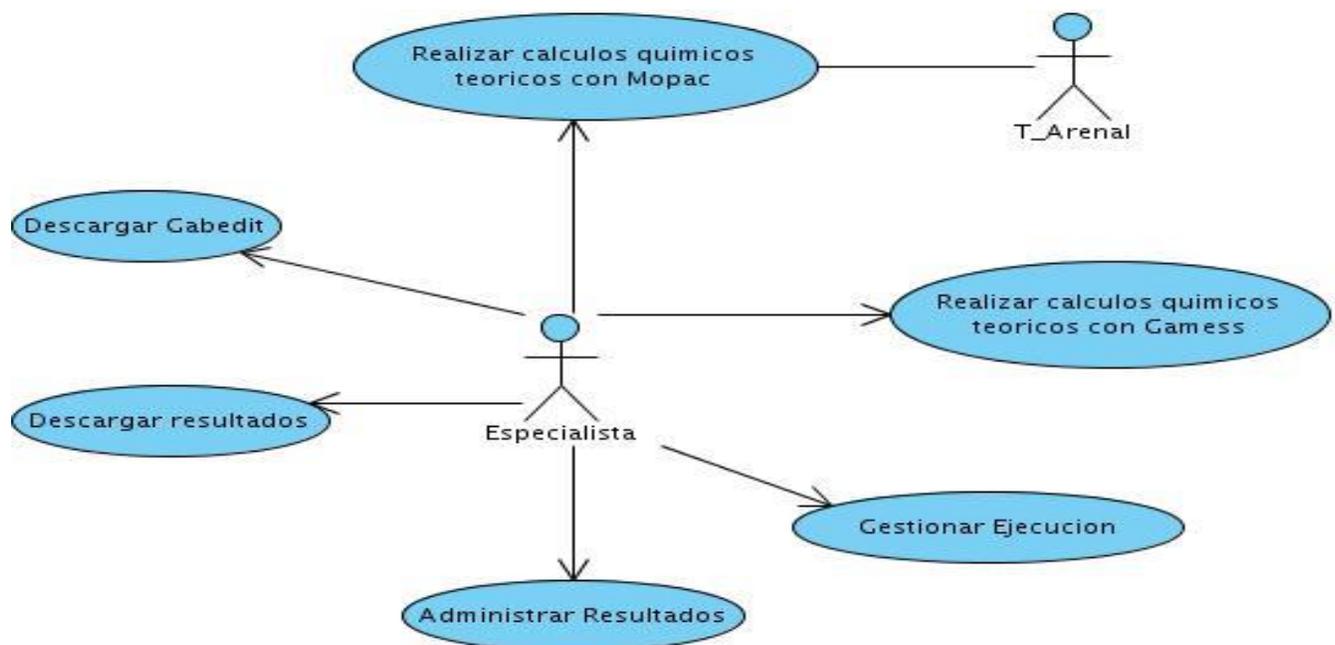


Figura 5. Diagrama de Casos de Uso del sistema

2.5. Definición de los Casos de Uso del Sistema

Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema cuando el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad de funcionalidad coherente, son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del actor. (32)

2.5.1. Actores del sistema

Los actores pueden ser sistemas, hardware externo o personas que interactúan con el sistema, ya sea para inicializar una funcionalidad o brindarle información al mismo. Son los que se benefician de las funcionalidades dentro de los casos de uso.

Tabla 1. Actores del Sistema

Actor	Descripción
Especialista	Persona que tiene los privilegios suficientes y accede a la aplicación para realizar cálculos químicos teóricos con Mopac o Gamess.
T_Arenal	Plataforma de tareas distribuidas integrada a la Plataforma de Servicios Bioinformáticos.

2.5.2. Descripción de los Casos de Usos del sistema

La descripción textual de los casos de uso describe paso a paso la interacción entre el actor y el sistema, es cómo responde el sistema ante las peticiones de los diferentes actores. A continuación la descripción textual de los casos de uso *"Realizar cálculos químicos teóricos con Mopac"* y *"Realizar cálculos químicos teóricos con Gamess"*.

CU Realizar cálculos químicos teóricos con Mopac.

Tabla 2. Descripción del CU "Realizar cálculos químicos teóricos con Mopac"

Objetivo	Poder realizar cálculos químicos teóricos con Mopac.
Actores	Especialista(inicia)
Resumen	El caso de uso inicia cuando el especialista selecciona una ejecución para realizar el cálculo.
Complejidad	Alta
Prioridad	Crítico
Precondiciones	Tener creada una ejecución.
Referencias	RF1
Flujo de eventos	
Flujo básico <Realizar cálculos químicos teóricos con Mopac>	
Actor	Sistema
1.1 Accede a la "Interfaz Ejecuciones" y selecciona una ejecución a ejecutar.	1.2 Llama al servicio web Realizar cálculos con Mopac. 1.3 El servicio se conecta a la Plataforma de Cálculos Distribuidos y envía la ejecución.

	<p>1.4 La Plataforma de Cálculos Distribuidos realiza los cálculos y retorna el resultado al servicio.</p> <p>1.5 Obtiene el resultado y actualiza la ejecución en la base de datos.</p> <p>1.6 Muestra en una página con el resultado de la ejecución y la opción “Descargar resultado “.</p>
--	--

CU Realizar cálculos químicos teóricos con Gamess.

Tabla 3. Descripción del CU "Realizar cálculos químicos teóricos con Gamess"

Objetivo	Poder realizar cálculos químicos teóricos con Gamess.
Actores	Especialista(inicia)
Resumen	El caso de uso inicia cuando el especialista selecciona una ejecución para realizar el cálculo.
Complejidad	Alta
Prioridad	Crítico
Precondiciones	Tener creada una ejecución.
Referencias	RF2
Flujo de eventos	
Flujo básico <Realizar cálculos químicos teóricos con Gamess>	
Actor	Sistema
1.1 Accede a la “Interfaz Ejecuciones” y selecciona una ejecución a ejecutar.	<p>1.2 Llama al servicio web Realizar cálculos con Gamess.</p> <p>1.3 El servicio ejecuta el Gamess en el servidor de aplicaciones.</p> <p>1.3 Obtiene el resultado y actualiza la ejecución en la base de datos.</p> <p>1.4 Muestra en una página con el resultado de la ejecución y la opción “Descargar resultado “.</p>

Conclusiones del Capítulo

En este capítulo se definieron los artefactos correspondientes a los requisitos de la metodología de desarrollo OpenUP. Se definió el modelo conceptual del dominio y se describieron las clases del dominio. Se definieron 12 requisitos funcionales y 10 requisitos no funcionales. Los RF identificados se agruparon en el modelo de casos de uso del sistema, destacando los CU: “Realizar cálculos químicos teóricos con Mopac” y “Realizar cálculos químicos teóricos con Gamess” por tener gran impacto en la arquitectura.

CAPÍTULO 3

Diseño del sistema

En este capítulo se traducen los requisitos a una especificación que describe cómo implementar el sistema, dando paso al diseño de la aplicación generando los artefactos necesarios para cada fase. Se explican los principales patrones arquitectónicos y de diseño utilizados, además se realizan los diagramas de clases y de interacción para uno de los casos de usos más relevantes definidos en el capítulo anterior. También se muestra el modelo datos y la vista de despliegue.

3.1. Arquitectura del sistema

La arquitectura de *software* consiste en la estructura o sistema de estructuras, que comprenden los elementos de *software*, las propiedades externas visibles de esos elementos y la relación entre ellos. (32)

La arquitectura de *software* define la estructura de un sistema. Esta estructura se constituye de componentes, las relaciones entre ellos, el contexto en el que se implantarán y los principios que orientan su diseño y evolución. Para que un sistema sea extensible y reutilizable, su arquitectura tiene que estar diseñada en base a ello. Para tal propósito y buen uso de la misma se utilizan los estilos y patrones arquitectónicos.

3.1.1. Estilos y patrones arquitectónicos

Los estilos de arquitectura guían a la organización del sistema de *software*, expresan a la arquitectura de *software* en el nivel de abstracción más elevado, en el sentido más formal y teórico (32).

Los patrones expresan un paradigma fundamental para estructurar u organizar un sistema de *software*. Un patrón arquitectónico es la expresión de un esquema estructural de organización para sistemas de *software*, que proporciona un conjunto de subsistemas predefinidos, especifica sus responsabilidades e incluye normas y directrices para las relaciones entre ellos (34).

Los estilos expresan a la arquitectura de *software* en el nivel de abstracción más elevado, en el sentido más formal y teórico, mientras que los patrones se ocupan de cuestiones que están más cerca del diseño, la práctica, la implementación, el proceso, el refinamiento y el código.

Patrón Modelo Vista Controlador

El Patrón Modelo Vista Controlador (Model View Controller por sus siglas inglés) divide una aplicación en tres módulos claramente identificables y con funcionalidades bien definidas, el modelo, que especifica la información con la cual el sistema opera, las vistas, encargadas de manejar la presentación visual de los datos representados por el modelo y el controlador, que procesa las peticiones que vienen desde la capa de presentación y donde está encapsulada la lógica del negocio (figura 6).

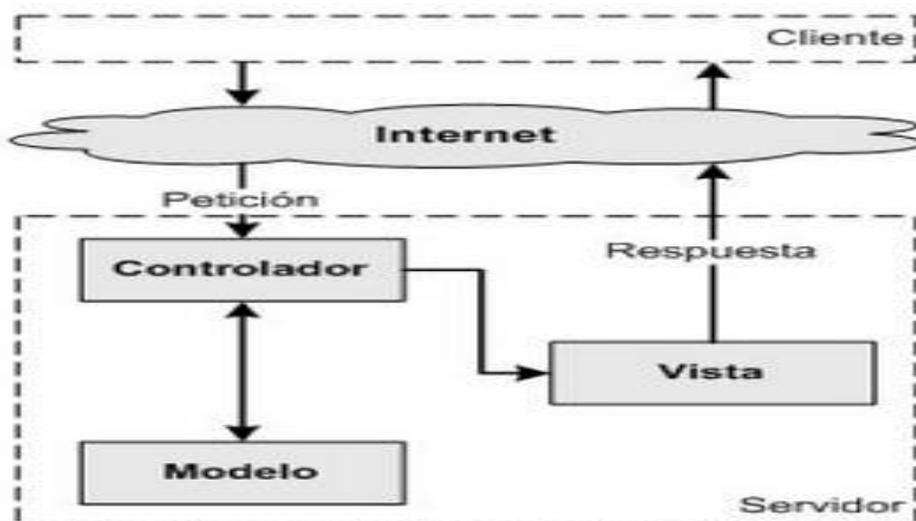


Figura 6. Patrón Modelo Vista Controlador

El servicio de Química Computacional presenta los patrones arquitectónicos Cliente-Servidor, Arquitectura Orientada a Servicios y Modelo Vista Controlador. El primero se manifiesta en el paradigma petición-respuesta, dado que es una aplicación web a la que se accede por el protocolo HTTP. El segundo representada en el servicio web implementado que está disponible a través del protocolo SOAP e intercambian información usando XML. El tercero se aplica en el desarrollo del portlet, cuando una petición llega al controlador frontal (archivo XML), este mediante un proceso de búsqueda encuentra el controlador destinado a atender dicha petición, invoca al controlador, finalmente se envía al usuario la página con la información a mostrar.

3.1.2. Patrones de diseño

Un patrón describe un problema que ocurre varias veces y describe también el núcleo de la solución al problema, de forma que puede utilizarse en ilimitadas ocasiones sin tener que hacer dos veces lo mismo.

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. (34)

En otras palabras, un patrón de diseño no es más que una solución estándar para un problema común de programación. Una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios. Es decir, los patrones de diseño ayudan a un diseñador a conseguir un diseño correcto rápidamente. Entre ellos podemos mencionar los patrones de diseño para la Asignación de Responsabilidades (General Responsibility Assignment Software Patterns GRASP, por sus siglas en inglés) y los patrones *GoF*⁶.

Patrones GRASP

El nombre se eligió para sugerir la importancia de aprender estos principios para diseñar con éxito el *software* orientado a objetos (35). Los patrones *GRASP* son utilizados para describir los principios fundamentales del diseño y la asignación de responsabilidades.

❖ Experto en Información

Un Modelo de diseño podría definir cientos o miles de clases *software*, y una aplicación podría requerir que se realicen cientos o miles de responsabilidades. Durante el diseño de objetos, cuando se definen las interacciones entre los objetos, se toman decisiones sobre la asignación de responsabilidades a las clases. Haciendo un resumen de lo anteriormente planteado este patrón consiste en asignar la responsabilidad a la clase que tiene la información necesaria para realizarla.

❖ Creador

La creación de instancias es una de las actividades más comunes en un sistema orientado a objetos. En consecuencia, es útil contar con un principio general para la asignación de las responsabilidades de creación. Si se asignan bien, el diseño puede soportar un bajo acoplamiento, mayor claridad y reutilización. El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, una tarea muy común. La intención básica del patrón es encontrar un creador que necesite conectarse al objeto creado en alguna situación. Eligiéndolo como el creador se favorece el bajo acoplamiento.

⁶ Conocidos como patrones de la pandilla de los cuatro (GoF, siglas en inglés de Gang of Four).

❖ Alta Cohesión

La cohesión es la medida de la fuerza que une a las responsabilidades de una clase. Una clase con baja cohesión hace muchas cosas no relacionadas, o hace demasiado trabajo. Tales clases no son convenientes ya que son difíciles de mantener, de reutilizar y de entender. Una clase con alta cohesión mejora la claridad y la facilidad de su uso, su mantenimiento se simplifica y es fácil de reutilizar.

❖ Bajo Acoplamiento

El acoplamiento es una medida de la fuerza con que un elemento está conectado a otro, un elemento con bajo acoplamiento no depende demasiado de otros elementos. Una clase con alto acoplamiento confía en muchas otras clases, tales clases podrían no ser deseables ya que son muy complicadas de entender, son difíciles de reutilizar y los cambios realizados en las clases relacionadas fuerzan cambios locales.

