

Universidad del Bío-Bío
Facultad de Ciencias Empresariales
Departamento de Sistemas de Información



UNIVERSIDAD DEL BÍO-BÍO

**INCORPORACIÓN DE HERRAMIENTAS PARA ANÁLISIS
MASIVO DE DOCKING MOLECULAR A UN ENTORNO
GALAXY/COMPUTACIÓN DE ALTO RENDIMIENTO**

Felipe A. Gutiérrez Medina

Memoria para optar al título de Ingeniero Civil en Informática

Profesores Guías:

Prof. Patricio Galvez Galvez

Dr. Alexis Salas Burgos

Concepción - Chile

Octubre 2013

Agradecimientos

Quiero agradecer a todos quienes de alguna u otra forma hicieron posible la realización de este proyecto. En primer lugar quiero agradecer a Dios. Le doy las gracias por haber puesto en mi camino todos los elementos que pueden ayudar a alguien a desarrollarse como persona y profesional. Quiero dar las gracias, de todo corazón a las personas que siempre han estado a mi lado. Aquellos seres que lo han dado todo por mí y me han apoyado desde que nací, mi familia. Sin duda las personas más importantes en mi vida y lejos los mejores compañeros durante todo este tiempo. También quiero agradecer a Antonio, Edgardo, Enrique, Ernesto, Diego, Gianni y Ociel. Grandes amigos que me acompañaron en esta etapa y a lo largo de toda la carrera. Gracias por su compañía, ayuda, consejos y por todos los momentos que pasamos juntos. A Guiselle agradezco todo el apoyo durante este tiempo, sobre todo en último periodo. También agradecer a Danilo por su disposición y consejos al inicio de este trabajo. Por supuesto dar las gracias al Dr. Alexis Salas, quien desde el primer día desde la Práctica Profesional I hasta ahora, me dio todo su apoyo, confianza y ayuda no tan sólo en lo académico, sino que también en lo personal. Gracias por todas las gestiones que ha realizado para haber hecho posible este proyecto y gracias por todo lo que me ha enseñado como persona y profesional. Agradecer también al profesor Patricio Galvez por su disposición y haber confiado en este proyecto. Y agradecer a todas las personas de la Universidad Del Bío-Bío, todos sus trabajadores y colaboradores a todos ustedes muchas gracias por todo.

Antentamente,
Felipe A. Gutiérrez Medina.

Resumen

El docking molecular es una metodología ampliamente utilizada en el diseño racional de fármacos y permite la predicción de complejos de proteínas y ligandos, que son ordenados por su energía de interacción. Existen experimentos de docking donde se han calculado complejos con algunas moléculas contenidas en bibliotecas de hasta 100 moléculas y en bibliotecas masivas de varios cientos de miles a millones, este último necesita de cavidades acotadas en la estructura 3D del blanco terapéutico. Los programas utilizados son autodock/vina y DOCK6, el primero es utilizado principalmente para realizar docking de pequeñas bibliotecas y el último para docking de bibliotecas masivas. Hemos desarrollado algunos script en python que facilitan la realización de estas tareas de docking masivo de compuestos. Estas herramientas nos permiten el diseño de metodología complementarias y automatizadas de preparación de archivos para docking regulares y masivos, permitiendo su automatización y comparación. El análisis de los resultados de docking, la predictibilidad de átomos en los ligandos es fundamental para determinar farmacóforos (elementos moleculares mínimos de interacción), así como la identificación en el receptor de los aminoácidos involucrados en las interacciones y su contribución energética. La utilización de estas herramientas en python no es trivial por lo que nos propusimos disponer de este entorno en Galaxy, esta es una herramienta informática que ofrece el marco para crear una interfaz de usuario flexible, para permitir la ejecución de nuestras aplicaciones en un entorno Web. La implementación en Galaxy nos permitió una interfaz web de ejecución de las herramientas creadas en python en forma transparente para el usuario de un clúster de computadoras, realizar docking moleculares con compatibilidad entre interfaces, caracterizar los complejos y analizarlos en una forma rápida y detallada, entregando informe de los resultados que nos permite diseñar y proponer experimentos de validación de los complejos propuestos.

Abstract

The molecular docking methodology is a widely used in rational drug design and allows the prediction of protein-ligand complexes, which are sorted by their interaction energy. Docking experiments where there are calculated complex molecules contained in some libraries of up to 100 small molecules and libraries of hundreds of thousands to millions, the latter needs cavity bounded on the 3D structure of the therapeutic target. The programs used are AutoDock/Vina and DOCK6, the former is used mainly for docking of small libraries and one for docking massive library. We have developed some python scripts that facilitate the realization of these massive docking tasks compounds. These tools allow us complementary design methodology and automated docking preparing files for regular and massive, allowing automation and comparison. The analysis of the results of docking, the predictability of atoms in the ligands is essential for determining pharmacophores (minimum molecular interaction elements) as well as the identification of the amino acids involved in receptor interactions and energy contribution. The use of these tools in python is not trivial so we decided to have this environment Galaxy, this is a tool that provides the framework to create a flexible user interface, to allow the execution of our applications in a Web environment. Galaxy implementation in web interface allowed us to run python tools created transparently to the user of a computer cluster, perform molecular docking compatibility between dock software, characterize and analyze the complex in a fast and detailed, providing report of the results. This we allows to design and propose experiments to validate the proposed complex and increase the molecular drug developments.

Índice general

1. Introducción	1
2. Fundamentos	4
2.1. Conceptos básicos	4
2.1.1. Aminoácidos y proteínas	4
2.1.2. Receptor, ligando, sitio de unión y complejo	6
2.2. Biofísica de las interacciones proteína - ligando o proteína – proteína	6
2.2.1. ¿Cómo trabajan los fármacos?	6
2.2.2. Hipótesis de interacciones proteína ligando	9
2.2.3. Afinidad del ligando	10
2.2.4. Componentes de ΔG	12
2.3. Diseño de fármacos basado en bio-estructuras.	15
2.3.1. Droga anti-influenza	16
2.4. Diseño de drogas basado en ligando.	17
2.5. ¿Cómo se realiza el diseño racional de fármacos?	19
2.5.1. Relevancia del docking molecular	19
2.5.2. Docking molecular	19
2.5.3. Superficies y cavidades	23
2.6. Algoritmos de docking molecular.	26
2.6.1. Métodos de búsqueda.	26
2.6.2. Vina v/s DOCK6	28
2.7. Galaxy	29
2.7.1. Historial, flujo de trabajo, conjunto de datos y páginas	30

2.7.2.	Disponibilidad de Galaxy	31
3.	Análisis del problema	32
3.1.	Planteamiento del problema	33
3.1.1.	Situación actual	34
3.1.2.	Objetivo general	35
3.1.3.	Objetivos específicos	35
3.2.	Definición de requisitos	35
3.2.1.	Requisitos funcionales	36
3.2.2.	Requisitos no funcionales	37
3.2.3.	Diagrama de Flujo de Datos (DFD)	39
3.2.4.	Casos de Uso	41
3.3.	Estudio de viabilidad	44
3.3.1.	Viabilidad operacional	45
3.3.2.	Viabilidad técnica	45
3.3.3.	Viabilidad económica	46
4.	Solución	51
4.1.	Flujo de ejecución de los programas docking a través de Galaxy	51
4.2.	Instancia de Galaxy	53
4.2.1.	Añadir los programas a Galaxy	54
4.2.2.	Parámetros de los programas de docking.	56
4.2.3.	Interfaz de usuario y navegación	56
4.2.4.	Visualización de estructuras 3D.	58
4.3.	Software	60
4.3.1.	Dependencias de software	61
4.3.2.	Requisitos de hardware	61
5.	Pruebas	63
5.1.	Pruebas Unitarias	63
5.1.1.	Párametros de entrada	63
5.1.2.	Generar archivos docking.	64

5.1.3. Generar gráficos	65
5.1.4. Visualizar estructura 3D.	65
5.2. Pruebas de integración	66
5.2.1. Inicialización de programas en Galaxy	66
5.2.2. Ejecución de programas desde Galaxy	67
6. Conclusión	68
7. Trabajos futuros y derivados	70
A. Archivo de configuración de los programas	71
A.0.3. Código XML Programa1	71
A.0.4. Código XML Programa2	72
A.1. Interfaz de usuario	74
A.1.1. Interfaz de usuario programa1	74
A.1.2. Interfaz de usuario programa2	75
A.1.3. Historial	76
B. Bibliografía	77

Índice de figuras

2.1. Estructuras de distintas clases de aminoácidos.	5
2.2. Ejemplo de fármacos, estructuras reales del PDB.	8
2.3. Hipótesis “cerradura y llave”(1894) que luego evolucionó a la hipótesis de “encaje inducido”(1958)	10
2.4. Equivalencias del ΔG	11
2.5. Variaciones de energía libre	11
2.6. Cambios conformacionales en ligando y proteína	12
2.7. (S)-Glutamato unido a iGluR	13
2.8. Puentes de hidrogeno vs interacción hidrofóbica	14
2.9. Curva de energia de vdW	15
2.10. Celecoxid unido a COX-2	15
2.11. Oseltamivir unido a Neuraminidasa	16
2.12. La Neuraminidasa, Flu Española, Flu Aviar y el virus	17
2.13. Identificación de un farmacóforo	18
2.14. AZT, primer fármaco utilizado contra el VIH	19
2.15. Docking proteína-proteína	20
2.16. Docking de una molécula pequeña	21
2.17. Diferentes conformaciones para un ligando	22
2.18. Diferentes poses para un ligando	22
2.19. Diferentes configuraciones para un ligando	23
2.20. Docking Ciego. Una grilla para la estructura completa. Las moléculas se prueban en todas las conformaciones.	24
2.21. Docking dirigido. Una grilla localizada en una cavidad específica.	24

2.22. Centros de átomos de un ligando y superficie de van der Waals	25
2.23. Agrupación de Esferas de Kuntz et al	25
2.24. Formas Alfa	26
2.25. Gráfico que representa los Software de docking más utilizados en publicaciones	29
3.1. Diagrama de Flujo a nivel de contexto	39
3.2. Diagrama de Flujo a nivel de superior	40
3.3. Diagrama de Casos de Uso	41
4.1. Diagra flujo de la ejecución de programas a través de Galaxy	53
4.2. Esquema especificación de Interfaz.	57
4.3. Página de inicio del Galaxy.	58
4.4. Visualización 3D estructura y todas las grillas alineadas.	59
4.5. Visualización 3D de la selección de una grilla.	60
A.1. Interfaz para el ingreso de parámetros y la ejecución del programa1.	74
A.2. Interfaz para el ingreso de parámetros y la ejecución del programa2.	75
A.3. Historial con datos programa2	76

Índice de cuadros

3.1. Especificación de caso de uso CU01	42
3.2. Especificación de caso de uso CU02	42
3.3. Especificación de caso de uso CU03	43
3.4. Especificación de caso de uso CU04	43
3.5. Especificación de caso de uso CU05	44
3.6. Especificación de caso de uso CU06	44
3.7. Listado de recursos como inversión inicial	47
3.8. Listado de egresos anual del Proyecto	47
3.9. Listado de ingresos anual del Proyecto	48
3.10. Flujo de caja en un periodo de dos años	48
4.1. Configuración básica máquina virtual en etapa de desarrollo	54
4.2. Parámetros de entrada para el <i>Programa1</i>	56
4.3. Configuración básica para la utilización de aplicaciones Jmol	59
4.4. Dependencias de software	61
4.5. Requisitos de hardware. Clúster de computadoras	62

Capítulo 1

Introducción

La computación es una área extensa que interacciona con diversos campos de la ciencia. En este proyecto el interés lo centraremos en el papel que juega la computación en el campo del "Docking Molecular". El docking molecular es una metodología ampliamente utilizada en el diseño racional de fármacos, acota los candidatos a fármacos a decenas desde millones de moléculas disponibles en las bases de datos. Entonces, nos permite la identificación de interacciones líderes que son utilizadas en la construcción de molécula que interaccionan con un receptor modulando su actividad. El docking molecular es una de las principales herramientas para la caracterización y predicción de interacciones moleculares, a medida que se generan modelos más complejos la exactitud de estos aumenta y la predictibilidad se hace mayor.

No obstante, los formatos utilizados en los cálculos de docking molecular no son homogéneos y necesitan de una estandarización para realizar experimentos con programas que implementan metodologías análogas, pero no similares y sin poder realizar análisis comparativos.

La preparación de los experimentos de docking necesitan del cálculo de cavidades el cual puede ser estandarizado y asegurar la reproductibilidad de los cálculos desarrollados. Algo similar sucede con los parámetros de docking que podrían ser almacenados para ensayos posteriores.

Este es un campo que ha crecido mucho en los últimos años, algunos de los contribuyentes son David S. Goodsell y Irwin Kuntz creadores de los programas AutoDock y DOCK respectivamente, cuyo objetivo es la predicción de un complejo molecular.

Si bien existe una madurez en el cálculo de docking masivo, la distribución del cálculo no ha sido exhaustiva y existen nuevos protocolos diseñados que necesitan implementación.

Herramientas de análisis avanzado de los resultados como cálculos de farmacóforos (elementos moleculares mínimos de interacción), determinación de agrupamiento de residuos con teoría de grafos y evaluación electrostática posterior de la energía de complejos aún no han sido implementados.

Nosotros hemos desarrollado pequeños programas que son prototipo de los cálculos que se pueden implementar y desarrollar un entorno de subversiones con GIT, para profesionalizar la generación de nuevas subversiones y manejo de pruebas de implementación.

Las herramientas de docking han sido divididas en 4 scripts escritos en python que realizan las siguientes tareas:

- Preparar los ligandos y moléculas.
- Realización de los docking.
- Análisis con teoría de grafos y de mutaciones sitio-dirigidas.
- Análisis electrostático de interacciones.

Por otra parte el entorno de accesibilidad web por medio de Galaxy, asegura la usabilidad de las herramientas , la generación de protocolos y de análisis por personas con menor experiencia en la ejecución de los programas, estandarizando los resultados y generando un entorno de aplicabilidad y diseño racional de fármacos. La implementación de este proyecto nos permitirá homogenizar los cálculos de docking molecular, implementar una interfaz web para la creación de nuevos protocolos, desarrollar nuevas herramientas de análisis de docking molecular para realizar experimentos de validación de resultados por medio de otras técnicas de análisis de complejos. Implementar los prototipos desarrollados en forma robusta para ser utilizado por los usuarios sin habilidades informáticas a través de una interfaz de usuario que distribuya los trabajos en el clúster y monitoree la carga del sistema.

Este proyecto es parte del proyecto PharmScan Transporter, dirigido por el Dr. Alexis Salas del Departamento de Farmacología de la Universidad De Concepción cuyo objetivo es desarrollar una plataforma de negocios para la identificación de moléculas pequeñas con actividad biológica sobre proteínas transportadoras asociadas a enfermedades humanas crónicas de alta prevalencias. PharmScan Transporter es una plataforma accesible vía web para la realización y análisis de docking molecular masivo aplicado a proteínas transportadoras. Esta plataforma involucrará una manera

atractiva de acceder a la información calculada a través de acceso por la red con sistemas de cotización en línea y pagos de forma similar.

Para el autor ha resultado un desafío no menor aceptar este proyecto, ha sido un privilegio, pero a la vez una gran responsabilidad. Las principales motivaciones fueron haber trabajado antes en los laboratorios desarrollando las prácticas profesionales I y II (2011 - 2012). Esto permitió al autor conocimientos en varias herramientas que se utilizaran en este proyecto, además de una introducción a lo que es la Bioinformática y Biología Computacional.

Capítulo 2

Fundamentos

Este capítulo tiene como objetivo introducir al lector al ambiente en el cual se desarrollan los experimentos y simulaciones en el Laboratorio de Diseño Racional de Fármacos de la Universidad de Concepción. Es importante entender la base de este proyecto. Primero, se introduce al lector con algunas definiciones básicas de conceptos que serán mencionados durante este trabajo. Luego, se habla de los diseños de fármacos, en qué consiste, como trabajan, indicando su importancia y por qué es necesario la creación de mejores y nuevos medicamentos. Posteriormente, se define con mayor detalle el Docking Molecular: métodos de búsqueda, su objetivo, y el software comúnmente utilizado para realizar docking. Finalmente, se habla de la herramienta que se utilizará en este trabajo. Galaxy, se define y se describen sus principales componentes.

2.1. Conceptos básicos

Para comprender mejor esta sección es necesario, definir algunos conceptos que serán frecuentemente mencionados a lo largo de este documento:

2.1.1. Aminoácidos y proteínas

Los aminoácidos son la unidad básica de una proteína. Éstas son moléculas orgánicas compuestas por un grupo amino ($-NH_2$) y un grupo carboxilo ($-COOH$) además de un grupo R (resto de la molécula). Son 20 aminoácidos diferentes los que son componentes esenciales de las proteínas. De acuerdo con su capacidad para interactuar con el agua los aminoácidos se pueden clasificar en

cuatro clases: apolares, polares, ácidos y básicos (ver figura 2.1).

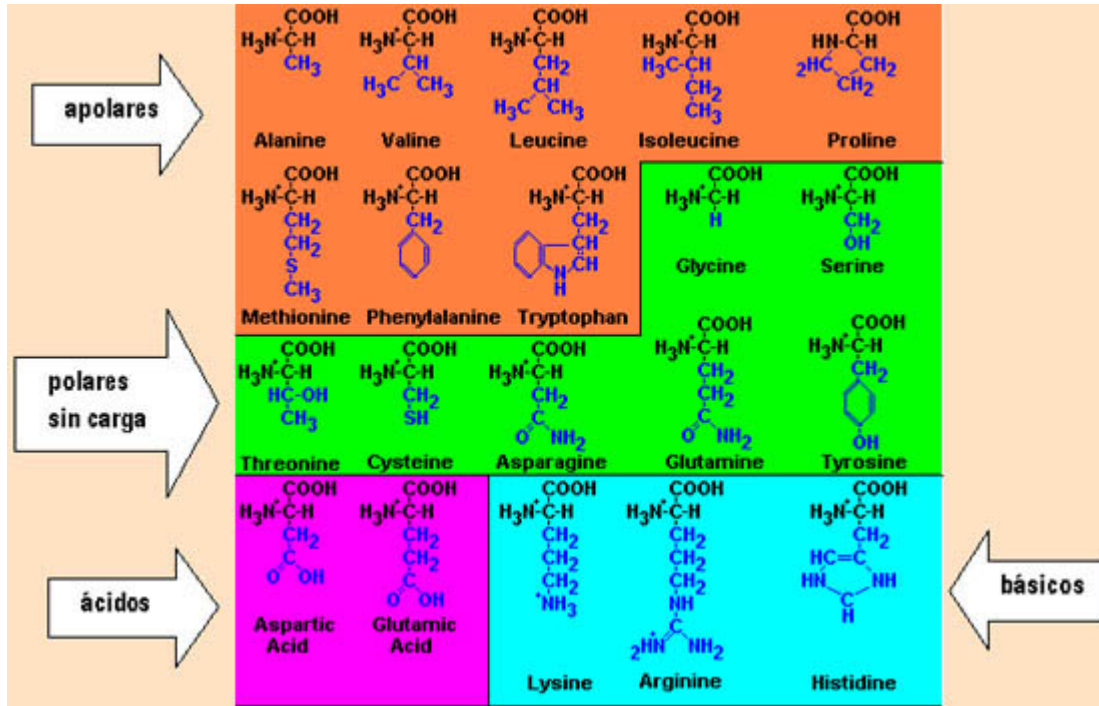


Figura 2.1: Estructuras de distintas clases de aminoácidos.

La imagen 2.1 muestra ejemplos de algunos aminoácidos estándar. En la parte izquierda de cada uno se aprecia el grupo amino, a la derecha el grupo carboxilo y en celeste se destaca una cadena lateral o radical R de cada aminoácido.

Las proteínas son macromoléculas formadas por cadenas lineales de aminoácidos. Son las encargadas de realizar la mayor parte de las funciones en las células de los seres vivos, pues están involucradas en todos los procesos biológicos que ocurren en el cuerpo. Participan en la regulación metabólica, el transporte, la defensa y la catálisis de reacciones químicas, etc.

