Universidad de las Ciencias Informáticas

Facultad 5

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

Programa para la búsqueda de similitud de fragmentos en grafos químicos ponderados por propiedades químico-físicas

Autor:

Aylin María Rodríguez Jiménez

Tutor:

M.Sc. Aurelio Antelo Collado

Co-tutor:

Dr. Ramón Carrasco Velar

Ing. Álvaro Luis Maceo Pixa

La Habana

2013

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de este trabajo y reconocemos la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo con carácter exclusivo. Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

_____	_____
Aylin María Rodríguez Jiménez.	M.Sc. Aurelio Antelo Collado
(Autor)	**(Tutor)**
_____	_____
Dr. Ramón Carrasco Velar	Ing. Álvaro Luis Maceo Pixa
(Co-tutor)	**(Co-tutor)**

DATOS DE CONTACTO

Tutor: M.Sc. Aurelio Antelo Collado.

Ciudadanía: cubano.

Institución: Universidad de las Ciencias Informáticas (UCI).

Título: Ingeniero Industrial. Master en Matemática Aplicada.

Categoría Docente: Asistente, con 7 años de experiencia. Actualmente, Vicedirector de Investigaciones y Posgrado del ISEC.
Correo electrónico: aantelo@uci.cu

Co-Tutor: Dr. Ramón Carrasco Velar.

Ciudadanía: cubano.

Institución: Universidad de las Ciencias Informáticas (UCI).

Título: Dr. Ciencias Químicas.

Categoría Docente: Auxiliar.

Correo electrónico: rcarrasco@uci.cu

Co-Tutor: Ing. Álvaro Luis Maceo Pixa.

Ciudadanía: cubano.

Institución: Universidad de las Ciencias Informáticas (UCI).

Título: Ingeniero en Ciencias Informáticas.

Correo electrónico: almaceo@uci.cu

AGRADECIMIENTOS

A mis tíos Tony, Elena, Nayra por apoyarme siempre en todas las decisiones de mi vida.

A mi hermano Alejandro por ser amigo y mi mayor confidente.

A mi abuela tetén, espero que estés orgullosa de mí.

A mi papá Frank, eres el mejor padre del mundo.

A mis primos Ayled, Chuchy, Dany, Cory y Amy

A mis tíos Delia y Santiago por ser tan dulces y quererme mucho.

A mi hermana Elizabeth y a su mamá

A dos grandes amigas Lucy y Alina por ser tan preocupados por mis estudios.

A un gran amigo Adrián Peña, gracias por ser especial.

A Jairo por compartir su vida conmigo y socorrerme en todo momento.

A mi suegris Ileana por ayudarme en todo momento y enseñarme cosas tan lindas.

A mis amigos Karel Piorno, Karel Rodríguez, Adrian Trujillo, Adrian Hernández, Félix, Yudy, Ridel, Hebert, Heydis, Alex Pardo, Lidises y todos ustedes que compartieron estos cinco años, gracias por darme aliento frente a las vicisitudes.

A mis tutores por dedicarme tiempo y ayudarme con la tesis.

A mis compañeras del apartamento por soportarme todos los días y reír conmigo a pesar de todas nuestras diferencias.

A todo aquel que aportó su granito de arena en esta travesía.

DEDICATORIA

A mi hermano Alejandro, por ser mi mamá, mi papá y mi mejor amigo.

A mi abuela tetén, por haber sido una gran confidente y la mejor abuela del mundo.

A mi tía Elena por haber confiado en mi hasta el último momento de su vida y se que un día como hoy estarías orgullosa de mi.

RESUMEN

En la actualidad, el auge de la informática ha provocado la automatización de diversos procesos en todas las áreas de la sociedad. En el caso de la Química existe una comunidad activa de profesionales que valiéndose del empleo de la informática, realizan investigaciones para mejorar los diagnósticos y encontrar nuevos compuestos que ayuden a contrarrestar diferentes enfermedades.

En el presente trabajo se muestra un software de visualización de moléculas para el proyecto **"Visualización y Minería de Grafos Ponderados**". La herramienta está diseñada principalmente para ayudar al especialista químico a realizar búsqueda de similitudes de forma automática, debido a que las búsquedas manuales pueden ser erróneas y poner en riesgo la elaboración de fármacos.

Como método, se implementó un modelo de búsqueda basado en una función de similitud en el cotejo de grafos. En la primera etapa del modelo se recibe una entrada con las propiedades a comparar, posterior a esto se calcula la similitud mediante una función de probabilidad. Luego de aplicar la función se verificó que el resultado fuese válido en un rango dado, probando que comparten una mayor cantidad de propiedades.

Finalmente se logró integrar el algoritmo de búsqueda de similitud a la arquitectura diseñada, demostrando la efectividad del algoritmo. Se mostró los resultados de las pruebas de aceptación realizadas al software por el cliente y de las pruebas unitarias realizadas al código por el equipo de desarrollo.

Palabras claves: Búsqueda, función, similitud, grafos, ponderados.

ÍNDICE DE TABLAS

Tabla 1 "Roles de la metodología XP" 18

Tabla 2"HU Abrir fichero que contiene información de la molécula". 19

Tabla 3"HU Cerrar fichero que contiene información de la molécula". 20

Tabla 4 " HU Calcular los índices topográficos híbridos de la molécula" 20

Tabla 5 " HU Comparar Moléculas" 21

Tabla 6 " HU Visualizar Molécula Balls" 22

Tabla 7 " HU Visualizar Molécula Van Der Walls" 23

Tabla 8 " HU Visualizar Molécula Wireframe" 24

Tabla 9 “HU Visualizar Índices Electrotopográfico” 25

Tabla 10 “HU Visualizar Índices Electrotopográfico” 26

Tabla 11 “HU Visualizar Índices Lipotopográfico”..... 27

Tabla 12 “HU Visualizar Índices Mixtos". 28

Tabla 13"HU Insertar etiquetas a los átomos de la molécula. " 29

Tabla 14 "HU Eliminar etiquetas a los átomos de la molécula". 30

Tabla 15 "HU Aumentar o Disminuir el tamaño de la molécula." 31

Tabla 16"HU Mostrar átomos de Hidrógeno a la molécula"..... 32

Tabla 17"Ocultar átomos de Hidrógeno a la molécula." 33

Tabla 18 "HU Seleccionar los átomos de la molécula." 34

Tabla 19 "HU Eliminar sombreado de la molécula." 35

Tabla 20 "Estimación de esfuerzos por HU" 36

Tabla 21 "Plan de duración de las iteraciones" 38

Tabla 22Tarjeta CRC “Molecule”. 39

Tabla 23 Tarjeta CRC “MolQuery" 39

Tabla 24 Tarjeta CRC “Reduced Graph”. 40

Tabla 25 Tarjeta CRC “Centro Descriptor”. 40

Tabla 26 Tarjeta CRC “Secuencia”. 41

Tabla 27 Tarjeta CRC “Tipo Centro Descriptor”. 41

Tabla 28 "Tarea 1 Iteración 1" 43

Tabla 29 "Tarea 2 Iteración 1" 44

Tabla 30 "Tarea 1 Iteración 2" 45

Tabla 31 "Tarea 2 Iteración 2" 45

Tabla 32 "Tarea 1 Iteración 3" 46

Tabla 33 "Tarea 2 Iteración 3" 46

Tabla 34 "Tarea 3 Iteración 3" 46

Tabla 35 "Tarea 4 Iteración 3" 47

Tabla 36 "Prueba de aceptación No 1." 47

Tabla 37 "Prueba de aceptación No 2." 48

ÍNDICE DE FIGURAS

Figura 1"Relación entre la Informática, las Ciencias de la Salud, la Biología Molecular, la Genética y la Biotecnología" 7

Figura 2 "Modelo conceptual" 13

Figura 3 "Componentes visuales de Java en la aplicación"..... 15

Figura 4 "Diagrama de clases del diseño"..... 42

Figura 5 "Prueba unitaria realizada con JUnit". 49

ÍNDICE

DECLARACIÓN DE AUTORÍA ..... **II**

DATOS DE CONTACTO ..... **III**

AGRADECIMIENTOS ..... **IV**

DEDICATORIA ..... **V**

RESUMEN ..... **VI**

ÍNDICE DE TABLAS ..... **VII**

ÍNDICE DE FIGURAS ..... **VIII**

ÍNDICE ..... **IX**

INTRODUCCIÓN **1**

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA **6**

Introducción **6**

1.1 La Bioinformática y su relevancia en el desarrollo de la industria farmacéutica. **6**

1.2 La búsqueda de similitud molecular **7**

1.3 Cotejo de grafos **8**

1.4 Ejemplos de técnicas de cotejo inexacto **9**

1.4.1 Distancia de edición de grafos 9

1.4.2 β arista isomorfismo 9

1.4.3 Homeomorfismo con vértice/arista disjuntas 9

1.4.4 Basadas en las probabilidades de sustitución 9

1.4.5 Función de distancia utilizada: Tanymoto 10

1.5 Algoritmos **10**

1.6 Selección de la metodología y el ambiente de desarrollo **11**

1.7 Modelo conceptual **13**

1.8 Componentes visuales en Java **13**

1.8.1 Flamingo 14

1.8.2 SwingX 14

1.8.3 MyDoggy 14

Conclusiones del capítulo **15**

CAPÍTULO 2 ANÁLISIS Y DISEÑO **16**

Introducción **16**

2.1 Análisis de la solución propuesta **16**

2.2 Requisitos del sistema **16**

2.2.1 Requisitos funcionales 16

2.2.2 Requisitos no funcionales 17

2.4 Roles de la metodología XP **18**

2.5 Fase de planificación **18**

2.5.1 Historias de usuario (HU) 18

2.5.2 Estimación de esfuerzos por historias de usuario 35

2.5.3 Plan de iteraciones 36

2.5.4 Plan de duración de las iteraciones 36

2.6 Fase de diseño **38**

2.6.1 Tarjetas CRC 38

2.6.2 Diagrama de clases del diseño 41

Conclusiones del capítulo **42**

CAPÍTULO 3 IMPLEMENTACIÓN Y PRUEBA **43**

Introducción **43**

3.1 Fase de implementación **43**

3.1.1Iteración 1 43

3.1.2Iteración 2 44

3.1.3 Iteración 3 45

3.2 Fase de pruebas **47**

3.2.1 Pruebas de aceptación 47

3.2.2 Pruebas unitarias 48

Conclusiones del capítulo **49**

CONCLUSIONES **50**

RECOMENDACIONES **51**

BIBLIOGRAFÍA **52**

REFERENCIAS BIBLIOGRÁFICAS **54**

INTRODUCCIÓN

En la actualidad, el constante desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) ha permitido intervenir en un sin número de procesos. Con mayor o menor grado estas se han convertido en una herramienta esencial para la conformación y comprensión de nuestro mundo. No sólo contribuyen a la rápida difusión de conocimientos, sino también a la rápida evolución tecnológica, lo cual provoca continuas transformaciones en las estructuras económicas, sociales, culturales y mejoran sustancialmente la calidad de vida.

Cuando se habla de las ciencias de la vida se refiere a todos los campos que se ocupan del estudio de los seres vivos. Además de la biología abarca también otros campos relacionados como la medicina, biomedicina y bioquímica. Estas ciencias y la biotecnología son consideradas por muchos como la próxima gran revolución de la economía del conocimiento que después de las tecnologías de la información, crearán nuevas oportunidades.

El ritmo con que se desarrollan los conocimientos en biotecnología y ciencias de la vida a escala mundial, generan cambios en esferas como la medicina, agricultura, producción alimentaria así como en la protección del medio ambiente, aportando nuevos descubrimientos científicos.

El creciente desarrollo alcanzado por los actores de las ciencias de la vida ha ido produciendo una dependencia cada vez mayor de las tecnologías que generan, almacenan, procesan y analizan la información. La industria farmacéutica, las grandes empresas genéticas, las dedicadas a la medicina genética y las compañías biotecnológicas, tienen una gran necesidad y una estrecha relación con la bioinformática. Los grandes laboratorios cuentan con grupos multidisciplinarios que cooperan y se retroalimentan constantemente. La biomedicina y bioinformática ayudan al desarrollo de procesos terapéuticos con menos efectos secundarios, permitiendo además, mejores métodos de diagnóstico.

Es en este punto que se hace importante el diseño de fármacos, centrados en el descubrimiento de nuevos agentes terapéuticos más específicos y con menos efectos colaterales; para lograrlo, es necesario encontrar nuevos compuestos que se unan y ayuden a contrarrestar virus, tumores malignos, entre otras enfermedades. Para almacenar y recuperar información relacionada con las complejas estructuras químicas de estos compuestos, se crearon grandes bases de datos que la recopilan generalmente en forma de grafos químicos.

En los últimos años se ha puesto de manifiesto la necesidad de convertir estos grandes volúmenes de datos en información útil. Para ello se han ido desarrollando técnicas y métodos que permiten procesar el cúmulo de datos en los repositorios. Ejemplo de tales técnicas lo constituye el descubrimiento de patrones frecuentes y la minería de grafos que consiste en encontrar elementos o patrones representativos dentro de un grafo.

Una vez descubierta una subestructura, puede usarse para simplificar el grafo original mediante el reemplazo de dicha subestructura por un vértice que la represente. Debido a que existen muchos fenómenos diferentes que pueden ser representados con grafos (ejemplo: la composición química de un elemento, la estructura de una proteína, etc.) se hace importante poder extraer información implícita de dichos fenómenos (Chakrabarti, March 2006).

Muchos repositorios de datos químicos están o pueden ser representados mediante colecciones de grafos, las cuales se pueden clasificar, entre otros aspectos, por las moléculas que lo representan o los tipos de grafos que contienen.

Algunas de estas colecciones han sido utilizadas para el análisis de estructuras bioquímicas. El descubrimiento de patrones frecuentes, en especial la detección de subgrafos frecuentes (SF) en colecciones de grafos, es un importante problema en la minería de grafos.

Debido al crecimiento de las soluciones a muchos problemas de Bioquímica varios autores han trabajado en el desarrollo de algoritmos eficientes para el descubrimiento de SF. Los algoritmos que abordan el problema de identificar subgrafos frecuentes en grafos, difieren entre sí por:

- La estrategia de búsqueda que emplean (en amplitud o en profundidad).
- La forma en que generan candidatos a patrones (extender o combinar patrones encontrados).
- La naturaleza de los grafos que examinan (si es una colección de grafos o un solo grafo) y el conjunto de subgrafos que encuentran (todos o parte de ellos).