Patrones *GoF*

Dentro de los patrones de diseño existen los patrones *GoF* los cuales juegan un papel muy importante dentro de los patrones de diseño de forma general y son utilizados en innumerables ocasiones cuando se desarrollan aplicaciones informáticas (35). Entre los patrones *GoF* más utilizados se pueden citar el de Agente Remoto y el Comando.

❖ Agente Remoto

El patrón Agente Remoto define como el sistema debe comunicarse con un componente situado en otro espacio o contexto a través de una clase de *software* local que represente al componente externo y asignarle la responsabilidad de contactar al componente real.

❖ Comando

Otro de los patrones *GoF* es el Comando, este patrón soluciona el problema de cuando un objeto o sistema recibe varias peticiones o comandos, para ello cada comando define una clase que lo represente y le asigna la responsabilidad de ejecutarse el mismo, de esta forma reduce la responsabilidad del receptor en el manejo de los comandos, aumenta la facilidad con que pueden agregarse otros comandos y ofrece las bases para registrar los comandos, formar colas de espera con ellos y cancelarlos.

Después de vistos los patrones *GRASP* y *GoF* se verán otros patrones utilizados en la aplicación.

DAO

El patrón *DAO* (por sus siglas en inglés *Data Access Object*) es un patrón de diseño que centraliza todo el acceso a datos en una capa independiente, aislándolo del resto de la aplicación. Su principal beneficio es que reduce la complejidad de los objetos de negocio al abstraerlos de la implementación real de la comunicación con la fuente de datos y de esta forma permitir una migración más fácil de fuente de datos.

3.2. Diagrama de clases del diseño

Los diagramas de clases del diseño se utilizan para modelar la vista de diseño estática de un sistema, estos describen gráficamente las especificaciones de las clases además de visualizar las relaciones entre estas.

A continuación se muestra el diagrama de clases del diseño para los casos de usos: “Realizar cálculos químicos teóricos con Mopac” y “Realizar cálculos químicos teóricos con Gamess”. Donde se identifican cuatro paquetes funcionales: Vista, Controlador, Objeto de Acceso a Datos, los que corresponden a las capas del patrón de diseño MVC, y por último el Cliente de Servicios Web.

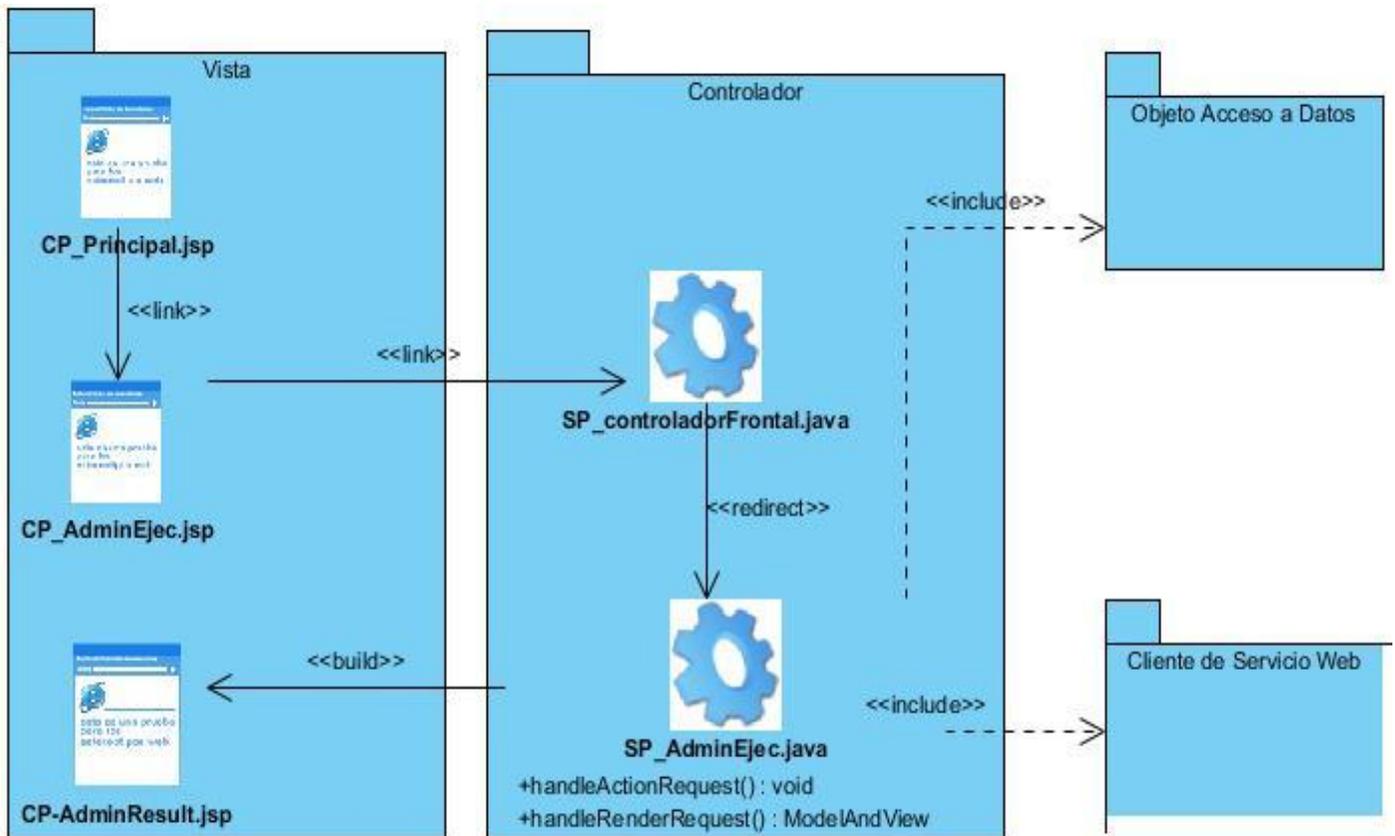


Figura 7. Diagrama de clases del diseño Realizar cálculos químicos teóricos con Mopac

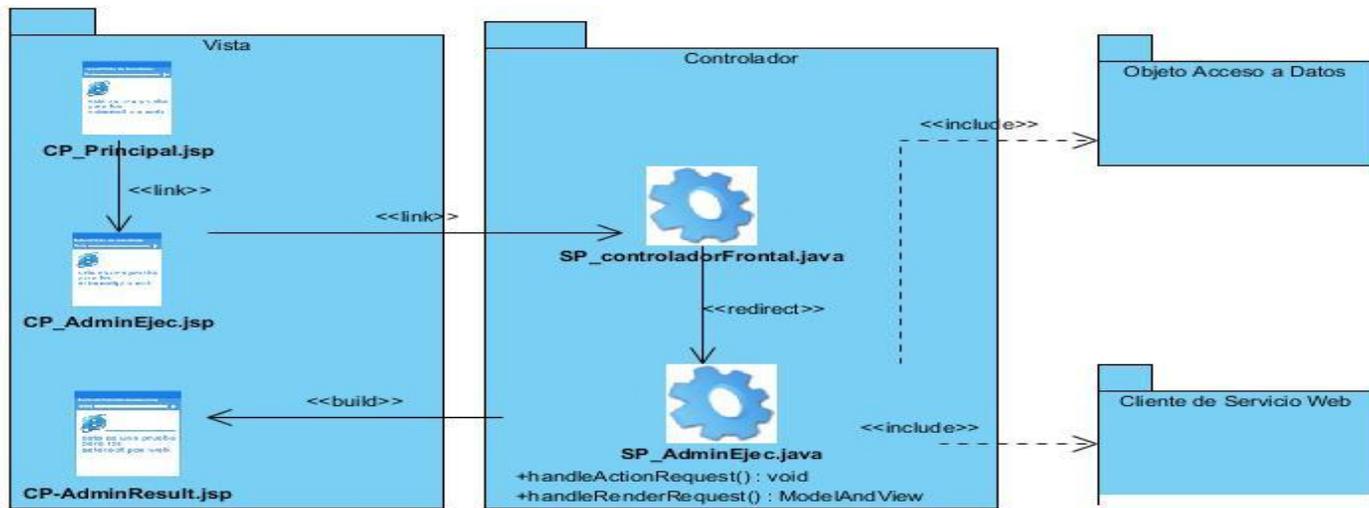


Figura 8. Diagrama de clases del diseño Realizar cálculos químicos teóricos con Games

En el diagrama de clases del diseño correspondiente al paquete Objeto de Acceso a Datos (Figura 5) se define la clase `ConexionBD.java` donde se encuentran todos los métodos necesarios para gestionar la base de datos y utiliza las clases `Ejecucion.java`, `Especialista.java`, `Games.java`, `Mopac.java` y `Archivo.java` para acceder a las tablas del modelo. Esta clase (`ConexionBD.java`) implementa el patrón DAO, ya que ella se encarga de realizar la gestión de los datos entre las clases que contienen la lógica de negocio y las entidades. También se evidencia el Bajo Acoplamiento ya que las clases de acceso a los datos tienen bastante independencia de las clases de abstracción de datos. Hay poca dependencia entre esas clases lo que permite una mayor reutilización.

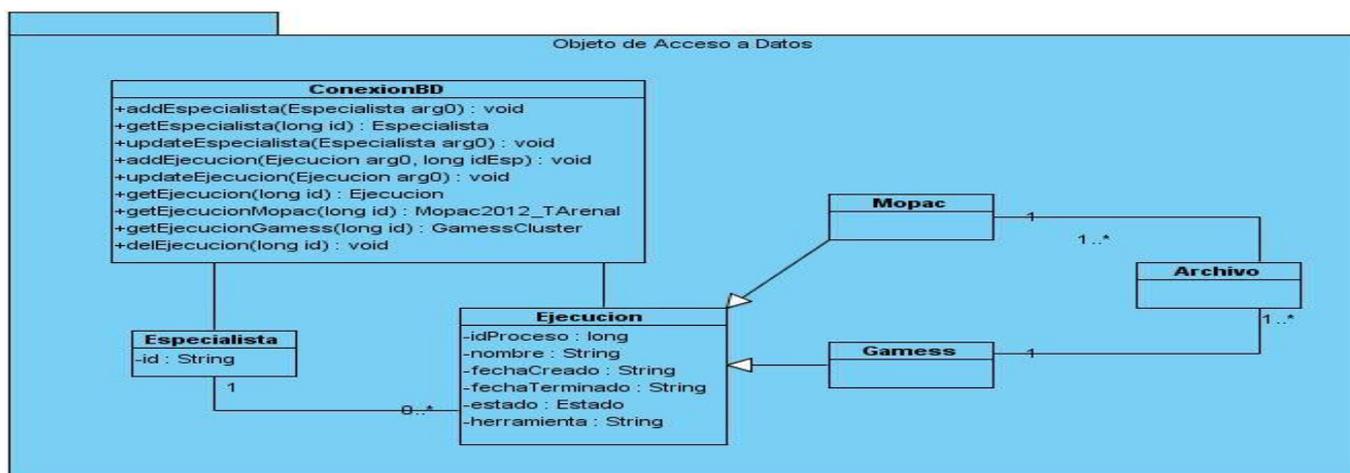


Figura 9. Diagrama de clases del diseño para el paquete Objeto de Acceso a Datos

El diagrama de clase del diseño correspondiente al paquete Cliente de Servicios Web (figura 10 y 11) muestra los componentes encargados de realizar las llamadas a las funciones remotas del sistema. En este paquete la interfaz `BioSoftWSPortType.java` declara las funciones que invocan los servicios mientras que la clase `BioSoftWSPortTypeProxy.java` define las mismas, para ello utiliza otras clases, las que están en correspondencia con los parámetros necesitados por el servicio en cuestión, en este caso, el servicio web “RealizarCálculoConMopac” requiere un objeto de la clase `Mopac.java`. El subsistema Servicios Web Biosoft representa los servicios albergados en el servidor de servicios web. En este diagrama se manifiesta el patrón Agente Remoto de la familia GoF a través de la clase `BioSoftWSPortTypeProxy.java`.

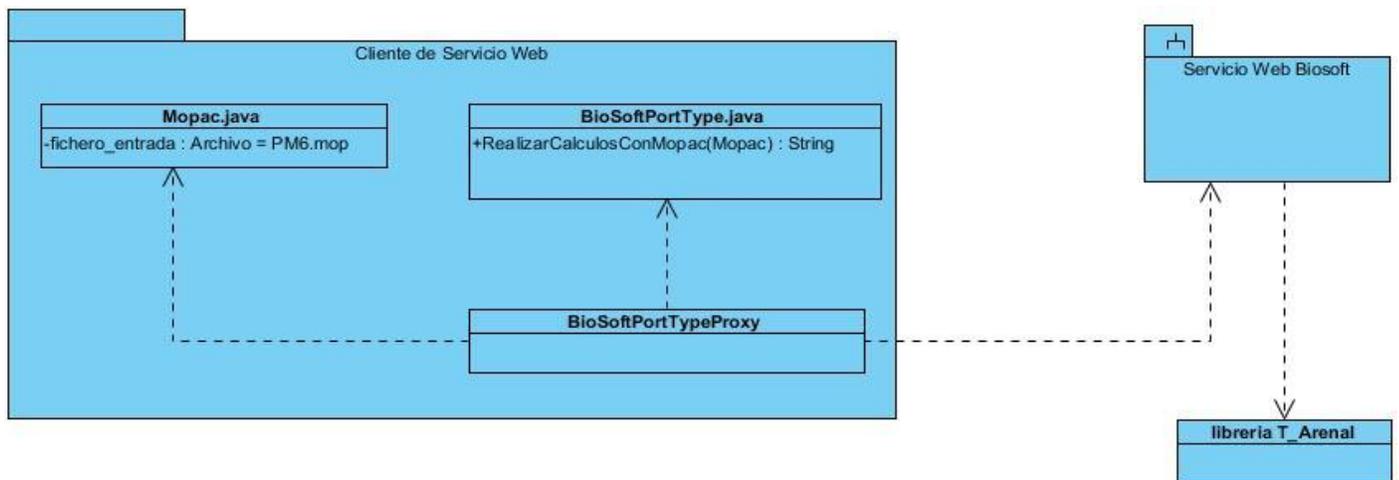


Figura 10. Diagrama de clases del diseño para el paquete Cliente de Servicios Web con Mopac