Los bioquímicos diferencian cuatro niveles de organización estructural en las proteínas. En la estructura primaria, los residuos de aminoácido están conectados mediante enlaces peptídicos. La estructura secundaria de los polipéptidos está estabilizada por enlaces de hidrógeno. La estructura terciaria es la conformación tridimensional única que asume una proteína debido a las interacciones entre las cadenas laterales de los aminoácidos. Varios tipos de interacciones estabilizan la estructura terciaria: el efecto hidrófobo, las interacciones electrostáticas, los enlaces de hidrógeno y determinados enlaces covalentes. Las proteínas que constan de dos o más polipéptidos poseen estructura

cuaternaria. Las subunidades se mantienen unidas por enlaces no covalentes y covalentes. La actividad biológica de las proteínas complejas con varias subunidades con frecuencia se regula por interacciones alostéricas en las que ligandos pequeños se unen a la proteína. Los cambios de la actividad de la proteína se deben a cambios de las interacciones entre las subunidades de la proteína. Los efectores pueden aumentar o disminuir la función de una proteína. (Mckke & Mckee, 2010, 3^a edición). Las estructuras de proteínas y ligandos tridimensionales determinadas están depositadas en la base de datos Protein Data Bank (PDB).

2.1.2. Receptor, ligando, sitio de unión y complejo

Cuando hablamos de Docking Molecular debemos manejar algunos conceptos. Más adelante se dedica una sección completa para definir y destacar la importancia del Docking Molecular. Por ahora es importante saber que:

Receptor es usualmente una proteína que contiene el sitio activo de unión.

Ligando es la estructura que interactúa con el sitio de unión, puede ser una molécula pequeña o una proteína.

Sitio de unión es el conjunto de aminoácidos que interactúa físicamente con el ligando.

Complejo es la estructura resultante de la unión receptor - ligando.

2.2. Biofísica de las interacciones proteína - ligando o proteína – proteína

2.2.1. ¿Cómo trabajan los fármacos?

Todos los fármacos a nivel molecular, trabajan a través de la interacción de una molécula pequeña, con una molécula mayor (“Blanco terapéutico”). La interacción que se produce es muy particular, hay energías súper específicas, donde el acople de las moléculas es muy preciso. Es esta búsqueda de complejos la que realizamos con el docking molecular; podemos tener millones de moléculas, y a través del docking las podemos filtrar por energías, podemos predecir que moléculas estarían interactuando. Las moléculas pequeñas interactúan y modulan la actividad del blanco terapéutico, que es una proteína.

A continuación se presentan ejemplos de fármacos que actúan en:

- **Antibióticos y antivirales:** Son venenos específicos, ellos necesitan eliminar a un agente patogénico sin afectar al paciente, generalmente estas bloquean maquinaria necesaria para la sobrevivencia o multiplicación del agente.

1. D-alanil-D-alanina carboxypeptidasea con penicilina (1pwc).

2. Proteasa del VIH con saquinavir (1hxb).

- **Quimioterapia contra el cáncer:** Células cancerígenas crecen descontroladamente y los fármacos van dirigidos contra blancos que actúan en célula creciendo activamente como bleomicina, que actúa clivando la cadena de ADN y paclitaxel (Taxol) que se une a tubulina.

3. DNA con bleomicina (1mxk).

4. Tubulin con taxol (1jff).

- **Efecto sobre proteínas de señalización:** Loratidina (Claritin) es usada para el tratamiento de alergias debido a que bloquea el receptor de histamina; losartan (Cozaar) es usado para el tratamiento de la alta presión sanguínea, debido a que bloquea el receptor de angiotensina II; y carazolol es uno de los bloqueadores beta que se une al receptor adrenérgico. Aspirina bloquea el dolor inhibiendo la enzima COX, que media la liberación de moléculas de prostaglandinas.

5. Receptor adrenérgico con carazolol (2rh1).

6. Prostaglandina H2 sintasa con aspirina (1pth).

- **Estilo de vida:** El orlistat (Xenical o alli) bloquea la acción de la lipasa pancreática, entonces reduce la cantidad de grasa absorbida desde la comida. La atorvastatin (Lipitor) y simvastatina (Zocor) bajan el colesterol por bloqueo de la acción de la reductasa de HMG-CoA.

7. Lipasa Pancreática con un inhibidor (similar a orlistat) (1lpb).

8. Reductasa de HMG-CoA con atorvastatina (1hwk).

- **Metabolismo de drogas:** Enzimas como el citocromo P450 continuamente buscan drogas y las destruyen. Esto es importante para evitar intoxicaciones de la dieta, medioambiente o medicación.

9. Citocromo P450 3A4 con eritromicina (2j0d).

Para todos los fármacos, mencionados en estos ejemplos, se conoce específicamente donde interaccionan. La figura 2.2 muestra las estructuras reales, para cada uno de ellos.



Figura 2.2: Ejemplo de fármacos, estructuras reales del PDB.

Cada estructura tiene un código de cuatro caracteres. Ejemplo: 9. Citocromo P450 3A4 con

eritromicina (**2j0d**). Solo con este código y utilizando algún visor 3D, tenemos acceso al PDB (Protein Data Bank) (<http://www.rcsb.org>).

La identificación y diseño de nuevos fármacos es muy importante, debido a que nos encontramos con una serie de dificultades que hacen que su acción no perdure en el tiempo, tales como: mutación de virus que provocan mayor resistencia a antibióticos, nuevos tipos de cáncer, necesidad de fármacos con mayor eficiencia, fármacos con menor toxicidad, nuevos fármacos que modulen ciertas condiciones en el organismo y la necesidad constante de la industria farmacéutica por innovar. Por ejemplo: en el caso de los antibióticos y antivirales, se producen resistencias, debido a: un cambio en la bacteria, alguna mutación que hace que el fármaco ya no la dañe, algún cambio en una enzima; que es capaz de degradar al fármaco, etc.

2.2.2. Hipótesis de interacciones proteína ligando

Todos los fármacos son moléculas pequeñas que interaccionan con un blanco, esta interacción es determinada por fuerzas intermoleculares que guían la unión entre ligando y receptor. Estas fuerzas hacen una unión específica, la cual es factible a una concentración determinada. La fuerza de unión está relacionada con una concentración de la molécula pequeña.

Estas interacciones de moléculas y macromoléculas se basan en una hipótesis bastante antigua, que es la “cerradura y llave” sugerida por Emil Fisher en 1894, quien se refería a una enzima como una cerradura y al sustrato como una llave que encaja perfecto en dicha cerradura (ver figura 2.3). Esta hipótesis fue evolucionando de una cerradura y llave que calzaba de forma perfecta, al “encaje inducido” sugerido por Koshland en 1958, en donde ya no eran tan complementarios los sitios de interacción, pero ambos con la capacidad de generar cambios conformacionales locales que finalmente hacían que la complementariedad de ambas superficies fuera perfecta (ver imagen 2.3).

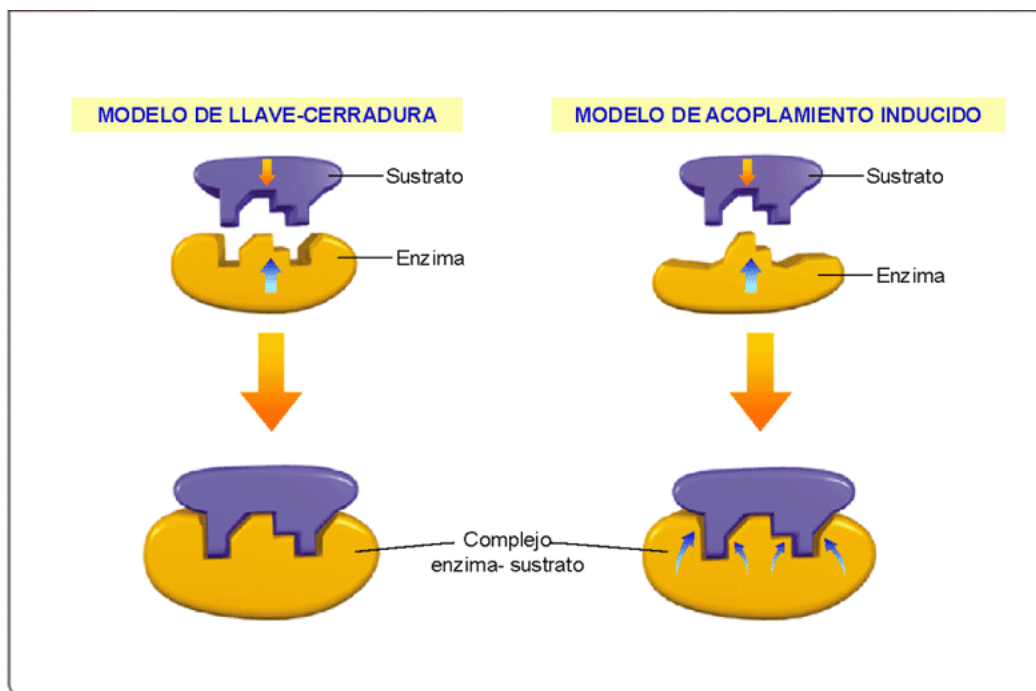
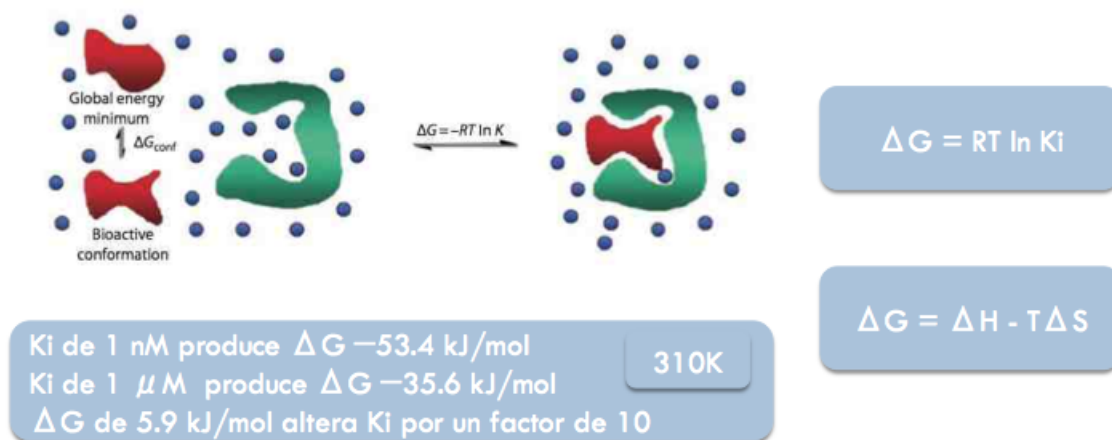


Figura 2.3: Hipótesis “cerradura y llave”(1894) que luego evolucionó a la hipótesis de “encaje inducido”(1958)

2.2.3. Afinidad del ligando

Las interacciones moleculares definen una propiedad importante que es la “Afinidad del Ligando”, la cual está determinada por la diferencia ΔG . La afinidad es medida en diseño de fármacos como una constante de inhibición (K_i) o por un valor de concentración inhibitoria que desplaza al 50% del ligando unido (IC_{50}). Los valores se encuentran generalmente en el orden de μM o nM, lo que produce valores de ΔG cada vez más negativos. Ligandos en este rango ya pueden ser usados terapéuticamente o en algún experimento, más altos ya no es una interacción específica. Mientras más baja la concentración, el valor ΔG es más negativo, por lo tanto más estable la interacción.


 Figura 2.4: Equivalencias del ΔG

La Figura 2.4 muestra que un cambio en la K_i de una molécula en 1 nM generalmente está relacionado con $-53,4$ kJ/mol. Se conoce cuál es la relación directa de energía con constante de inhibición, Además por cada ΔG de $5,9$ kJ/mol se altera la K_i por un factor de 10.

ΔG depende de la interacción de la molécula. La proteína y ligando sufren cambios conformacionales, también hay desplazamiento de moléculas de agua (ver figura 2.4). Lo mismo ocurre en el caso de proteína/proteína y proteína/ADN.

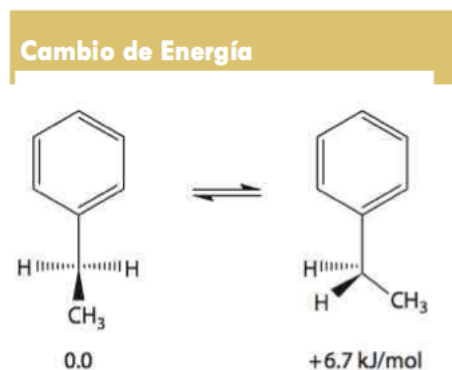


Figura 2.5: Variaciones de energía libre

La figura 2.5 muestra cómo cambios sutiles en los ligandos, pueden producir grandes variaciones de energía libre. Sólo con la rotación de uno de sus enlaces aumenta en $+6,7$ kJ/mol, suficiente energía para producir grandes variaciones en la constante de inhibición (K_i).

2.2.4. Componentes de ΔG

El ΔG se basa principalmente en los siguientes cinco componentes:

$$\Delta G = \Delta G_{transl + rot} + \Delta G_{conf} + \Delta G_{polar} + \Delta G_{hidrofob} + \Delta G_{vdW}$$

$\Delta G_{transl+rot}$: "La inmovilización global de las moléculas". Cambio de energía utilizado al perder los grados de libertad del ligando, no depende de la naturaleza del ligando. Es de 1 2kJ/mol para un complejo débil hasta 45 kJ/mol para un complejo fuerte.

ΔG_{conf} : "Cambios conformacionales del ligando y el receptor". Cambio de energía utilizado al perder la flexibilidad el ligando o la proteína. Se estima de 1-6 kJ/mol.

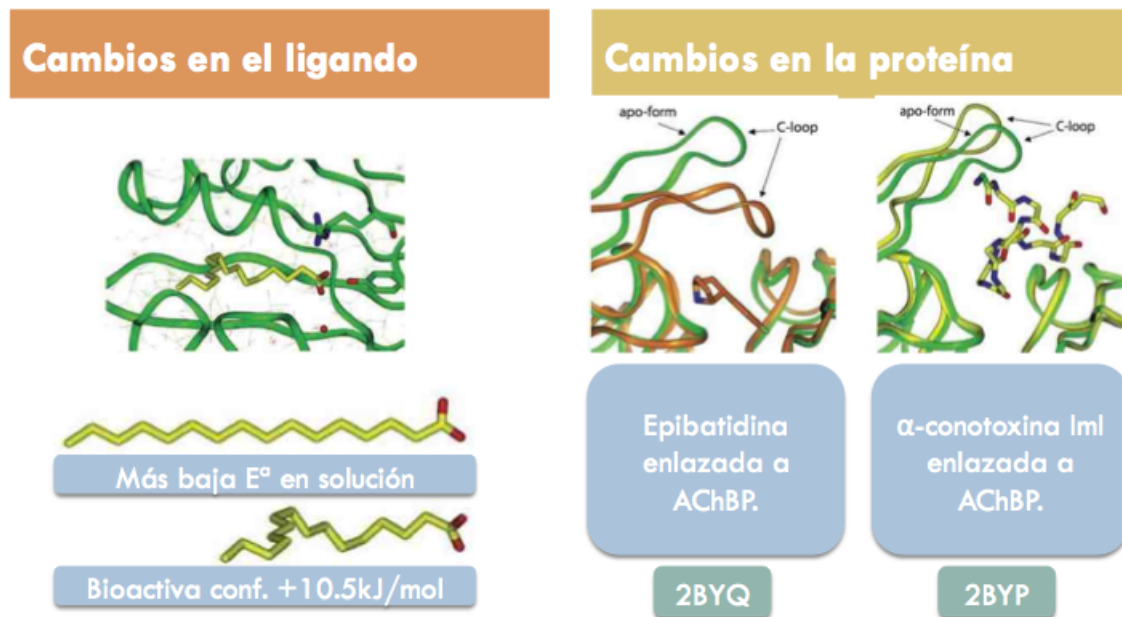


Figura 2.6: Cambios conformacionales en ligando y proteína

Ambas moléculas libres al unirse reducen sus grados de libertad. La Figura 2.6 muestra algunos ejemplos de cambios producidos en ligando y en proteína.

ΔG_{polar} : "Interacciones electrostáticas y puentes de hidrógenos". Cambio de energía debido a interacciones entre grupos funcionales polares. Frecuentemente también se observan interacciones mediadas por moléculas de agua. Se pueden destruir o producir nuevas interacciones electrostáticas y puentes de hidrógenos, debido a que las moléculas siempre están acompañadas de

una capa de solvatación, más aún si éstas son hidrofílicas. Como muestra la Figura 2.7 una vez que ingresan al sitio de reconocimiento de su receptor se generan nuevos puentes de hidrógenos que pueden tener energía diferente a las que se tenían inicialmente.

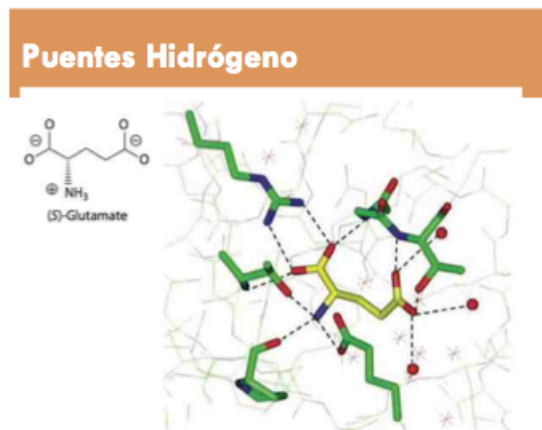


Figura 2.7: (S)-Glutamato unido a iGluR

La Figura 2.7 muestra Glutamato (amarillo) que está siendo reconocido por su receptor (verde), se aprecia la formación de varios puentes de hidrógeno que lo estabilizan con el receptor y además con moléculas de agua (circunferencia roja).

$\Delta G_{\text{hidrofob}}$: “El efecto hidrofóbico”. Es la energía de transferencia de una molécula o grupo no polar desde el agua a una fase orgánica. La energía depende del área de la superficie hidrofóbica y puede ir desde 0.1 hasta 0.24 kJ/Å² mol.

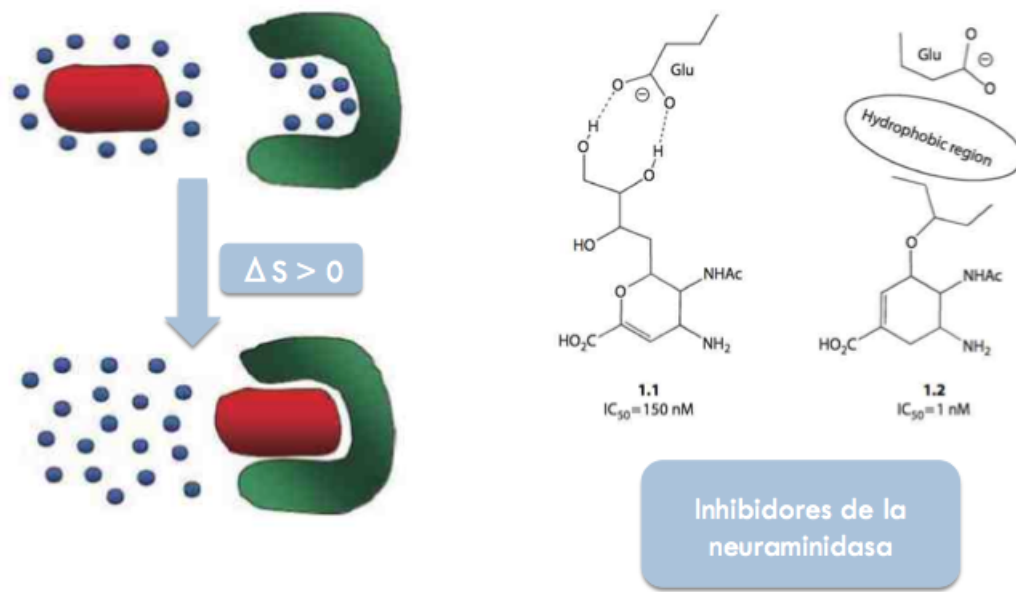


Figura 2.8: Puentes de hidrógeno vs interacción hidrofóbica

El agua se encuentra organizada y produce un cambio de la entropía. La Figura 2.8 muestra que la interacción hidrofóbica que se genera es mucho más fuerte que la de los puentes de hidrógeno con una diferencia de 150 veces; pasando de un IC_{50} de 150 nM a 1 nM haciendo éste último factible a utilizar como fármaco.