Los trabajos reportados en la literatura para la minería de SF muestran el avance logrado en esta temática. Sin embargo, aún queda mucho por hacer para lograr mejores desempeños computacionales en tiempo y espacio. Esto se debe a la existencia de algoritmos con gran complejidad computacional y otros que necesitan de mayor capacidad de almacenamiento para mantener sus estructuras internas.

En Cuba, donde se ha desarrollado un elevado potencial humano en el campo de las ciencias Bioinformáticas y Quimioinformáticas, se investigan estas disciplinas, que se

iniciaron en las primeras décadas del siglo XX; al igual que en otros países, los avances de la ingeniería genética y las nuevas tecnologías de la información, condicionaron su surgimiento, lo que conllevó a la creación de vínculos indisolubles entre la Informática, y las ciencias biológicas y químicas.

En la Universidad de las Ciencias Informáticas (UCI), las investigaciones en este campo se realizan específicamente, mediante el proyecto **Visualización y Minería de Grafos Ponderados** que se desarrolla de conjunto con la participación de los centros como el Centro de Desarrollo de Informática Industrial (CEDIN) y el Centro de Informatización de Seguridad Ciudadana (ISEC); el mismo se encarga de crear soluciones informáticas en el campo de la visualización de moléculas y la minería de grafos que ayuden a los especialistas de la Química en los diferentes campos que esta abarca.

En el trabajo del proyecto, se definió que los vértices del grafo molecular se ponderan con propiedades químico-físicas particionadas sobre el grafo, las cuales modulan el valor de diferentes índices topográficos para dar lugar a los llamados índices híbridos; es decir, no a través de estructuras, como se realiza comúnmente, pero está en desarrollo el algoritmo de búsqueda de subgrafos de interés a partir de este nuevo principio. Por lo cual es necesario crear un software capaz de realizar el cálculo, la búsqueda y la visualización de grafos y subgrafos para la realización de búsquedas de subgrafos químicos que contribuyan a la realización de estudios de relación estructura-actividad y diseño de fármacos. Una vez resuelto ese problema, se facilitaría el trabajo de los especialistas que se dedican al diseño de fármacos asistido por computadora.

Por tanto, la **situación problémica** es la siguiente:

- Carencia de un software que implemente un algoritmo de búsqueda de grafos y subgrafos ponderados por propiedades químico-físicas para la realización de estudios de la relación estructura-actividad y diseño de fármacos.

Por todo lo anterior, se plantea como **problema a resolver** de este trabajo:

¿Cómo detectar semejanzas entre grafos y subgrafos químicos?

Por lo que el **objeto de estudio** es: Sistemas informáticos para la búsqueda de semejanzas moleculares.

Para dar respuesta al problema antes mencionado se traza, como **objetivo general**: Desarrollar una aplicación informática para la búsqueda de similitud de fragmentos en grafos químicos, empleando descriptores topográficos híbridos.

Todo lo anterior permite definir el **campo de acción** como: Sistemas informáticos para la búsqueda de semejanzas moleculares basadas en la similitud en grafos ponderados por propiedades químico-físicas.

Para alcanzar dicho objetivo se determina desarrollar las siguientes **tareas:**

1. Elaboración del marco teórico de la investigación a partir del estado del arte existente sobre el tema.
2. Estudio del lenguaje de programación java para desarrollar el algoritmo de búsqueda.
3. Selección de los índices topográficos híbridos, que se emplearán en el algoritmo.
4. Estudio de la biblioteca CDK, para realizar el cálculo de descriptores y su relación cuantitativa estructura-actividad.
5. Estudio del visualizador molecular JMol, para la representación visual de las moléculas.
6. Implementación de los algoritmos para el cálculo de los índices topográficos híbridos para átomos en la biblioteca CDK.
7. Implementación del algoritmo para la detección de los centros descriptores de un grafo químico en la biblioteca CDK.
8. Implementar la fragmentación de la molécula en centros descriptores en la biblioteca CDK
9. Implementación del algoritmo de fragmentación de un grafo químico en la biblioteca CDK.
10. Integración de la biblioteca CDK a la aplicación.
11. Implementación del algoritmo de búsqueda de fragmentos similares utilizando una función de similitud seleccionada.
12. Validación del algoritmo de búsqueda implementado en diferentes ensayos biológicos.
13. Implementación de los algoritmos para la visualización de los índices híbridos en el visualizador molecular JMol.
14. Integración del visualizador molecular JMol a la aplicación.
15. Validación de la aplicación desarrollada.

El presente trabajo hará uso durante el proceso de desarrollo e investigación, de los siguientes **métodos de investigación**:

Métodos Teóricos:

Analítico-Sintético: se emplea para buscar información acerca del problema propuesto y para extraer los elementos que están relacionados con el objeto de estudio.

Modelación: se usa para representar el conocimiento adquirido durante la investigación y en el diseño de la aplicación.

Métodos Empíricos:

Consulta de las fuentes de información: se emplea en la selección de la información importante y en la elaboración del marco teórico.

Consulta de especialistas: para que las personas calificadas en el tema evalúen la aplicabilidad y utilidad del software desarrollado.

Pruebas: se utilizan para comprobar si la aplicación funciona correctamente.

Por todo lo anterior se propone como **Idea a defender:** Con el uso de la aplicación informática desarrollada para la búsqueda de similitud de fragmentos, empleando descriptores topográficos híbridos se facilitaría al especialista en química la detección de semejanzas en los subgrafos químicos a partir de la relación estructura-actividad.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

Introducción

En este capítulo se presentan las tendencias actuales de la Bioinformática y su relevancia en el desarrollo de la Industria Farmacéutica; se dan a conocer algunos conceptos para comprender en qué consiste el análisis de similitud molecular, la búsqueda de similitud; se hará mención de algunas técnicas de similitud estructural y algunos algoritmos de búsqueda que lo utilizan; se definirá la metodología y herramientas que se utilizarán en la construcción del software y se arribará a conclusiones parciales.

1.1 La Bioinformática y su relevancia en el desarrollo de la industria farmacéutica.

La Bioinformática, llamada también Biología Molecular Computacional, corresponde como tal a una disciplina científica que utiliza tecnología de la información para organizar, analizar y distribuir información de Biomoléculas con la finalidad de responder preguntas complejas (Altschul SF, 1994).

Según la definición del NCBI[1] en el año 2001: *"Bioinformática es un campo de la ciencia en el cual confluyen varias disciplinas tales como: biología, computación y tecnología de la información…"* Una de las principales aplicaciones es la simulación, el análisis y la minería de datos a los resultados obtenidos en el estudio de moléculas relevantes para la vida. A pesar de ser relativamente joven está continuamente produciendo nuevas tecnologías para obtener datos en nuevos experimentos, y que requieren el desarrollo de nuevas estrategias o algoritmos y su implementación en servidores Web que es la forma tradicional en el área de ofrecer los servicios a la comunidad científica.

Esta ciencia se nutre de dos grandes áreas del conocimiento: las ciencias biológicas y las ciencias de la computación. Los datos obtenidos a partir de los estudios realizados en sí mismos no son comerciables, pero la información relevante e implícita en ellos sí lo es.

Cuando se habla de la genómica se refiere a todas las ciencias y técnicas dedicadas al estudio integral del funcionamiento, el contenido, la evolución y el origen de la información genética que posee un organismo o una especie en particular. Al comienzo de su evolución, el concepto de bioinformática se refería sólo a la creación y

[1] **NCBI:** siglas del inglés *National Center for Biotechnology Information* o Centro Nacional para la Información Biotecnológica en español.

mantenimiento de base de datos donde se almacena información biológica, tales como secuencias de nucleótidos y aminoácidos. El desarrollo de este tipo de base de datos no solamente significaba el diseño de la misma sino también el desarrollo de interfaces complejas donde los científicos pudieran acceder a los datos existentes y suministrar o revisar datos. Luego toda esa información debía ser combinada para contribuir a formar una idea lógica de las actividades celulares normales, de tal manera que los investigadores pudieran estudiar cómo estas actividades se veían alteradas en presencia de una enfermedad. El proceso de analizar e interpretar datos biológicos es conocido como biocomputación. Dentro de estas áreas del conocimiento está: la informática, las Ciencias de la Salud, la Biología Molecular, la Genética y la Biotecnología.

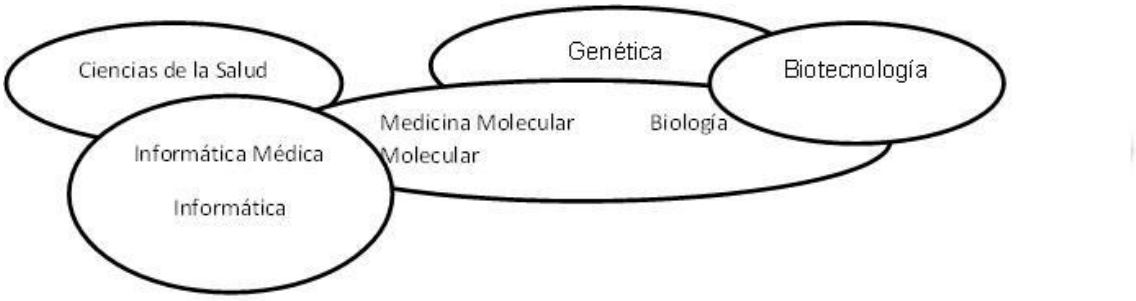


Figura 1"Relación entre la Informática, las Ciencias de la Salud, la Biología Molecular, la Genética y la Biotecnología"

Algunas consecuencias de este trabajo que asocia diferentes ciencias, son la identificación de las causas moleculares de las enfermedades. Junto con el desarrollo de la industria Biotecnológica y Farmacéutica; se han desarrollado mejores métodos de diagnóstico, la identificación de avisos terapéuticos y el desarrollo de fármacos personalizados, así como una mejor medicina preventiva. Realmente, una computadora no puede reemplazar un laboratorio y es por esto que los aportes tradicionales químicos y farmacológicos continuarán siendo importantes.

1.2 La búsqueda de similitud molecular

La búsqueda de similitud molecular es aquella técnica de recuperación de la información, mediante la cual a partir de una estructura química definida por un especialista, son identificadas aquellas moléculas en una base de datos que son más similares a la molécula, usando medidas cuantitativas de similitud intermolecular.

Esta técnica de extracción de la información es aplicable a las bases de datos bidimensionales (2D) y tridimensionales (3D); y dos clases de medidas de similitud de estructuras químicas han sido desarrolladas, las medidas globales y locales. Las

medidas de clase global son aquellas que evalúan un valor numérico para la completa similitud entre dos moléculas. Las medidas del tipo local son aquellas medidas que proveen información física como resultado de la alineación de una molécula con otra; por ejemplo estos sistemas producen un mapeo de las características de la molécula con aquellas características estructurales de las moléculas existentes en la base de datos, de forma que este proceso constituya una superposición de una molécula sobre otra (Poliakov,2004).

Es importante señalar que los sistemas de información en su mayoría han trabajado en aquellas búsquedas que abarcan un gran número de registros o estructuras químicas en las bases de datos, estando estas enmarcadas en las de clase Global. Las conocidas por locales conllevan procesamientos más lentos y no pueden incluirse gran número de registros a procesar de acuerdo con el estado actual del hardware y software disponible; generalmente son sistemas desarrollados para trabajos específicos de similitud entre un grupo muy limitado de moléculas (Poliakov,2004). Por otro lado el cálculo de similitud entre moléculas debe ser efectivo, su aplicación debe dar resultados útiles al usuario; y debe ser eficiente, que se traduce en utilizar los requerimientos de computación necesarios para lograr búsquedas de similitud en bases de datos de tamaño considerable en tiempos razonables.

Los trabajos de búsquedas de similitud crean las bases para un mejor desarrollo de los proyectos de investigación a la hora de evaluar nuevas variantes de moléculas para ser utilizadas en sustitución de otras; creando todo un campo de gestión de información dentro de las estructuras químicas con directa implicación en los proyectos de investigación y desarrollo.

1.3 Cotejo de grafos

El cotejo de grafos (en inglés, graph matching) (Conte, 2004) es una técnica de similitud estructural, la cual es una de las tareas más costosas en tiempo en la minería de SF. Algunas de las formas de cotejo de grafos son NP-Completo (ejemplo, homeomorfismo (Xiao,2008)), otras son NP-Duro (ejemplo, sub-isomorfismo (Gago,2010)) y otras como el caso del isomorfismo no se ha podido determinar si es P^2 o NP^3 (Conte, 2004). Existen dos formas de abordar el cotejo de grafos: cotejo exacto y cotejo inexacto. El cotejo exacto consiste en determinar si las etiquetas y la estructura de dos grafos son idénticas

[2] **P:** siglas del inglés *polynomial time* o tiempo polinomial en español

[3] **NP:** siglas del inglés *nondeterministic polynomial time* o tiempo polinomial no determinista en español

y ha sido utilizada con éxito en muchas aplicaciones. Sin embargo, existen aplicaciones donde esta forma exacta de describir las correspondencias no es aplicable con éxito. Debido a que hay que tolerar cierto nivel de distorsiones geométricas, variaciones semánticas o desajustes entre vértices o aristas, mientras se realiza la búsqueda de los subgrafos frecuentes. Es por eso que se ha hecho necesario evaluar la similitud entre grafos admitiendo algunas diferencias estructurales, o sea, mediante técnicas de cotejo inexacto. El cotejo inexacto consiste en encontrar la mejor adecuación entre vértices o aristas de dos grafos para determinar su similitud admitiendo diferencias entre estructuras y etiquetas. Sobre esta base, se plantea la necesidad de realizar la minería de SF utilizando cotejo inexacto de grafos.

1.4 Ejemplos de técnicas de cotejo inexacto

1.4.1 Distancia de edición de grafos

Una técnica utilizada para calcular la similitud entre dos grafos es la Distancia de Edición de Grafos (DEG); esta técnica se le conoce como el conjunto de operaciones de edición estándar sobre un grafo y el proceso a seguir sería reetiquetear o sustituir, e insertar un vértice o una arista del grafo seleccionado. Además existen también otras operaciones de edición como la fusión o división de vértices o aristas donde mientras más pequeña sea la DEG resultante el costo de edición será menor, siendo muy similares los grafos en término de estructura y etiquetas (Eichinger, 2010).

1.4.2 β arista isomorfismo

Esta función de similitud es definida en términos de β arista isomorfismo (Ver definición 1). Mediante esta función se calcula la similitud entre dos grafos tratando la falta de aristas entre grafos, manteniendo exactos los vértices (Zhang, 2007).

Definición 1 (Isomorfismo) Se dice que dos grafos son isomorfos si existe una correspondencia biunívoca entre V1 y V2 que mantiene las aristas y todas las etiquetas.