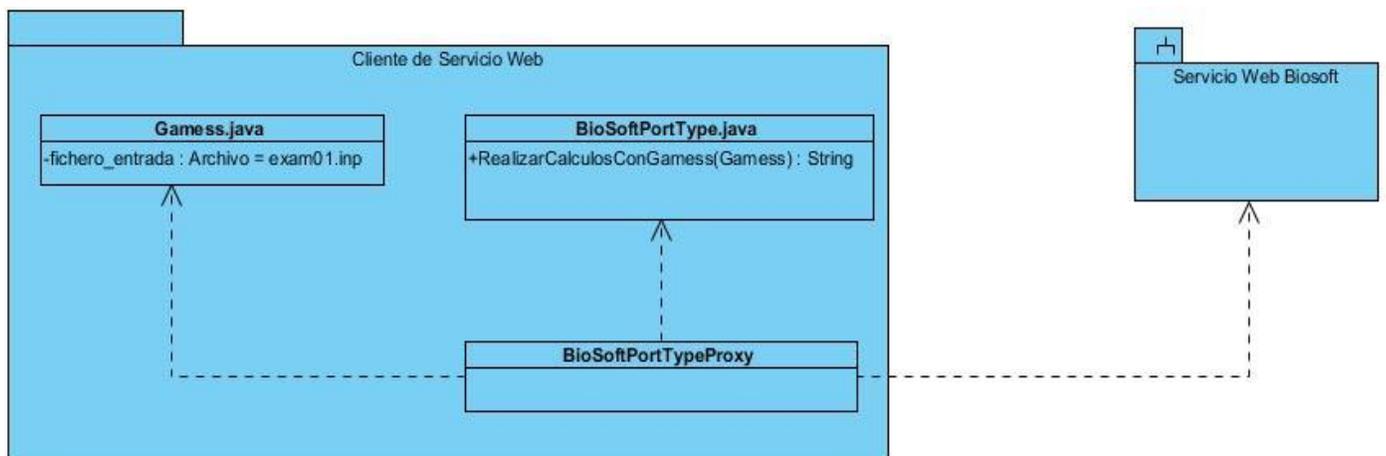


Figura 11. Diagrama de clases del diseño para el paquete Cliente de Servicios Web con Gamess

3.3. Diagramas de interacción

Un diagrama de interacción, representa la forma en cómo un Cliente (Actor) u Objetos (Clases) se comunican entre sí en petición a un evento. Representan el dinamismo que se modela en el diagrama de clases del diseño de manera estática, esto implica recorrer toda la secuencia de llamadas de donde se obtienen las responsabilidades. Hay dos tipos de diagramas de interacción: Diagramas de secuencia y Diagramas de colaboración. (32)

3.3.1. Diagramas de secuencia

Los diagramas de secuencia muestran las interacciones entre objetos ordenadas en secuencia temporal durante un escenario concreto. Proporciona una representación de la realización de casos de uso en términos de clases del diseño. En el diseño es más factible el empleo de los diagramas de secuencia ya que representan con más claridad el flujo de las acciones que debe realizar el sistema (32).

A continuación se muestran los diagramas de secuencia para para los casos de usos: *“Realizar cálculos químicos teóricos con Mopac”* y *“Realizar cálculos químicos teóricos con Gamess”*.

El diagrama de secuencia de la figura 12 y 13 muestra la comunicación de los componentes del *software*. Se define el `controladorFrontal` encargado de gestionar todas las peticiones del usuario y de determinar el controlador que le dará respuesta a dicha petición, en este caso `AdminEjec.java`. De esta manera se aplican patrones como el Controlador de GRASP y el Comando de los GoF, también están los patrones Bajo acoplamiento y Alta cohesión.

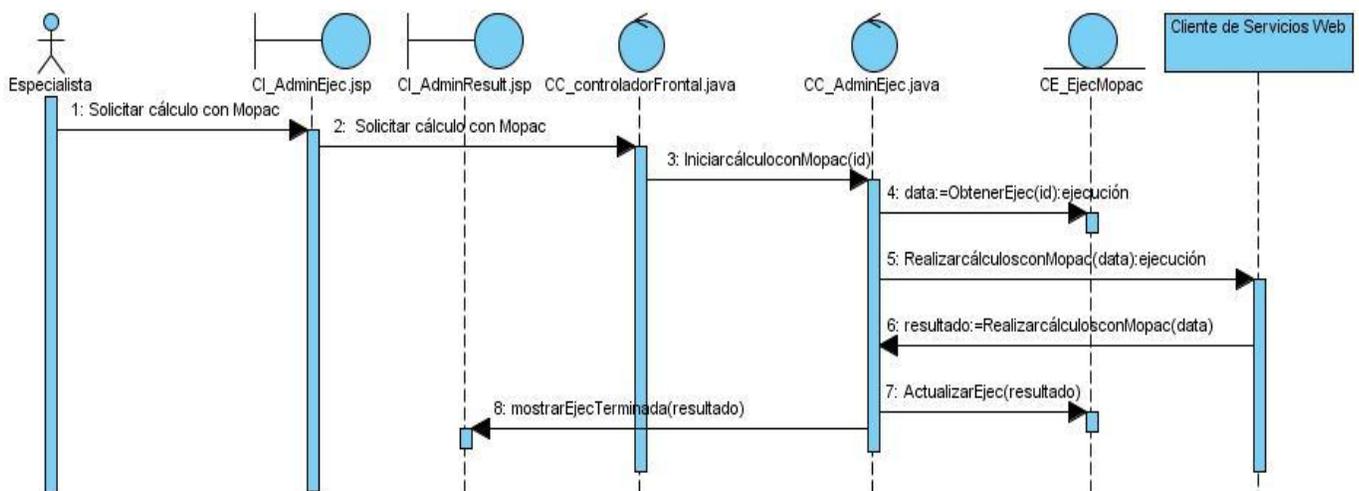


Figura 12. Diagrama de secuencia para Realizar cálculos químicos teóricos con Mopac

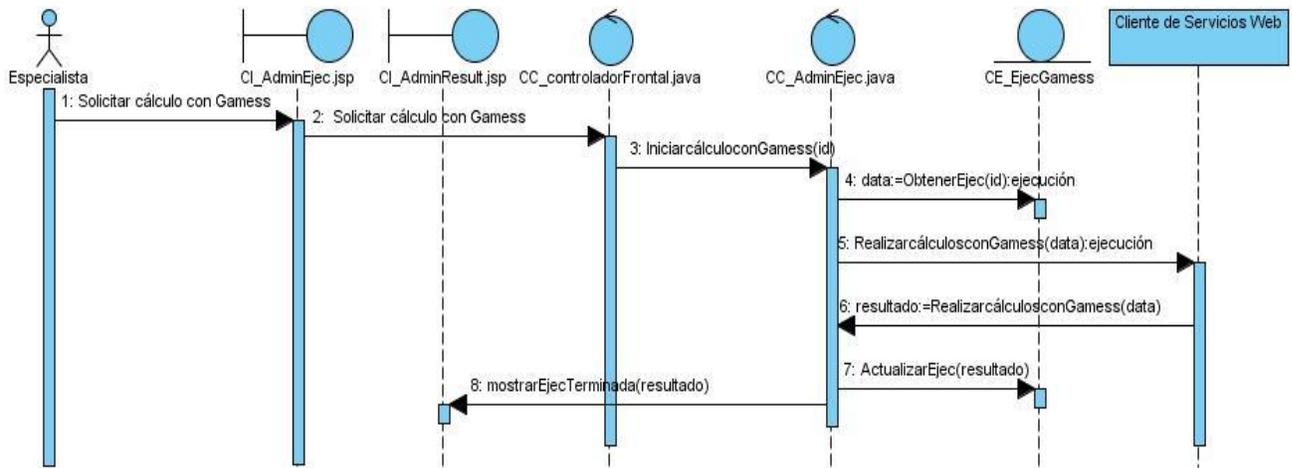


Figura 13. Diagrama de secuencia para Realizar cálculos químicos teóricos con Games

3.4. Modelo de datos

El modelo de datos es una colección de conceptos que sirven para describir la estructura de una base de datos, además de proporcionar los medios necesarios para conseguir la abstracción de los datos. Un modelo de datos es la estructura o representación física de las tablas de la base de datos (33). En la figura 14 se muestra el modelo de datos de la aplicación.



Figura 14. Modelo de Datos

3.4. Vista de despliegue

La Vista de despliegue se utiliza para capturar los elementos de configuración del procesamiento y las conexiones entre estos elementos. También se utiliza para visualizar la distribución de los componentes de *software* en los nodos físicos. (32)

La Vista de despliegue provee un modelo detallado de la forma en la que los componentes se desplegarán a lo largo de la infraestructura del sistema. Detalla las capacidades de red, las especificaciones del servidor, los requisitos de hardware y otra información relacionada con el despliegue del sistema propuesto. Esta vista representa el mapeo de componentes de *software* ejecutables con los nodos de procesamiento.

A continuación en la figura 15 se presenta la Vista de despliegue de la aplicación.

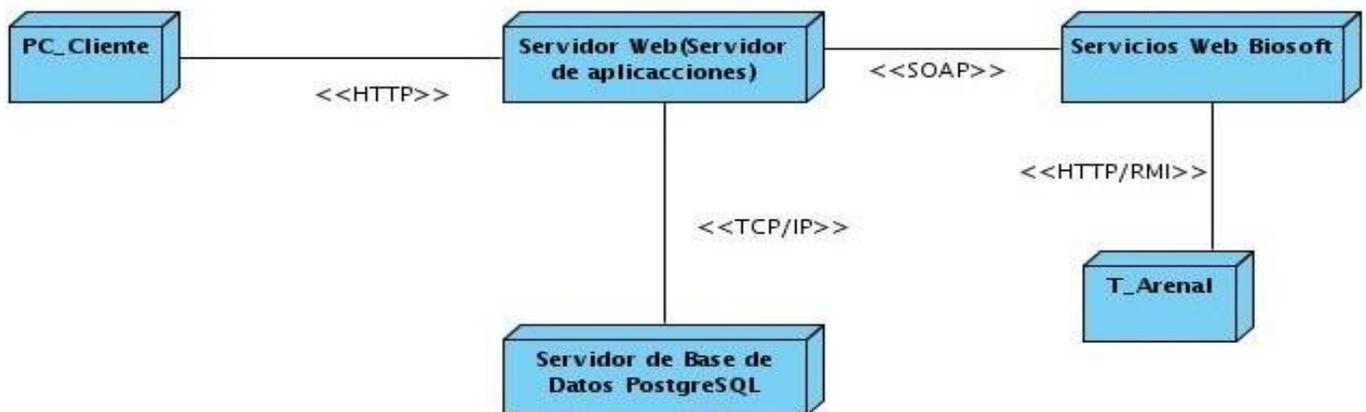


Figura 15. Vista de despliegue

Conclusiones del Capítulo

En el capítulo se analizaron diferentes patrones arquitectónicos y patrones de diseño que posibilitaron definir una mejor arquitectura, entre estos se definieron la arquitectura cliente-servidor y orientada a servicios, así como el MVC, Alta cohesión y Bajo acoplamiento y se analizó cómo se evidenciaban dentro de la aplicación. Se realizó el diagrama de clases del diseño de los Casos de Uso críticos: *“Realizar cálculos químicos teóricos con Mopac”* y *“Realizar cálculos químicos teóricos con Gamess”* y los diagramas de secuencia asociados a ellos. Además, se obtuvo el modelo de datos y la vista de despliegue del sistema.

CAPÍTULO 4

Implementación y Prueba

Durante este capítulo se realiza el flujo de trabajo de implementación, se analizan los artefactos principales como el Modelo de implementación que incluye componentes, subsistemas de implementación y diagramas de componentes. Además se determinan los tipos de pruebas a realizar y los casos de pruebas que serán aplicados al sistema.

4.1. Modelo de implementación

El Modelo de implementación representa la composición física de la implementación en términos de subsistemas de implementación y elementos de implementación. Describe cómo se organizan los componentes de acuerdo a los mecanismos de estructuración disponibles en el entorno de implementación y cómo dependen los componentes unos de otros. Dicho modelo se considera el artefacto más significativo del flujo de trabajo de implementación debido a la importancia que tiene para los desarrolladores. Este modelo está conformado por el diagrama de componentes. (33)

4.1.1. Diagrama de componentes

Dentro del Modelo de implementación se encuentran los diagramas de componentes. Un diagrama de componentes representa cómo un sistema es dividido en componentes y muestra las dependencias entre estos, además describe los elementos físicos y sus realizaciones en el entorno de implementación y en el lenguaje de programación utilizado (32). A continuación se representa el diagrama de componentes de los CU: *Realizar cálculos de químicos teóricos con Mopac* y *Realizar cálculos químicos teóricos con Mopac*.

En la figura 16 y 17 se muestran los diagramas de componentes del sistema.