ΔG_{vdW} : “Interacciones atractivas y repulsivas de vdW”. Es la energía de interacción no polar dependiente de la distancia de interacción, donde a corta distancia es repulsiva por el solapamiento de las nubes electrónicas y es responsable de la baja afinidad de moléculas con solapamientos de los radios de vdW. A mayor distancia la interacción es de estabilización con baja energía de 0.2 kJ/mol, pero el número de interacciones por molécula puede llegar a ser alto convirtiendo a este componente de la energía en un valor significativo (ver Figura 2.9)

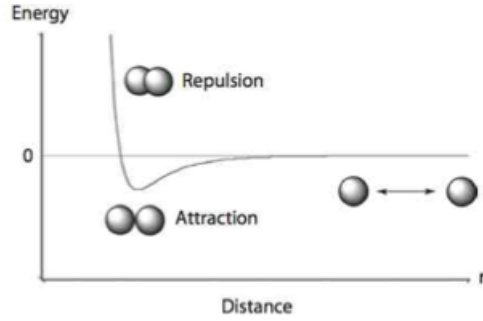


Figura 2.9: Curva de energía de vdW

2.3. Diseño de fármacos basado en bio-estructuras.

Cuando se habla de bio-estructura se refiere a que el diseño de fármacos se realiza conociendo la estructura de nuestro receptor. Se utiliza la información de la forma y propiedades del sitio de unión de la molécula blanco para diseñar moléculas con alta especificidad y afinidad. Este tipo de diseño racional necesita del conocimiento 3D de la proteína blanco por: cristalografía y rayos X, espectroscopia RMN ó por métodos computacionales de predicción como el modelaje molecular por homología. La resolución es importante: 1.2 ángstrom es ideal y resuelve H-bonds; 2.0 ángstrom es lo más común y se utiliza con frecuencia; mayor a 3.0 ángstrom se debe utilizar con precaución.

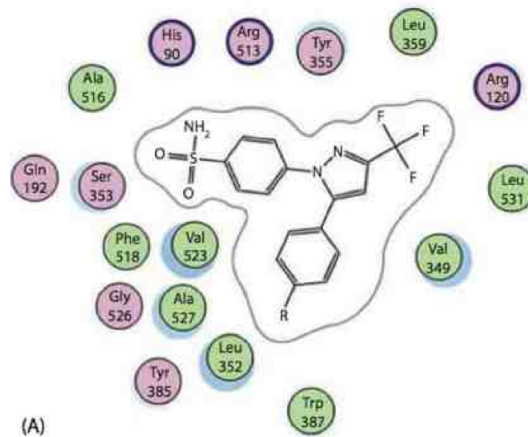


Figura 2.10: Celecoxid unido a COX-2

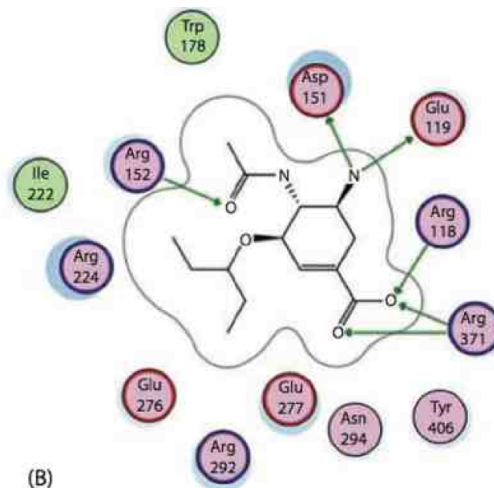


Figura 2.11: Oseltamivir unido a Neuraminidasa

Las Figuras 2.10 y 2.11 muestran el Oseltamivir y un inhibidor de la COX-2 los que fueron desarrollados gracias a que se conocía específicamente cuales eran los aminoácidos que estaban formando la cavidad. En base al conocimiento de las propiedades de estos aminoácidos se puede diseñar el tipo de molécula que tendría que interactuar en la cavidad.

2.3.1. Droga anti-influenza

Se ha mencionado la importancia del diseño de nuevos y mejores fármacos. Las drogas anti-influenza son un ejemplo de esto. El virus de la influenza está continuamente cambiando. Cada década aparece una nueva cepa con potencial de generar pandemias. En el 2009 fue la cepa H1N1. La influenza posee dos proteínas que son blancos terapéuticos, la hemaglutinina y la neuranimidasa.

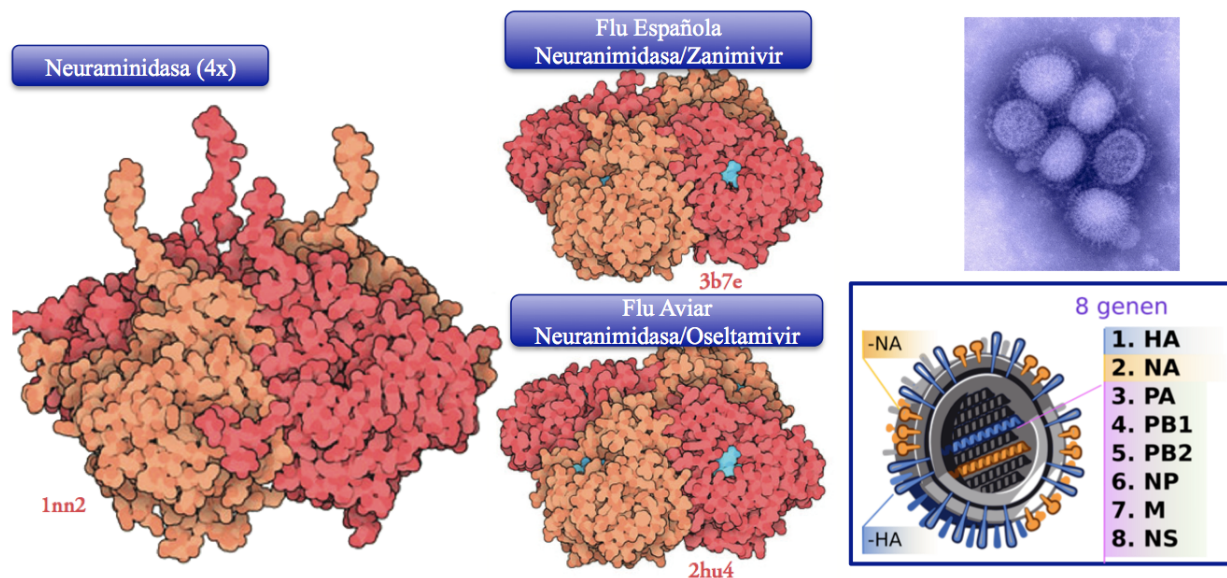


Figura 2.12: La Neuraminidasa, Flu Española, Flu Aviar y el virus

La Figura 2.12 muestra una de las proteínas más conocidas del virus. El Zanimivir se utilizó como terapia cuando se produjo la enfermedad. Luego al aparecer la H1N1 el virus era resistente al Zanimivir y a partir de éste se diseñó otro fármaco que interacciona en la misma zona, llamado Oseltamivir. El virus posee ocho genes, de los cuales 2 son los atacados por el fármaco, ya que están en la cubierta del virus. La inhibición de la enzima encargada de la síntesis del virus hace que éste muera.

2.4. Diseño de drogas basado en ligando.

Muchas proteínas que son blancos interesantes en enfermedades son difíciles de cristalizar y no se tienen sus estructuras 3D, pero se conoce la familia de ligandos que genera un efecto beneficioso con proteínas específicas con las que están interaccionando. Métodos de diseño de drogas basados en ligandos son “modelamientos de farmacóforos (elemento químico mínimo que permite la interacción funcional)” que son utilizados para desarrollar un modelo de farmacóforo que describa las interacciones entre los ligandos y la proteína blanco desde el punto de vista del ligando. Un modelo de estos podría ser adicionalmente utilizado como punto de partida para realizar un análisis de relación cuantitativa de la estructura actividad en 3D para ligandos (3D- QSAR). El análisis 3D-QSAR

analiza ligandos, combina metodologías obtenidas desde la modelación de farmacóforos, el campo de la interacción molecular y la quimiometría estadística para entregar predicciones cuantitativas de la afinidad de receptores como guías para la optimización de ligandos.

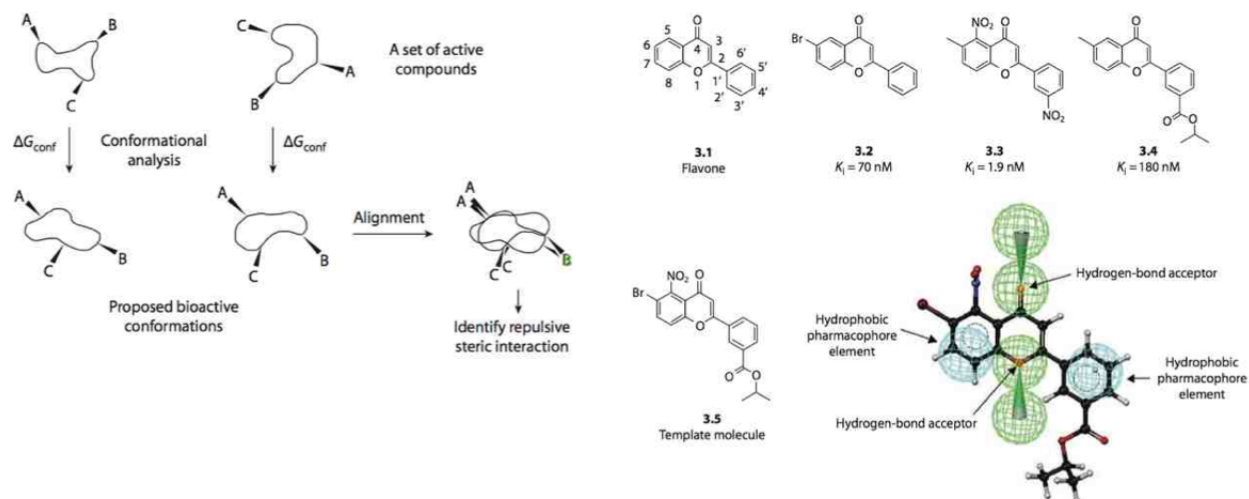


Figura 2.13: Identificación de un farmacóforo

La Figura 2.13 muestra un grupo de moléculas ligando con diferentes constantes de inhibición. A través de QSAR se pueden alinear todos los ligandos. Se destaca en la figura regiones receptoras de hidrógenos, regiones hidrofóbicas. A través de esto podemos determinar el farmacóforo y calcular sustituyentes en las zonas que pueden variar. Luego agregar otros sustituyentes que cambien los valores de K_i . Todo este proceso es solamente con moléculas ligando, sin las estructuras de los receptores.

Algunos ejemplos de fármacos diseñados a través de esta metodología fueron los inhibidores nucleósidos y nucleótidos para transcriptasa reversa del VIH. La transcriptasa reversa (RT) del VIH es uno de los principales blancos terapéuticos para luchar contra el SIDA.

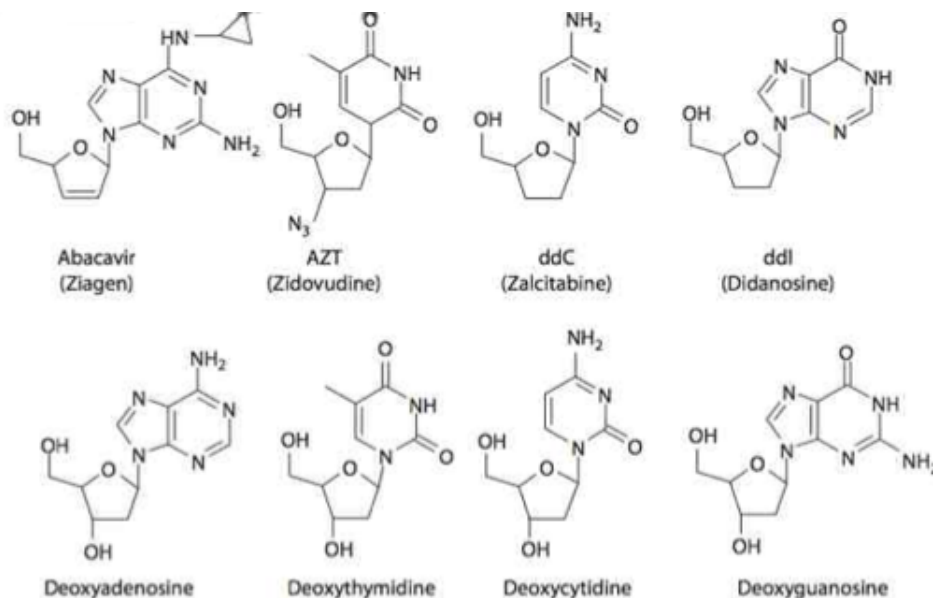


Figura 2.14: AZT, primer fármaco utilizado contra el VIH

La Figura 2.14 muestra los inhibidores basados en la estructura de nucleótidos y nucleósidos que se conocen, a los que se le hicieron modificaciones para llegar al diseño del primer fármaco utilizado contra el VIH, Zidovudina (AZT).

2.5. ¿Cómo se realiza el diseño racional de fármacos?

2.5.1. Relevancia del docking molecular

Las interacciones moleculares son esenciales para la mayoría de los procesos biológicos. Conocer como interaccionan y cuáles son los componentes energéticos que favorecen esa interacción. Esto es necesario para determinar donde y cómo se va a producir la interacción del fármaco y cual sería su efecto. El número de estructuras moleculares conocidas continua en crecimiento, el análisis computacional de las interacciones moleculares aumenta en importancia. La predicción de interacciones moleculares es una de las principales herramientas en el descubrimiento de nuevas drogas.

2.5.2. Docking molecular

Se denomina docking molecular a un procedimiento de la química *in silico*

utilizado para estudiar y predecir computacionalmente la estructura de un complejos ligando-receptor.

Las estructuras tridimensionales de proteínas representan complejos objetos con muchas, protusiones y cavidades. La interacción de ligandos con proteínas está directamente relacionada con una mayor superficie de interacción, en estas cavidades los aminoácidos están dispuestos en conformaciones muy concertadas en interacción con el ligando. El objetivo de este procedimiento es la de generar un conjunto suficiente de conformaciones del complejo ligando-receptor y entonces clasificarlas de acuerdo a su estabilidad. Para realizar este proceso existe diferentes métodos, todos basados en un procedimiento general, el cual de forma iterativa se ubica en una determinada posición y orientación el ligando en el sitio activo y se le calcula un valor de afinidad con una función de puntuación. Un protocolo de docking sería la combinación de una estrategia de búsqueda cuya función es la de elucidar todos los posibles modos de unión entre el ligando y receptor (en teoría la exploración de los seis grados de libertad translacional y rotacional del ligando junto con los grados de libertad conformacional interna del ligando y receptor), y una función de puntuación. En la práctica el uso de las estrategias de búsqueda es limitado debido al gran costo computacional que implicaría explorar las conformaciones con todos los grados de libertad, por lo tanto, se aplican una serie de restricciones y aproximaciones con el fin de simplificar el sistema. Por esto se han desarrollado diferentes métodos y algoritmos que tienen en cuenta la flexibilidad del ligando y del receptor en forma limitada.

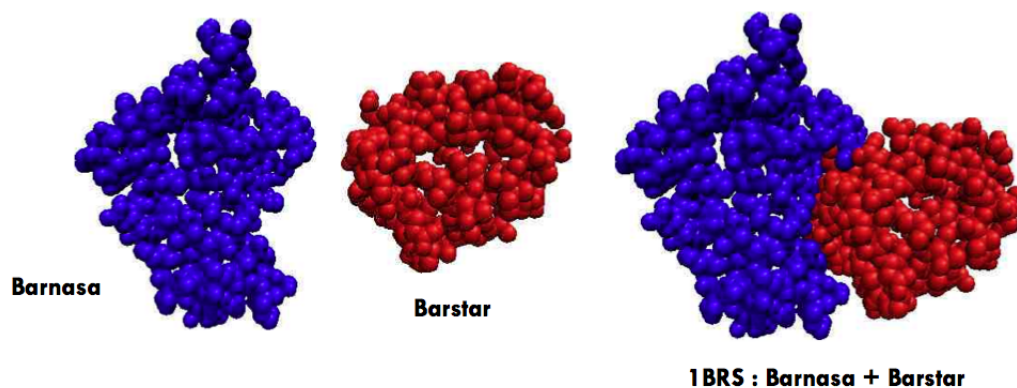


Figura 2.15: Docking proteína-proteína

La Figura 2.15 muestra un ejemplo de docking proteína-proteína. Barnasa ("BActerialRiboNucleASA") es una proteína bacteriana que consiste de 110 aminoácidos y posee una actividad

ribonucleasa. Es sintetizada y secretada por la bacteria *Bacillus amyloliquefaciens*, es letal para las células cuando es expresada sin su inhibidor barstar.

La flexibilidad de ligando o receptor es importante al momento de realizar el docking. En el caso de moléculas pequeñas podemos realizar docking con:

Receptor rígido – ligando rígido. Es menos costoso en recursos de computadora. Analiza una sola conformación.

Receptor rígido – ligando flexible. Analiza todas las conformaciones del ligando.

Receptor flexible – ligando flexible. Evalúa todas las conformaciones diferentes de ligando y receptor, por lo tanto demanda mucho más recursos de computadores. Es el más costoso.

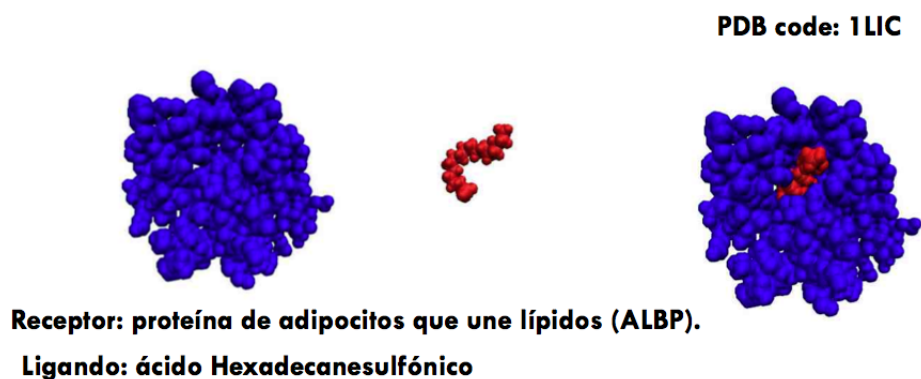


Figura 2.16: Docking de una molécula pequeña

En la Figura 2.16 se muestra un ejemplo de docking entre una proteína y una molécula pequeña. ALBP, las proteínas que unen lípidos son una familia de proteínas intracelulares de 15 kDa con la capacidad de unir con alta selectividad ligandos hidrofóbicos como ácidos grasos, sales biliares y ácidos retinoicos.

Ligandos (conformación, configuración y pose)

Conformación: es la posición relativa de los átomos en una estructura 3D de una molécula, independiente del sistema de coordenadas (ver Figura 2.17). Los programas de docking no realizan docking de una molécula rígida, sino de diferentes conformaciones de la molécula, dependiendo el software utilizado, pueden ser hasta 60 conformaciones diferentes.

Dos diferentes conformaciones para un ligando.

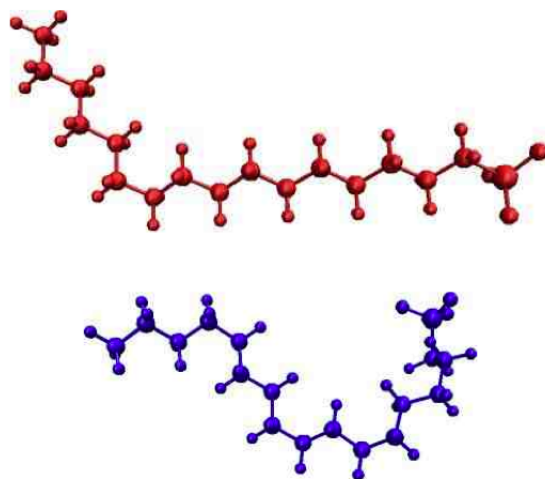


Figura 2.17: Diferentes conformaciones para un ligando

Pose: Es una configuración de una conformación de una molécula en un sistema de coordenadas. Es decir, la posición del ligando sin cambios de conformación (ver Figura 2.19). La pose varía dependiendo del ΔG Traslacional.

Dos diferentes poses para un ligando.

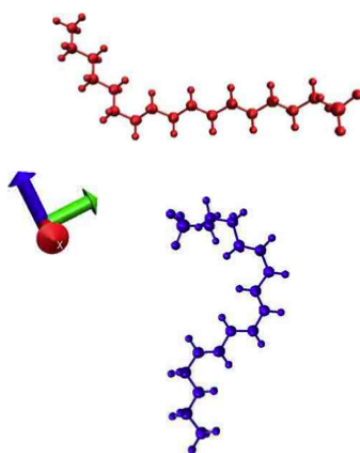


Figura 2.18: Diferentes poses para un ligando

Configuración/posicionamiento: las posiciones de los átomos de la molécula después una

transformación rígida (rotación y traslación) de un sistema de coordenadas. Una configuración es una relación entre conformación y una pose.