1.4.3 Homeomorfismo con vértice/arista disjuntas

Esta función calcula la similitud entre dos grafos basándose en la topología menor, que no es más que al generar del grafo, se reduzcan los caminos independientes de uno de sus subgrafos hasta transformarlos en aristas (XIAO,2008).

1.4.4 Basadas en las probabilidades de sustitución

Existe una función de similitud entre dos grafos que se define en término de sub-isomorfismo aproximado (Jia, 2009) y una función de diferencias definida en término de grado de aproximación. Estas se basan en las probabilidades de que las etiquetas puedan ser sustituidas por otras. Este enfoque permite distinguir las sustituciones de las

etiquetas de los vértices y aristas de los grafos, característica que no presentan las funciones de similitud mencionadas anteriormente. La distinción de las sustituciones entre etiquetas se ha realizado mediante el uso de una función de probabilidad por cada etiqueta sobre todo el dominio de estas. Esta función de probabilidad da como resultado cuán probable sería que una etiqueta pudiera tomar el valor de otra.

Definición 2 (Función de probabilidad). Una función de probabilidad de una variable aleatoria discreta, definida por p (v), es una función tal que, al sustituir v por un valor de la variable, el valor que toma la función es la probabilidad de que la variable u asuma el valor de v. Para ello la función debe cumplir las siguientes propiedades:

- $\forall$ v, p(v) ≥0,
- $\sum_v$ p(v)=1,
- P(v) = P(u=v)

1.4.5 Función de distancia utilizada: Tanymoto

$$\mathrm{f(A,B)} = \frac{A * B}{|\mathrm{A}|^2 + |\mathrm{B}|^2 - \mathrm{A} * \mathrm{B}}$$

Donde:

A: Propiedad

B: Propiedad a comparar

Nota: Las propiedades que se comparan son las electrotopográficas, refractotopográficas, las lipotopográficas y la combinación de las tres propiedades.

- ✓ Para ello la función debe cumplir:

Para obtener un Máximo Común de Propiedad la función debe arrojar un resultado entre 0.98 y 1, demostrando así que comparten una mayor cantidad de propiedades en común.

1.5 Algoritmos

Existen varios algoritmos para la minería de subgrafos frecuentes aproximados en colecciones de grafos que utilizan diferentes funciones de similitud en el cotejo de grafos. La minería de subgrafos aproximados puede dividirse en cinco según el enfoque del cotejo: SUBDUE (HOLDER L.B., 1992) y RNGV (SONG Y., 2006) están basados en distancia de edición de grafos, Monkey (ZHANG S., 2008) se basa en β-arista sub-isomorfismo; CSMiner (XIAO Y., 2008) utiliza el sub-homeomorfismo con vértice/arista

disjuntas; MUSE (ZOU, 2009) se basan en sub-isomorfismo; gApprox (CHEN C., 2007), APGM (JIA Y., 2011) y VEAM (Acosta Mendoza, y otros, junio 2011)están basados en probabilidades de sustitución. Estos últimos especifican cuáles vértices, aristas o etiquetas pueden reemplazar otras. De este modo, se defiende la idea de que no siempre una etiqueta de vértice o una etiqueta de arista puedan ser sustituidas por cualquier otra. En el algoritmo gApprox se realiza la minería de subgrafos frecuentes aproximados en un solo grafo siendo de interés en este trabajo el procesamiento de colecciones de grafos. Los algoritmos APGM y VEAM usan matrices de sustitución para realizar la minería de subgrafos frecuentes aproximados en colecciones de grafos. APGM solamente trata las variaciones entre el conjunto de etiquetas de los vértices. Por otro lado, VEAM realiza el proceso de minería con los conjuntos de etiquetas de vértices y aristas.

En este trabajo se utiliza la función de probabilidad de sustitución, basándose en el valor de las propiedades físico químicas que peticionan los átomos de la molécula. Dicha función es aplicada al algoritmo de búsqueda de semejanzas. Esto se debe a la necesidad de un algoritmo eficiente que permita algunas variaciones en los datos utilizando probabilidades.

1.6 Selección de la metodología y el ambiente de desarrollo

La metodología que guiará el proceso de desarrollo será XP[4], porque es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP centra las fuerzas en la implementación del software y solo documenta los artefactos más importantes. Se basa en la simplicidad, la comunicación, el reciclado continuo de código y funciona mejor en grupos pequeños (Beck, 2000)

El ambiente de desarrollo para la realización de la aplicación es el siguiente:

- Para gestionar la documentación del software, se selecciona Visual Paradigm for UML como herramienta CASE, pues soporta el ciclo completo de desarrollo

[4] **XP:** siglas del inglés *Extreme Programming* o programación extrema en español.

y genera el código en una amplia gama de lenguajes. Además, proporciona varios tutoriales, es multiplataforma y muy profesional.

- Java es un lenguaje de programación que se utiliza para la creación de aplicaciones informáticas, ha sido diseñado a modo de eliminar las complejidades de otros lenguajes como C y C++. Tiene muchas ventajas tales como: Lenguaje orientado a objetos, lo que hace que los programas se construyan a partir de módulos independientes, y que esos módulos se pueden transformar o ampliar fácilmente. Es multiplataforma ya que los programas pueden ser ejecutados en cualquier plataforma sin necesidad de hacer cambios. Es compatible a cualquier plataforma a nivel de código compilado, a nivel de fuente y a nivel de biblioteca. Flexible pues combina la robustez y legibilidad gracias a una revisión de tipos durante la compilación y durante la ejecución.

- Se escoge Eclipse porque es un entorno integrado de desarrollo (IDE) muy poderoso que permite la construcción de aplicaciones en Java. Tiene además, los beneficios siguientes: Es una herramienta de código abierto. Se puede ejecutar en una gran cantidad de sistemas operativos incluyendo Windows y Linux. Capacidad de ser soportado para distintas arquitecturas, resaltado de sintaxis, entre otros.

- Se utiliza el Chemistry Development Kit (CDK) pues esta es una biblioteca Java para la química computacional y la bio/quimioinformática. Es desarrollada por más de 40 programadores alrededor del mundo y usado en más de 10 proyectos académicos e industriales diferentes de todo el mundo. En los últimos cuatro años, la biblioteca de CDK ha evolucionado hasta convertirse en un potente paquete de quimioinformática completo con código que va desde los cálculos del descriptor QSAR[5] hasta la generación de modelos 2D y 3D.

- Se utiliza Jmol para la visualización de las moléculas pues este es un visor de estructuras químicas en 3D. Permite usarse como herramienta de enseñanza o de investigación; ejemplos de ciencia que pueden auxiliarse de sus beneficios están a Química y la Bioquímica. Es software libre y de código abierto, escrito en Java y por ello se puede ejecutar en Windows, Mac OS X, Linux y sistemas Unix. Existe una aplicación independiente y un kit de herramientas de desarrollo que puede integrarse en otras aplicaciones Java. La característica más notable es una miniaplicación (applet) que se puede integrar en páginas web para

[5] QSAR: siglas del inglés *quantitative structure-activity relationship* o relación cuantitativa estructura-actividad en español

mostrar las moléculas de muchas formas. Por ejemplo, las moléculas se pueden mostrar como modelos de "bola y palo", modelos "que llenan el espacio", modelos de "cinta", etc. Jmol es compatible con una amplia gama de formatos de archivo moleculares, incluyendo Protein Data Bank (PDB), archivo de información cristalográfico (CIF), MDL Molfile (mol) y Chemical Markup Language (CML).

1.7 Modelo conceptual

Un modelo conceptual es una herramienta importante para el análisis orientado a objetos, es por ello que se ha querido incluir aquí. En él se explican los conceptos más significativos en el dominio del problema, identificando los atributos y las asociaciones. Un modelo conceptual representa cosas del mundo real, no componentes del software, de ahí que sea de especial importancia para clientes y desarrolladores. En UML suele representarse mediante un grupo de diagramas de estructura estática donde no se define ninguna operación. En estos diagramas se muestran conceptos (objetos), asociaciones entre conceptos (relaciones) y atributos de conceptos (atributos).La **Figura 2** muestra el modelo conceptual del software a desarrollar para validar el algoritmo de búsqueda.

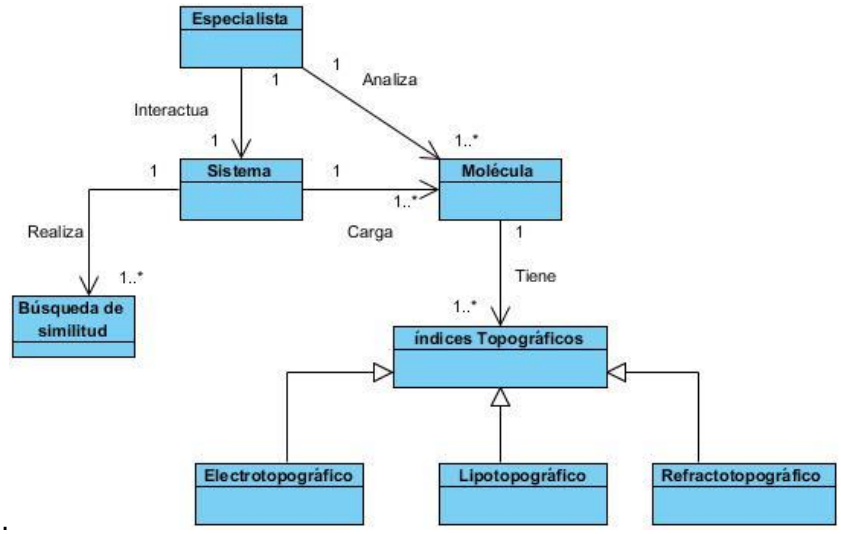


Figura 2 "Modelo conceptual"

Especialista: Persona capacitada para interactuar con el sistema.

1.8 Componentes visuales en Java

En la mayoría de los sistemas operativos actuales, se ofrece una cantidad de código para simplificar la tarea de programación. Este código toma la forma, normalmente, de

un conjunto de bibliotecas dinámicas que las aplicaciones pueden llamar cuando lo necesiten. Las bibliotecas proporcionan una interfaz abstracta para tareas que son altamente dependientes del hardware de la plataforma destino y de su sistema operativo. Tareas tales como manejo de las funciones de red o acceso a ficheros, suelen depender fuertemente de la funcionalidad nativa de la plataforma destino. En el caso concreto anterior, las bibliotecas java.net y java.io implementan el código nativo internamente, y ofrecen una interfaz estándar para que aplicaciones Java puedan ejecutar tales funciones. Finalmente, no todas las plataformas soportan todas las funciones de una aplicación Java. En estos casos, las bibliotecas pueden emular esas funciones usando lo que esté disponible, o bien ofrecer un mecanismo para comprobar si una funcionalidad concreta está presente.

1.8.1 Flamingo

Flamingo es una implementación Swing la cual es una biblioteca gráfica para Java. Incluye widgets para interfaz gráfica de usuario tales como cajas de texto, botones, desplegables y tablas, el cual tiene paquete de componentes para Windows, gratis y de gran utilidad, con una funcionalidad visualmente similar a la del menú superior de Microsoft Office 2010, denominado Ribbon. Los componentes incluidos, básicamente bibliotecas JAVA, aportan visualizaciones consistentes en el procesador y permiten nuevas configuraciones al usuario. Para que Flamingo funcione es necesario tener instalado la máquina virtual de Java 6.0 o superior

1.8.2 SwingX

SwingX es una extensión del propio Swing, donde se pretenden mejorar algunos de los controles existentes e incluir otros nuevos. Así, por ejemplo, SwingX añade funciones avanzadas de ordenación, filtrado, selección y resaltado a controles básicos de Swing, como tablas (JTable), árboles (JTree) y listas (JList). Añade nuevas funcionalidades a otros controles como cuadros de diálogo (JDialog) o paneles (JPanel), e incluye algunos controles nuevos no presentes en Swing como JXTreeTable, JXLoginPanel o JXDatePicker, para la presentación de tablas jerarquizadas, controles de autenticación o cuadros de selección de fecha.

1.8.3 MyDoggy

MyDoggy es una fuente abierta de Java desarrollada, para la gestión de ventanas secundarias, dentro de la ventana principal. Permite mover, cambiar el tamaño o la extracción de las ventanas secundarias. Además, presta apoyo a la gestión de contenidos de la ventana principal.

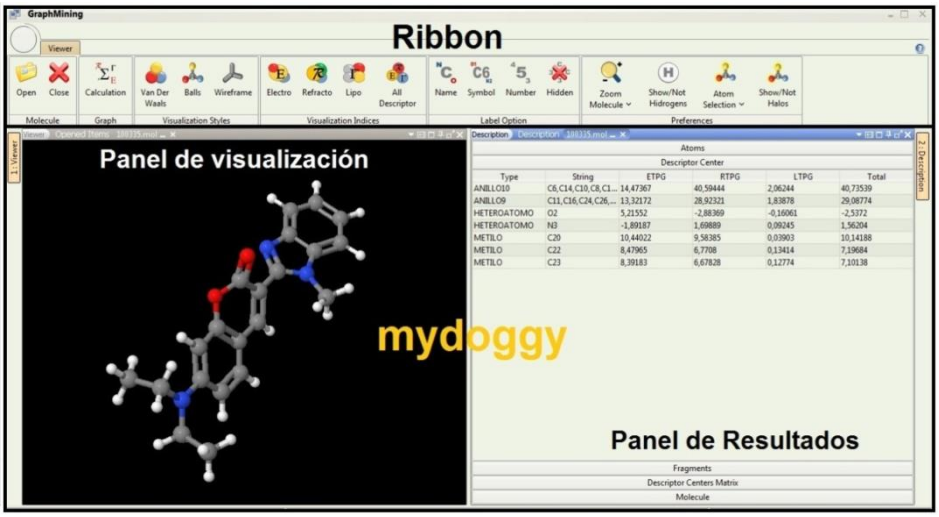


Figura 3 "Componentes visuales de Java en la aplicación"

Conclusiones del capítulo

A partir del análisis realizado en el capítulo se logró dar a conocer algunos conceptos para entender en que consiste el tema del trabajo. Se analizó técnicas de cotejo inexacto de grafos y fue seleccionado para el desarrollo del algoritmo, el de probabilidad de sustitución; específicamente como función de distancia Tanymoto y se dieron a conocer algunos algoritmos que utilizan esta técnica, pero todos se basan en la búsqueda estructural y no en cuanto a propiedades, lo cual motiva a desarrollar un algoritmo que cumpla las limitaciones de los que existen actualmente.