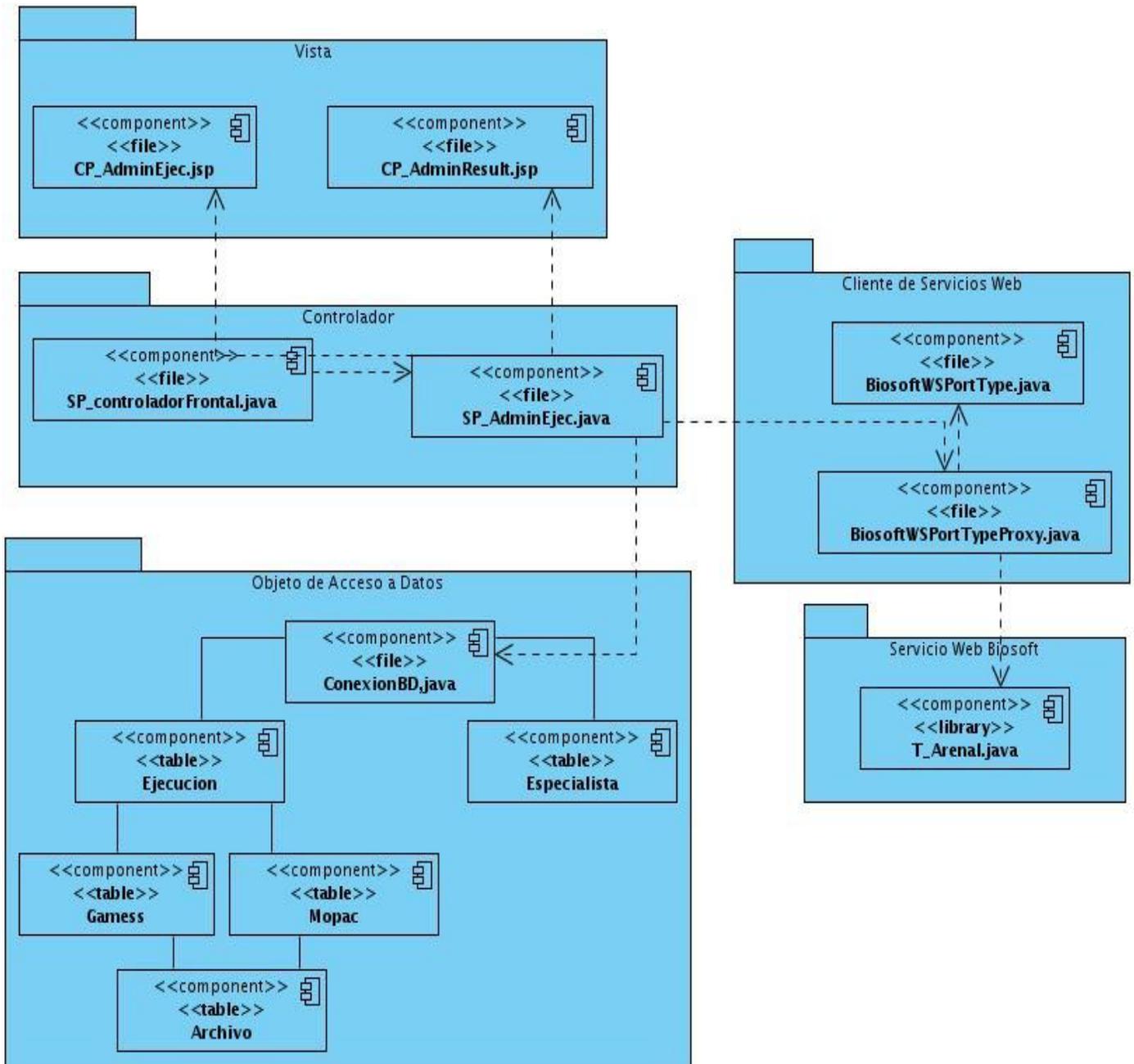


Figura 16. Diagrama de Componentes del CU Realizar cálculos químicos teóricos con Mopac

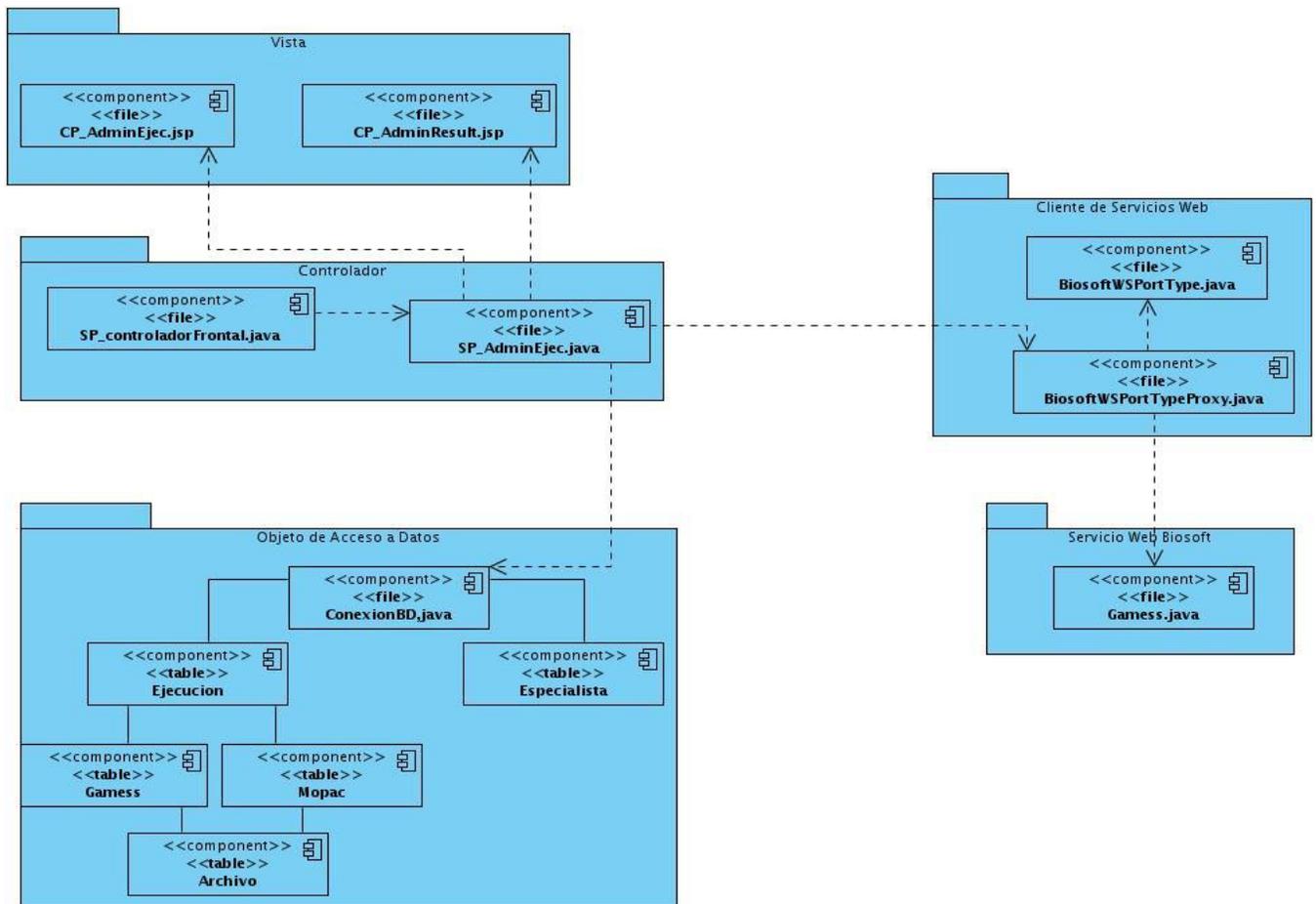


Figura 17. Diagrama de Componentes del CU Realizar cálculos químicos teóricos con Gamess

4.2. Pruebas

Un instrumento adecuado para determinar la calidad de un producto de *software* es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del *software* o al sistema de *software* en su totalidad, con el objetivo de medir el grado en que este cumple con los requisitos. Deben realizarse pruebas a todos los artefactos generados durante la construcción de un producto, lo que incluye especificaciones de requisitos, casos de uso, diagramas de diversos tipos y, por supuesto, el código fuente y el resto de productos que forman parte de la aplicación.

4.2.1. Casos de pruebas de caja negra

La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del *software*. O sea, los casos de prueba pretenden demostrar que las funciones del *software* son operativas, que la

entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene. Las pruebas realizadas a la aplicación son de caja negra al *software*, para realizar estas se diseñaron casos de pruebas a través de la descripción de los casos de usos. En la tabla 5 se muestra un ejemplo del Caso de Prueba realizado al Caso de Uso Gestionar Ejecución de la sección Crear Ejecución.

Tabla 4. Variables para el Caso de Prueba Crear Ejecución

No.	Nombre del campo	Clasificación	Valor Nulo	Descripción
1	Nombre de la ejecución	Campo de texto	No	Cadena de caracteres
2	Archivo de entrada	Archivo	No	Archivo de entrada

Tabla 5. Caso de prueba 1: Crear Ejecución

Escenario	Descripción	V1	V2	Respuesta del sistema	Resultado de las pruebas
EC 1.1 Introducir de forma correcta los datos para Crear una ejecución.	Se llenan los campos correctamente y se Crea la ejecución.	Ejecución1	Exam01.inp	El sistema muestra una interfaz que le permitirá al especialista Crear una ejecución en el sistema.	Se creó la ejecución en el sistema, se mostró la interfaz Ejecuciones y se listo las ejecuciones del especialista.
EC1.2 Introducir de forma errónea el nombre para Crear una ejecución.	Se llenan incorrectamente el nombre de la ejecución.	Ejecución 1 ó -	Exam01.inp	El sistema muestra una interfaz que le permitirá al especialista Crear una ejecución en el sistema.	El sistema muestra un mensaje indicando el error.
EC1.3 Introducir de forma errónea el archivo de entrada para Crear una	Se entra incorrectamente el archivo de entrada de la ejecución.	Ejecución 1	Exam01.lig	El sistema muestra una interfaz que le permitirá al especialista Crear una ejecución en el	El sistema muestra un mensaje indicando el error.

ejecución.				sistema.	
EC1.4 Cancelar la creación de una ejecución en el sistema.	Se cancela la creación de una ejecución.	-	-	El sistema muestra la interfaz Ejecuciones y lista las Ejecuciones del especialista.	Se canceló la creación de la ejecución en el sistema, se mostró la interfaz Ejecuciones y se listó las ejecuciones del especialista.

Tabla 6. Variables para el Caso de Prueba Modificar Ejecución

No.	Nombre del campo	Clasificación	Valor Nulo	Descripción
1	Nombre de la ejecución	Campo de texto	No	Cadena de caracteres
2	Archivo de entrada	Archivo	No	Archivo de entrada

Tabla 7. Caso de prueba 2: Modificar Ejecución

Escenario	Descripción	V1	V2	Respuesta del sistema	Resultado de las pruebas
EC 1.1 Selección correcta de la Ejecución a modificar.	Se selecciona una ejecución a modificar.	-	-	El sistema muestra una interfaz que le permitirá al especialista seleccionar una ejecución en el sistema.	El sistema muestra la interfaz para modificar la ejecución.
EC1.2 Selección incorrecta de la Ejecución a modificar.	Se selecciona varias o ninguna ejecución a modificar.	-	-	El sistema muestra una interfaz que le permitirá al especialista seleccionar una ejecución en el sistema.	El sistema muestra un mensaje indicando el error.

EC1.3 Introducir de forma correcta los datos necesarios para modificar una ejecución en el sistema.	Se llenan correctamente el nombre de la ejecución y el archivo de entrada de la ejecución a modificar.	Ejecución 1	Exam01. mop	El sistema muestra una interfaz que le permitirá al especialista Modificar una ejecución en el sistema.	Se modificó la ejecución en el sistema, se mostró la Interfaz Ejecuciones y se listó las ejecuciones del especialista.
EC1.4 Introducir de forma incorrecta el nombre de la ejecución a modificar.	Se entra incorrectamente el nombre de la ejecución a modificar.	Ejecución 1 ó	Exam01. mop	El sistema muestra una interfaz que le permitirá al especialista Modificar una ejecución en el sistema.	El sistema muestra un mensaje indicando el error.
EC1.5 Introducir de forma incorrecta el archivo de entrada de la ejecución a modificar.	Se entra incorrectamente el archivo de entrada de la ejecución a modificar.	Ejecución 1	Exam01. ki	El sistema muestra una interfaz que le permitirá al especialista Modificar una ejecución en el sistema.	El sistema muestra un mensaje indicando el error.
EC1.4 Cancelar la modificación de una ejecución en el sistema.	Se cancela la modificación de una ejecución.	-	-	El sistema muestra la interfaz Ejecuciones y lista las Ejecuciones del especialista.	Se canceló la modificación de la ejecución en el sistema, se mostró la interfaz Ejecuciones y se listó las ejecuciones del especialista.

Tabla 8. Variables para el Caso de Prueba Buscar Ejecución

No.	Nombre del campo	Clasificación	Valor Nulo	Descripción
1	Criterio de búsqueda	Campo de texto	No	Cadena de caracteres

Tabla 9. Caso de prueba 3: Buscar Ejecución

Escenario	Descripción	V1	Respuesta del sistema	Resultado de las pruebas
EC 1.1 Introducir un criterio válido de búsqueda para realizar una búsqueda de ejecuciones en el sistema	Se introduce un criterio de búsqueda válido.	Ejecución2	El sistema muestra una interfaz que le permitirá al especialista Buscar una ejecución en el sistema.	Se listó el resultado de la búsqueda.
EC1.2 Introducir un criterio inválido de búsqueda para realizar una búsqueda de ejecuciones en el sistema	Se introduce un criterio de búsqueda inválido.	Ejecución 2 ó -	El sistema muestra una interfaz que le permitirá al especialista Buscar una ejecución en el sistema.	El sistema muestra un mensaje indicando el error.

Tabla 10. Caso de prueba 4: Eliminar Ejecución

Escenario	Descripción	Respuesta del sistema	Resultado de las pruebas
EC 1.1 Selección correcta de la Ejecución a eliminar.	Se selecciona correctamente la ejecución a Eliminar.	El sistema muestra una interfaz que le permitirá al especialista seleccionar una ejecución en el sistema.	Se actualizó la lista de Ejecuciones.
EC1.2 Selección incorrecta de la Ejecución a modificar.	Se selecciona incorrectamente la ejecución a Eliminar.	El sistema muestra una interfaz que le permitirá al especialista seleccionar una ejecución en el sistema.	Se muestra un mensaje indicando el error.

4.3. Resultados

A continuación se exponen los resultados alcanzados con el desarrollo del trabajo de diploma, los cuales dan respuesta a los objetivos planteados.