**Dos diferentes configuraciones
(la misma conformación)**

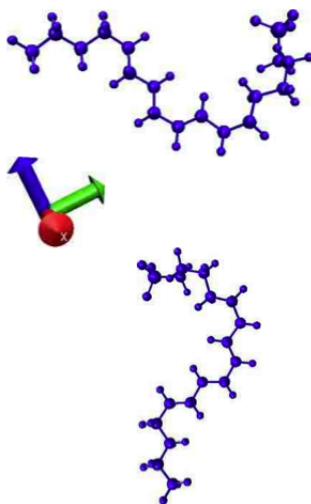


Figura 2.19: Diferentes configuraciones para un ligando

2.5.3. Superficies y cavidades

El problema de predecir como una molécula interacciona con el receptor e identificar el complejo más estable. Cuando tenemos de 1 a 100 ligandos no hay problema en la forma que realicemos el docking. Podríamos realizar un “Docking Ciego”; probamos las moléculas en la estructura completa, en todas sus posiciones (ver Figura 2.20). Girando en los 3 ejes, se rota y calcula la energías de interacción en todas las conformaciones. El problema se amplifica cuando queremos trabajar con más de 100 moléculas. Por ejemplo: ¿Qué sucede si queremos probar 100.000 moléculas?. En este caso la demanda de cálculo, memoria y procesamiento es demasiado alto, realizando un Docking Ciego tardaría años en realizar el Docking . Para optimizar esto realizamos un “Docking Dirigido”; Para esto localizamos el docking, se debe representar eficientemente la superficie de docking e identificar las regiones de interés. Al calcular estas cavidades se reduce el problema, y ya no aplicamos el docking completo, sino en forma localizada. De esta manera el tiempo de computo es mucho menor y podemos probar un millón de moléculas en una semana.

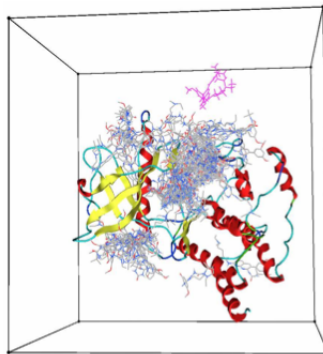


Figura 2.20: Docking Ciego. Una grilla para la estructura completa. Las moléculas se prueban en todas las conformaciones.

En el caso del "docking dirigido" la grilla es localizada en una cavidad específica. La Figura 2.21 muestra como se ve la estructura 3D en este caso.

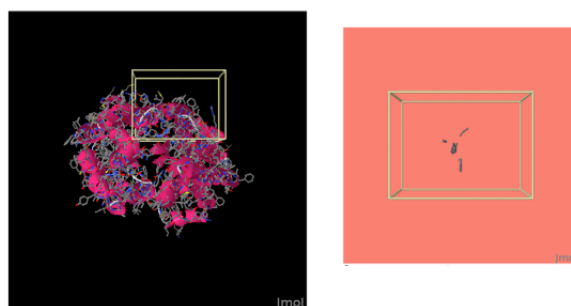


Figura 2.21: Docking dirigido. Una grilla localizada en una cavidad específica.

Determinación de superficies

Generalmente donde interaccionan las moléculas hay una superficie de complementariedad mayor, estas son las cavidades y protrusiones. Las cavidades se pueden calcular de diferentes formas, lo más común es utilizar esferas de prueba para hacer que éstas rueden sobre la superficie y al momento de solaparse se determina que existe una cavidad.

Superficie de Connolly. Cada esfera atómica esta dada por el radio de Van der Waals para cada átomo. Rodando una esfera de prueba sobre la superficie de Van der Waals genera la superficie accesible al solvente o superficie de Connolly.

Técnica de Lenhoff. Calcula una superficie "complementaria" para el receptor en lugar de la

superficie de Connolly, por ejemplo calcula las posibles posiciones para los centros de los átomos de un ligando.

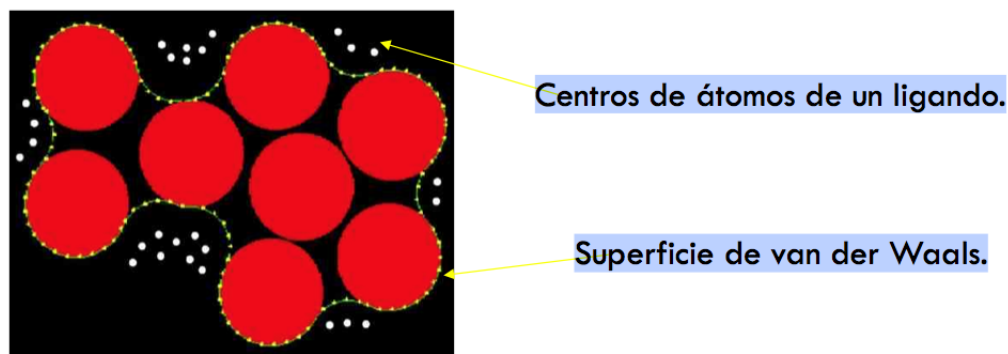


Figura 2.22: Centros de átomos de un ligando y superficie de van der Waals

Agrupación de esferas de Kuntz et al. Usa esferas agrupadas para identificar cavidades en un receptor y protrusiones en un ligando. Calcula una esfera por cada par de puntos de superficie, i y j , con el centro de la esfera en la normal desde el punto i (ver Figura 2.23). Las regiones donde muchas esferas se solapan son cavidades en el receptor o protrusiones en ligando.

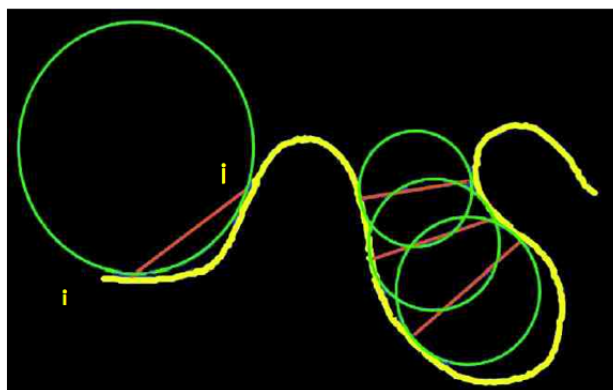


Figura 2.23: Agrupación de Esferas de Kuntz et al

Formas Alfa. Formaliza la idea de “forma”. En 2D un eje entre dos puntos es “alfa-expuesto” si existe un círculo de radio alfa así que dos puntos se unen en la superficie del círculo y el círculo no contiene otros puntos de la superficie (Ver Figura 2.24).

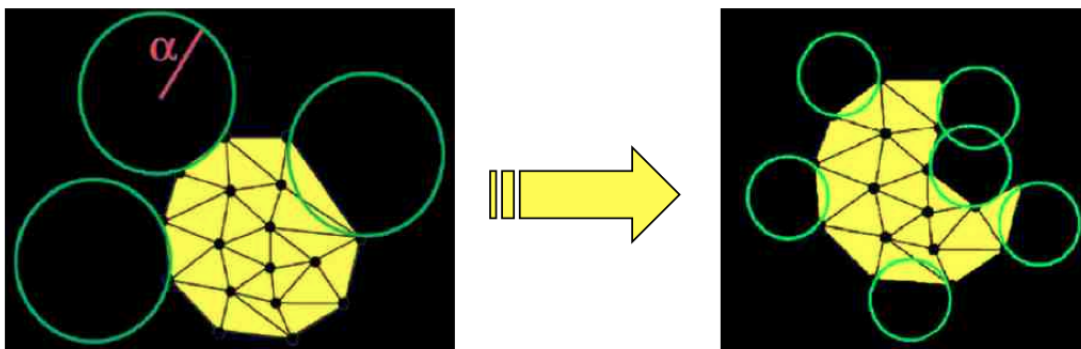


Figura 2.24: Formas Alfa

2.6. Algoritmos de docking molecular.

2.6.1. Métodos de búsqueda.

Nosotros podemos clasificar los algoritmos de búsqueda de acuerdo a los grados de libertad que ellos consideran. Docking rígido o docking flexible.

Docking rígido: Docking de enlace. El ligando es tratado como una estructura rígida durante el docking. Sólo se consideran los grados de libertad traslacional y rotacional (6 grados de libertad: 3 rotaciones y 3 traslaciones). Con respecto a las conformaciones del ligando, se genera un gran número de conformaciones de cada ligando las que son ensayados en forma individual. Menos costoso en términos de recursos computacionales. Menor precisión.

Docking flexible: Es la forma más común de docking. Las conformaciones de cada molécula son generadas por el algoritmo de búsqueda durante el proceso de docking. El algoritmo evita conformaciones que no calcen. Se realiza una búsqueda exhaustiva, por lo tanto, computacionalmente es muy costosa a medida que el espacio de búsqueda incrementa. Mayor precisión

Actualmente existen métodos de búsqueda que consideran flexibilidad del receptor, pero los métodos más desarrollados son los que realizan flexibilidad del ligando. Se pueden identificar tres clases de algoritmos de búsqueda, estas son:

Búsqueda sistemática.

Es computacionalmente costosa a medida que el espacio de búsqueda incrementa. El problema de estos métodos es el gran número de conformaciones al cual se ven enfrentados debido a que se consideran todos los grados de libertad de la molécula. Una alternativa es realizar una búsqueda incremental evitando una búsqueda demasiado exhaustiva. En una primera etapa el ligando se divide en un fragmento rígido y las cadenas laterales como un fragmento flexible. Luego estas ramificaciones son subdivididas en cada nuevo enlace rotatable que posean. Los fragmentos rígidos se van ubicando en el sitio activo y las cadenas laterales se conectan en forma secuencial acompañando el proceso con una búsqueda sistemática de los ángulos diedros. Ejemplos: *DOCK*, *FlexX*, *SLIDE*, *Surflex*.

Búsqueda aleatoria.

No garantizan una solución óptima, pero sí una buena solución dentro de un tiempo razonable. En esta categoría se encuentran métodos conocidos como lo son Algoritmos Genéticos y Monte Carlo. Estos métodos realizan una exploración mediante cambios aleatorios de la estructura del ligando. Se define una función de probabilidad para evaluar el ligando obtenido. Otros algoritmos son los de búsqueda tabú, éstos tienen la particularidad de considerar las áreas del espacio conformacional que ya han sido exploradas con anterioridad. Según el valor RMSD obtenido con respecto a cada una de las conformaciones almacenadas se acepta o rechaza una determinada conformación. Ejemplos: *Autodock*, *MCDock*, *ICM*, *GOLD*, *Glide*.

Métodos de simulación.

La aproximación más utilizada es dinámica molecular, y esta a menudo no alcanza altos estados energéticos en periodos de tiempo razonables de simulación y por lo tanto sólo puede acomodar el ligando en mínimos locales de la superficie energética. Para solucionar este problema a menudo se simulan diferentes partes de un sistema a diferentes temperaturas o comenzando los cálculos de dinámica molecular desde diferentes posiciones del ligando (Varas, I.A. 2011). Ejemplos: *FLOG*, *FRED*, *Yucca*. Muy costoso en tiempo computacional, pero pueden alcanzar gran precisión de los resultados.

2.6.2. Vina v/s DOCK6

Los programas mayormente utilizados para los cálculos de docking molecular son Autodock/Vina y DOCK6, el primero es utilizado principalmente para realizar docking de pequeñas bibliotecas y el último para docking de bibliotecas masivas. AutoDock Vina (Trott y Olson; 2010), programa de código abierto para el descubrimiento de fármacos, docking molecular y virtual screening, diseñado por el mismo laboratorio que desarrollo AutoDock (Garret y col., 2009). Vina es una evolución del AutoDock con mejoras tales como: mayor precisión de las predicciones de los modos de unión, además esta versión fue desarrollada en maquinas de 64-bits y cuenta con la capacidad de uso de múltiples núcleos permitiendo un mejor desempeño. Otra diferencia es haber facilitado el calculo de la grilla y haber agrupado en directorios los resultados. Además los archivos de entrada y salida son estandarizados a *pdbqt* facilitando el trabajo e interacción del programa con el usuario. Pero todo estos procesos se pueden optimizar aún más. Este punto será tratado más adelante en este documento.

El algoritmo de DOCK desarrollado por Kuntz y colaboradores es considerado uno de los mayores avances en el docking de proteínas ligandos. [Kuntz et al., JMB, 1982, 161, 269]. Las primera versiones del algoritmo de DOCK sólo consideraban docking de cuerpo rígido y fue diseñado para identificar moléculas con un alto grado de complementariedad de forma con el sitio de unión de ligandos. Las primeras versiones de DOCK incluían la construcción de una imagen negativa del sitio de unión consistente de una serie de esferas solapadas de radio variable, derivadas desde la superficie molecular de la proteína. Los átomos de ligando son luego superpuestos a los centros de las esferas de forma que la distancia entre los átomos sea igual a las distancias entre los correspondientes centros, dentro de alguna tolerancia. La conformación del ligando es luego orientada dentro del sitio de unión. Después se observa si existente impedimentos estéricos y luego es asignado un puntaje. Nuevas orientaciones son producidas por la generación de un nuevo grupo de átomos de ligandos que se superponen a los centros de las esferas. Tal procedimiento continua hasta que todo los posibles ligandos han sido considerados.

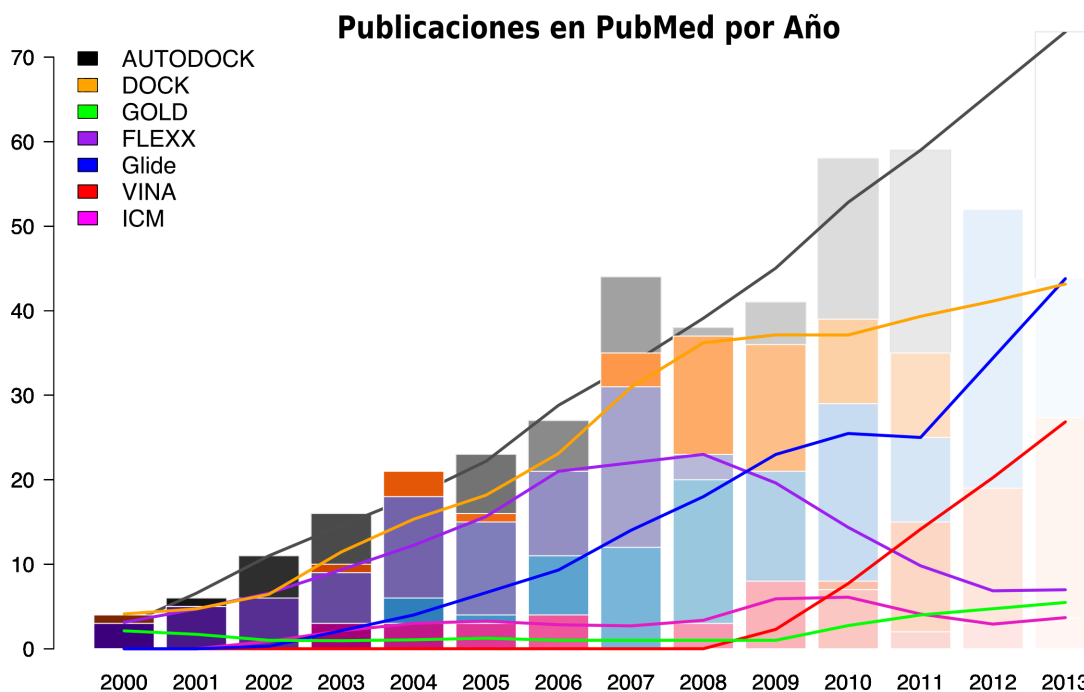


Figura 2.25: Gráfico que representa los Software de docking más utilizados en publicaciones

En el Gráfico de la Figura 2.25 se aprecia que los software más utilizado en publicaciones son DOCK y AutoDock, pero este último al final presenta una disminución, la que se debe a que está siendo reemplazado por Vina.

2.7. Galaxy

La computación es una de las principales herramientas utilizada en la investigación biológica. Sin embargo en ocasiones científicos e investigadores no cuentan con los conocimientos técnicos e informáticos necesarios para diseñar y/o utilizar métodos computacionales que muchas veces terminan por ser un problema no fácil de solucionar. Los obstáculos pueden partir desde la instalación de un programa, la manipulación de los parámetros de éste, o por la necesidad de combinar eficientemente múltiples herramientas informáticas. Además investigaciones revelan que menos de la mitad de los experimentos publicados en Nature Genetic (<http://www.nature.com/ng/index.html>) pudieron ser reproducibles, esto debido a la complejidad de los métodos, mayoritariamente computacionales, a

la difícil definición de datos y utilización de programas de análisis. Galaxy es una solución a estos problemas. “Galaxy is an open, web-based platform for accessible, reproducible, and transparent computational biomedical research. Accessible: Users without programming experience can easily specify parameters and run tools and workflows. Reproducible: Galaxy captures information so that any user can repeat and understand a complete computational analysis. Transparent: Users share and publish analyses via the web and create Pages, interactive, web-based documents that describe a complete analysis.” (<http://wiki.galaxyproject.org/>)

Con respecto a la *accesibilidad* el enfoque de Galaxy es una interfaz de usuario sencilla, simple y eficaz. La *reproducibilidad* es fundamental en la ciencia. Galaxy permite registrar todos los pasos de un análisis computacional, de esta manera un usuario podría volver a realizar exactamente el mismo procedimiento en cualquier momento. Además permite compartir a la comunidad científica todos los elementos desarrollados, con la opción de que sean estudiados, evaluados, nuevamente ejecutados por otros científicos investigadores apoyando a la *transparencia*.

2.7.1. Historial, flujo de trabajo, conjunto de datos y páginas

Historial: Galaxy puede ser utilizado simplemente ingresando los parámetros de entrada de la herramienta que se pretende ocupar, pero también los usuarios se pueden registrar en un sistema Galaxy. Esto tiene algunas ventajas, entre éstas la de tener un historial propio. Es una alternativa a llevar un registro de los datos e información utilizada en los distintos análisis computacionales realizados por un usuario. De esta manera un investigador puede ingresar con su cuenta en el futuro y tener un conjunto de datos intermedios y de resultados anteriores.

Flujos de Trabajo: Un usuario puede crear flujos de trabajo, que es la configuración de los pasos de un determinado análisis computacional. Estos incluyen la utilización de los programas disponible en la barra de herramientas. Un flujo de trabajo es ejecutado del mismo modo que se ejecutan las otras herramientas. Las salidas de un flujo de trabajo, también se gestionan en el historial. Los usuarios pueden agregar sus etiquetas a los flujos de trabajo. De esta forma es más claro identificar que hacen y así volver a utilizarlo.

Conjunto de datos: Es cualquier dato de entrada, intermedio o salida de un procedimiento de análisis computacional.

Páginas: El usuario tiene la capacidad de visualizar los historiales, conjunto de datos, detalle de

los flujos de trabajo, etc. A través de las páginas. Además para quienes trabajan con su propia instancia Galaxy pueden utilizar las páginas para reproducir cualquier tipo de resultados para que sean visualizados por el usuario, sin la necesidad de salir de Galaxy.

2.7.2. Disponibilidad de Galaxy

Galaxy esta disponible en un servidor web gratuito (<https://main.g2.bx.psu.edu/>) disponible para todo público. Este servidor web dispone de varias herramientas bioinformáticas útiles en una investigación. También el usuario se puede registrar y tener sus propios flujos de trabajos, historiales y todo lo anteriormente comentado con la posibilidad de reproducir, guardar y compartir con los demás. Pero también Galaxy esta disponible como software libre. Esto nos permite descargar el Software y tener nuestra propia instancia Galaxy con todos lo beneficios que esto incluye. Esta última es la opción que se pretende llevar a cabo durante el desarrollo de este trabajo. Si tenemos los conocimientos de programación e informática se pueden crear nuevas herramientas, en cualquier lenguaje de programación y luego agregar los programas a nuestra propia instancia Galaxy. Y por supuesto también nos permite compartir, hacer más accesible y reproducible cualquier trabajo de biología computacional.

Capítulo 3

Análisis del problema

Previo a la implementación de un proyecto es necesario un análisis detallado de éste. Se debe realizar un estudio, para responder a cualquier interrogante con respecto al tema que se está desarrollando, comprender el problema y los objetivos que se quieren alcanzar.