CAPÍTULO 2 ANÁLISIS Y DISEÑO

Introducción

En este capítulo se establecerá la solución al problema a través de especificaciones de requisitos que rigieron el desarrollo de la solución y de la definición de las historias de usuarios y de la planificación del tiempo de trabajo. Presenta como objetivo principal describir las características fundamentales que poseerá el sistema, así como establecer las iteraciones por las que tiene que pasar el proyecto para su desarrollo. Además se mostrará una planificación de cómo se trabajarán las iteraciones y se describirá el diseño de la aplicación.

2.1 Análisis de la solución propuesta

Se conoce que un requisito funcional define el comportamiento interno del software, es decir, funcionalidades específicas de la aplicación; son complementados por los requisitos no funcionales, que se enfocan en el cambio en el diseño o en la implementación.

A partir de la situación problémica y el estudio realizado, se definieron como principales requisitos funcionales (RF) del software a desarrollar, validando el algoritmo de búsqueda, los listados a continuación.

2.2 Requisitos del sistema

2.2.1 Requisitos funcionales

RF1 Abrir fichero que contiene información de la molécula.

RF2 Cerrar fichero que contiene información de la molécula.

RF 3. Calcular los índices topográficos híbridos de la molécula.

RF 4 Comparar Moléculas.

RF 5 Visualizar Molécula Balls.

RF 6 Visualizar Molécula Van Der Waals.

RF 7 Visualizar Molécula Wireframe.

RF 8 Visualizar Índices Electrotopográfico.

RF 9 Visualizar Índices Refractotopográfico.

RF 10 Visualizar Índices Lipotopográfico.

RF 11 Visualizar Índices mixtos.

RF 12 Insertar etiquetas a los átomos de la molécula.

RF 13 Eliminar etiquetas a los átomos de la molécula.

RF 14 Aumentar o Disminuir el tamaño de la molécula.

RF 15 Mostrar átomos de hidrógeno a la molécula.

RF 16 Ocultar átomos de hidrógeno a la molécula.

RF 17 Seleccionar los átomos de la molécula.

RF 18 Eliminar sombreado de la molécula.

2.2.2 Requisitos no funcionales

Usabilidad: La aplicación debe ser intuitiva y fácil de utilizar.

Software: El sistema debe funcionar sobre el Sistema Operativo Windows XP o Superior y en cualquier versión de Linux; Máquina Virtual de Java versión 6.0 o Superior.

Restricciones en el diseño y la implementación: El Lenguaje de programación a ser usado para la implementación es Java.

Hardware: El software debe instalarse en un ordenador con al menos 512 MB de memoria RAM y el micro AMD Athlon.

Rendimiento: El sistema no debe presentar los resultados necesariamente en tiempo real.

Soporte: Uso de tecnologías libres. El sistema debe ser implementado con tecnologías libres.

2.3 Roles de la metodología XP.

Se definió como persona relacionada con el sistema, a aquella que interactúa de una forma u otra con el mecanismo propuesto, dígase vinculado al proceso de desarrollo, así como a la persona que interactúa con la herramienta de gestión de script (Jeffries, 2001).

Roles	Descripción
Programador	Es la persona encargada de realizar la implementación de la herramienta y desarrollar todas las especificaciones que requiera el cliente.
Cliente (Especialista en química)	Es la persona encargada de escribir las historias de usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las historias de usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio.

Tabla 1 "Roles de la metodología XP"

2.4 Fase de planificación

Se define el alcance del proyecto, donde el cliente plantea sus necesidades a través de las historias de usuarios. Además, se estima la prioridad y el esfuerzo necesario para desarrollar cada historia de usuario (HU) y se realiza el cronograma de iteraciones de acuerdo a las mismas.

2.4.1 Historias de usuario (HU)

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento las historias de usuario pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas (Jeffries, 2001).

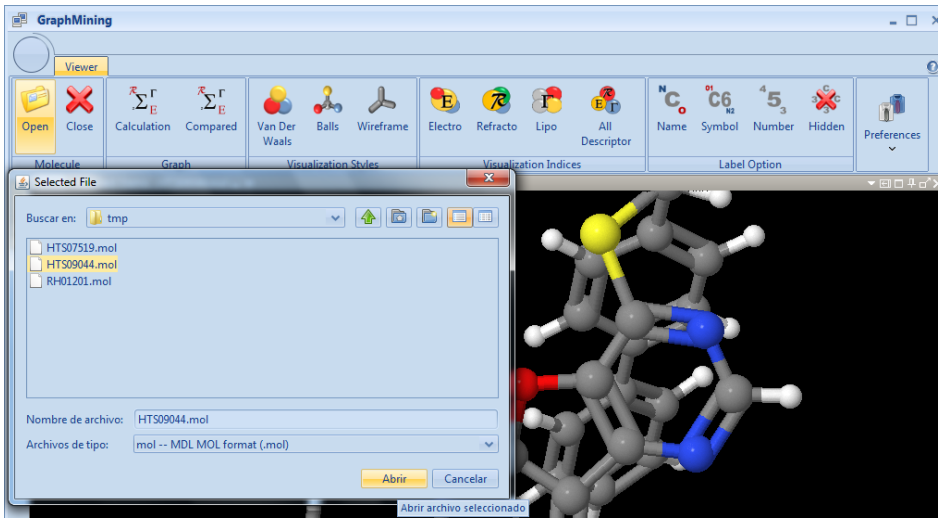
Historia de Usuario	
Número: 1	**Usuario:** Especialista
Nombre historia: Abrir fichero que contiene información de la molécula.	
Prioridad en el negocio: Media	**Nivel de complejidad:** Media
Tiempo de Estimación: 1	**Iteración asignada:** 1
Descripción: Se debe abrir el fichero *.mol, el cual contiene la información necesaria de la molécula para que el sistema sea capaz de visualizarla y representarla mediante los índices electrotopográfico, refractotopográfico y lipotopográfico, para que el cliente (Especialista) pueda analizar los resultados de la visualización obtenida.	
Observaciones: 	

Tabla 2"HU Abrir fichero que contiene información de la molécula".

Historia de Usuario	
Número: 2	**Usuario:** Especialista
Nombre historia: Cerrar fichero que contiene información de la molécula.	
Prioridad en el negocio: Media	**Nivel de complejidad:** Baja
Tiempo de Estimación: 1	**Iteración asignada:** 1
Descripción: Se debe cerrar el fichero *.mol, luego que el usuario (Especialista) haya analizado los resultados de la visualización obtenida.	

Observaciones:

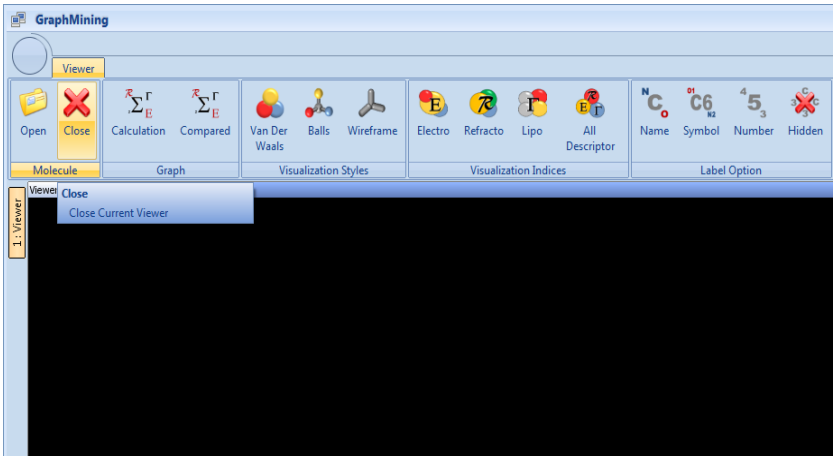
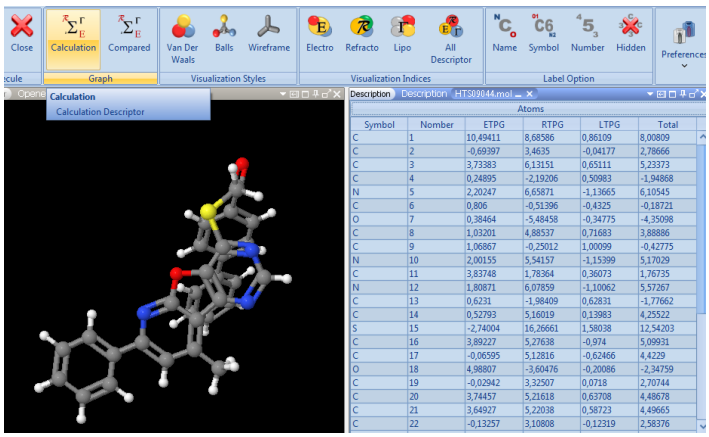


Tabla 3"HU Cerrar fichero que contiene información de la molécula".

Historia de Usuario	
Número: 3	**Usuario:** Programador
Nombre historia: Calcular los índices topográficos híbridos de la molécula	
Prioridad en el negocio: Alta	**Nivel de complejidad:** Alta
Tiempo de Estimación: 2	**Iteración asignada:** 1
Descripción: Se realizará el cálculo de los índices electrotopográfico, refractotopográfico y lipotopográfico asociados a los átomos, centros descriptores, fragmentos de la molécula y una matriz de distancia asociada a los centros descriptores de la molécula.	
Observaciones: 	

Symbol	Nomber	ETPG	RTPG	LTPG	Total
C	1	10,49411	8,68586	0,86109	8,00809
C	2	-0,69397	3,4635	-0,04177	2,78666
C	3	3,73383	6,13151	0,65111	5,23373
C	4	0,24895	-2,19206	0,50983	-1,94868
N	5	2,20247	6,65871	-1,13665	6,10545
C	6	0,806	-0,51396	-0,4325	-0,18721
O	7	0,38464	-5,48458	-0,34775	-4,35098
C	8	1,03201	4,88537	0,71683	3,88886
C	9	1,06867	-0,25012	1,00099	-0,42775
N	10	2,00155	5,54157	-1,15399	5,17029
C	11	3,83748	1,78364	0,36073	1,76735
N	12	1,80871	6,07859	-1,10062	5,57267
C	13	0,6231	-1,98409	0,62831	-1,77662
C	14	0,52793	5,16019	0,13983	4,25522
S	15	-2,74004	16,26661	1,58038	12,54203
C	16	3,89227	5,27638	-0,974	5,09931
C	17	-0,06595	5,12816	-0,62466	4,4229
O	18	4,98807	-3,60476	-0,20086	-2,34759
C	19	-0,02942	3,32507	0,0718	2,70744
C	20	3,74457	5,21618	0,63708	4,48678
C	21	3,64927	5,22038	0,58723	4,49665
C	22	-0,13257	3,10808	-0,12319	2,58376

Tabla 4 " HU Calcular los índices topográficos híbridos de la molécula"

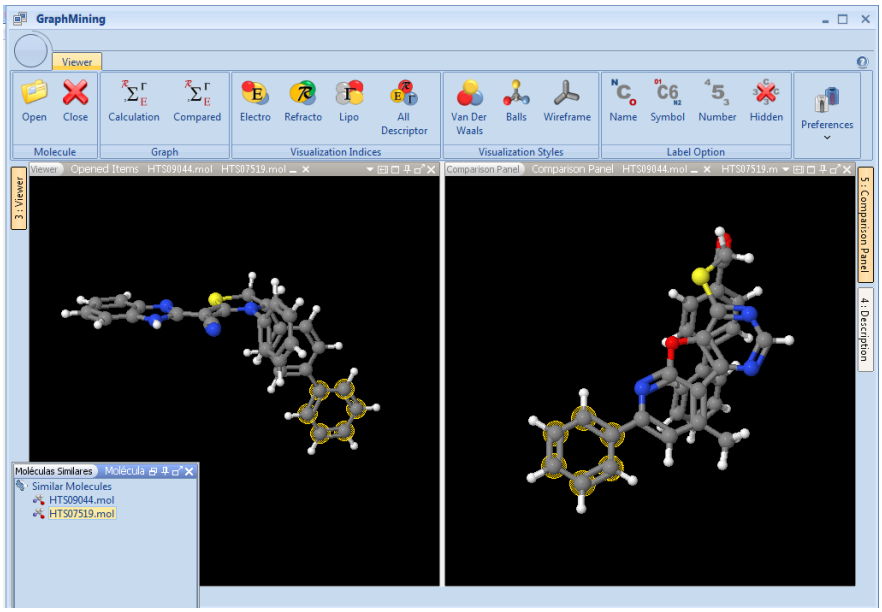
Historia de Usuario	
Número: 4	**Usuario:** Programador
Nombre historia: Comparar Moléculas	
Prioridad en el negocio: Alta	**Nivel de complejidad:** Alta
Tiempo de Estimación: 2	**Iteración asignada:** 1
Descripción: Después que el especialista carga la molécula, se pueden realizar comparaciones con las otras moléculas que se encuentran en el ensayo o con una en específico y mostrar en un árbol todas aquellas que tienen propiedades semejantes con la seleccionada.	
Observaciones: 	

Tabla 5 " HU Comparar Moléculas"

Historia de Usuario	
Número: 5	**Usuario:** Programador
Nombre historia: Visualizar Molécula Balls.	
Prioridad en el negocio: Media	**Nivel de complejidad:** Baja
Tiempo de Estimación: 1	**Iteración asignada:** 2

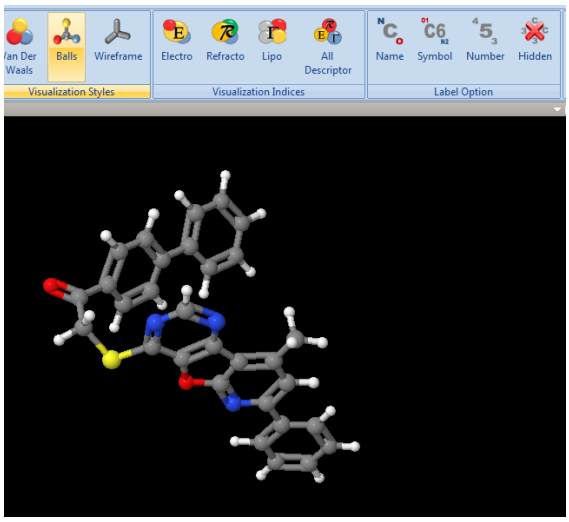
Descripción: Después que el especialista carga la molécula, la misma tiene diferentes estilos de visualización y en este caso, lo hace mediante Balls, que sería visualizando los enlaces como varillas y los átomos pequeñas esferas. Este estilo de visualización no refleja ni el tamaño ni la forma real de la molécula, pero permite distinguir claramente los diferentes átomos y enlaces y observar los cambios en el panel de visualización.
Observaciones: 

Tabla 6 " HU Visualizar Molécula Balls"

Historia de Usuario	
Número: 6	**Usuario:** Programador
Nombre historia: Visualizar Molécula Van der Waals	
Prioridad en el negocio: Baja	**Nivel de complejidad:** Baja
Tiempo de Estimación: 1	**Iteración asignada:** 2
Descripción: Después que el especialista carga la molécula, la misma tiene diferentes estilos de visualización y en este caso, lo hace mediante Van der Waals, que sería representando todos los átomos como esferas sólidas con sus radios de Van der Waals (es lo más semejante al volumen real ocupado por el átomo).En este caso se muestra el tamaño, la forma reales de la molécula, pero dificulta la percepción de su estructura y observar los cambios que presenta este estilo en el panel de visualización.	