4.3.1. Servicio de química computacional con Gamess

Gamess es un programa de Química Computacional, especializado en métodos *ab initio* (métodos que consumen gran cantidad de CPU debido a su fundamento teórico). Contiene paralelizaciones de sus métodos, por lo que se ejecuta en sistemas paralelos para reducir el tiempo de espera de sus resultados. Por lo que se pasa a la implementación de un Servicio con Gamess, en el clúster Beowulf del Dpto. Bioinformática, con las siguientes características:

- ❖ Clúster Beowulf con el sistema operativo GNU/Linux Debian Lenny 5.0 y con ocho nodos con dos procesadores Intel(R) Core (TM) 2Duo CPU E4500 2.20GHz. Los nodos están interconectados mediante Ethernet Switch 10/100 Mbps, con una topología Beowulf o de anillo y cada nodo consta de 1 Gigabyte de memoria RAM.

A continuación se describen los pasos realizados en el proceso de implementación del Servicio con Gamess en el clúster Beowulf:

Paso 1:

- ❖ Se pasó a la instalación en una pc del Gamess. Para ello se realizaron las siguientes instrucciones:
 - Instalar los paquetes `gfortran libatlas-dev tcsh`.
 - Ejecutar el fichero (`./config`) que configura el programa, este crea el fichero `install.info` que es el que define como compilar todo.
 - Compilar los ficheros básicos `./compall ./compddi ./lked`
 - Ejecutar `./rungms exam01.inp >& exam01.log`.

Resultados del Paso 1: Se logró la instalación del Gamess en una pc, obteniendo resultados satisfactorios.

Paso 2:

- ❖ Se pasó a la instalación en el clúster Beowulf del Dpto. Bioinformática. Para ello se realizaron las siguientes instrucciones:
 - Se realizaron las instrucciones explicadas en el Paso 1, especificando en el fichero install.info, la comunicación entre los nodos por MPI.

Resultados del Paso 2: No se logró la instalación del Gamess en el clúster Beowulf, obteniendo resultados insatisfactorios. Debido a que en su proceso de instalación en el clúster Beowulf no reconoció la comunicación entre los nodos por MPI.

Se realizaron búsquedas bibliográficas para encontrar soluciones al problema de instalación del Gamess utilizando un clúster Beowulf. Se encontró en un foro de discusión de personas, donde explicaban que tuvieron problemas con la versión MPI del Gamess y realizaban la distribución de sus ficheros de entrada (.inp).

Paso 3:

- ❖ Se implementó un programa en lenguaje C el cual se encarga de la distribución del runrms en el clúster Beowulf.
 - Instalar el Gamess secuencial en todos los nodos del clúster Beowulf.
 - Se ejecutó el programa creado en C en el intérprete de BASH.

Resultados del Paso 3: El programa implementado arrojó un error al ejecutarlo en el intérprete de BASH, que no pudo ser solucionado. Por lo que no se pudo realizar la distribución del runrms.

Paso 4:

- ❖ Se implementó un Servicio Web invocando al Gamess secuencial en el clúster Beowulf.
 - Se realizó la conexión al servicio RMI del clúster.
 - Se envió la tarea a ejecutar.
 - Se descargó los resultados del servicio RMI del clúster.
 - Resultados insatisfactorios debido a que nunca se encontró el fichero de salida del programa (.log)

Resultados del Paso 4: El Servicio Web implementado no brindó resultados satisfactorios debido a que no se encontró el fichero de salida (.log), pues en el Servicio RMI del clúster no se supo donde copió las salidas, las cuales estaban especificadas en una dirección.

Por lo que se pasó a la implementación del Servicio Web en una máquina, sin la utilización del clúster de computadoras del Dpto. Bioinformática.

4.3.2. Servicio de química computacional con Mopac

Se desarrolló en servicio web de química computacional utilizando Mopac programa de química computacional especializado en métodos *Semiempíricos* sobre la Plataforma de cálculo Distribuido T_Arenal.

Tiempos de ejecución

Para comprobar la influencia del hardware en los tiempos de respuesta de Mopac2012 se realizaron ejecuciones en 3 ordenadores con distintas prestaciones cada uno, Intel (R) Pentium 4 CPU 3.0, memoria principal 1GB, Intel Core(TM) 2 Duo T5670 CPU 1.8GHz, memoria principal 2GB, Intel (R) Core(TM) i3-2120 CPU 3.30GHz, memoria principal 2GB, todos con sistema operativo Linux. Como datos de entrada se utilizó un fichero que contiene una molécula de aminoácido alifático compuesta por 25 átomos (figura 18).

En cada ejecución se varió únicamente el método de cálculo a utilizar. Mopac2012 puede realizar uno de los siguientes métodos *Semiempíricos* por cada molécula: MNDO, MNDOD, AM1, PM3, PM6, RM1. La tabla 11 muestra los tiempos obtenidos.

Tabla 11. Tiempos de respuesta para una molécula utilizando varios métodos *Semiempíricos*.

PC	S.O	Procesador	RAM	AM1	PM3	PM6	RM1	MNDO	MNDOD
1	Linux	Pentium 4 3.0 GHz	1 GB	25 s	23 s	15 s	10 s	7 s	7 s
2	Linux	Core2duo 1.8 GHz	2GB	20 s	9 s	7 s	4 s	3 s	3 s
3	Linux	i3-2120 3.3 GHz	2GB	8 s	8 s	3 s	2 s	2 s	2 s

```
(palabras clave) am1 ef precise xyz t=10d nointer nolog mmok
(nombre de la molécula) Aminoácidos_Alifáticos
(comentario)
C
C 1.55231 1
O 1.24782 1 120.85182 1
C 3.11799 1 48.24168 1 38.59111 1 3 2 1
N 2.48159 1 59.80745 1 277.17117 1 4 3 2
H 2.08279 1 56.08826 1 94.98437 1 5 4 3
H 2.51701 1 73.03939 1 27.53169 1 6 5 4
H 1.81548 1 67.80913 1 239.26026 1 7 6 5
H 1.81507 1 60.05199 1 104.34559 1 8 7 6
H 2.73163 1 113.09108 1 288.30444 1 9 8 7
N 3.99864 1 64.46852 1 320.79764 1 10 9 8
C 1.42814 1 131.46821 1 251.95985 1 11 10 9
H 1.12363 1 109.86766 1 138.19928 1 12 11 10
H 1.82931 1 35.48028 1 239.17196 1 13 12 11
H 1.82812 1 59.86525 1 323.70068 1 14 13 12
C 5.91806 1 78.93050 1 109.97207 1 15 14 13
C 1.50871 1 132.04387 1 77.43856 1 16 15 14
O 2.41115 1 26.10054 1 111.55324 1 17 16 15
H 4.24467 1 120.69427 1 331.85860 1 18 17 16
H 6.42765 1 13.89973 1 340.05020 1 19 18 17
H 1.81879 1 83.80559 1 305.87706 1 20 19 11
H 1.81919 1 60.00955 1 67.94602 1 21 20 19
O 7.96657 1 53.06252 1 273.54872 1 22 21 20
H .96127 1 38.99675 1 221.71735 1 23 22 21
H 1.51012 1 38.23506 1 231.72186 1 24 23 22
O
```

Figura 18. Estructura del fichero de entrada para Mopac

Como puede observarse los tiempos de respuesta son pequeños cuando se trata de una molécula, esto es debido a que los métodos *Semiempíricos* incluyen parametrizaciones de varias moléculas lo que posibilita una reducción en la complejidad de sus algoritmos en comparación con los métodos *ab initio*. Sin embargo cuando se realizan estudios de grandes sistemas compuestos por cientos de moléculas, los tiempos pueden tornarse extensos. Una solución a este problema es la utilización de un entorno de cálculo distribuido, donde un sistema de numerosas moléculas puede dividirse en otros más pequeños y utilizar varias máquinas para resolverlo.

Las pruebas fueron realizadas mediante la Plataforma de Tareas Distribuidas T-Arenal v2.0 usando hasta 60 estaciones de trabajo conectadas a una red de 100Mbps, 42 Intel (R) Pentium 4 CPU 3.0GHz, memoria

principal 1GB, 14 Intel (R) Core(TM) 2 Duo CPU E4500 2.20GHz, memoria principal de 1GB, 3 Intel (R) Core(TM) i3-2100 CPU 3.10GHz, memoria principal 2GB y 1 Intel (R) Core(TM) i3-2120 CPU 3.30 GHz, memoria principal 2GB, todos con sistema operativo Linux. Como dato de entrada se utilizaron varias moléculas del tipo que se muestra en la figura 17 y como método *Semiempírico* de cálculo el AM1. Puede apreciarse en la figura 19 los tiempos de respuesta obtenidos realizando los cálculos en 3 ordenadores independientes y en la Plataforma T-Arenal, variando la cantidad de moléculas de entrada en cada ejecución.

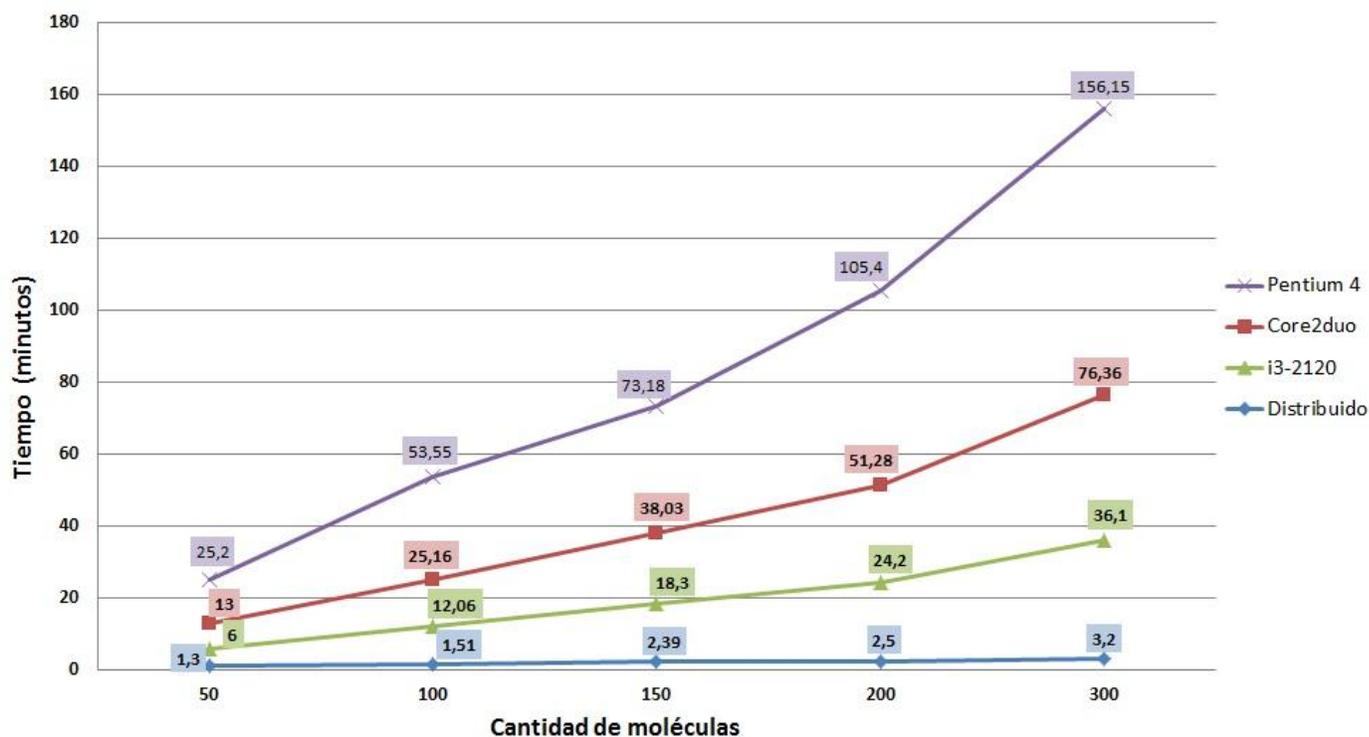


Figura 19. Tiempos de respuesta de Mopac en distintos entornos de ejecución

Como puede observarse, mediante el uso de un entorno distribuido se reduce aceptablemente el tiempo de respuesta para los cálculos sobre sistemas con gran cantidad de moléculas.

Ganancia de velocidad

Para comprobar la ganancia de velocidad (*Speed-up*) de los cálculos distribuidos en T-Arenal se realizaron varias ejecuciones variando la cantidad de PC a utilizar para cada ejecución en 1, 5, 15, 30 y 60 respectivamente, tomando como dato de entrada una muestra de 100 moléculas del tipo que se muestra en la figura 18 y como método *Semiempírico* de cálculo el AM1. Puede apreciarse en la figura 20 como a

medida que se aumenta la cantidad de estaciones de trabajo a utilizar disminuye el tiempo de respuesta de los cálculos.

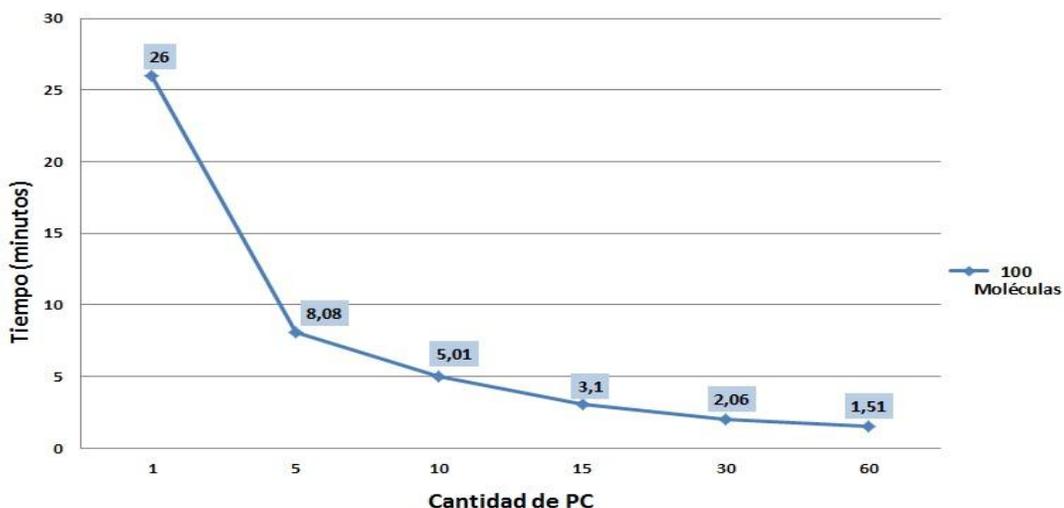


Figura 20. Tiempos de respuesta variando la cantidad de PC

A partir de los tiempos obtenidos en la prueba anterior puede obtenerse una aproximación del comportamiento del *Speed-up* mediante la fórmula $Sp = Tu/Td$, donde Sp es el *Speed-up*, Tu es el tiempo de ejecución unitario o secuencial en una PC y Td es el tiempo de ejecución distribuido. En la figura 21 se observa el *Speed-up* obtenido por el sistema durante las pruebas, donde los tiempos de respuesta fueron reducidos hasta 17,14 veces usando 60 estaciones de trabajo.