En la primera parte de este capítulo se presenta la base del problema. Se realiza un análisis de la situación actual, dando a conocer los problemas específicos que llevan a la implementación de este proyecto. Se define el objetivo general y la descripción de cada uno de los objetivos específicos.

Además, va a ver en detalle el análisis de requerimientos y el modelamiento del problema. Luego se muestran los resultados de los estudios de viabilidad que fueron realizados para este proyecto, a nivel operacional, técnico y económico.

Este capítulo es enfocado desde un punto de vista analítico, con el objetivo de comprender y definir el contexto en el cual se desarrolla el proyecto, pues así, determinar la mejor solución e implementación.

3.1. Planteamiento del problema

El descubrimiento de fármacos es principalmente representado como un proceso lineal consecutivo, que comienza con la identificación de sitios dianas moduladores de la función de las proteínas (target to hit o target discovery), seguido de un proceso de identificación de novedosos compuestos que actúan sobre el sitio diana (hit-to-lead o lead discovery), para terminar con los estudios *in vivo* en animales y humanos (Terstappen y Reggiani, 2001; Jayaram et al., 2008).

Tradicionalmente, las drogas han sido descubiertas mediante síntesis de compuestos testeando (screening) un portafolio/batería de moléculas en estudios *in vivo*, en donde los candidatos más prometedores siguen a la siguiente etapa (Jayaram et al., 2008). Este proceso de desarrollo ha dado lugar a altas tasas de deserción con fracasos atribuidos a pobre farmacocinética (39%), a falta de eficacia (30%), a toxicidad en animales (11%), y a efectos adversos en humanos (10%) (Jayaram et al., 2008). Actualmente, existe una limitada habilidad para predecir si el compuesto químico va a tener el efecto terapéutico esperado y si va tener efectos adversos en proteínas transportadoras debido a la carencia de modelos tridimensionales. En comparación, los ingenieros aeronáuticos pueden predecir el rendimiento de un nuevo avión a través simulación computacional, porque ellos conocen las ecuaciones de flujo de aire que permiten hacer la simulación (President's council of advisors on S&T, 2012). Pero los equipos científicos del área de proteínas transportadoras, carecen de adecuadas herramientas de predicción, para determinar efectos adversos y eficacia terapéutica. Debido a lo anterior, de cada 5.000 compuestos que ingresan a estudio preclínico, en promedio sólo 5 son probados en estudio clínico, y de estos sólo 1 es aprobado para usos terapéuticos (Kraljevic S, et al., 2004). Además, de cada 1.000 moléculas candidatas, sólo 1 llega al mercado en un período de 15 años, costo que es traspasado al consumidor para solventar los gastos de investigación, producción y royalties.

La carencia de estudios a nivel estructural y funcional (herramientas de predicción) se debe a problemas en la sobreexpresión, estabilidad, conservación del plegamiento nativo, purificación y cristalización de proteínas de membrana (Caffrey 2003). Dado estos problemas, menos del 1% de las proteínas de membrana son conocidas a nivel atómico, generándose un vacío en el conocimiento de la relación entre estructura y función que es clave para el desarrollo de nuevos medicamentos (Tusnády y col. 2004). En atención a los antecedentes expuesto anteriormente, y a objeto de optimizar el proceso de diseño racional de fármacos para proteínas transportadoras, los científicos demandan

el desarrollo de plataformas bioinformáticas para el diseño de las estructuras tridimensionales de estas proteínas y la búsqueda masiva de fármacos con actividad en las proteínas transportadoras.

3.1.1. Situación actual

Actualmente en el Laboratorio de Diseño Racional de Fármacos de la Universidad de Concepción se llevan a cabo una serie de trabajos en el área de docking molecular. Los programas utilizados para el cálculo de docking son: Autodock/Vina y DOCK6, el primero utilizado para realizar docking de pequeñas bibliotecas y el último para docking de bibliotecas masivas. Si bien, gracias a estos software existe una madurez en el cálculo de docking masivo, la distribución del cálculo no ha sido exhaustiva y existen nuevos protocolos diseñados que necesitan implementación. Por ejemplo: al preparar los ligandos y moléculas, manualmente no podemos comparar ni complementar resultados entre ambos programas, puesto que, es imposible generar dos grillas idénticas para ambos software. Se necesitan herramientas que faciliten las tareas de docking masivo de compuestos, automatizar el proceso de preparación de archivos para docking regulares y masivos, disminuyendo los tiempos y permitiendo su comparación.

Los programas Vina y DOCK6 sólo realizan los cálculos de docking, el problema continua cuando queremos hacer estudios de los resultados de docking. No se dispone de software que permitan el análisis avanzado de los resultados; como cálculos de fármacosforos (elemento mínimo de interacción), determinación de clúster de aminoácidos con teoría de grafos y evaluación electrostática posterior de la energía de complejos.

En el laboratorio se han desarrollado algunas herramientas, estos trabajos han sido dirigidos por el Dr. Alexis Salas Burgos académico del departamento de Farmacología de la Universidad de Concepción. El problema es que el uso de estos algoritmos no se ha masificado. Principalmente esto se debe a que existe un problema entre los usuarios que realizan los experimentos (alumnos, académicos, investigadores), los que generalmente no cuenta con conocimientos para la ejecución de programas a través de terminal. Se necesita desarrollar estas herramientas y estandarizarlas para un acceso fácil, rápido y eficiente sin la necesidad de que los usuarios requieran conocimientos técnicos de computación e informática.

Para dar solución a todos estos problemas, anteriormente mencionados, se propone llevar a cabo el proyecto que se documenta en este texto. Éste deberá satisfacer un conjunto de necesidades, las

cuales definirán sus funciones. A continuación se definen los objetivos generales y específicos los cuales se deben seguir a lo largo de la implementación.

3.1.2. Objetivo general

Desarrollar para el Laboratorio de Diseño Racional de Fármacos de la Universidad de Concepción, la incorporación de herramientas para análisis masivo de docking molecular a un entorno Galaxy/HPC.

3.1.3. Objetivos específicos

1. Identificar, conocer y analizar la realidad actual y los problemas en el Laboratorio de Diseño Racional de Fármacos de la UDEC.
2. Identificar y especificar requerimientos, a través de encuestas, entrevistas, reuniones, etc.
3. Implementar programas robustos en python para ser utilizados en el sitio accesible vía web.
4. Evaluar e implementar algoritmos multiproceso y/o procesamiento en paralelo para tareas masivas de docking molecular.
5. Implementar y desarrollar Galaxy para ser utilizado como marco de ejecución de programas y de tuberías entre procesos.
6. Diseñar interacción web para gráficas de resultados y visualización de estructuras tridimensionales.

3.2. Definición de requisitos

Esta sección se define con el objetivo de definir cada uno de los requerimientos que el software debe cumplir. Se detallan los requisitos funcionales y los requisitos no funcionales del sistema.

3.2.1. Requisitos funcionales

1. El usuario puede subir sus archivos desde la computadora, indicando las URL o simplemente pegando el contenido del archivo e indicando su formato.
2. Desde un historial el usuario puede consultar los archivos que ha subido y visualizar su contenido en pantalla.
3. El usuario puede iniciar el análisis de los resultados de docking, enviando el archivos de salida del calculo de docking y el archivo pdb del receptor.
4. El usuario puede visualizar los gráficos de los resultados en la interfaz de Galaxy.
5. Desde el historial el usuario puede descargar, guardar localmente los distintos tipos de archivos que éste registre, incluyendo los gráficos de resultado, las moléculas, receptor, etc.
6. El usuario puede eliminar todo tipo de archivos de su Historial o eliminar un historial completo de su cuenta Galaxy.
7. El usuario puede loguearse o registrarse para poder ingresar al Galaxy y tener su propio historial, de lo contrario puede usar el sistema pero no tiene su propio historial.
8. El software debe generar los gráficos de los resultados de docking.
 - Cada una de las poses debe estar representada en un esquema.
 - Se pueden visualizar los átomos que interaccionan directamente.
9. El usuario puede visualizar las estructuras 3D y las grillas (alineadas).
10. El usuario puede seleccionar una por una las distintas cavidades y su grilla.
11. El usuario puede manipular la imagen 3D, realizar zoom, girar, cambiar color de fondo, desplazar en los distintos ejes, visualizar puentes de hidrogeno, ver y ocultar los átomos, etc.
12. EL software debe generar los archivos de configuración para realizar los docking en los distintos software (Vina/AutoDock y Dock).
 - Alinear la estructura en el eje de coordenadas.
 - Calcular y lanzar las grillas a utilizar.

3.2.2. Requisitos no funcionales

Del software

1. El software debe ser implementado en el lenguaje de programación Python.
2. El software debe generar gráficos de forma rápida y eficiente.
3. El código debe ser modular, flexible y estar comentado, para una fácil legibilidad, inserción y modificación de código fuente.
4. El software debe ser paralelizable, con el fin de calcular y procesar grandes volúmenes de datos.

De la interacción con Galaxy

1. Fácil de usar.
 - El tiempo para instruir a los usuarios será de:
 - 8 horas para el administrador del laboratorio.
 - 6 horas para alumnos.
 - 4 horas para docentes e investigadores.
 - Para el manejo correcto del sistema el usuario no deberá tardar más de:
 - 5 días para el administrador del laboratorio.
 - 4 días para alumnos.
 - 3 horas para docentes e investigadores.
 - Estable
 - En un rango de 6 meses no debería fallar más de dos veces.
 - Confiable
 - La información entregada será fidedigna con respecto a los análisis de los resultados de docking molecular, grafos, datos e informes de salida.
 - Poseer un buen rendimiento.
2. De la organización

- El usuario administrador, será el único que puede ingresar nuevas cuentas o dar de baja antiguas cuentas de usuario.
- Todo tipo de usuario, podrá monitoriar la carga del sistema y ver el estado de las tareas que se estan ejecutando.

3.2.3. Diagrama de Flujo de Datos (DFD)

Gracias al Diagrama de Flujo de Datos (DFD) podremos establecer una relación entre la interacción del sistema con las distintas entidades que presenta éste y cada uno de los procesos asociados a la ejecución de los programas de docking molecular en el sistema Galaxy.

Diagrama de contexto

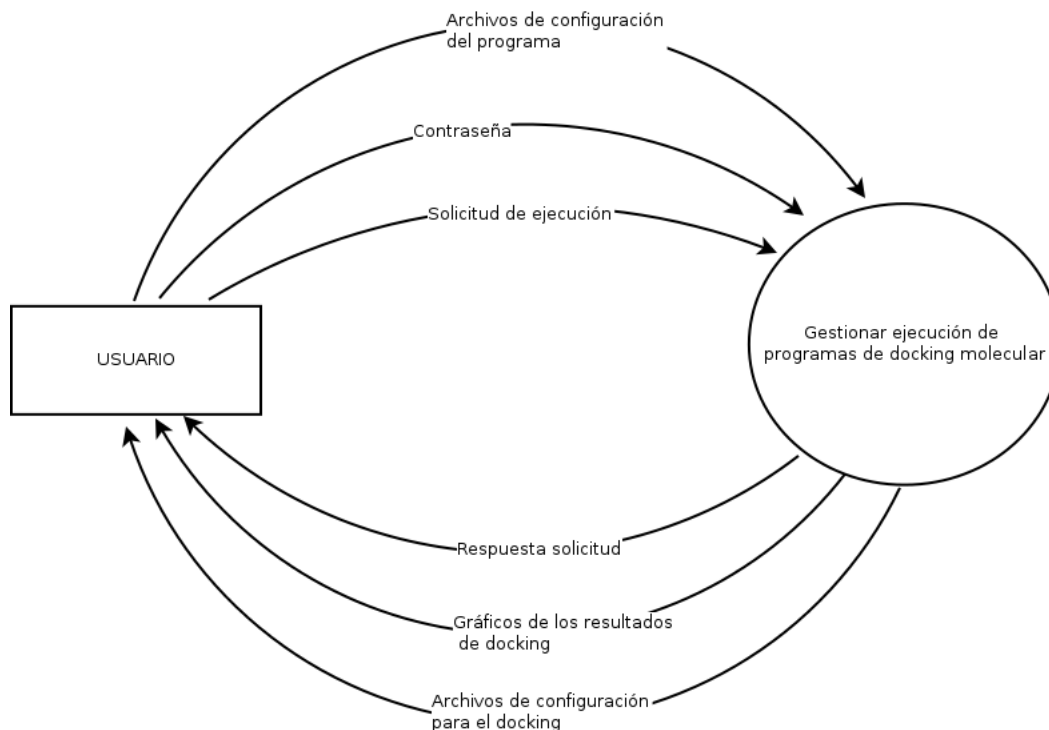


Figura 3.1: Diagrama de Flujo a nivel de contexto

Diagrama de nivel superior

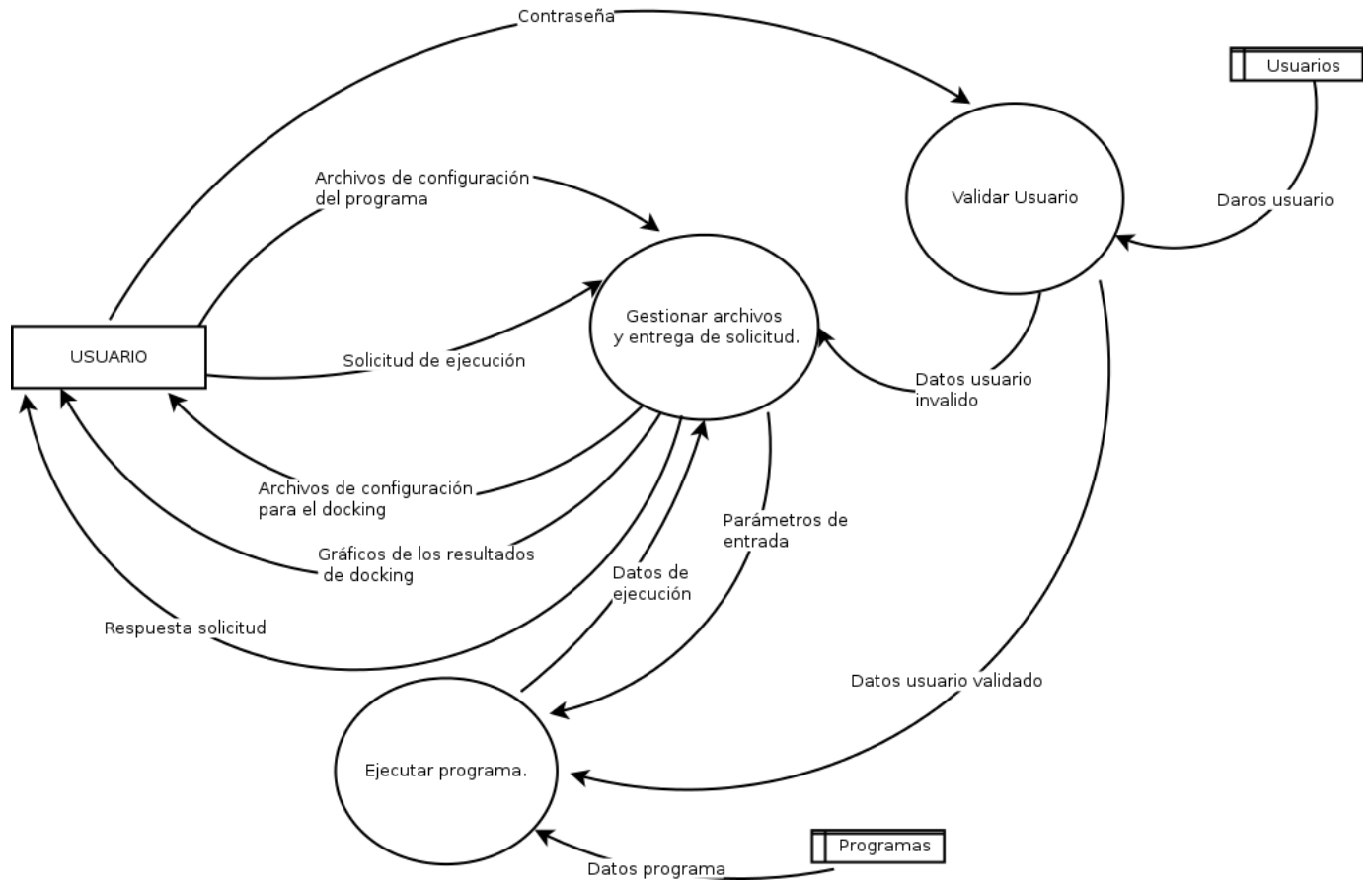


Figura 3.2: Diagrama de Flujo a nivel de superior

3.2.4. Casos de Uso

Los Diagrama de Casos de Uso nos ayudan a identificar la forma en que los usuarios utilizaran el sistema, esta herramienta nos facilita la captura de los requerimientos del proyecto. A continuación se describen las acciones del sistema Galaxy desde el punto de vista del usuario. Se muestran los casos de uso y la descripción de cada uno de ellos.

Diagrama de casos de uso

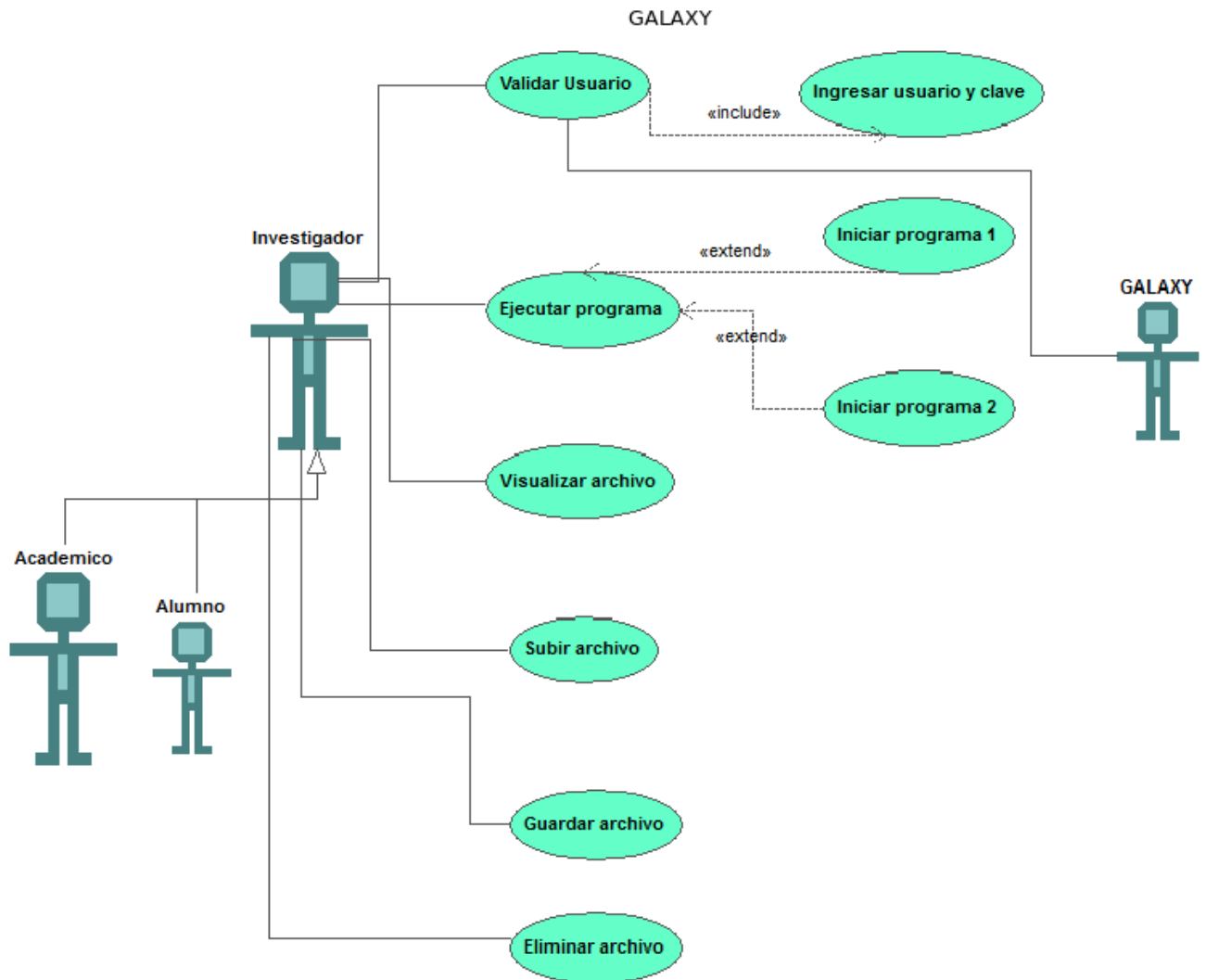


Figura 3.3: Diagrama de Casos de Uso

Especificación de casos de uso

Nombre	Validar usuario
Identificador	CU01
Descripción	Permite al usuario validar su cuenta para el ingreso a Galaxy. Además permite al Galaxy gestionar las cuentas de usuario, validando la información de éstos.
Actores	Investigador, Galaxy
Pre-Condiciones	–
Flujo de eventos básicos	a) El sistema muestra un formulario con los campos de usuario y contraseña. b) EL usuario ingresa sus datos. c) El sistema Galaxy chequea los datos y se efectua la validación. d) Galaxy muestra la página de bienvenida. e) Se ejecuta CU04
Flujo de eventos alternativo	a) Si los datos o el usuario no son válidos, el sistema Galaxy lo indicará por pantalla.
Post-Condición	Usuario válido - cuenta activa.