Observaciones:

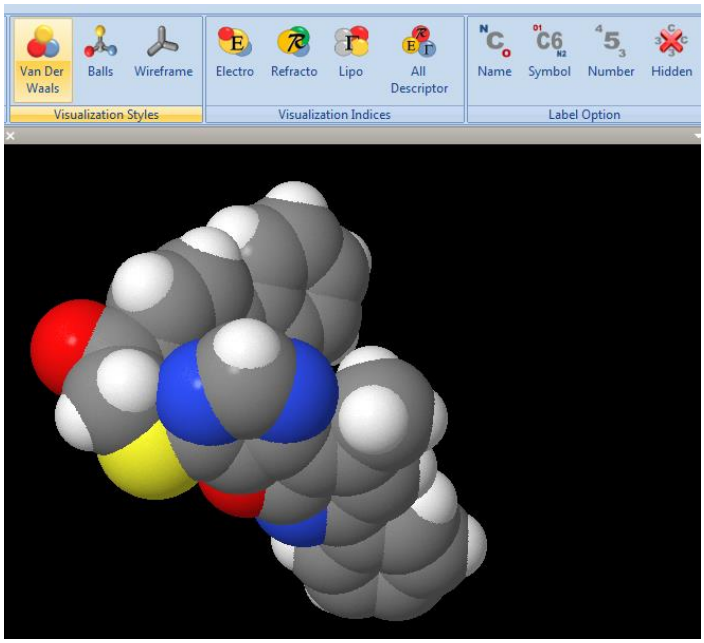


Tabla 7 " HU Visualizar Molécula Van Der Walls"

Historia de Usuario	
Número: 7	**Usuario:** Programador
Nombre historia: Visualizar Molécula Wireframe.	
Prioridad en el negocio: Media	**Nivel de complejidad:** Baja
Tiempo de Estimación: 1	**Iteración asignada:** 2
Descripción: Después que el especialista carga la molécula, la misma tiene diferentes estilos de visualización y en este caso, lo hace mediante Wireframe, que sería mostrando sólo los enlaces de la molécula y observar los cambios en el panel de visualización.	

Observaciones:

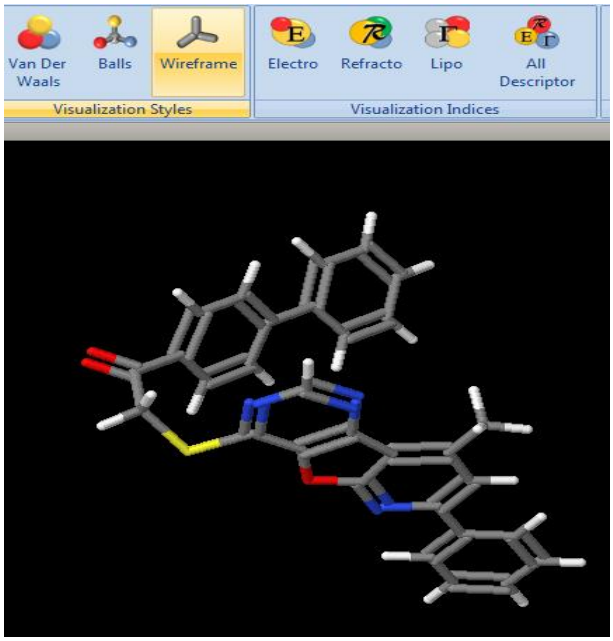


Tabla 8 " HU Visualizar Molécula Wireframe"

Historia de Usuario	
Número: 8	**Usuario:** Programador
Nombre historia: Visualizar Índice Electrotopográfico	
Prioridad en el negocio: Media	**Nivel de complejidad:** Baja
Tiempo de Estimación: 1	**Iteración asignada:** 2
Descripción: Después que el especialista carga la molécula, la misma puede ser visualizada en su estado simple o de acuerdo a las propiedades: electrotopográfico, refractotopográfico, lipotopográfico y aplicada las tres propiedades, este caso particular decidió visualizarla mediante la propiedad electrotopográfico.	

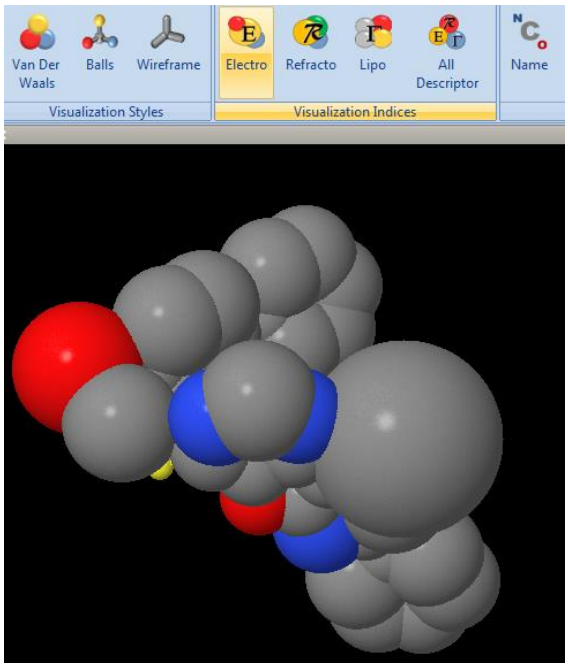
Observaciones:


Tabla 9 "HU Visualizar Índices Electrotopográfico"

Historia de Usuario	
Número: 9	**Usuario:** Programador
Nombre historia: Visualizar Índice Refractotopográfico.	
Prioridad en el negocio: Media	**Nivel de complejidad:** Media
Tiempo de Estimación: 1	**Iteración asignada:** 2
Descripción: Después que el especialista carga la molécula, la misma puede ser visualizada en su estado simple o de acuerdo a las propiedades: electrotopográfico, refractotopográfico, lipotopográfico y aplicada las tres propiedades, este caso particular decidió visualizarla mediante la propiedad refractotopográfico.	

Observaciones:

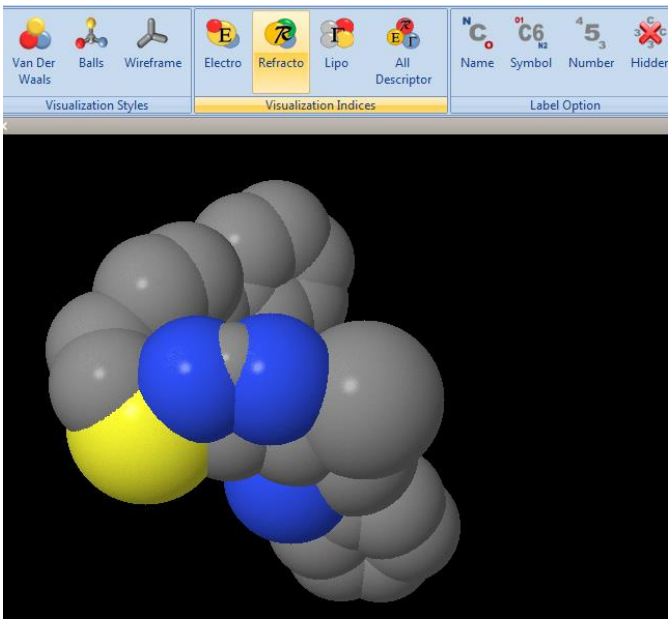


Tabla 10 "HU Visualizar Índices Electrotopográfico"

Historia de Usuario	
Número: 10	**Usuario:** Programador
Nombre historia: Visualizar Índice Lipotopográfico.	
Prioridad en el negocio: Media	**Nivel de complejidad:** Media
Tiempo de Estimación: 1	**Iteración asignada:** 2
Descripción: Después que el especialista carga la molécula, la misma puede ser visualizada en su estado simple o de acuerdo a los valores calculados de los índices: electrotopográfico, refractotopográfico, lipotopográfico y aplicada las tres propiedades, este caso particular decidió visualizarla mediante la propiedad lipotopográfico.	

Observaciones:

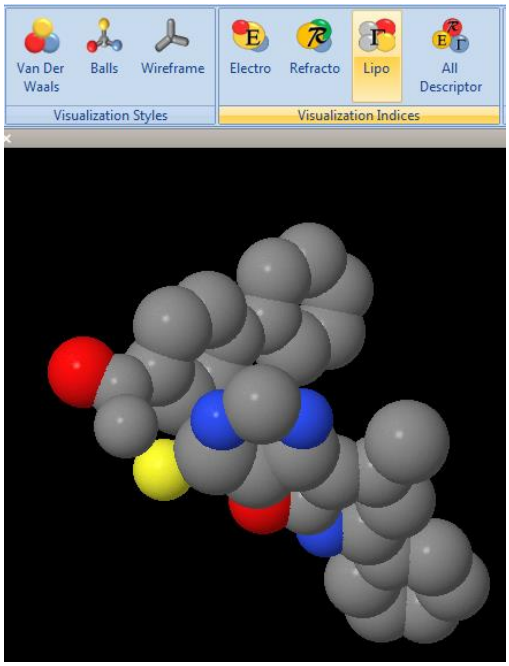


Tabla 11 "HU Visualizar Índices Lipotopográfico".

Historia de Usuario	
Número: 11	**Usuario:** Programador
Nombre historia: Visualizar Índices mixtos.	
Prioridad en el negocio: Media	**Nivel de complejidad:** Media
Tiempo de Estimación: 2	**Iteración asignada:** 2
Descripción: Después que el especialista carga la molécula, la misma puede ser visualizada en su estado simple o de acuerdo a las propiedades: electrotopográfico, refractotopográfico, lipotopográfico y aplicada las tres propiedades, este caso particular decidió visualizarla aplicando las tres propiedades	

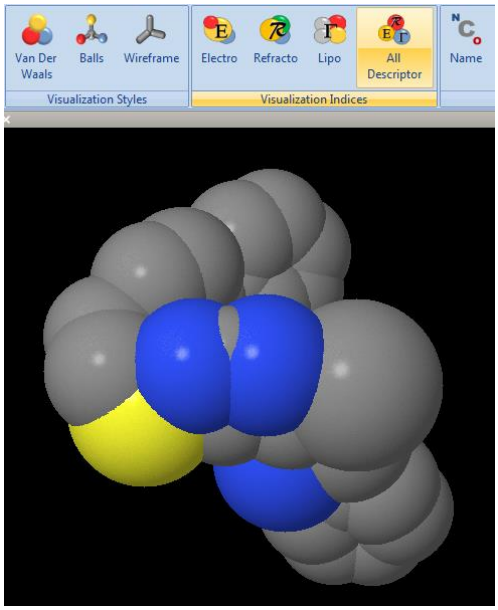
Observaciones:


Tabla 12 "HU Visualizar Índices Mixtos".

Historia de Usuario	
Número: 12	**Usuario:** Programador
Nombre historia: Insertar etiquetas a los átomos de la molécula.	
Prioridad en el negocio: Baja	**Nivel de complejidad:** Baja
Tiempo de Estimación: 1	**Iteración asignada:** 3
Descripción: Después que el especialista carga la molécula, tiene la opción de ponerle etiquetas a los átomos de la molécula, dichas etiquetas pueden ser el nombre, el símbolo y el número y observar los cambios en el panel de visualización.	

Observaciones:

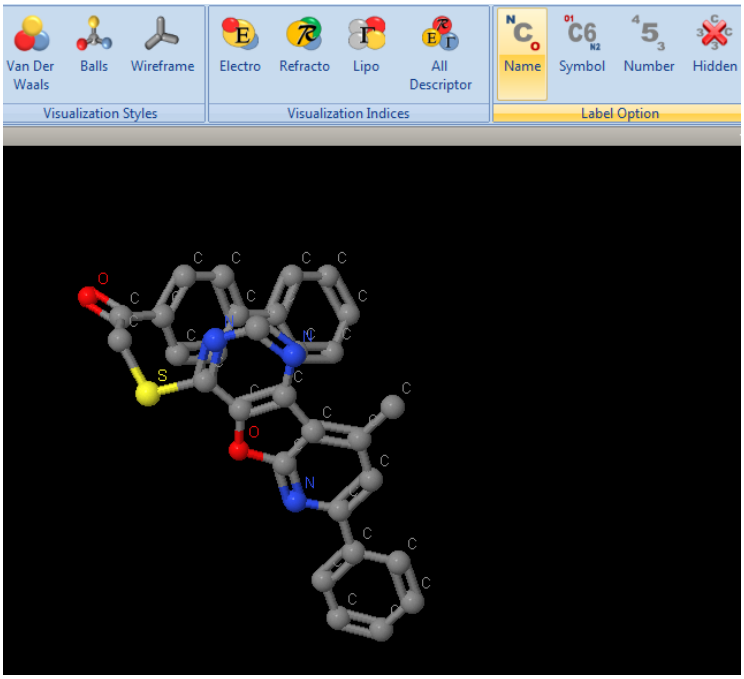


Tabla 13"HU Insertar etiquetas a los átomos de la molécula. "

Historia de Usuario	
Número: 13	**Usuario:** Programador
Nombre historia: Eliminar etiquetas a los átomos de la molécula.	
Prioridad en el negocio: Baja	**Nivel de complejidad:** Baja
Tiempo de Estimación: 1	**Iteración asignada:** 3
Descripción: Después que el especialista carga la molécula, y le pone etiquetas a los átomos de la molécula; tiene la opción de quitárselas y observar los cambios en el panel de visualización.	