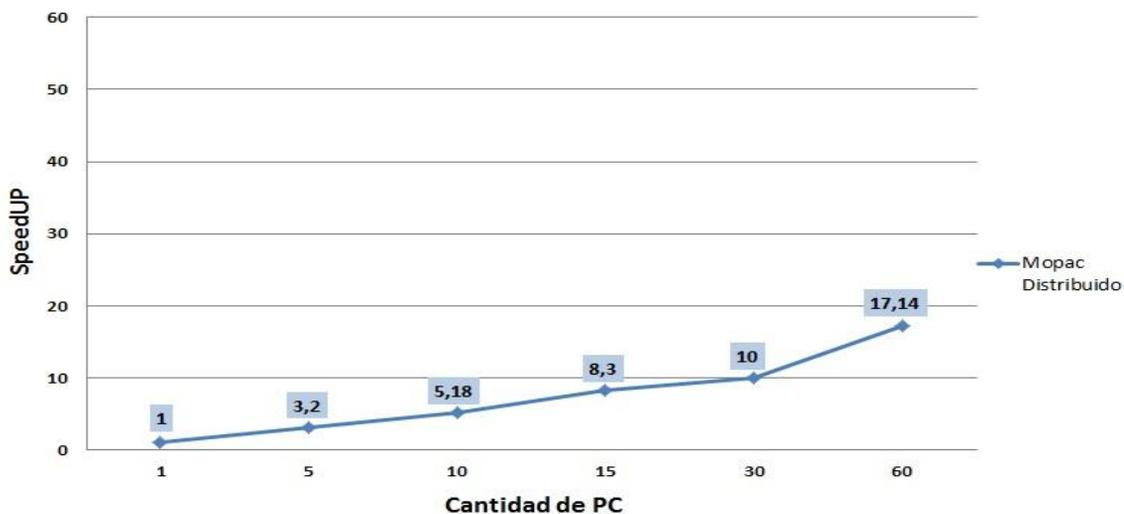


Figura 21. Speed-up para un problema de 100 moléculas

Eficiencia

Para comprobar el aprovechamiento del tiempo de procesamiento en las PC utilizadas en el entorno distribuido se realizan las pruebas de eficiencia.

Puede apreciarse en la figura 22 como haciendo uso de 5 y 15 estaciones de trabajo se aprovecha un 64 y un 55 por ciento respectivamente el tiempo de procesamiento, sin embargo a partir de las 15 PC comienza a disminuir gradualmente la eficiencia del sistema, esto ocurre debido a que se trata de un entorno distribuido donde las estaciones de trabajo no son dedicadas y con características heterogéneas, y donde factores externos como el apagado de las computadoras que estén siendo utilizadas para el cálculo influye negativamente en el rendimiento. Teniendo en cuenta lo antes planteado puede afirmarse que Mopac distribuido muestra una eficiencia aceptable en el aprovechamiento de los recursos de cómputo a través de la Plataforma de Tareas Distribuidas T-Arenal.

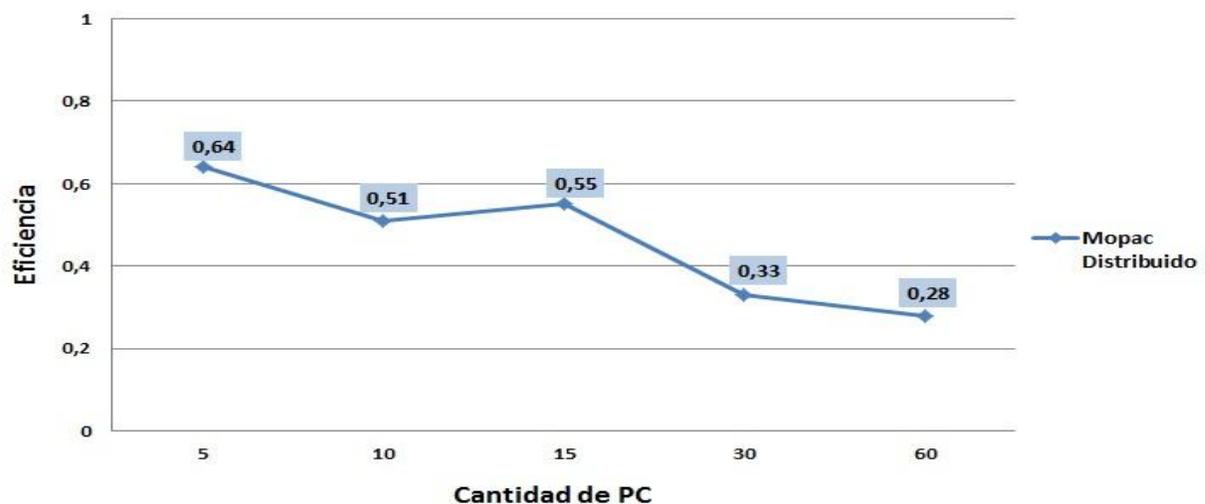


Figura 22. Eficiencia en el entorno distribuido

4.4. Interfaces de la aplicación

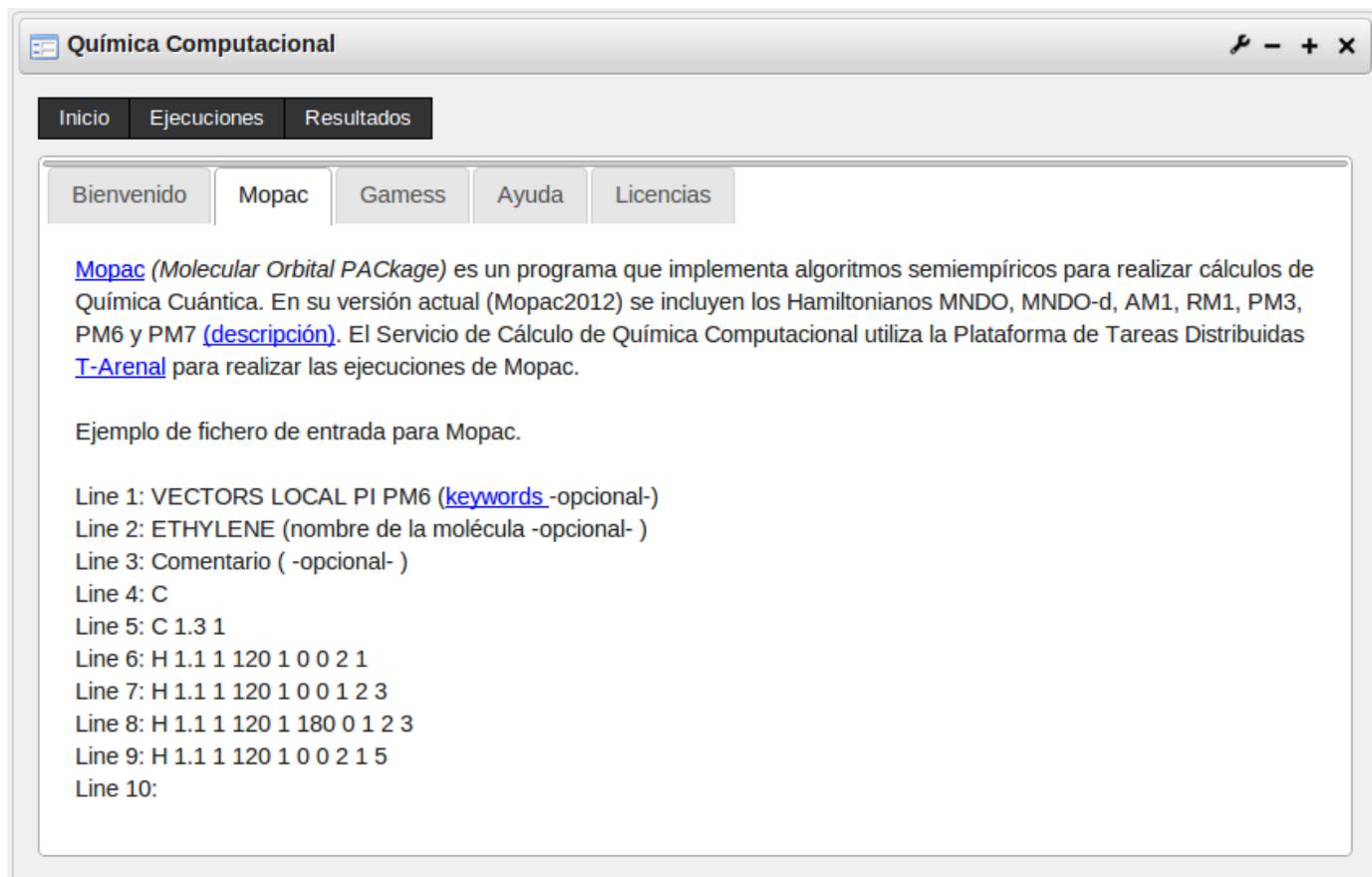


Figura 23. Interfaz Principal



Figura 24. Interfaz Administrar Ejecuciones

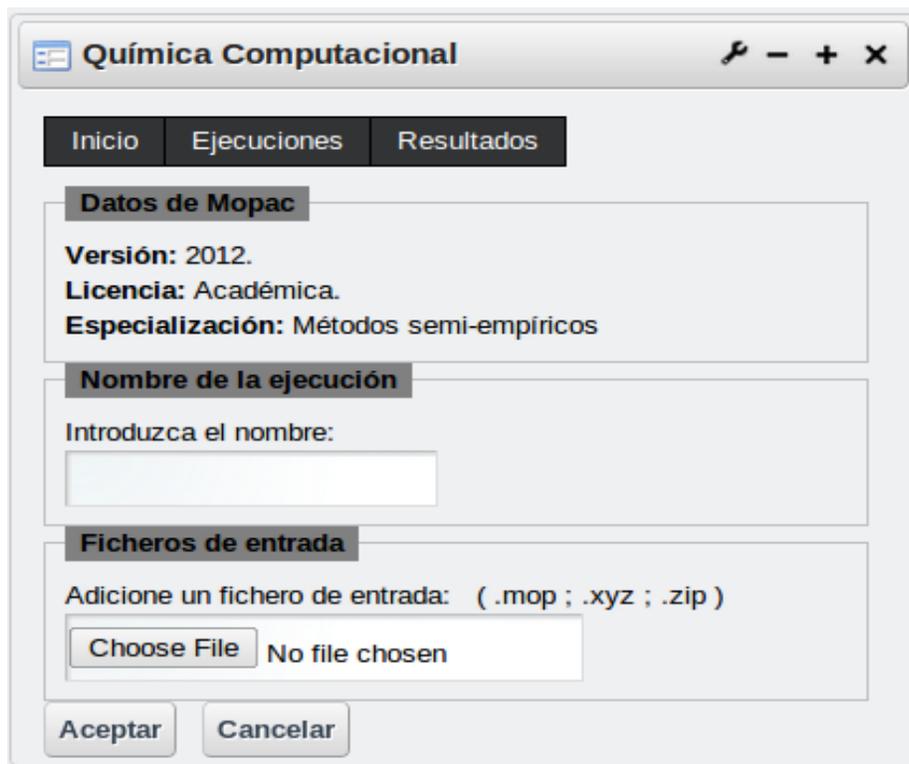


Figura 25. Interfaz Crear Ejecución con Mopac



Figura 26. Interfaz Administrar Resultados

Conclusiones del Capítulo

En este capítulo se construyó el modelo de implementación, para ello se realizó el diagrama de componentes del CU: “Realizar cálculos químicos teóricos con Mopac” y “Realizar cálculos químicos teóricos con Gamess”. Además se definió el modelo del despliegue que muestra los elementos necesarios

para realizar el despliegue e instalación del sistema: al menos un ordenador cliente, un servidor de aplicaciones, un servidor de bases de datos y un servidor de servicios web. Se definieron los casos de prueba de caja negra para la técnica de partición de equivalencia al CU: “*Gestionar Ejecuciones*”, los que se aplicaron a la sección: “*Crear Ejecución*”, “*Modificar Ejecución*”, “*Buscar Ejecución*”, “*Eliminar Ejecución*”. Se implementó un Servicio Web que ejecuta el programa Gamess en el servidor de aplicaciones. Se implementó además un Servicio Web que ejecuta el programa Mopac en un entorno distribuido de computadoras logrando una reducción aceptable del tiempo de respuesta de los cálculos.

Conclusiones generales

Una vez culminado el trabajo es posible afirmar se cumplieron los objetivos trazados y puede concluirse que:

- ❖ Se determinó el uso de los programas de Química Computacional Mopac2012 y Gamess para realizar los cálculos en el servicio web y se brindó la posibilidad de descargar la aplicación Gabedit como herramienta para modelar los datos de entrada para el servicio.
- ❖ Se definieron los requisitos que debía tener la aplicación y se logró el desarrollo del Servicio de Química Computacional.
- ❖ La implementación del Servicio de Química Computacional posibilitará realizar cálculos químicos-teóricos a los usuarios de la Plataforma de Servicios Bioinformáticos.
- ❖ Se utilizó la Plataforma de Cálculos Distribuidos T_Arenal sobre un conglomerado de estaciones de trabajo lográndose una reducción aceptable del tiempo de respuesta de los cálculos realizados.
- ❖ Las pruebas realizadas para validar las funcionalidades que fueron implementadas, demuestran que la aplicación cumple satisfactoriamente con los requisitos que garantizan su correcto funcionamiento.