Cuadro 3.1: Especificación de caso de uso CU01

Nombre	Ejecutar programa
Identificador	CU02
Descripción	Permite al usuario ejecutar un programa de docking molecular.
Actores	Investigador
Pre-Condiciones	CU04
Flujo de eventos básicos	a) El sistema muestra un menú con los programas de docking. b) El usuario selecciona un programa e ingresa los parámetros de entrada. c) Se ejecuta el programa. d) Salidas guardadas en el historial de Galaxy. e) Se ejecuta CU03
Flujo de eventos alternativo	a) Si los archivos de configuración o parámetros no son correctos, el sistema lo indicará por pantalla. b) Si existe un error mientras se ejecuta el programa, el error se mostrará en pantalla.
Post-Condición	Programa ejecutado - archivos de salida en Historial.

Cuadro 3.2: Especificación de caso de uso CU02

Nombre	Visualizar archivo
Identificador	CU03
Descripción	Permite al usuario visualizar un archivo de las salidas de cada programa (Gráficos, estructura 3D), o un archivo subido anteriormente por el usuario a Galaxy.
Actores	Investigador
Pre-Condiciones	CU2 - CU04
Flujo de eventos básicos	a) El sistema muestra el historial con todos los archivos en el sistema. b) EL usuario selecciona un archivo. c) Se ejecuta el programa. d) Se muestra el contenido del archivo en pantalla.
Flujo de eventos alternativo	a) Si el archivo no es de un formato compatible con Galaxy, se da la opción de descargar o guardar el archivo en otra unidad (CU05)
Post-Condición	Vizualización del archivo en la pantalla del sistema Galaxy.

Cuadro 3.3: Especificación de caso de uso CU03

Nombre	Subir archivo
Identificador	CU04
Descripción	Permite al usuario subir los archivos al sistema Galaxy.
Actores	Investigador
Pre-Condiciones	-
Flujo de eventos básicos	a) El sistema muestra un formulario, con las opciones de escribir URL o subir directamente desde la computadora. b) EL usuario selecciona una opción e indica sus archivos. d) Se cargan los archivos en el historial. e) Ejecutar CU02
Flujo de eventos alternativo	a) Si el archivo por alguna razón no es cargado en el servidor, Galaxy muestra el archivo de un color diferente advirtiendo la situación.
Post-Condición	Archivo cargado en el sistema Galaxy.

Cuadro 3.4: Especificación de caso de uso CU04

Nombre	Guardar archivo
Identificador	CU05
Descripción	Permite al usuario guardar un archivo que esta en el historial de Galaxy en una unidad externa o en la computadora.
Actores	Investigador
Pre-Condiciones	CU02 - CU04
Flujo de eventos básicos	a) El sistema para cada archivo muestra un icono de diskette indicando la opcion de guardar el archivo. b) EL usuario selecciona la opción de guardar un archivo y se realiza la descarga de éste. d) El sistema muestra cuando finaliza la descarga.
Flujo de eventos alternativo	a) Si hay algún problema en el proceso de descarga, el sistema lo indicará por pantalla.
Post-Condición	Archivo guardado en la unidad de destino.

Cuadro 3.5: Especificación de caso de uso CU05

Nombre	Eliminar archivo
Identificador	CU06
Descripción	Permite al usuario Eliminar un archivo que se encuentra en el historial de Galaxy.
Actores	Investigador
Pre-Condiciones	CU02 - CU04
Flujo de eventos básicos	a) El sistema para cada archivo muestra un icono (X) indicando la opcion de eliminar el archivo. b) EL usuario selecciona la opción de eliminar un archivo y se efectúa la eliminación de éste. d) El sistema muestra el historial sin el archivo eliminado.
Flujo de eventos alternativo	a) Si hay algún problema en el proceso de eliminación, el sistema lo indicará por pantalla.
Post-Condición	Archivo eliminado del historial de Galaxy.

Cuadro 3.6: Especificación de caso de uso CU06

3.3. Estudio de viabilidad

A continuación se muestra el estudio de viabilidad del proyecto a nivel operacional, técnico y económico. Un análisis basado en los requerimiento y en la solución que se desea implementar.

En esta sección se define la viabilidad del proyecto. Sabremos si está disponible la tecnología, los recursos económicos y todas las condiciones como para que se pueda llevar a cabo este proyecto.

3.3.1. Viabilidad operacional

Debido al poco conocimiento de los usuarios (alumnos, docentes e investigadores) en la ejecución de programas a través de terminal y en ocasiones conocimiento nulo de sistemas tipo Unix, nace este proyecto como una necesidad de masificar el uso de las herramientas en el laboratorio de Diseño Racional de Fármacos. Se pretende cambiar este escenario implementado Galaxy, generando una interfaz web para nuestras herramientas. Por lo tanto, para el uso y ejecución de las herramientas, todo usuario deberá tener conocimientos mínimos de navegación web y por supuesto definir los parámetros de acuerdo a la tarea a realizar.

El proyecto es totalmente viable, desde el punto de vista operacional, debido al conocimiento, manejo e interacción diaria que tienen los usuarios con sistemas web aún más complejos que el que se pretende implementar.

3.3.2. Viabilidad técnica

La viabilidad técnica se ha estudiado desde dos enfoques. Primero, un análisis sobre los conocimientos que debe tener el desarrollador para implementar el proyecto y luego, un análisis enfocado a la infraestructura a nivel de hardware computacional requerido para la implementación del proyecto.

Desarrollo

El desarrollo de este proyecto se realiza en conjunto con el Dr. Alexis Salas Burgos académico del Departamento de Farmacología y del programa de Magister en Bioquímica y Bioinformática de la Universidad de Concepción. Se ha desarrollado un entorno de subversiones con GIT, lo que permite profesionalizar la generación de nuevas versiones y manejo de pruebas de implementación.

Para obtener los conocimientos necesarios y comprender el tema que se está realizando, se definen horarios para asistir a diversos cursos, clases, seminarios y charlas realizadas por el Dr. Alexis Salas o por otras personas con mayor conocimiento en las áreas de la biología y bioquímica; específicamente en el área del docking molecular, diseño de fármacos y/o bioinformática. Con esta

metodología y los tiempos destinados para la implementación del Proyecto hacen considerar viable la obtención de conocimientos, con respecto a la teoría.

Por otra parte las herramientas a utilizar, al ser de código libre se tienen a disposición comunidades, grupos que dan soporte y documentación completa a estas herramientas. Además, considerando la gran cantidad de tutoriales en línea y la bibliografía a utilizar, se considera viable obtener las capacidades necesarias para llevar a cabo este proyecto.

Hardware

Los cálculos de docking molecular requieren de gran cantidad recursos computacionales debido a la búsqueda exhaustiva que realizan los algoritmos. Además la implementación de proyectos como este demanda procesamiento gráfico 3D, lo que hace necesario contar con potentes tarjetas de video, adicional a la gran cantidad de recursos de memoria y procesamiento. Computadoras básicas o de escritorio no serían suficiente para la ejecución optima de este proyecto. Sí, lo serían varias computadoras organizadas de tal manera que brinden capacidades muy altas. Estamos hablando de un entorno de computación de alto rendimiento (High Performance Computing (HPC)). Este tipo de ambientes de trabajo, sistemas clúster son primordiales en laboratorios de desarrollo e investigación, como el Laboratorio de Diseño Racional de Fármacos de la Universidad de Concepción. Tomando en cuenta esto y la relación costo versus utilidad que representan estos equipos, se considera viable la adquisición de este tipo de Hardware.

3.3.3. Viabilidad económica

El objetivo es ordenar y sistematizar la información monetaria recabada para su análisis y evaluación de rentabilidad. La viabilidad económica del proyecto va a depender de la rentabilidad que este pueda tener. A continuación se analiza mediante los criterios de Valor Actual Neto (VAN) y la Tasa Interna de Retorno (TIR).

Inversión inicial

Todos los egresos necesarios para el inicio de este proyecto corresponden a la inversión inicial. La tabla 3.7 muestra los recursos que son requisito para partir con la implementación. Responden a una estación de trabajo y el hardware necesario para la ejecución de cálculos que demandan una

gran cantidad de memoria y procesamiento.

Recurso	Monto(\$)
Escritorio Computación	80.000
Silla de Escritorio	150.000
Computadores de Alto Rendimiento	5.000.000
Pantalla Led de Alta Resolución	700.000
Periféricos (Teclado, mouse, audio)	200.000
UPS y reguladores	200.000
Total	6.330.000

Cuadro 3.7: Listado de recursos como inversión inicial

Egresos de operación

Se consideran como egresos el arriendo del lugar de trabajo. Mensualmente se deberá pagar un monto de \$300.000. Además, se cancelará la suma de \$200.000 mensuales, equivalentes a los servicios básicos necesarios para la implementación del proyecto.

Egreso	Monto(\$)
Pago arriendo lugar de trabajo	3.600.000
Egresos servicios basicos (Luz, agua, internet, etc)	2.400.000
Total	6.000.000

Cuadro 3.8: Listado de egresos anual del Proyecto

Ingresos de operación

Los ingresos serán considerados después del primer año de implementación del proyecto y corresponden al 15 % de los ingresos del proyecto PharScan Transporter, del cual es parte este trabajo. A continuación se detallan los ingresos totales del proyecto PharScan Transporter:

Los productos ofrecidos están dentro de dos categorías: 1) producto finales y 2) servicios de desarrollo en conjunto con laboratorios de investigación de transportadores. Producto 1: Informes Técnicos. La cantidad de informes disponibles al final del desarrollo de la plataforma corresponderá a 100. Con un valor de 1.200 US\$/reporte. Bajo el supuesto de que en el primer año el 5% de los grupos de trabajo compra al menos 5 informes por año, se traduce en 180.000 US\$/año. Cabe destacar que cada año será desarrollada la estructura de una nueva familia de proteínas transportadoras, sumando al año 5 un portafolio de 7 familias de proteínas transportadoras, y por consiguiente un mercado potencial de 1.500 grupos de investigación.

Producto 2. Venta de paquetes de moléculas pequeñas identificadas. Esto se realizará a través del uso de la aplicación de barrido masivo de base de datos. Se ofrecerá a los equipos científicos el reporte que incluya las mejores 50 moléculas que interactúen con el transportador específico. Se estima una venta de 10 reportes a un precio de 10.000 US\$/reporte. Nos daría ingresos por 100.000 US\$/año.

Ingreso	Monto(\$)
15 % del Ingreso total Proyecto PharmScan Transporter	\$21,000,000
Total	\$21,000,000

Cuadro 3.9: Listado de ingresos anual del Proyecto

Flujo de caja

Concepto	0	1	2
Ingresos	0	21,000,000	21,000,000
Egresos	0	-6.000.000	-6.000.000
Inversión	-6.330.000	0	0
Flujo	-6.330.000	15,000,000	15,000,000

Cuadro 3.10: Flujo de caja en un periodo de dos años

La tabla 3.10 muestra el flujo de caja del proyecto en relación a 2 años proyectados. Se aprecia que rápidamente se recupera la inversión inicial del proyecto.

Valor Actual Neto (VAN)

A partir de la comparación entre el Valor Actual de los flujos futuros del proyecto y la inversión realizada, y de acuerdo a los Ingresos (Y), Egresos (E) e Inversión (I_0) detallados anteriormente, se determinará si se acepta el proyecto. Se determina en un periodo de 2 años a la tasa de 5% anual.

$$VAN = \sum_{t=1}^2 \frac{Y_t}{(1 + 0,05)^t} - \sum_{t=1}^2 \frac{E_t}{(1 + 0,05)^t} - I_0 \quad (3.1)$$

$$VAN = \sum_{t=1}^2 \frac{Y_t - E_t}{(1 + 0,05)^t} - I_0 \quad (3.2)$$

$$VAN = 27891156,29 - 6330000 \quad (3.3)$$

$$VAN = 21561156,47 \quad (3.4)$$

A través del Valor Actual Neto (VAN) se han considerado todos los flujos del proyecto y el Valor del dinero en el tiempo. El VAN es mayor a cero, por lo tanto, el proyecto es aceptado. Se muestra una ganancia sobre lo invertido por el equipo de trabajo.

Tasa Interna de Retorno (TIR)

La Tasa Interna de Retorno (TIR) permitirá medir la rentabilidad del proyecto como un porcentaje, ésta es la tasa más alta exigible. Si la TIR es mayor o iguala a la tasa de descuento aplicada al proyecto 5%, diremos que el proyecto es recomendable.

A partir de las ecuaciones definidas por el Valor Actual Neto (VAN), igualando los ingresos con los egresos más la inversión inicial y dejando la tasa de interés como una variable (r) se puede calcular la Tasa Interna de Retorno (TIR).

$$VAN = \sum_{t=1}^2 \frac{Y_t - E_t}{(1 + r)^t} - I_0 = 0 \quad (3.5)$$

Despejando lo anterior se obtiene una expresión para la tasa r :

$$\frac{15000000}{(1 + r)} + \frac{15000000}{(1 + r)^2} - 6330000 = 0 \quad (3.6)$$

$$6330000(r + 1)^2 - 15000000(r + 1) - 15000000 = 0 \quad (3.7)$$

reemplazando $x = (1 + r)$:

$$6330000x^2 - 15000000x - 15000000 = 0 \quad (3.8)$$

x tiene el valor:

$$x = \frac{-(-15000000) \pm \sqrt{-15000000^2 - 4 * 6330000 * (-15000000)}}{2 * 15000000} \quad (3.9)$$

$$x_1 = 1,3197 \quad (3.10)$$

$$x_2 = -0,31975 \quad (3.11)$$

retornando $x = (1 + r)$

$$r_1 = 1,3197 - 1 = 0,3197 = 31,97\% \quad (3.12)$$

$$r_2 = -0,31975 - 1 = -1,31975 = -131,975\% \quad (3.13)$$

Considerando la mayor tasa $r_1 = 31,97\%$, la Tasa Interna de Retorno (TIR) nos indica que el proyecto es rentable, puesto que, la TIR supera la tasa de descuento aplicada al proyecto 5% . Podemos decir que el proyecto debería ser recomendado.

Capítulo 4

Solución

El objetivo es llevar a cabo una solución, a través de la implementación y desarrollo de Galaxy. Incorporando algoritmos generalmente optimizados y que permitan el diseño de metodologías complementarias y automatizadas de preparación de archivos para docking regulares y masivos, permitiendo su automatización y comparación. Así como también la incorporación de algoritmos de análisis de resultados de docking.

El marco Galaxy debe permitir una ejecución fácil y eficiente de los programas que se incorporan, para esto es necesario una interfaz web que además permita la visualización gráfica de resultados y de las estructuras tridimensionales. Este capítulo muestra una visión general sobre la incorporación de los algoritmos de docking a un sistema Galaxy, con el objetivo de dar a conocer cómo se plantea cada uno de los elementos asociados.

4.1. Flujo de ejecución de los programas docking a través de Galaxy

El sistema Galaxy será el encargado de gestionar los datos y procesos de ejecución de los programas, por lo que a continuación se describe la forma en la que se propone regular este proceso. En un ambiente general, el usuario de a cuerdo a sus necesidades tendrá que seleccionar uno de los programas incorporados a Galaxy. Independiente de la herramienta que seleccione deberá también ingresar los archivos de configuración para ésta. La ejecución de los programas de docking, estará condicionada a ingresar al sistema los correctos archivos de configuración de cada progra-

ma . Si los archivos y los formatos de éste no son los correctos o no existen, no se permitirá la ejecución de ningún programa de docking. Si el ingreso de los archivos de configuración son los correctos se procederá a la ejecución de un programa. Si el programa seleccionado es el *Programa 1*, se mostrará la estructuras 3D (con todas sus grillas y cavidades). La continuidad de ejecución del *Programa 1* estará condicionada al ingreso de un último dato por parte del usuario. Éste es el número de la cavidad seleccionada. Si el número de la cavidad es ingresado correctamente, el *Programa 1* continuará con la ejecución. Finalmente la ejecución de este programa llega a su fin una vez que termina con la preparación de los archivos para el docking. Si el número de la cavidad no es ingresado correctamente, el usuario sólo podrá seguir manipulando la estructura 3D en pantalla. En el caso de que el programa seleccionado sea el *Programa 2*, éste terminará con la ejecución una vez que finalice con la creación de los gráficos de los resultados de docking.

La Figura 4.1 muestra un el diagrama de flujo que representa el flujo de ejecución de los programas docking a través de Galaxy. Las siguientes secciones, documentan el enfoque resolutivo para implementar lo anteriormente expuesto.

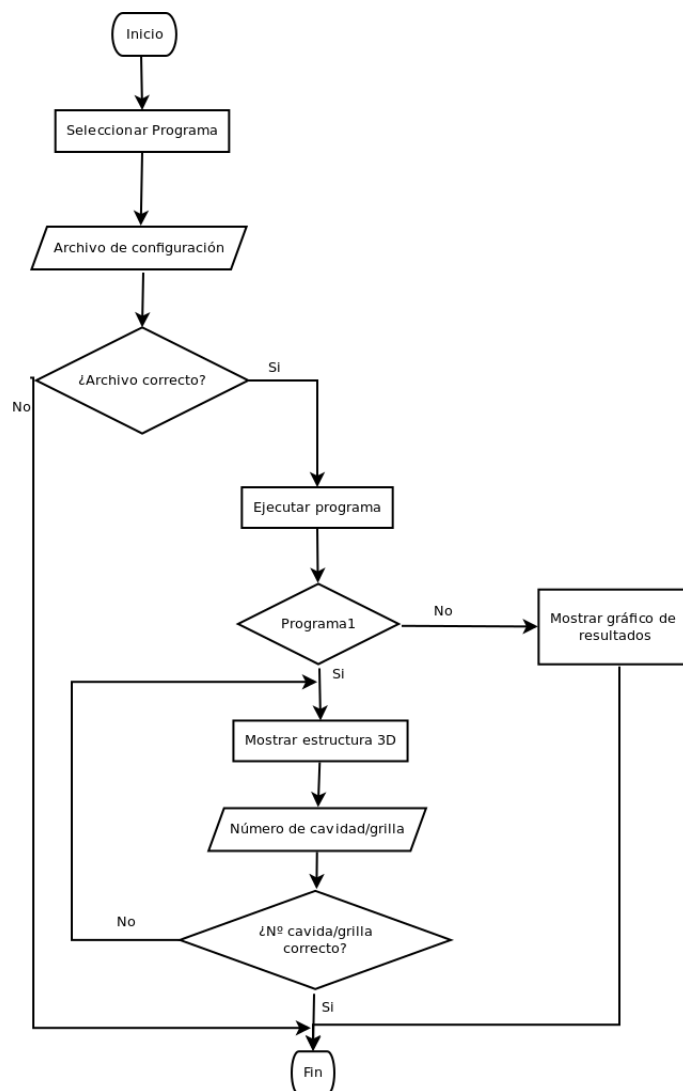


Figura 4.1: Diagra flujo de la ejecución de programas a través de Galaxy

4.2. Instancia de Galaxy

Para desarrollar, añadir nuestras herramientas y ejecutar en un servidor local Galaxy, es necesario instalar nuestra propia instancia de Galaxy. Para esto se ha dispuesto crear una maquina virtual, utilizando alguna de las herramientas de virtualización existentes. Para el caso de este proyecto se pretende configurar una maquina virtual con las características necesarias para el período de desarrollo del sistema. Una vez finalizada la etapa de desarrollo la herramienta de virtualización permite migrar nuestras maquinas virtuales a otros entornos de trabajo. En este caso, finalizada

esta etapa se pretende trasladar la maquina virtual de Galaxy a un entorno HPC (High Performance Computing) y de esta forma hacer frente a los cálculos con mayor demanda de los programas de docking. El usuario puede usar un clúster de computadoras previamente configurado. Todo esto en forma transparente para él. Se estima que la mejor herramienta de virtualización para este proyecto es VM VirtualBox, esta herramienta además de contar con una versión que es software libre, ya ha sido utilizada en otros proyectos por el autor de este trabajo. La tabla 4.1 muestra la configuración básica para la máquina virtual en el período de desarrollo.