Observaciones:

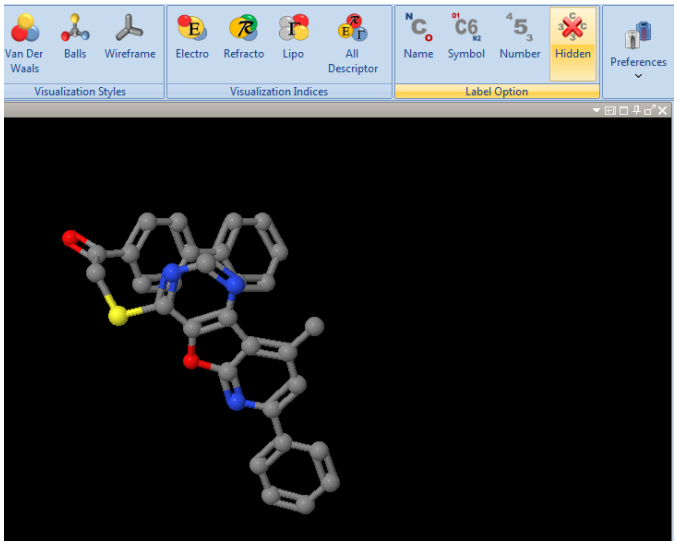


Tabla 14 "HU Eliminar etiquetas a los átomos de la molécula".

Historia de Usuario	
Número: 14	**Usuario:** Especialista
Nombre historia: Aumentar o Disminuir el tamaño de la molécula.	
Prioridad en el negocio: Baja	**Nivel de complejidad:** Baja
Tiempo de Estimación: 1	**Iteración asignada:** 3
Descripción: Después que el especialista carga la molécula, puede interactuar con la misma aumentando o disminuyendo su tamaño y observar cómo se visualizará en el panel.	

Observaciones:

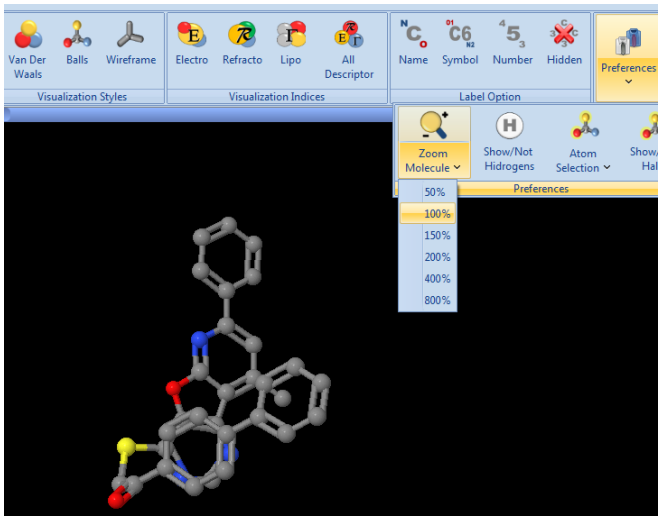


Tabla 15 "HU Aumentar o Disminuir el tamaño de la molécula."

Historia de Usuario	
Número: 15	**Usuario:** Programador
Nombre historia: Mostrar átomos de hidrógeno a la molécula.	
Prioridad en el negocio: Media	**Nivel de complejidad:** Media
Tiempo de Estimación: 1	**Iteración asignada:** 3
Descripción: Después que el especialista carga la molécula, puede interactuar con la misma poniéndole los átomos de hidrógenos a molécula y observar cómo se visualizará en el panel.	

Observaciones:

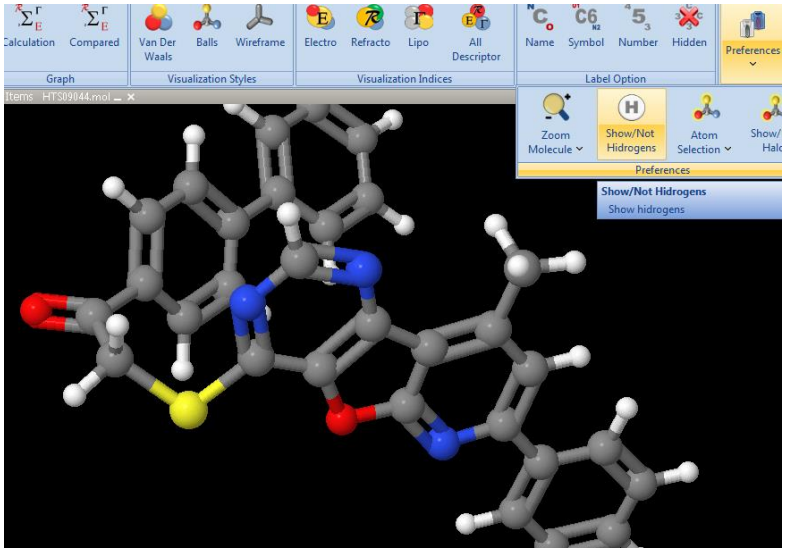


Tabla 16"HU Mostrar átomos de Hidrógeno a la molécula".

Historia de Usuario	
Número: 16	**Usuario:** Programador
Nombre historia: Ocultar átomos de hidrógeno a la molécula.	
Prioridad en el negocio: Baja	**Nivel de complejidad:** Baja
Tiempo de Estimación: 1	**Iteración asignada:** 3
Descripción: Después de que el especialista carga la molécula, puede interactuar con la misma ocultándole los átomos de hidrógenos a molécula y observar cómo se visualizará en el panel.	

Observaciones:

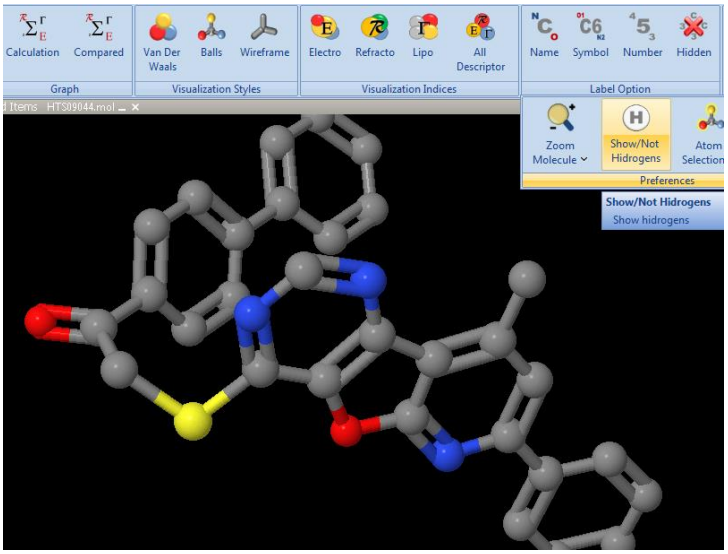


Tabla 17"Ocultar átomos de Hidrógeno a la molécula."

Historia de Usuario	
Número: 17	**Usuario:** Programador
Nombre historia: Seleccionar los átomos de la molécula.	
Prioridad en el negocio: Media	**Nivel de complejidad:** Baja
Tiempo de Estimación: 2	**Iteración asignada:** 3
Descripción: Después que el especialista carga la molécula, puede interactuar con la misma sombreando los átomos, centros descriptores y fragmentos que conforman la molécula y observar cómo se visualizará en el panel.	

Observaciones:

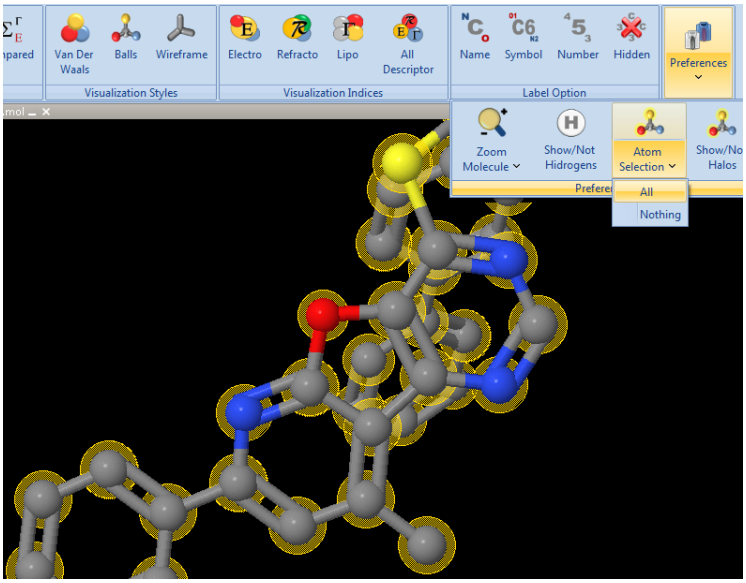


Tabla 18 "HU Seleccionar los átomos de la molécula."

Historia de Usuario	
Número: 18	**Usuario:** Programador
Nombre historia: Eliminar sombreado de la molécula.	
Prioridad en el negocio: Media	**Nivel de complejidad:** Baja
Tiempo de Estimación: 2	**Iteración asignada:** 3
Descripción: Después que el especialista carga la molécula, puede interactuar con la misma seleccionando las conexiones que conforman la molécula y observar cómo se visualizará en el panel.	

Observaciones:

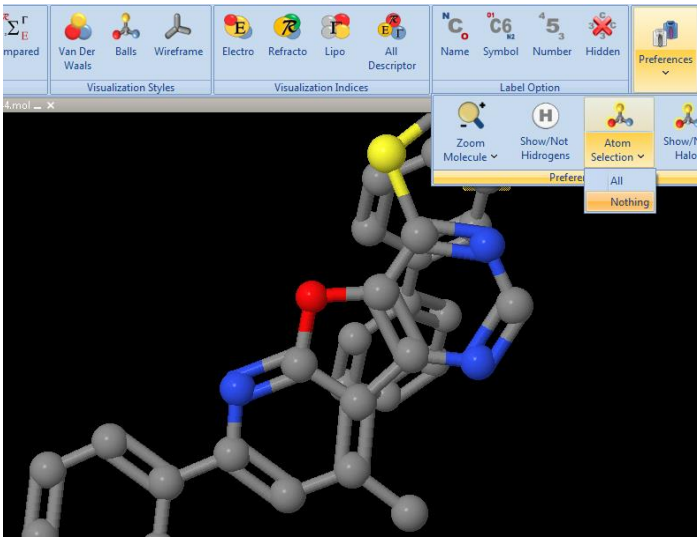


Tabla 19 "HU Eliminar sombreado de la molécula."

2.4.2 Estimación de esfuerzos por historias de usuario

Al estimar el esfuerzo que se necesita para realizar las historias de usuario, se mide la velocidad con que se desarrollará el software y el progreso existente en la implementación del mismo. A continuación se muestra la estimación del esfuerzo requerido para cada HU:

Historia de usuario	Puntos de estimación(semanas)
Abrir fichero que contiene información de la molécula.	1
Cerrar fichero que contiene información de la molécula.	1
Calcular los índices topográficos híbridos de la molécula.	2
Comparar Moléculas.	3
Visualizar Molécula Balls.	1
Visualizar Molécula Van Der Waals.	1
Visualizar Molécula Wireframe.	1
Visualizar Índices Electrotopográfico.	1
Visualizar Índices Refractotopográfico.	1
Visualizar Índices Lipotopográfico.	1

Visualizar Índices mixtos.	1
Insertar etiquetas a los átomos de la molécula.	1
Eliminar etiquetas a los átomos de la molécula.	1
Aumentar o Disminuir el tamaño de la molécula.	1
Mostrar átomos de Hidrógeno a la molécula.	1
Ocultar átomos de Hidrógeno a la molécula.	2
Seleccionar los átomos de la molécula.	1
Eliminar sombreado de la molécula**.**	2

Tabla 20 "Estimación de esfuerzos por HU"

2.4.3 Plan de iteraciones

Después de identificar y detallar las historias de usuario, se procede a la elaboración del plan de iteraciones, donde se definen qué historias de usuario serán implementadas en cada iteración. Para el desarrollo del presente software se estableció la realización de tres iteraciones:

Iteración 1: Esta iteración tiene como objetivo dar cumplimiento a las HU que se consideraron con una mayor importancia para el desarrollo de la herramienta. Al concluir dicha iteración se contó con todas las funcionalidades descritas en las HU 1, 2, 3 y 4 las cuales ayudan al trabajo con fichero, el cálculo de los índices topográficos híbridos y a la comparación entre las moléculas del ensayo.

Iteración 2: Esta iteración tiene como objetivo dar cumplimiento de las HU que tienen relación con la visualización y con la trabajo del visor molecular Jmol. Al concluir la iteración se debe haber cumplido con las funcionalidades descritas en las HU 5, 6, 7, 8, 9, 10 y 11.Una vez concluida esta iteración el especialista podrá realizar todos los estilos de visualización que desee ya sea estructuralmente o en cuanto a sus propiedades.

Iteración 3: Se implementan las historias de usuario con prioridad media en el negocio y las que se dan respuesta a la interacción del especialista con las moléculas.

2.4.4 Plan de duración de las iteraciones

El plan de duración de las iteraciones muestra cuánto se estima que dure cada iteración a realizar, además del orden en que serán implementadas las historias de usuario en cada una de las iteraciones. (Ver en la tabla 10)

Iteración	Historias de usuarios	Duración total (semanas)
1	• Abrir fichero que contiene información de la molécula. • Cerrar fichero que contiene información de la molécula. • Calcular los índices topográficos híbridos de la molécula. • Comparar Moléculas.	7
2	• Visualizar Molécula Balls. • Visualizar Molécula Van der Waals. • Visualizar Molécula Wireframe. • Visualizar Índice Electrotopográfico. • Visualizar Índice Refractotopográfico. • Visualizar Índice Lipotopográfico. • Visualizar Índices mixtos	7
3	• Insertar etiquetas a los átomos de la molécula. • Eliminar etiquetas a los átomos de la molécula. • Aumentar o Disminuir el tamaño de la molécula. • Mostrar átomos de hidrógeno a la molécula. • Ocultar átomos de hidrógeno a la molécula. • Seleccionar los átomos de la molécula. • Eliminar sombreado de la molécula	9

Tabla 21 "Plan de duración de las iteraciones"

2.5 Fase de diseño

Se realiza el diseño del sistema mediante las tarjetas CRC (Contenido, Responsabilidad, Colaboración) y se añade un diagrama de clases para facilitar la comprensión del mismo. Además, se describe el estándar de codificación que se usará en la implementación.

2.5.1Tarjetas CRC

Las tarjetas CRC son fichas en las que se escriben las responsabilidades (funcionalidades) de la clase y los objetos (clases que involucra) con los que colabora para llevar a cabo esas responsabilidades (Jeffries,2001). A continuación se detalla cada clase:

Tarjeta CRC	
Clase: Molecule	
Responsabilidad	**Colaboración**
Funciones que realiza son: • getAtomsSelected (Contener todos los átomos que conforman la molécula). • getListDescriptorCenter (Contenedor de los centros descriptores que conforman la molécula). • getListFragments (Contiene los fragmentos que conforman la molécula). • calcularIndicesTopologicos (Realiza el cálculo de los índices topológicos) • calcularIndicesTopograficos (Realiza el cálculo de los índices topográficos).	• MolQuery • ReducedGraph

• tanymotoSimilarity (Realiza el cálculo de distancia, según Tanymoto). • getSimilarFragments (Realiza la búsqueda de similitud de fragmentos). • getSimilarDescriptorCenters (Realiza la búsqueda de similitud de centros descriptores).	