Recomendaciones

Después de haber logrado los objetivos que se trazaron al principio de este trabajo y como el producto informático se encuentra en su primera versión se plantean las siguientes recomendaciones:

- ❖ Introducir paulatinamente nuevos servicios que abarquen otras metodologías de química computacional no tenidas en cuenta para esta versión del servicio.
- ❖ Continuar la búsqueda de otras herramientas de Química Computacional que tengan sus métodos paralelizados y su arquitectura se ajuste al clúster del Dpto. de Bioinformática con el objetivo de introducirlas al Servicio.

Referencias bibliográficas

1. YOUNG, D. Computational Chemistry: A practical guide for applying techniques to real world problems, Wiley & Sons, 2001, ISBN: 0471333689. 360.
2. RAHA, K., PETERS, M.B., NING YU, B.E., WOLLACOTT, A.M., WESTERHOFF, L.M, KM. The role of quantum mechanics in structure –based drug desing, Drug Discovery Today, 2007, 12:17-18.
3. CRAMER, C. J. Essentials of Computational Chemistry: theories and models, Wiley & Sons, 2002, ISBN: 0471485527.
4. SZABO, A. Modern Quantum Chemistry. Introduction to Advanced Electronic Structure Theory. First Edition, Macmillan Publishing Co., 1982, ISBN: 0-07-062739-8.
5. YOUNG, D. "Appendix A. A.2.4 GAUSSIAN". Computational Chemistry, Wiley& Sons, 2001, ISBN 0471333689. 360.
6. The official Gaussian Website. Gaussian 09. [en línea], 2012. [Consultado el: 20 de noviembre del 2012]. Disponible en: [http://www.gaussian.com/g_prod/g09.htm]
7. YOUNG, D. "Appendix A. A.2.6 MOLPRO". Computational Chemistry, Wiley& Sons, 2001, ISBN 0471333689. 360.
8. Molpro quantum chemistry package. Molpro. [en línea], 2012. [Consultado el: 20 de noviembre del 2012].Disponible en:[<http://www.molpro.net/>]
9. Q_Chem. Q_Chem [en línea] ,2012. [Consultado el: 20 de noviembre del 2012] Disponible en: [<http://www.q-chem.com/>]
10. Amsol Home Page. Amsol- version7.1. [en línea], 2011. [Consultado el: 20 de noviembre del 2012]. Disponible en: [<http://comp.chem.umn.edu/amsol/>]
11. YOUNG, D. "Appendix A. A.3.2 MOPAC". Computational Chemistry, Wiley& Sons, 2001, ISBN 0471333689. 360.
12. Mopac. What is MOPAC2012™? [en línea], 2012. [Consultado el: 20 de noviembre del 2012]. Disponible en: [<http://www.openmopac.net/MOPAC2012.html>].

13. YOUNG, D. "Appendix A. A.2.3 GAMESS". Computational Chemistry, Wiley& Sons, 2001, ISBN 0471333689. 360.
14. Mark Gordon's Quantum Theory Group. Summary of GAMESS Capabilities. [en línea], 2012. [Consultado el: 20 de noviembre del 2012]. Disponible en: [<http://www.msg.ameslab.gov/gamess/capabilities.html>].
15. Gabedit. What is Gabedit? [en línea], 2011. [Consultado el: 20 de noviembre del 2012]. Disponible en: [<http://gabedit.sourceforge.net/home.html>].
16. CESGA. Centro de Supercomputación de Galicia. [en línea] ,2010. [Consultado el: 20 de noviembre del 2012].Disponible en: [<http://www.cesga.es/>]
17. BSC-CNS. Centro Nacional de Supercomputación. [en línea] ,2013. [Consultado el: 5 de enero del 2013].Disponible en: [<http://www.bsc.es/>]
18. THURASINGHAM, B. Secure Semantic Service-Oriented Systems, Taylor & Francis, 2010, ISBN: 9781420073324. 463.
19. Mora Luján, Sergio. Programación de aplicaciones web: Historia, principios básicos y clientes web, Editorial Club Universitario, 2002, ISBN: 8454847589654. 349.
20. DURÁN, F., GUTIÉRREZ, F., PIMENTEL, E., Programación orientada a objetos con Java, Editorial Paraninfo, 2007, ISBN: 9788497325721, 311.
21. AUMAILLE,B. J2EE desarrollo de aplicaciones web. Ediciones ENI, 2002, ISBN: 9782746019126. 357.
22. BRITAIN, J., DARWIN, I. Tomcat: The Definitive Guide, Second Edition, O'Reilly Media, Inc., 2008, ISBN: 9780596554941, 496.
23. PEAK, P., HEUDECKER, N. Hibernate quickly, Dreamtech Press, 2005, ISBN: 9788177226416, 425.
24. JAYASINGH, D., AZEEZ A. Apache Axis2 Web Services, Packt Publishing Ltd, 2011, ISBN: 9781849511575, 308.
25. WALLS, C. Spring in Action, Manning Publications Company, 2011, ISBN: 9781935182351,400.
26. SARIN, A. Portlets in Action. Manning Publications Company, 2011, ISBN: 9781935182542. 612.
27. SEZOV, R. Liferay in Action. Manning Publications Company, 2011, ISBN: 9781935182825. 351.

28. MOMJIAN, B. PostgreSQL: introduction and concepts, Addison-Wesley, 2001, ISBN: 9780201703313,462.
29. The Eclipse Foundation. Eclipse. [en línea], 2013. [Consultado el: 10 de enero del 2013]. Disponible en: [<http://www.eclipse.org/>].
30. OpenUp. Introduction to OpenUp. [en línea], 2013. [Consultado el: 10 de enero del 2013]. Disponible en: [<http://epf.eclipse.org/wikis/openup/index.htm>].
31. OSCAR, S. Visual Paradigm for Uml, International Book Market Service Limited, 2013, ISBN: 9786139166534, 88.
32. PRESSMAN, R. Ingeniería de *Software*. Un enfoque práctico, Mac Graw-Hill, 1992, ISBN: 9789684226746, 628.
33. JACOBSON, I., BOOCH, G., RUMBAUGH,J., El proceso unificado de desarrollo de *software*, Félix Varela, 2004.
34. BUSCHMAN, F., MEUNIER, R., ROHNERT, H. Pattern-Oriented *Software* Architecture. A Sustum of Patterns. John Wiley & Sons, 1996. 512.
35. LARMAN, C. UML y Patrones, Félix Varela, 2004.

Bibliografía

RAHA, K., PETERS, M.B., NING YU, B.E., WOLLACOTT, A.M., WESTERHOFF, L.M, KM. The role of quantum mechanics in structure –based drug desing, *Drug Discovery Today*, 2007, 12:17-18.

YOUNG, D. *Computational Chemistry: A practical guide for applying techniques to real world problems*, Wiley & Sons, 2001, ISBN: 0471333689. 360.

CRAMER, C. J. *Essentials of Computational Chemistry: theories and models*, Wiley & Sons, 2002, ISBN: 0471485527.

SZABO, A. *Modern Quantum Chemistry. Introduction to Advanced Electronic Structure Theory*. First Edition, Macmillan Publishing Co., 1982, ISBN: 0-07-062739-8.

YOUNG, D. "Appendix A. A.2.4 GAUSSIAN". *Computational Chemistry*, Wiley& Sons, 2001, ISBN 0471333689. 360.

The official Gaussian Website. Gaussian 09. [en línea], 2012. [Consultado el: 20 de noviembre del 2012]. Disponible en: [http://www.gaussian.com/g_prod/g09.htm]

YOUNG, D. "Appendix A. A.2.6 MOLPRO". *Computational Chemistry*, Wiley& Sons, 2001, ISBN 0471333689. 360.

Molpro quantum chemistry package. Molpro. [en línea], 2012. [Consultado el: 20 de noviembre del 2012]. Disponible en: [<http://www.molpro.net/>]

Q_Chem. Q_Chem [en línea] ,2012. [Consultado el: 20 de noviembre del 2012] Disponible en: [<http://www.q-chem.com/>]

Amsol Home Page. Amsol- version7.1. [en línea], 2011. [Consultado el: 20 de noviembre del 2012]. Disponible en: [<http://comp.chem.umn.edu/amsol/>]

YOUNG, D. "Appendix A. A.3.2 MOPAC". *Computational Chemistry*, Wiley& Sons, 2001, ISBN 0471333689. 360.

Mopac. What is MOPAC2012™? [en línea], 2012. [Consultado el: 20 de noviembre del 2012]. Disponible en: [<http://www.openmopac.net/MOPAC2012.html>].

YOUNG, D. "Appendix A. A.2.3 GAMESS". Computational Chemistry, Wiley& Sons, 2001, ISBN 0471333689. 360.

Mark Gordon's Quantum Theory Group. Summary of GAMESS Capabilities. [en línea], 2012. [Consultado el: 20 de noviembre del 2012]. Disponible en: [<http://www.msg.ameslab.gov/games/capabilities.html>].

Gabedit. What is Gabedit? [en línea], 2011. [Consultado el: 20 de noviembre del 2012]. Disponible en: [<http://gabedit.sourceforge.net/home.html>].

CESGA. Centro de Supercomputación de Galicia. [en línea] ,2010. [Consultado el: 20 de noviembre del 2012].Disponible en: [<http://www.cesga.es/>]

BSC-CNS. Centro Nacional de Supercomputación. [en línea] ,2013. [Consultado el: 5 de enero del 2013].Disponible en: [<http://www.bsc.es/>]

THURASINGHAM, B. Secure Semantic Service-Oriented Systems, Taylor & Francis, 2010, ISBN: 9781420073324. 463.

Mora Luján, Sergio. Programación de aplicaciones web: Historia, principios básicos y clientes web, Editorial Club Universitario, 2002, ISBN: 8454847589654. 349.

AUMAILLE,B. J2EE desarrollo de aplicaciones web. Ediciones ENI, 2002, ISBN: 9782746019126. 357.

SARIN, A. Portlets in Action. Manning Publications Company, 2011, ISBN: 9781935182542. 612.

SEZOV, R. Liferay in Action. Manning Publications Company, 2011, ISBN: 9781935182825. 351.

BRITAIN, J., DARWIN, I. Tomcat: The Definitive Guide, Second Edition, O'Reilly Media, Inc., 2008, ISBN: 9780596554941, 496.

PEAK, P., HEUDECKER, N. Hibernate quickly, Dreamtech Press, 2005, ISBN: 9788177226416, 425.

WALLS, C. Spring in Action, Manning Publications Company, 2011, ISBN: 9781935182351,400.

JAYASINGH, D., AZEEZ A. Apache Axis2 Web Services, Packt Publishing Ltd, 2011, ISBN: 9781849511575, 308.

MOMJIAN, B. PostgreSQL: introduction and concepts, Addison-Wesley, 2001, ISBN: 9780201703313,462.

DURÁN, F., GUTIÉRREZ, F., PIMENTEL, E., Programación orientada a objetos con Java, Editorial Paraninfo, 2007, ISBN: 9788497325721, 311.

The Eclipse Foundation. Eclipse. [en línea], 2013. [Consultado el: 10 de enero del 2013]. Disponible en: [<http://www.eclipse.org/>].

OpenUp. Introduction to OpenUp. [en línea], 2013. [Consultado el: 10 de enero del 2013]. Disponible en: [<http://epf.eclipse.org/wikis/openup/index.htm>].

OSCAR, S. Visual Paradigm for Uml, International Book Market Service Limited, 2013, ISBN: 9786139166534, 88.

PRESSMAN, R. Ingeniería de *Software*. Un enfoque práctico, Mac Graw-Hill, 1992, ISBN: 9789684226746, 628.

JACOBSON, I., BOOCH, G., RUMBAUGH, J., El proceso unificado de desarrollo de *software*, Félix Varela, 2004.

BUSCHMAN, F., MEUNIER, R., ROHNERT, H. Pattern-Oriented *Software* Architecture. A System of Patterns. John Wiley & Sons, 1996. 512.

LARMAN, C. UML y Patronos, Félix Varela, 2004.