Requerimiento	Configuración
Sistema Operativo	Linux
Procesador	2 núcleos de procesador
Memoria	2 GB de RAM
Almacenamiento	Disco Duro 30 Gb.
Gráficos	Acelerar gráficos 3D (Admitir OpenGL 2.1 y Xrender)
Red	Utilizar misma tarjeta S.O anfitrión

Cuadro 4.1: Configuración básica máquina virtual en etapa de desarrollo

4.2.1. Añadir los programas a Galaxy

Para añadir los programas de docking molecular a nuestra instancia Galaxy el programador debe crear un archivo de configuración para cada programa, llamado “archivo de configuración del programa”. Este archivo es muy importante, es el encargado de proporcionar la información a Galaxy para que identifique y pueda invocar las nuevas herramientas. El archivo de configuración del programa, es un archivo XML el cual debe contener atributos de la herramienta que se desea agregar. Entre estos atributos, hay algunos que deben ser incluidos obligatoriamente, de otro modo Galaxy no reconoce los nuevos programas y no permite su ejecución.

Listing 4.1: Ligplotdock6.xml

```

1 <tool id="Programa1" name="Ligplotdock6" version="0.3">
2 <description>From a molecular docking perform a graph
3 analysis in aminoacids or atom levels.</description>
```



```

4
5 ...
6 <command> Definicion de los parametros de entrada</command>
7 ...
8
9 ...
10 Definicion de interfaz de usuario
11 ...
12
13 ...
14 Definicion de las salidas
15 ...
16
17 fin

```

Listing 4.2: Pydock6.xml

```

1 <tool id="Programa1" name="pydock6" version="2.3">
2 <description>Prepare files for molecular docking for Vina and Dock6.</description>
3
4 ...
5 <command> Definicion de los parametros de entrada</command>
6 ...
7
8 ...
9 Definicion de interfaz de usuario
10 ...
11
12 ...
13 Definicion de las salidas
14 ...
15
16 fin

```

Los trozos de código anteriores muestran cómo deben ser definidos los archivos de configuración de los programas del proyecto. Esta es la estructura general para cada herramienta (para detalles A

ver configuración básica de cada programa) . Para el caso del Programa1, se asigna un identificador (id=Programa1) el cual es único y distinto para cada programa. A través de este identificador Galaxy identifica el programa al momento de ser solicitado. En la cabecera también se define el nombre y la versión de la herramienta. Luego se agrega una descripción la cual puede ser omitida, pero se recomienda no hacer esto, ya que son de ayuda para el usuario al momento de iniciar Galaxy. Lo mismo sucede para el programa2. El archivo de configuración del programa también contiene otras definiciones, las que serán abordadas en las siguientes secciones.

4.2.2. Parámetros de los programas de docking.

La definición de los parámetros no es una tarea fácil para el programador. Galaxy necesita saber cuáles son los parámetros de entrada y además cuáles son los parámetros de salida. Pero exige que el programador le pase los parámetros de salida (pueden ser vacíos) como parámetros de entrada al programa que se desea ejecutar. De esta forma Galaxy guarda las salidas de los programas en un Historial. Para el usuario esto es transparente, ya que la única misión de él es ingresar los parámetros de entrada y luego puede ver las salidas en su Historial. La tabla 4.2 muestra los parámetros de uno de los programas de docking.

Parámetro	Dominio
Receptor.	$\{(x, y, z) : x, y, z \in \mathbb{R}\}$
Ligando.	$\{(x, y, z) : x, y, z \in \mathbb{R}\}$
El espacio.	$\{x : x \in \mathbb{R} \wedge x > 0\}$
Número de la cavidad seleccionada.	$\{x : x \in \mathbb{Z} \wedge x > 0\}$
Número de cavidades.	$\{x : x \in \mathbb{Z} \wedge x > 0\}$

Cuadro 4.2: Parámetros de entrada para el *Programa1*

4.2.3. Interfaz de usuario y navegación

Uno de los requerimientos de este proyecto es una interfaz de usuario sencilla con una página de inicio, pero manteniendo la estructura y diseño de una instancia Galaxy.

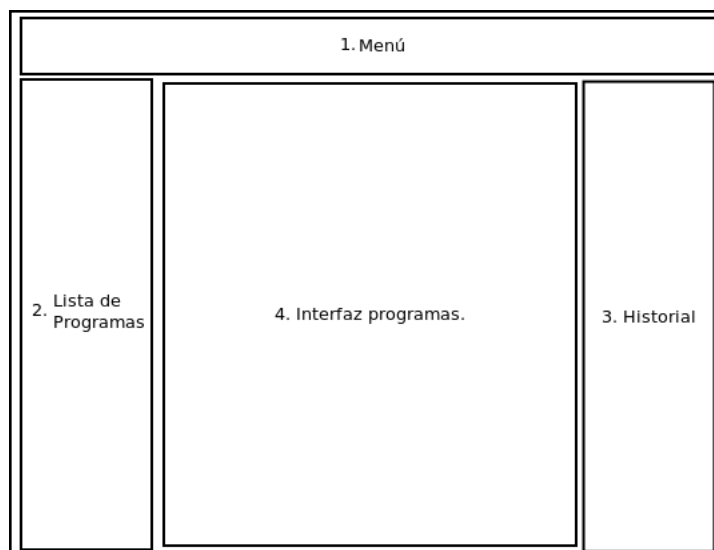


Figura 4.2: Esquema especificación de Interfaz.

El esquema de la figura 4.2 muestra como es en términos generales una interfaz de un sistema Galaxy. Se destacan las siguientes areas:

El área 1: Menú. Incluye opciones de volver a la página de inicio, ingresar y salir de una cuenta, entre otras.

El área 2: Muestra la lista de programas que se agregan al sistema. Para el caso de este proyecto son *Programa1* y *Programa2*.

El área 3: Historial. Es muy importante. Incluye una lista de todos los archivos subidos a Galaxy, además de los archivos de salida de cada programa. También están los botones con las opciones para cada archivo (Guardar, eliminar, mostrar en pantalla.)

El área 4: Esta área a diferencia de las otras es la única que cambia por completo. Incluye la interfaz de los programas y también es el área de despliegue de datos.

Para definir la interfaz de usuario se utilizará el archivo de configuración de cada programa. Para el desarrollo de la esta página de inicio se utilizará HTML y CSS. Conservando la estructura y diseño de una instancia Galaxy y agrgando algunas imagenes de los mismos programas de docking la página de inicio debería ser similar a la que se presenta en la Figura 4.3.

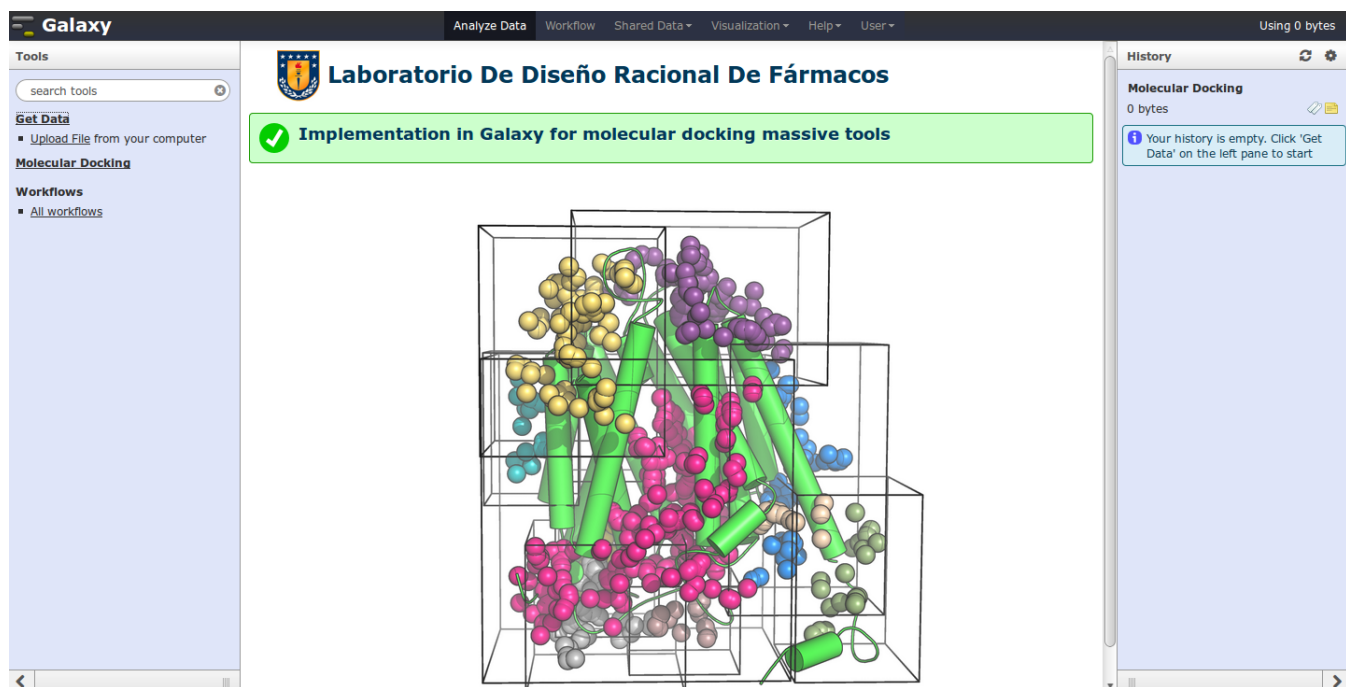


Figura 4.3: Página de inicio del Galaxy.

Conservar la estructura del marco Galaxy tiene varias ventajas. Galaxy se encarga de administrar las cuentas de acceso, gestiona el historial de cada usuario, es muy intuitivo y fácil de usar. La correcta programación del archivo de configuración del programa condicionará una correcta navegación del usuario.

4.2.4. Visualización de estructuras 3D.

Para la construcción de la aplicación que permita la visualización de estructuras 3D, se pretende implementar la biblioteca Javascript Jmol.js. Jmol es un visor Java de código abierto para estructuras químicas en tres dimensiones. <http://www.jmol.org/>. La biblioteca Jmol.js facilita el diseño y creación de una página web que con tenga una aplicación Jmol. Proporciona las funciones necesarias para la inserción de una aplicación y para la gestión de sus controles y diseño a través de HTML y CSS. Esta biblioteca también será usada para la creación de una interfaz de usuario que permita su interacción con las estructuras 3D en la aplicación Jmol. Otra de las ventajas de utilizar Jmol es que no requiere OpenGL o hardware 3D específico para usar las aplicaciones. Cuenta con su propio motor gráfico un z-buffer implementado en Java y diseñado para representar moléculas.

Esto no quiere decir que una tarjeta gráfica 3D no mejore el rendimiento ni tampoco que los requerimientos de hardware descritos en la tabla 4.3 no puedan ser mayores. Como para el caso de este proyecto que son muy superiores. A continuación se presentan los requerimientos de hardware y software básicos para la utilización de aplicaciones Jmol.

Requerimiento	Configuración
Sistema Operativo	Cualquier sistema que permita Java 1.4. Preferencia Linux
Biblioteca	Jmol.js
Navegador	Cualquiera que cuente con comunicación LiveConnect entre javascript y las miniaplicaciones Java.
Procesador	1 procesador 750 MHz
Memoria	256 MB de RAM
Almacenamiento	Disco Duro 30 Gb.

Cuadro 4.3: Configuración básica para la utilización de aplicaciones Jmol

Disponiendo con un entorno de trabajo con la configuración de la tabla anterior y añadiendo las salidas del Programa1 (Estructura y grillas alineadas), más la correcta programación de la página. El usuario debería ver y manipular estructura 3D como las que se presentan a continuación.

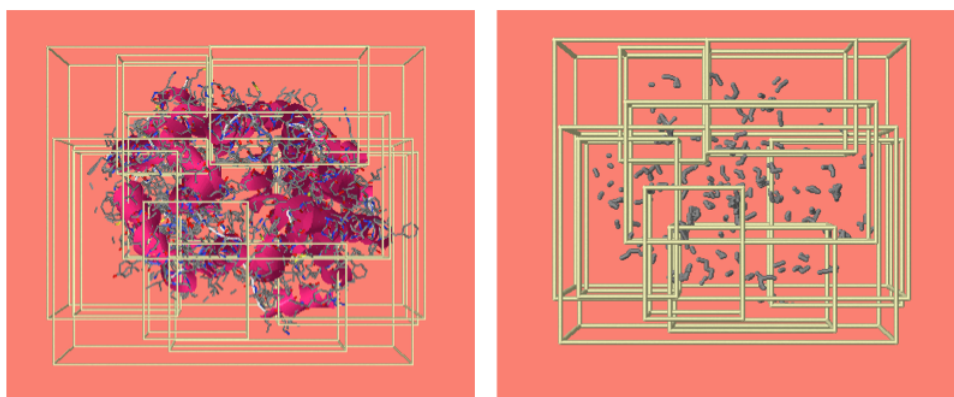


Figura 4.4: Visualización 3D estructura y todas las grillas alineadas.

A través de esta aplicación el usuario tiene la opción de seleccionar una grilla/cavidad (ver Figura 4.5). De esta forma el programa1 termina la ejecución exitosa y con los correspondientes

archivos de configuración para el docking.

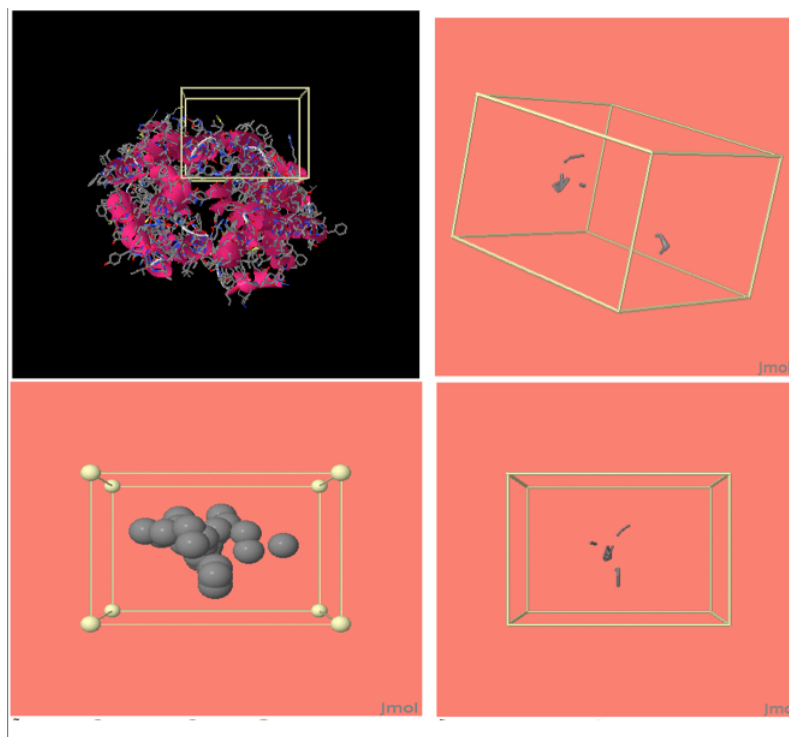


Figura 4.5: Visualización 3D de la selección de una grilla.

4.3. Software

El desarrollo de los programas de docking molecular (*Programa1* y *Programas2*) ha sido realizado mediante la programación de bibliotecas y módulos utilizando el lenguaje de programación Python. Python destaca en la Bioinformática y en la Biología computacional. Por ser un lenguaje de alto nivel con el mayor desarrollo en el área biológica. Si lo comparamos con otros lenguajes de programación como por ejemplo: Ruby, Java, C, etc. Python va un paso adelante en el desarrollo de herramientas bioinformática. Científicos e investigadores lo prefieren ya que la curva de aprendizaje es muy rápida, la utilización de módulos no es compleja y se puede encontrar variada documentación de estos módulos en la web y la literatura.

4.3.1. Dependencias de software

Si bien es cierto gran parte de las funcionalidades de estos programas se ha desarrollado, pero también se ha hecho uso de algunas herramientas existentes y módulos externos que permiten el exitoso desarrollo de los algoritmos de docking molecular. La tabla 4.4 muestra dependencias de los programas necesarias para una correcta ejecución.

Software/ Módulos	Versión	Descripción
Pymol	1.6	Visor molecular 3D. (<i>Programa1</i>)
Jmol.js	13.0	Biblioteca Jmol para gestionar página con visor 3D. (<i>Programa1</i>)
AutoDockTools	4.2	Herramientas de Autodock. Utilizado para preparar receptor y ligando <i>Programa1</i>)
DMS	3.	Utilizada para generar archivo ejecutable en Pymol. Cálculo de superficie molecular. (<i>Programa1</i>)
NumPy	1.7	Módulo matemático que permite tener la funcionalidad de Matlab en Python. Usado en el manejo de las coordenadas x y z <i>Programa1</i>)
Matplotlib	1.2	Biblioteca de graficos 2D. Incluye dibujo vectorial (SVG) (Gráficos (<i>Programa2</i>))
NetworkX	1.7	Biblioteca de python utilizada en la construcción de los modelos de redes. Grafos. (<i>Programa2</i>)

Cuadro 4.4: Dependencias de software

4.3.2. Requisitos de hardware

A continuación se definen condiciones de hardware optimas para obtener una ejecución de alta calidad, además para el cálculo de docking se considera la ejecución de los Software Vina y DOCK6. En este caso la necesidad de hardware se relaciona directamente con la cantidad de moléculas de prueba. Los requerimientos descritos en la tabla 4.5 responden a cálculos de bibliotecas masivas.

Hardware	Requerimiento
Procesador de Cálculo	2 x 2 CPU 2,4 GHz (16 cores)
Memoria	2 x 24 GB SDRAM ECC DDR3 a 1,333 MHz
Almacenamiento	2 x 2TB a 3 GB/S y 7200 RPM
Gráficos	2 x Radeon HD 5770 ATI 16b GDDRS
Red	Swith 16 port 1GB/s

Cuadro 4.5: Requisitos de hardware. Clúster de computadoras

Capítulo 5

Pruebas

Durante el desarrollo de este proyecto se llevaron a cabo diferentes pruebas, las que fueron supervisadas por el Dr. Alexis Salas. A continuación se describen la definición de pruebas necesarias para constatar el correcto funcionamiento tanto de los módulos individuales y del sistema completo.

5.1. Pruebas Unitarias

Los nombres de los programas de docking son *Pydock6* (Programa1) y *Ligplotdock6* (Programa2). El primero es el encargado de generar los archivos de docking molecular para Vina y Dock6. El segundo es el encargado de realizar análisis de los resultados de docking (Gráficos).

5.1.1. Parámetros de entrada

La prueba unitaria de clave *Parámetros de Entrada* tiene por objetivo probar la correcta lectura de parámetros al ejecutar los programas a través de terminal.

Programa involucrado	Pasos
<i>Pydock6</i>	Ejecutar el programa <i>Pydock6</i> enviando los archivos y datos de entrada.
<i>Ligplotdock6</i>	Ejecutar el programa <i>Ligplotdock6</i> enviando los archivos y datos de entrada.

Resultados esperados

Los archivos y datos deben ser recibidos en forma correcta y luego comenzar la asignación de variables del programa.

Resultados incorrectos

La definición de parámetros del programa se ha realizado de forma inadecuada y no acorde a la notación estándar definida. Se aborta la ejecución.

Resultados obtenidos

Se envía un mensaje de ayuda indicando la correcta definición de parámetros del programa.

5.1.2. Generar archivos docking.

La prueba unitaria de clave *generar archivos docking* tiene por objetivo probar la correcta creación de los archivos para el docking.

Programa involucrado	Pasos
<i>Pydock6</i>	Ejecutar el programa <i>Pydock6</i> creando las carpetas y los correspondientes archivos de configuración.

Resultados esperados

Los archivos y datos deben ser generados correctamente por el programa.

Resultados incorrectos

La creación de de los archivos de docking no se ha completado. Se aborta la ejecución del programa.

Resultados Obtenidos

Se envía un mensaje con la descripción del error ocurrido en la creación de los archivos.

5.1.3. Generar gráficos

La prueba unitaria de clave *Generar gráficos* tiene por objetivo probar la correcta creación de los gráficos de los análisis para los resultados de docking molecular.