Tabla 22Tarjeta CRC "Molecule".

Tarjeta CRC	
Clase: MolQuery	
Responsabilidad	**Colaboración**
Funciones que realiza son: • MolQuery (Realiza consulta, para tener en cuenta todos los centros descriptores que pertenecen a una molécula).	• Molecule

Tabla 23 Tarjeta CRC "MolQuery"

Tarjeta CRC	
Clase: Reduced Graph	
Responsabilidad	**Colaboración**
Funciones que realiza son: • ReducedGraph (Reduce el grafo y lo convierte en uno simple). • addCentroDescriptor (Adiciona los Centros Descriptores que conforman la molécula). • actualizarCantidades (Actualiza la cantidad de Centros Descriptores	• Molecule • Centro Descriptor

por lo que está compuesto la molécula).	

Tabla 24 Tarjeta CRC "Reduced Graph".

Tarjeta CRC	
Clase: Centro Descriptor	
Responsabilidad	**Colaboración**
Funciones que realiza son: • crearSecuencia Muestra la secuencia de centros descriptores por lo que está compuesto una molécula). • calcularCentroMasa (Calcula el centro de masa que tienen los centros descriptores). • calcularIndicesTopologico (Realiza el cálculo de los índices topológicos). • calcularIndicesTopograficos (Realiza el cálculo de los índices topográficos).	• Molecule • Secuencia • Tipo Centro Descriptor

Tabla 25 Tarjeta CRC "Centro Descriptor".

Tarjeta CRC	
Clase: Secuencia	
Responsabilidad	**Colaboración**
Funciones que realiza son: • GetCodigo (Tiene un código asignado para identificar cada molécula). • codificarAtomo (Contiene la secuencia de átomos por la cual está conformada una molécula).	• Centro Descriptor

Tabla 26 Tarjeta CRC "Secuencia".

Tarjeta CRC	
Clase: Tipo Centro Descriptor	
Responsabilidad	**Colaboración**
Funciones que realiza son: • Es un tipo de dato enumerativo. • TipoCentroDescriptor (Muestra información de todos los tipos de centros descriptores que puede tener una molécula).	• Centro Descriptor

Tabla 27 Tarjeta CRC "Tipo Centro Descriptor".

2.5.2 Diagrama de clases del diseño

Un diagrama de clases del diseño "representa las especificaciones de las clases e interfaces software" (Larman, 2006). La Figura muestra el diagrama de clases del diseño correspondiente al sistema desarrollado.

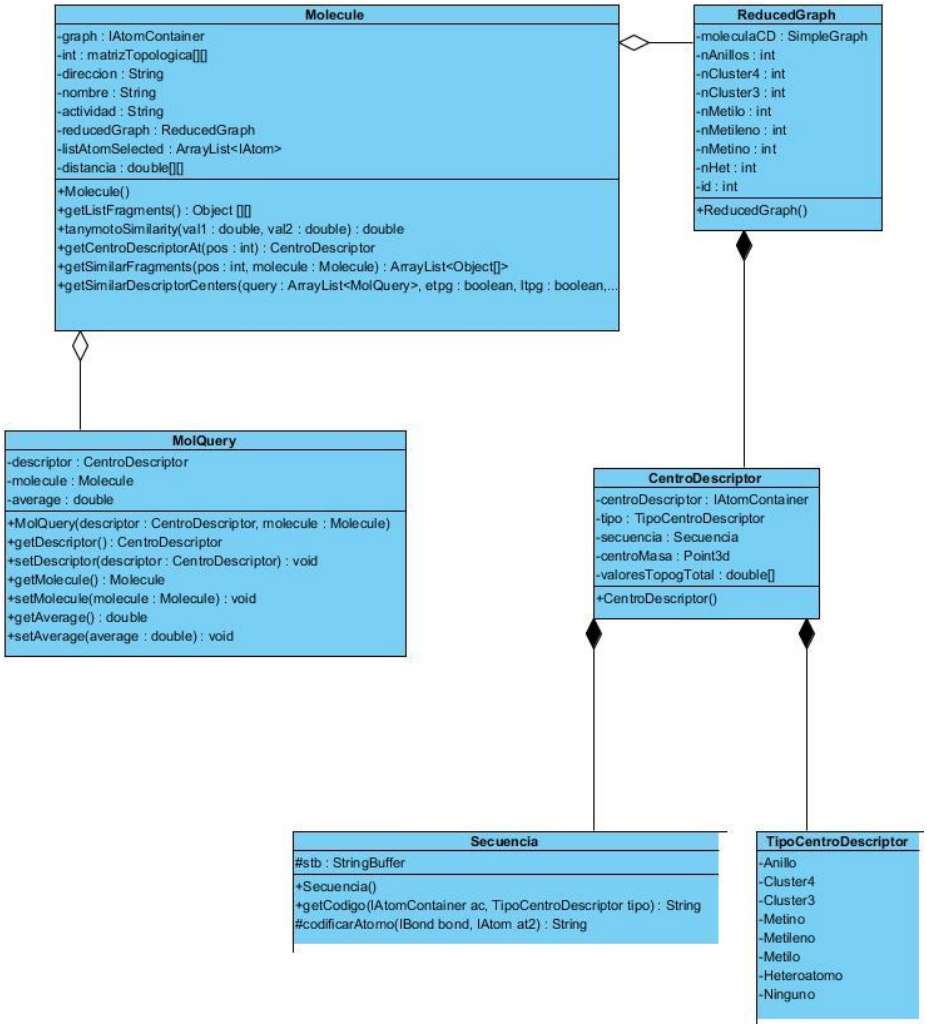


Figura 4 "Diagrama de clases del diseño"

Conclusiones del capítulo

Se lograron analizar y diseñar satisfactoriamente todos los requisitos funcionales previstos teniendo en cuenta las restricciones planteadas por el cliente y se describieron las clases que se utilizarán en la confección del mismo, con lo que se sientan las bases para la etapa de implementación.

CAPÍTULO 3 IMPLEMENTACIÓN Y PRUEBA

Introducción

El objetivo de este capítulo es implementar la propuesta de solución elaborada anteriormente y comprobar que el sistema desarrollado cumple con cada requerimiento planteado por el cliente. Para ello, se llevan a cabo las tareas de ingeniería y se realizan las pruebas de aceptación.

3.1 Fase de implementación

La metodología XP propone comenzar la implementación de la solución a partir de una arquitectura lo más flexible posible, con el propósito de que los desarrolladores puedan reestructurar el sistema sin cambiar su comportamiento y así remover duplicaciones de código, mejorar la comunicación, simplificar el código o agregar flexibilidad.

Además se realizan las tareas de ingeniería necesarias para materializar cada HU y se chequea el plan de iteraciones para saber si se ha afectado el mismo. El proceso de implementación se muestra en cinco iteraciones, como se propuso en el capítulo anterior (Jeffries, 2001).

3.1.1 Iteración 1

El objetivo de esta iteración es dar cumplimiento a las HU que se consideraron con una mayor importancia para el desarrollo de la herramienta. Al concluir dicha iteración se contó con todas las funcionalidades descritas en las HU 1, 2, 3 y 4 las cuales ayudan al trabajo con fichero, el cálculo de los índices topográficos híbridos y a la comparación entre las moléculas del ensayo.

Para ello se trazaron (2) tareas, que se indican a continuación:

- Tarea No.1: Incorporar la biblioteca CDK.
- Tarea No. 2: Integrar los algoritmos para el cálculo de los índices topográficos híbridos para átomos.

No. de la tarea: 1.	**Número de HU:** 3,4
Nombre de la tarea: Incorporar la biblioteca CDK.	
Tipo de tarea: Desarrollo	**Estimación:** 5 días
Programador responsable: Aylin María Rodríguez Jiménez	
Descripción: Incorporar al software la biblioteca CDK, para lograr realizar el cálculo de los índices topográficos híbridos para átomos.	

Tabla 28 "Tarea 1 Iteración 1"

No. de la tarea: 2	**Número de HU:** 3,4
Nombre de la tarea: Integrar los algoritmos para el cálculo de los índices topográficos híbridos para átomos.	
Tipo de tarea: Desarrollo	**Estimación:** 4 días
Programador responsable: Aylin María Rodríguez Jiménez	
Descripción: Programar el algoritmo para el cálculo de los índices topográficos híbridos para átomos. (El cual es implementado en la clase Molecule y se llama calcularIndicesTopograficos)	

Tabla 29 "Tarea 2 Iteración 1"

3.1.2 Iteración 2

El objetivo de esta iteración es darle cumplimiento a las HU que tienen relación con la visualización y con la trabajo del visor molecular Jmol. Al concluir la iteración se debe haber cumplido con las funcionalidades descritas en las HU 5, 6, 7, 8, 9, 10 y 11.Una vez concluida esta iteración el especialista podrá realizar todos los estilos de visualización que desee ya sea estructuralmente o en cuanto a sus propiedades.

Para ello se trazaron (2) tareas, que se indican a continuación:

- Incluir biblioteca Jmol.
- Integrar el algoritmo para la visualización de los grafos y subgrafos moleculares a partir de los índices topográficos híbridos.

No. de la tarea: 1	**Número de HU:** 4
Nombre de la tarea: Incluir biblioteca Jmol.	
Tipo de tarea: Desarrollo	**Estimación:** 4 días
Programador responsable: Aylin María Rodríguez Jiménez	
Descripción: Incorporar al software la biblioteca Jmol, para lograr el objetivo planteado en la iteración.	

Tabla 30 "Tarea 1 Iteración 2"

No. de la tarea: 2	**Número de HU:** 4 y 5
Nombre de la tarea: Integrar el algoritmo para la visualización de los grafos y subgrafos moleculares a partir de los índices topográficos híbridos.	
Tipo de tarea: Desarrollo	**Estimación:** 6 días
Programador responsable: Aylin María Rodríguez Jiménez	
Descripción: Programar los algoritmos para visualizar la molécula y otro aplicando los índices topográficos híbridos: electrotopográfico, refractotopográfico y lipotopográfico, el cual es implementado en la clase VisualizationIndices, haciendo uso de la biblioteca Jmol.	

Tabla 31 "Tarea 2 Iteración 2"

3.1.3 Iteración 3

Iteración 3: Se implementan las historias de usuario con prioridad media en el negocio y las que se dan respuesta a la interacción del especialista con las moléculas.

Para ello se trazaron (4) tareas, que se indican a continuación:

- Integrar el algoritmo de reducción de grafo.
- Incorporar el icono y la acción al XML del menú Ribbon
- Incorporar la acción para reducir el grafo, calcular los descriptores topográficos y mostrar resultados.
- Implementación del algoritmo de búsqueda de fragmentos similares utilizando una función de similitud.

No. de la tarea: 1	**Número de HU:** 6 y 7

Nombre de la tarea: Integrar el algoritmo de reducción de grafo.	
Tipo de tarea: Desarrollo	**Estimación:** 7 días
Programador responsable: Aylin María Rodríguez Jiménez	
Descripción: Programar el algoritmo de reducción del grafo en la clase Molecule (el cual se llama getReducedGraph).	

Tabla 32 "Tarea 1 Iteración 3"

No. de la tarea: 2	**Número de HU:** 8
Nombre de la tarea: Incorporar el icono y la acción al XML del menú Ribbon.	
Tipo de tarea: Desarrollo	**Estimación:** 4 días
Programador responsable: Aylin María Rodríguez Jiménez	
Descripción: Una vez Implementado el algoritmo de reducción de grafo y el cálculo de los descriptores topográficos incorporar esa acción al XML (el cual se encuentra en la carpeta ext.) del menú Ribbon.	

Tabla 33 "Tarea 2 Iteración 3"

No. de la tarea: 3	**Número de HU:** 8
Nombre de la tarea: Incorporar la acción para reducir el grafo, calcular los descriptores topográficos y mostrar resultados.	
Tipo de tarea: Desarrollo	**Estimación:** 4 días
Programador responsable: Aylin María Rodríguez Jiménez	
Descripción: Implementar el algoritmo en la vista (CalculationDescriptorViewsAcctionListener), la cual nos permitirá ver los resultados	

Tabla 34 "Tarea 3 Iteración 3"

No. de la tarea: 4	**Número de HU:** 8
Nombre de la tarea: Implementación el algoritmo de búsqueda de fragmentos similares utilizando una función de similitud.	
Tipo de tarea: Desarrollo	**Estimación:** 6 días
Programador responsable: Aylin María Rodríguez Jiménez	

Descripción: Implementar el algoritmo de búsqueda de fragmentos similares, utilizando Tanymoto como función de similitud; e incorporando las vistas asociadas a estas funcionalidades.

Tabla 35 "Tarea 4 Iteración 3"

3.2 Fase de pruebas

El proceso de pruebas es uno de los pilares fundamentales de la metodología XP, el cual ayuda al cliente a verificar y concretar las funcionalidades de las HU, por lo que favorece la comunicación entre el cliente y el equipo de desarrollo. Esta filosofía ayuda a identificar y corregir fallos u omisiones cometidas en las mismas, por lo que se reduce el número de errores no detectados así como el tiempo entre la introducción de éste en el sistema y su detección (Letelier,2003).

3.2.1 Pruebas de aceptación

Las pruebas de aceptación son realizadas por el propio cliente en compañía de uno de los representantes del equipo de desarrollo y se orientan a las funcionalidades del sistema. Su objetivo es comprobar, desde la perspectiva del usuario final, el cumplimiento de las especificaciones de la lista de reservas del producto. A continuación, aparecen las pruebas de aceptación realizadas a la solución propuesta:

Caso de Prueba de Aceptación	
Código: H3P1	**Número de HU:** 3
Nombre: Calcular los índices topográficos híbridos de la molécula.	
Descripción: Una vez cargada la molécula, se pueden realizar el cálculo de los índices topográficos y mostrarlo en una tabla.	
Condiciones de Ejecución: 1. El fichero.*mol debe estar cargado. 2. Debe seleccionarse la opción"*Calculation*".	
Entrada/Pasos de ejecución: 1. Verificar que la molécula está cargada. 2. Presionar el botón "*Calculation*". 3. Verificar que se activa una vista con los cálculos realizados.	
Resultado esperado: Una vez que se verifica que hay algún fichero cargado, y se desea realizar la operación de calcular se deben mostrar los resultados en una tabla.	
Evaluación de la prueba: Satisfactoria	

Tabla 36 "Prueba de aceptación No 1."