Descripción textual de Casos de Usos

Tabla 12. Descripción del CU Gestionar Ejecuciones

Objetivo	Poder crear, listar, modificar, buscar y eliminar las ejecuciones realizadas por el especialista.
Actores	Especialista (inicia).
Resumen	El caso de uso inicia cuando el especialista selecciona la opción de Ejecuciones en la interfaz principal.
Complejidad	Alta
Prioridad	Crítico
Precondiciones	
Flujo de eventos	
Flujo básico <Gestionar Ejecuciones>	
Actor	Sistema
1.1 Accede a la opción Ejecuciones en la Interfaz Principal del Módulo de Química Computacional.	1.2 Lista las corridas del especialista y muestra las opciones: <ul style="list-style-type: none"> • "Crear ejecución". • "Modificar ejecución". • "Buscar ejecución". • "Eliminar ejecución".
35.3 Selecciona una opción.	1.4 Si escoge: <ul style="list-style-type: none"> • Crear ejecución, ir a la sección "Crear ejecución" • Modificar ejecución, ir a la sección "Modificar ejecución". • Buscar ejecución, ir a la sección "Buscar ejecución". • Eliminar ejecución, ir a la sección "Eliminar ejecución".
Sección 1: "Crear Ejecución"	
Flujo básico Gestionar Ejecuciones.	
Actor	Sistema
1.1 Selecciona la opción "Crear Ejecución".	1.2 Muestra la interfaz para que el especialista introduzca los datos necesarios para crear una ejecución.
1.3 Entra los datos (nombre de la ejecución, archivos necesarios) de la ejecución que desea	1.4 Valida los datos entrados por el especialista, si los datos son correctos crea la ejecución en

crear y presiona el botón Aceptar.	la base de datos.
Flujos alternos	
Nº Evento <Los datos son incorrectos>	
Actor	Sistema
1. Entra los datos incorrectos.	2. Muestra en la interfaz un dialogo mostrando el error.
Sección 2: “Modificar Ejecución”	
Flujo básico Gestionar Ejecuciones.	
Actor	Sistema
1.1 Selecciona la opción “Modificar Ejecución”.	1.2 Valida que se haya seleccionado al menos una ejecución y muestra la interfaz con los datos de la ejecución para que el investigador los modifique.
1.3 Modifica los datos de la ejecución y presiona el botón Aceptar.	1.4 Valida los datos entrados por el investigador y si son correctos modifica los datos de la ejecución en la base de datos.
Flujos alternos	
Nº Evento <No selecciona una ejecución>	
Actor	Sistema
1. No selecciona la ejecución a “Modificar”.	2. Si no hay ejecuciones seleccionadas muestra en la interfaz un diálogo mostrando el error.
Nº Evento <La ejecución a modificar está ejecutándose>	
Actor	Sistema
	2. Si la ejecución ya está en ejecución, muestra en la interfaz un diálogo mostrando el error.
Nº Evento <Los datos son incorrectos>	
Actor	Sistema
1. Entra los datos incorrectos.	2. Muestra en la interfaz un diálogo mostrando el error.
Sección 3: “Buscar Ejecución”	
Flujo básico Gestionar Ejecuciones.	
Actor	Sistema
1.1 Introduce el criterio de búsqueda en el campo de texto y selecciona la opción de buscar ejecución, “Buscar”.	1.2 Válida el criterio de búsqueda y si es correcto, lista en la interfaz las ejecuciones que cumplan con el criterio.
Flujos alternos	
Nº Evento <Los datos son incorrectos>	
Actor	Sistema
1. Entra los datos incorrectos.	2. Muestra en la interfaz un diálogo mostrando el

	error.
Sección 3: “Eliminar Ejecución”	
Flujo básico “Gestionar Ejecuciones”.	
Actor	Sistema
1.1 El especialista selecciona una o varias ejecuciones a eliminar y luego selecciona la opción de eliminar corrida,” Eliminar”.	1.2 Valida que se haya seleccionado al menos una ejecución y elimina la(s) ejecución(es) de la base de datos.
Flujos alternos	
Nº Evento <No selecciona las ejecuciones>	
Actor	Sistema
1. No selecciona las ejecuciones a Eliminar.	2. Muestra en la interfaz un diálogo mostrando el error.
Nº Evento < La ejecución a eliminar está ejecutándose >	
Actor	Sistema
	2.Muestra en la interfaz un diálogo mostrando el error.

Tabla 13. Descripción del CU Administrar Resultados

Objetivo	Poder listar, buscar, descargar o eliminar los resultados obtenidos por el sistema a las ejecuciones del especialista.
Actores	Especialista(inicia)
Resumen	El caso de uso inicia cuando el especialista selecciona la opción de Resultados en la interfaz principal.
Complejidad	Alta
Prioridad	Crítico
Precondiciones	Ejecutada una ejecución
Flujo de eventos	
Flujo básico <Administrar Resultados>	
Actor	Sistema
1.1 Accede a la opción Resultados en la Interfaz Principal del Módulo de Química Computacional.	1.2 Lista los resultados del especialista y muestra las opciones: <ul style="list-style-type: none"> • Buscar resultados. • Eliminar resultados. • Descargar resultados.
1.3 Selecciona una opción.	1.4 Si escoge: <ul style="list-style-type: none"> • Buscar resultados, ir a la sección Buscar resultados. • Eliminar resultados, ir a la sección Eliminar resultados.

	<ul style="list-style-type: none"> • Descargar resultados, ir a la sección Descargar resultados.
Sección 1: “Ver resultados”	
Flujo básico Administrar resultados.	
Actor	Sistema
1. Selecciona la opción Ver resultados.	2. Muestra en la interfaz correspondiente los resultados del especialista.
Flujos alternos Nº Evento <No hay resultados seleccionados>	
Actor	Sistema
1. No selecciona el resultado para visualizar.	2. Muestra en la interfaz un dialogo mostrando el error.
Nº Evento <Más de un resultado seleccionado>	
1. Selecciona más de un resultado para visualizar.	2. Muestra en la interfaz un dialogo mostrando el error.
Sección 2: “Buscar resultados”	
Flujo básico Administrar resultados.	
Actor	Sistema
1.1 Introduce el criterio de búsqueda en el campo de texto y selecciona la opción de Buscar.	1.2 Valida el criterio de búsqueda y si es correcto, lista en la interfaz los resultados que cumplan con el criterio.
Flujos alternos	
Nº Evento <Criterio de búsqueda incorrecto>	
Actor	Sistema
1. Entra el criterio de búsqueda incorrecto.	2. Muestra en la interfaz un dialogo mostrando el error.
Sección 3: “Eliminar resultados”	
Flujo básico Administrar resultados.	
Actor	Sistema
1.1 Selecciona uno o varios resultados a eliminar y luego selecciona la opción Eliminar.	1.2 Valida que se haya seleccionado al menos un resultado y elimina el (los) resultado (s) de la base de datos.
Flujos alternos	
Nº Evento <No hay resultados seleccionados>	
Actor	Sistema
1. No selecciona el resultado para eliminar.	2. Muestra en la interfaz un dialogo mostrando el error.

Diagrama de clases del Diseño

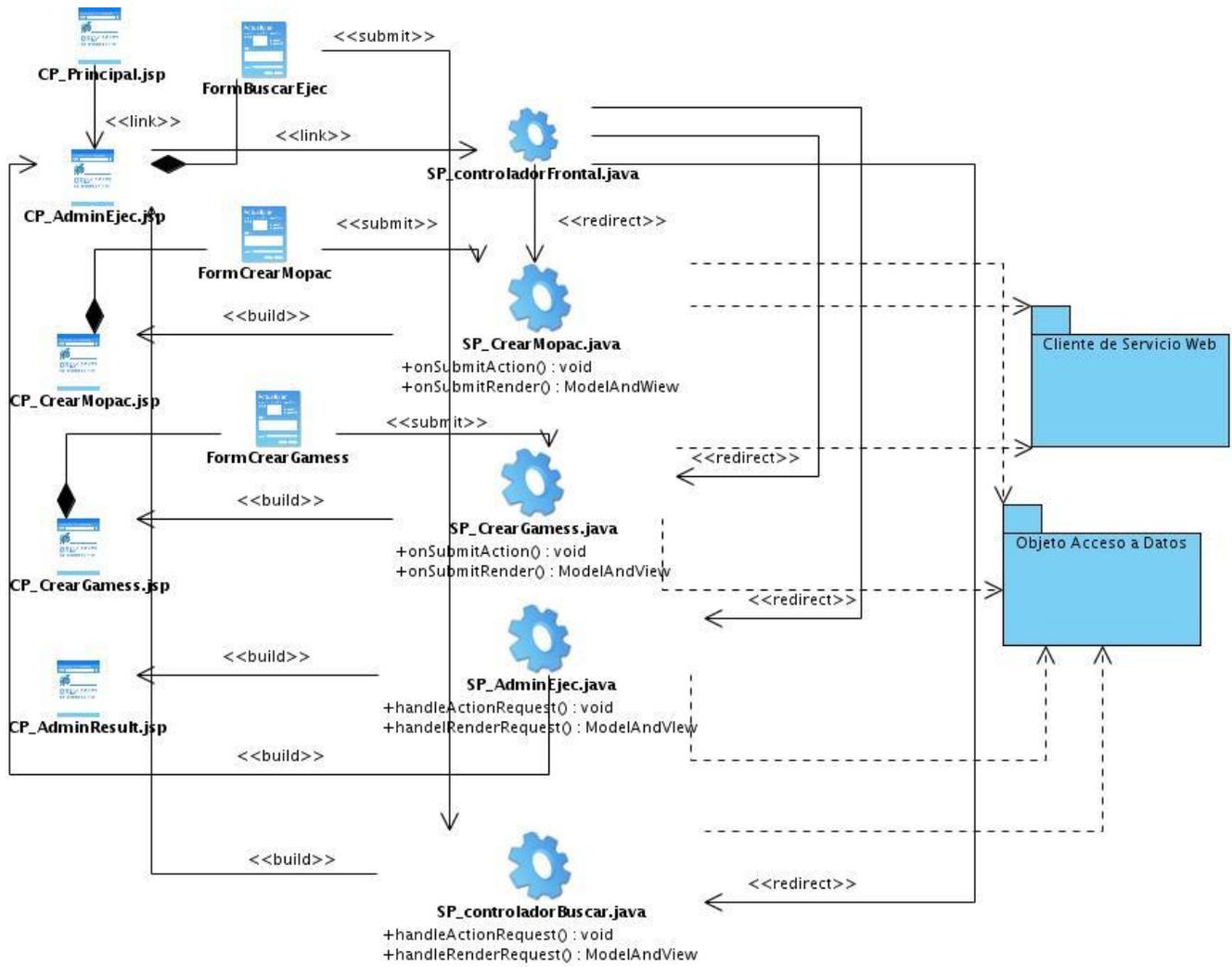


Figura 27. Diagrama del Diseño del CU Gestionar Ejecución

Diagrama de Secuencias

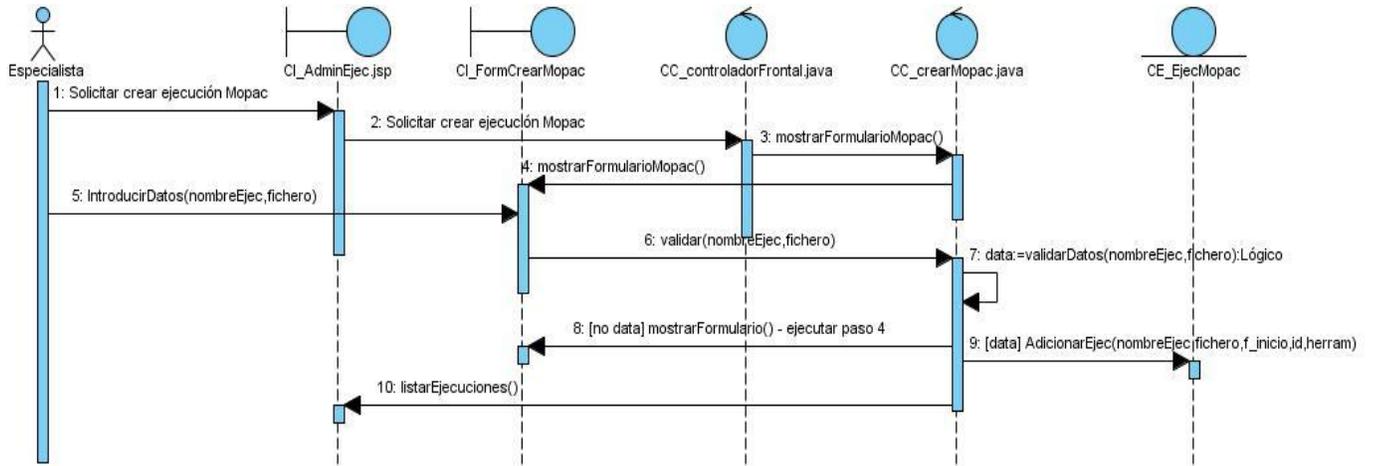


Figura 28. Diagrama de secuencia del CU Gestionar Ejecución de la sección Crear Ejecución

Diagrama de Componente

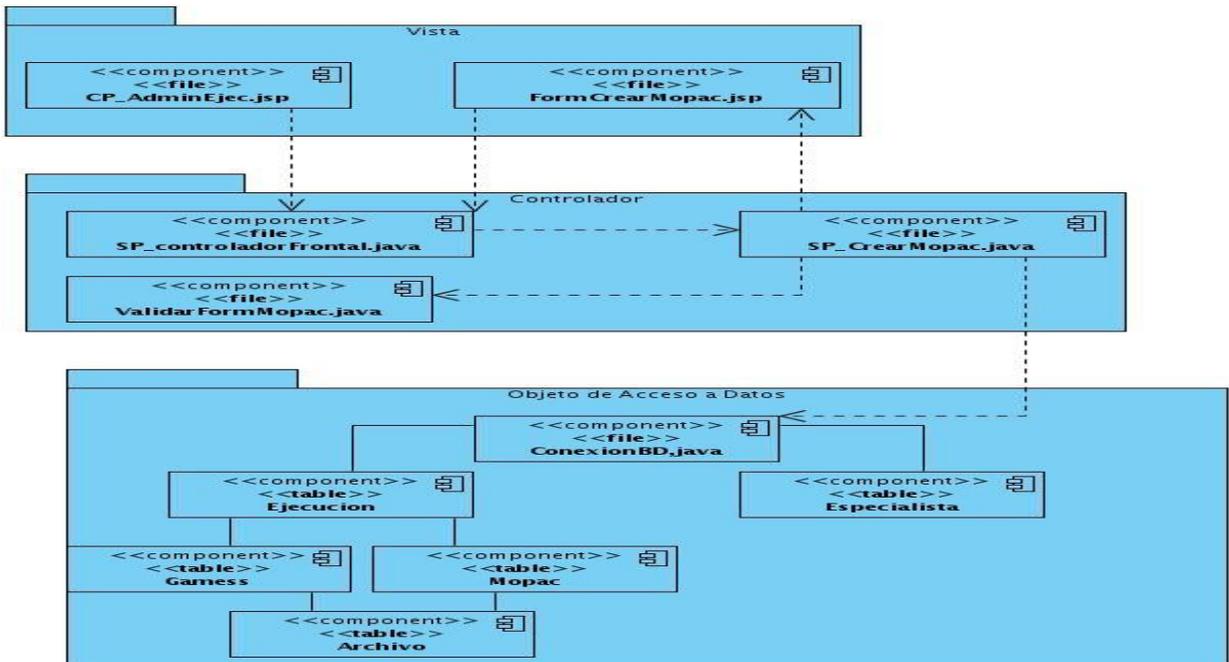


Figura 29. Diagrama de componentes del CU Gestionar Ejecución de la sección Crear Ejecución