Programa involucrado	Pasos
<i>Ligplotdock6</i>	Ejecutar el programa <i>Ligplotdock6</i> creando los gráficos de resultados.

Resultados esperados

Los gráficos deben ser creados en forma correcta por el programa.

Resultados Incorrectos

La creación de de los graficos no se ha completado. Se aborta la ejecución del programa.

Resultados obtenidos

Se envía un mensaje con la descripción del error ocurrido en la creación de los graficos.

5.1.4. Visualizar estructura 3D.

La prueba unitaria de clave *Visualizar estructura 3D* tiene por objetivo probar la correcta visualización de los modelos 3D

Programa involucrado	Pasos
<i>Pydock6</i>	Ejecutar el programa <i>Pydock6</i> Elaborar el archivo definiendo los modelos que deben ser cargados.
<i>Jmol</i>	Leer el archivo de los modelos y cargarlos en la aplicación.

Resultados esperados

Se deben visualizar correctamente las estructuras 3D.

Resultados Incorrectos

No se cargan todos los modelos y no se visualiza la estructura 3D.

Resultados Obtenidos

Se envía un mensaje con el nombre del modelo que no ha cargado.

5.2. Pruebas de integración

En esta Sección se presentarán pruebas de integración para validar la correcta comunicación entre módulos del sistema Galaxy.

5.2.1. Inicialización de programas en Galaxy

La prueba de integración de clave *Inicialización de programas en Galaxy* tiene por objetivo probar la correcta lectura de parámetros al ejecutar los programas desde Galaxy.

Programa involucrado	Pasos
<i>Galaxy</i>	Ejecutar <i>Galaxy</i> permitiendo el ingreso de los archivos y datos de entrada.
<i>Pydock6</i>	Ejecutar el programa <i>Pydock6</i> enviando los archivos y datos de entrada.
<i>Ligplotdock6</i>	Ejecutar el programa <i>Ligplotdock6</i> enviando los archivos y datos de entrada.

Resultados esperados

Los archivos y datos deben ser recibidos en forma correcta y luego comenzar la asignación de los valores a las variables de los programas.

Resultados incorrectos

No se recibieron en forma correcta los datos y archivos.

Resultados Obtenidos

Los parámetros ingresados por el usuario a través de la interfaz, han sido asignados correctamente.

5.2.2. Ejecución de programas desde Galaxy

La prueba de integración de clave *Ejecución de programas desde Galaxy* tiene por objetivo probar la correcta ejecución de los programa desde Galaxy.

Programa involucrado	Pasos
<i>Galaxy</i>	Ejecutar <i>Galaxy</i> enviando la orden de ejecución.
<i>Pydock6</i>	Ejecutar el programa <i>Pydock6</i> enviando los archivos de docking para Vina y DOCK6.
<i>Ligplotdock6</i>	Ejecutar el programa <i>Ligplotdock6</i> enviando los gráficos de los análisis de resultados.

Resultados esperados

Los programas deben ser ejecutados correctamente desde Galaxy y cargar las salidas en el historial.

Resultados incorrectos

Los archivos de salida se cargan con errores en el historial. La ejecución no es la correcta.

Resultados obtenidos

Archivos cargado en forma exitosa en el historial.

Capítulo 6

Conclusión

Es importante el diseño e implementación de software que permita tener un control sobre cada una de las etapas del proceso de docking molecular. Pero también es importante poder masificarlo. Este trabajo se ha centrado en aquello y deja la base para trabajos futuros que puedan complementarlo. El trabajar en un proyecto como éste, no es algo común para un estudiante del Departamento de Sistemas de Información. Sin embargo, para el autor ha sido una tarea enriquecedora desde todos los puntos de vista, ampliando de gran manera los conocimientos en distintas disciplinas, encontrando un campo laboral muy interesante. La investigación realizada durante todo este tiempo, no fue un trabajo fácil, pero que dio muchos resultados para el autor. Aquí se detallan algunas de las nuevas herramientas adquiridas después de haber trabajado en este proyecto:

- Conocimientos de la biología, bioquímica y específicamente de docking molecular.
- Conocimientos en el área de Diseño Racional de Fármacos
- Conocimiento y aplicación de herramientas Bioinformáticas.
- Conocimiento y desarrollo de plataforma Galaxy.
- Conocimiento y aplicación de herramientas de programación con Python.
- Conocimiento sobre Clúster de computadoras.
- Conocimiento e implementación de visor 3D.

- Conocimiento de áreas de trabajo científico a través de lectura de artículos científicos e investigación.
- Mayor conocimientos y práctica en el idioma inglés.
- Composición de textos a nivel profesional mediante la herramienta $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.
- Utilización de la herramientas como *Autodock/Vina* para realizar Docking Molecular.
- Trabajo colaborativo a través de la herramienta gestor de versiones *Git*.

El resultado final de este proyecto permite la utilización de métodos computacionales por personas que no necesariamente necesiten conocimientos de programación e informática. Gracias a Galaxy se permitió una interfaz Web para la ejecución de herramientas creadas en python en forma transparente para el usuario de un clúster de computadoras. Es posible la realización de docking moleculares con compatibilidad entre interfaces, la caracterización de los complejos y analizarlos en una forma rápida y detallada entregando informe de los resultados que nos permiten diseñar y proponer experimentos de validación de los complejos propuestos. Además, Galaxy permitirá la accesibilidad, transparencia y reproducibilidad de los procesos que se lleven a cabo. Lo que es muy importante al momento de realizar una publicación.

Capítulo 7

Trabajos futuros y derivados

Galaxy entrega muchas posibilidades de personalización. La instancia desarrollada para el Laboratorio de Diseño Racional de Fármacos de la Universidad de Concepción se pretende seguir mejorando. Incorporando los programas de los cálculos de docking Vina y DOCK6 a nuestra plataforma. Además de la incorporación de nuevos programas como por ejemplo; herramientas que realizan cálculos de energía electrostática de complejos, ya que los software de docking no incluyen análisis específicos de estas energías que son de gran relevancia en las interacciones moleculares. Este trabajo puede servir como base para futuros alumnos memoristas. El sistema Galaxy desarrollado en este proyecto se puede seguir personalizando. Se podrían diseñar y añadir nuevos programas que ayuden a optimizar no tan solo las tareas de docking molecular, sino que también a otras tareas asociadas al diseño de fármacos. Se pueden seguir diseñando páginas con herramientas para la visualización de resultados, reportes e informes. Galaxy no restringe un lenguaje en particular. Alumnos con conocimientos en Perl, Ruby, Java, etc. Pueden también diseñar y añadir sus herramientas. De esta forma poder crear nuevos entornos y flujos de trabajo que puedan ser reutilizados por personas sin experiencia en programación e informática .

Apéndice A

Archivo de configuración de los programas

El archivo de configuración del programa en Galaxy cumple un rol fundamental. Aquí se han definido los parámetros de entrada, y las salidas de cada programa, además una interfaz para el ingreso de los archivos y parámetros de cada uno. A continuación se muestra el código XML de una configuración básica. En el código se aprecia como los parámetros de salida de los programas deben ser indicados como parámetros de entrada. De esta manera Galaxy carga las salidas en el Historial.

A.0.3. Código XML Programa1

Listing A.1: Pydock6.xml

```
1 <tool id="programa1" name="Pydock6" version="2.3">
2   <description>Prepare files for molecular docking for Vina and Dock6.</description>
3   <parallelism method="basic"></parallelism>
4   <command interpreter="python">
5     "pydock6.py"
6     "$pdbname2"
7     "$pdbname"
8     "$ncav"
9     "$spacing"
```

```

10     "$output1"
11     "$output2"
12     "$output3"
13     "$output4"
14 <requirements>
15     <requirement type="package"></requirement>
16 </requirements>
17 <inputs>
18     <param name="pdbname2" type="select" label="PDB file" help="" optional="true">
19
20         <option value="PDB" selected="true">PDB</option>
21     </param>
22     <param name="pdbname" type="select" label="Ligand DB" help="" optional="true">
23
24         <option value="PDB" selected="true">File</option>
25     </param>
26     <param name="ncav" type="integer" label="Number of Cavities" help="" optional="true"
27         value="10" />
28     <param name="spacing" type="float" label="Spacing" help="" optional="true" value="0.3
29         " />
30 </inputs>
31 <outputs>
32     <data name="output1" format="pdbqt" />
33     <data name="output2" format="pdbqt" />
34     <data name="output3" format="pdb" />
35     <data name="output4" format="pdb" />
36 </outputs>
37 <help>

```

A.0.4. Código XML Programa2

Listing A.2: Ligplotdock6.xml

```

1 <tool id="Programa1" name="Ligplotdock6" version="0.3">
2   <description>From a molecular docking perform a graph analysis in aminoacids or atom
   levels.</description>

```

```
3 <!-- <parallelism method="basic"></parallelism> -->
4 <command interpreter="python">
5   ligplotdock6.py
6   "$mol2file"
7   "$pdb_receptor"
8   "$molecule"
9   "$output2"
10  "$output3"
11 </command>
12 <inputs>
13   <param name="mol2file" type="data" format="mol2" label="Output Molecular docking"
14     help="" optional="true"/>
15   <param name="pdb_receptor" type="data" format="pdb" label="Receptor PDB file" help=""
16     optional="true"/>
17   <!-- <param name="molnumber" type="text" label="Molecule Filter (1-100)" help=""
18     optional="true" value="" /> -->
19   <!-- <param name="molnumber2" type="integer" label="Pose Filter (1-30)" help=""
20     optional="true" value="" /> -->
21 </inputs>
22 <outputs>
23   <data name="molecule" format="tmp" />
24   <data name="output2" format="png" />
25   <data name="output3" format="svg" />
26 </outputs>
27 <help>
28 }
```

A.1. Interfaz de usuario

A.1.1. Interfaz de usuario programa1

Pydock6 (version 2.3)

PDB file:
PDB ▲▼

Ligand DB:
File ▲▼

Number of Cavities:

Spacing:

Execute

Generated By:
felgutier

Based On:
Definition of "ligplotdock6"
2013
Felipe Gutiérrez M. felgutie@alumnos.ubiobio.cl

Figura A.1: Interfaz para el ingreso de parámetros y la ejecución del programa1.

A.1.2. Interfaz de usuario programa2

Ligplotdock6 (version 0.3)

Output Molecular docking:
Selection is Optional ▾

Receptor PDB file:
Selection is Optional ▾

Execute

Generated By:
felgutier

Based On:
Definition of "ligplotdock6"
2013
Felipe Gutiérrez M. felgutie@alumnos.ubiobio.cl

Figura A.2: Interfaz para el ingreso de parámetros y la ejecución del programa2.

A.1.3. Historial

History

Molecular Docking
3.9 MB

- 50: Ligplotdock6 on data 3 and data 38**
- 49: Ligplotdock6 on data 3 and data 38**
- 48: Ligplotdock6 on data 3 and data 38**
- 38: topsort_GridScore_200.mol2**
- 3: Glut16ns_nh_orient.pdb**
3,817 lines
format: txt, database: 2
uploaded txt file

```

CRYST1    0.000    0.000    0.000  90.
ATOM      1  N   MET P  1   -20.502
ATOM      2  CA  MET P  1   -20.987
ATOM      3  CB  MET P  1   -19.827
ATOM      4  CG  MET P  1   -20.188
ATOM      5  SD  MET P  1   -21.382
    
```

Figura A.3: Historial con datos programa2

Bibliografía

Implementaciones Bioinformáticas en Galaxy

(1) Blankenberg, D.; Gordon, A.; Von Kuster, G.; Coraor, N.; Taylor, J.; Nekrutenko, A. & G. T. (2010). Manipulation of FASTQ data with Galaxy. *Bioinformatics* 26(14), 1783-1785. [en línea] USA: Huck Institute for the Life Sciences. Recuperado el 5 septiembre del 2012 de: <http://bioinformatics.oxfordjournals.org/content/26/14/1783.full.pdf+html>

(2) Blankenberg, D.; Taylor, J.; Nekrutenko, A. & , G. T. (2011). Making whole genome multiple alignments usable for biologists. *Bioinformatics*. 27(17), 2426- 2428. [en línea] Recuperado el enero del 2013 de: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3157923/>.

(3) Blankenberg, D.; Von Kuster, G.; Coraor, N.; Ananda, G.; Lazarus, R.; Mangan, M.; Nekrutenko, A. & Taylor, J. (2010), Galaxy: a web-based genome analysis tool for experimentalists. *Curr Protoc Mol Biol* 19(10),1-21.[en línea] Pennsylvania: Pennsylvania State University. Recuperado el 30 de agosto del 2012 de: <https://www.google.cl/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&ved=0CD4QFjAB&url=http%3A%2F%2Fwww.researchgate.net%2Fpublication%2F41002147%2Ffile%2F0b495153563f4dd96.pdf&ei=g2TpUeGuCMjxiQLjw4GYAw&usq=AFQjCNECqEgvjNpgXq37eArwEOcSvlSp7Q&sig2=pFoVyczFbUzzY9j8AAGFsw&bvm=bv.49478099,d.cGE>

(4) Cesaroni, M.; Cittaro, D.; Brozzi, A.; Pelicci, P. G. & Luzi, L. (2008), CARPET: a web-based package for the analysis of ChIP-chip and expression tiling data. *Bioinformatics* 24(24), 2918-2920. [en línea] Oxford Index. Recuperado el 5 de enero del 2013 de: CARPET: a web-based

package for the analysis of ChIP-chip and expression tiling data. *Bioinformatics* 5) Giardine, B.; Riemer, C.; Hardison, R. C.; Burhans, R.; Elnitski, L.; Shah, P.; Zhang, Y.; Blankenberg, D.; Albert, I.; Taylor, J.; Miller, W.; Kent, W. J. & Nekrutenko, A. (2005), Galaxy: a platform for interactive large-scale genome analysis. *Genome Res* 15(10), 1451- 1455. [en línea] EE.UU: Centro de Genómica Comparativa y Bioinformática. Recuperado el 15 marzo del 2013 de: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1240089/>

(6) Schultheiss, S. J.; Busch, W.; Lohmann, J. U.; Kohlbacher, O. & Ratsch, G. (2009), KIRMES: kernel-based identification of regulatory modules in euchromatic sequences. *Bioinformatics* 25(16), 2126- 2133. [en línea] Alemania: Friedrich Miescher Laboratory of the Max Planck Society. Recuperado el 1 marzo del 2013 de: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2722996/>

Docking **molecular**

(7) Durrant, J. D. & McCammon, J. A. (2011), BINANA: a novel algorithm for ligand-binding characterization. *J Mol Graph Model* 29(6), 888- 893. [en línea] USA: University of California San Diego. Recuperado el 13 septiembre del 2012 de: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3099006/>

(8) Durrant, J. D. & McCammon, J. A. (2011), NNScore 2.0: a neural-network receptor-ligand scoring function. *J Chem Inf Model* 51(11), 2897- 2903. [en línea] USA: University of California San Diego. Recuperado el 18 septiembre del 2012 de: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3225089/ffnsectitle>

(9) Knox, C.; Law, V.; Jewison, T.; Liu, P.; Ly, S.; Frolkis, A.; Pon, A.; Banco, K.; Mak, C.; Neveu, V.; Djoumbou, Y.; Eisner, R.; Guo, A. C. & Wishart, D. S. (2011), DrugBank 3.0: a comprehensive resource for ómics' research on drugs. *Nucleic Acids Res* 39(Database issue), D1035- D1041. [en línea] Canada :University of Alberta. Recuperado el 16 de octubre del 2012 de: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3013709/>

(10)Koes, D. R. & Camacho, C. J. (2012), ZINCPharmer: pharmacophore search of the ZINC database. *Nucleic Acids Res.* 40 W409- W414. [en línea] USA: University of Pittsburgh. Recuperado el 05 noviembre del 2012 de: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3394271/>

(11) Morris, G. M.; Huey, R.; Lindstrom, W.; Sanner, M. F.; Belew, R. K.; Goodsell, D. S. & Olson, A. J. (2009), AutoDock4 and AutoDockTools4: Automated docking with selective receptor

flexibility. *J Comput Chem.* 30(16), 2785- 2791. [en línea] USA: The Scripps Research Institute. Recuperado el 14 de noviembre del 2012 en: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2760638/>

(12) McKee, T. and McKee J.R., (2003). *Bioquímica. La Base Molecular de la Vida* (3 ed) Extraído de: <http://librosgratisparauniversitarios.blogspot.com/2012/11/bioquimica-la-base-molecular-de-la-vida.html>

(13) Yedidi, R.S, Liu, Z; Wang, Y; Brunzelle, J.S; Kovari, I. A; Woster, P.M; Kovari, L. C; Gupta, D. 2012. Crystal structures of multidrug-resistant HIV-1 protease in complex with two potent anti-malarial compounds. *Biochem Biophys Res Commun.* 421(3) 413- 417. USA: Wayne State University School of Medicine. Recuperado el 20 de abril del 2013 de: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3351498/>

(14) Wang, Y.; Liu, Z.; Brunzelle, J.S.; Kovari, I. A.; Dewdney, T. G.; Reite, S. J.; Kovari, L. C. 2011. The higher barrier of darunavir and tipranavir resistance for HIV-1 protease. *Biochem Biophys Res Commun.* 412(4) 737-742. [en línea] USA: Wayne State University School of Medicine. Recuperado el 20 de abril del 2013 de: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3188455/>

(15) Trott, O. & Olson, A. J. (2010). AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J Comput Chem* 31(2), 455-461. [en línea] CA: El Instituto de Investigación Scripps. Recuperado el 15 de junio del 2013 de: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3041641/>

(16) Xie, L.; Xie, L.; Kinnings, S. L. & Bourne, P. E. (2012). Novel computational approaches to polypharmacology as a means to define responses to individual drugs. *Annu Rev Pharmacol Toxicol* 52, 361-379. [en línea] New York: The City University of New York Recuperado el 7 de noviembre del 2012 de: <http://www.annualreviews.org/doi/full/10.1146/annurev-pharmtox-010611-134630>

Tesis

(17) Silva, J.L., Venegas, R.A. (2012). Cluster de alto rendimiento para renderizado gráfico utilizando CPU y GPU: Proyecto de Software Aplicado. Tesis de licenciado. Universidad del Bío-Bío, Chile.

(18) Varas, I.A. (2011). Desarrollo de Modelos Tridimensionales para el Transportador de Hexosas

de Plasmodium falciparum: Análisis de sitios alostéricos. Tesis de licenciado. Universidad de Talca, Chile.

(19) Sandoval, D.A. (2012). Desarrollo de Software Simulador Híbrido vía Dinámica Molecular para sistemas con interacción Fluido-Grano. Tesis de licenciado. Universidad de Bío-Bío, Chile.

Web

(20) Galaxy (s.f) Data intensive biology for everyone. [en línea] recuperado el 30 de septiembre del 2012 de: <http://galaxyproject.org/>

(21) RCSB Protein Data Bank (2013). Biological Macromolecular Resource. [en línea] Recuperado el 3 de octubre del 2012 de: <http://www.rcsb.org/pdb/home/home.do>

(22) Nature Publishing Group (2013). Nature.com [en línea] Recuperado del 3 de octubre del 2012 de: <http://www.nature.com/>

(23) Jmol: un visor Java de código abierto para estructuras químicas en tres dimensiones con prestaciones para compuestos químicos, cristales, materiales y biomoléculas. (s.a, s.f) [en línea] Recuperado el 24 de septiembre de: <http://www.jmol.org/>

(24) Python Software Foundation (1990-2013). Python v3.3.2 documentation [en línea] Recuperado el 7 de octubre del 2012 de: <http://docs.python.org/3/>

(25) NetworkX Developers (2010) NetworkX [en línea] Recuperado el 5 septiembre del 2012 de <http://networkx.github.io/documentation/latest/>

(26) Hunter, J.; Dale, D.; Firing, E.; Droettboom, M. and the matplotlib development team. (2013) Matplotlib [en línea] Recuperado el 2 de octubre del 2012 de: <http://matplotlib.org/>

(27) Pymol (s.a, s.f) [en línea] recuperado el 7 de octubre del 2012 de <http://www.pymol.org/>

(28) AutoDock Vina (s.a, s.f) [en línea] Recuperado el 25 de septiembre del 2012 de <http://vina.scripps.edu/>

(29) El Instituto de Investigación Scripps (1989). AutoDock [en línea] Recuperado el 3 de mayo del 2013 de <http://autodock.scripps.edu/>

(30) Nuevo y mejorado: la liberación de muelle 6,6 (s.a,s.f) [en línea] Recuperado el 2 de octubre de 2012 de: <http://dock.compbio.ucsf.edu/>