Caso de Prueba de Aceptación	
Código: H4P2	**Número de HU:** 4
Nombre: Comparar Moléculas.	
Descripción: Una vez cargada la molécula, se pueden comparar moléculas, de acuerdo a las propiedades de los índices topográficos y mostrar la comparación.	
Condiciones de Ejecución: 1. El fichero.*mol debe estar cargado. 2. Seleccionar la opción ”*Calculation*”. 3. Seleccionar las propiedades electrotopográfico, refractotopográfico, lipotopográfico o las tres propiedades por las cuales se desea comparar y el fragmento de la molécula. 4. Debe seleccionarse la opción ”*Compared*”.	
Entrada/Pasos de ejecución: 1. Verificar que la molécula está cargada. 2. Presionar el botón “*Calculation”.* 3. Buscar en la tabla, el fragmento de la molécula que se desea comparar. 4. Presionar el botón “*Compared*”. 5. Verificar que se activa un árbol con las moléculas similares a la cargada inicialmente. 6. Seleccionar del árbol un nuevo fichero con información de las moléculas que tiene propiedades semejantes y poderlas visualizarla en un nuevo panel y establecer las comparaciones.	
Resultado esperado: Una vez que se verifica que hay algún fichero cargado, y se desea realizar la operación de comparar se deben mostrar los resultados en la tabla de comparación.	
Evaluación de la prueba: Satisfactoria	

Tabla 37 "Prueba de aceptación No 2."

3.2.2 Pruebas unitarias

Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse. En este contexto de desarrollo evolutivo y de énfasis en pruebas constantes, la automatización para apoyar esta actividad es crucial (Letelier, 2003).

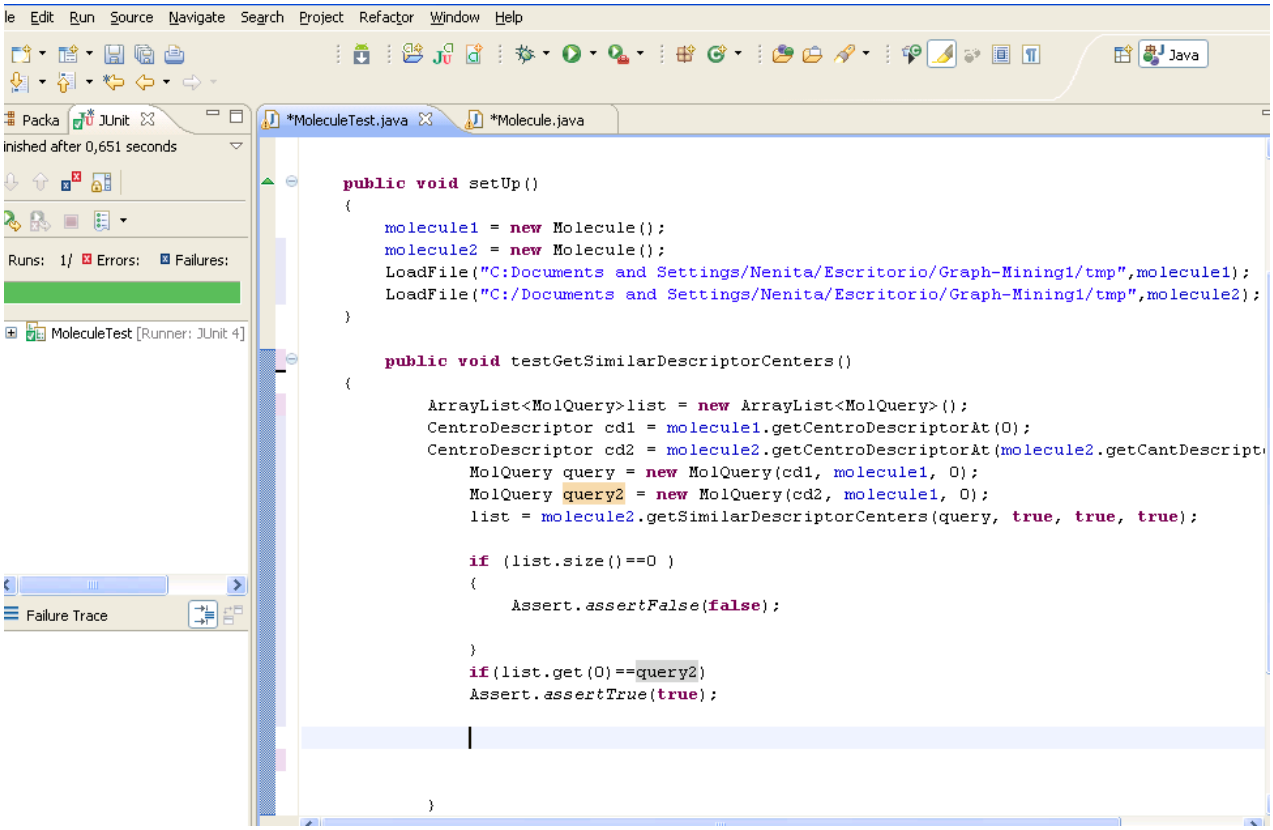


Figura 5 "Prueba unitaria realizada con JUnit".

Conclusiones del capítulo

Se implementaron los requisitos funcionales y se realizaron las pruebas de aceptación y unitarias, lo cual permitió validar el software de acuerdo a las acciones que debe realizar en el momento esperado.

CONCLUSIONES

1. Se implementó el algoritmo de búsqueda, cumple con las expectativas del usuario, facilitando un trabajo cómodo con su utilización.
2. Se logró desarrollar una aplicación informática para la búsqueda de similitud de fragmentos en grafos químicos, empleando descriptores topográficos híbridos.

RECOMENDACIONES

1. Optimizar el algoritmo de búsqueda propuesto, aplicando la función de similitud Tanymoto a N-Centros Descriptores, sin afectar la eficiencia del método de forma considerable.

BIBLIOGRAFÍA

A, Coulson. *High performance searching of biosequence databases.* s.l. : Trends Biotechnol, 1994. págs. 76-80.

Bensmail H, Haoudi A. *Postgenomics: proteomics and bioinformatics in cancer researc. .* Chile : J Biomed Biotechno, 2003.

Definition of a novel atomic index for QSAR: the refractotopological state. . **R.Carrasco-Velar, J.A.Padrón-Garcia, J.Gálvez.** 7, s.l. : J Pharm Pharmaceut Sci, 2004, págs. 19-26.

Dopazo J, Zanders E, Dragoni I, Amphlett G, Falciani F. . *Methods and approaches in the analysis of gene expression data. J Immunol Methods.* 2001. págs. 93-112.

Escalona Arranz, Julio César y Carrasco Velar, Ramón. *Introducción al diseño de fármacos. .* La Habana : Editorial Universitaria, 2008.

Faulon, Jean-Loup and Bender, Andreas. *Handbook of chemoinformatics algorithms.* London : CRC Press, 2010. págs. 37-61.

Friddle CJ, Koga T, Rubin EM, Bristow J. *Expression profiling reveals distinct sets of genes altered during induction and regression of cardiac hypertrophy.* USA : Proc Natl Acad Sc, 2000.

Medina-Franco, J. L, López Vallejo, F y Castillo. *Diseño de fármacos asistido por computadora.* s.l. : Edu. Quím, 2007. págs. 452-457. Vol. 17.

Thompson JD, Higgins DG, Gibson TJ. *CLUSTALW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting position specific gap penalties and weight matrix choice. s.l. : Nucleic Acids Res.* 1994. págs. 4673-80.

Walker, John M. . Bonn. *METHODS IN MOLECULAR B IOLOGY.* s.l. : Humana Press, 2010.

Steinbeck C., Hoppe C., Kuhn S., Floris M., Guha R., Willighagen. *"Recent Developments of the Chemistry Development Kit (CDK) - An Open-Source Java Library for Chemo- and Bioinformatics".* s.l. : Curr. Pharm. Des, 2006. 2174/13816120677758527.

Steinbeck C., Han Y., Kuhn S., Horlacher O., Luttmann E., Willighagen E.L., J. *"The Chemistry Development Kit (CDK): An Open-Source Java Library for Chemo- and Bioinformatics".* s.l. : Chem. Inf. Comput. Sci. , 2003 Mar-Apr. págs. 493-500 . 10.1021/ci025584.

Gago-Alonso, A. Minería de subgrafos conexos frecuentes en colecciones de grafos etiquetados. México : s.n., Enero 2010. PhD thesis. : Instituto Nacional de Astrofísica, .

Jmol. [En línea] http://jmol.sourceforge.net/.

Chemistry Development Kit. [En línea] [Citado el: 5 de 5 de 2013.] http://cdk.sourceforge.net/.

MyDoggy. [En línea] [Citado el: 4 de 26 de 2013.] :http://mydoggy.sourceforge.net/.

Javalobby. *MyDoggy 1.3.1 - My Java Docking Framework .* [En línea] [Citado el: 2 de 5 de 2013.] http://www.javalobby.org/java/forums/t102112.html.

Calderón, Marcela y Davis, Emilio. Swing, la solución actual de Java para crear GUIs. [En línea] 21 de 4 de 2013. http://users.dcc.uchile.cl/~lmateu/CC60H/Trabajos/edavis/swing.html.

Nuevas versiones de SwingX, Flamingo y Substance. [En línea] [Citado el: 22 de 4 de 2013.] http://www.sgoliver.net/blog/?p=451.

javaHispano. [En línea] [Citado el: 23 de 4 de 2013.] http://www.javahispano.org/contenidos/es/%20flamingo_4_0_rc/.

REFERENCIAS BIBLIOGRÁFICAS

Acosta Mendoza, Niusvel, Gago Alonso, Andres y Medina Pagola, Jose E. junio 2011. *Minería de subgrafos frecuentes utilizando cotejo inexacto.* junio 2011. 2072/6260.

Altschul SF, Boguski MS, Gish W, Wootton JC. 1994. *Issues in searching molecular.* 1994. págs. 119-29. Vol. 6.

Beck, K. TAddison Wesley. 2000. *Extreme Programming Explained. Embrace Change.* s.l. : Pearson Education, 2000.

Brudno M, Poliakov A, Salamov A, Cooper GM, Sidow A, Rubin EM, Solovyev V, Batzoglou S, Dubchak I . 2004. Automate whole-genome multiple alignment of rat, mouse, and human. s.l. : Suppl 1, 2004, págs. 685–692.

Chakrabarti, Deepayan. March 2006. *Graph Mining: Laws, Generators, and Algorithms.* New York, USA : s.n., March 2006. Vol. 38 .

CHEN C., YAN X, ZHU F., HAN J. 2007. gApprox: Mining frequent approximate patterns from a massive network. In: International Conference on Data Mining. IEEE Computer Society. 2007, págs. 445–450.

Conte, D., Foggia, P., Sansone, C., Vento, M. 2004. *Thirty years of graph matching in pattern recognition.* 2004. págs. 265-298.

Conte, D., Foggia, P., Sansone. 2004. *Thirty years of graph matching in pattern recognitio.* s.l. : IJPRAI 18, 2004. págs. 265–298.

Coutin, Adrian. 2006. La información de estructuras químicas y su implicación en el desarrollo de la bioinformática. [En línea] 2006. [Citado el: 10 de 3 de 2013.] http://www.congreso-info.cu/UserFiles/File/Info/Info97/Ponencias/110.pdf..

Eichinger, F., B¨ohm, K.In Aggarwal, C.C., Wang, H. 2010. *Software-Bug Localization with Graph Mining. Managing and Mining Graph Data. of Advances in Database Systems.* Verlag New York : s.n., 2010. Vol. 40 .

Fernández, Gerardo. 2002. Introducción a Extreme Programming. Introducción a Extreme . . [En línea] 2002. [Citado el: 11 de 4 de 2013.] http://www.infoab.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion-XP.pdf..

Gago-Alonso, A. Enero 2010. Minería de subgrafos conexos frecuentes en colecciones de grafos etiquetados. México : s.n., Enero 2010. PhD thesis. : Instituto Nacional de Astrofísica, .

HOLDER L.B., COOK D.J., BUNKE H. 1992. *Fuzzy substructure discovery. In: Proceedings of the ninth international workshop on Machine learning. Morgan Kaufmann Publishers Inc.* San Francisco, CA, USA : s.n., 1992. págs. 218–223. .

Jeffries, R., Anderson, A., Hendrickson, C.. Addison-Wesley. 2001. *"Extreme Programming Installed".* 2001.

JIA Y., ZHANG J., HUAN J. 2011. *An efficient graph-mining method for complicated and noisy data with realworld applications. Knowledge Information System.* 2011. págs. 423–447.

Jia, Y., Huan, J., Buhr, V., Zhang, J., Carayannopoulos, L. 2009. *Towards comprehensive structural motif mining for better fold annotation in the "twilight zone"of sequence dissimilarity. .* s.l. : BMC Bioinformatics , 2009.

Letelier, Patricio y Penadés, Ma. Carmen. 2003. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). [ed.] Conference on eXtreme Programming and Agile. Australia : s.n., 2003.

Medina-Franco, J. L, López Vallejo, F y Castillo. 2007. *Diseño de fármacos asistido por computadora.* s.l. : Edu. Quím, 2007. págs. 452-457. Vol. 17.

SONG Y., CHEN S.S. 2006. *Item sets based graph mining algorithm and application in genetic regulatory networks.* s.l. : Data Mining, IEEE International Conference, 2006. págs. 337–340.

XIAO Y., WU W., WANG W., HE Z. 2008. *Efficient algorithms for node disjoint subgraph homeomorphism determination. In: Proceedings of the 13th international conference on Database systems for advanced applications.* Berlin, Heidelberg : Springer-Verlag, 2008.

ZHANG S., YANG J. RAM ICSSDM. 2008. *Randomized approximate graph mining. In: Proceedings of the 20th International Conference on Scientific and Statistical Database Management.* 2008. págs. 187–203.

Zhang, S., Yang, J., Cheedella. 2007. *Monkey: Approximate graph mining based on spanning trees.* Los Alamitos, CA, USA : In IEEE 23rd International Conference on Data Engineerin, 2007.

ZOU Z., LI J., GAO H., ZHANG S. 2009. *Frequent subgraph pattern mining on uncertain graph data. In: Proceeding of the 18th ACM conference on Information and knowledge managemen.* New York, USA : s.n., 2009. págs. 583–592.

Larman, Craig. 2006. UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado. s.l. : Segunda. s.l.: Prentice Hall, 2006. pág. 590.